11–6 Memory Expansion

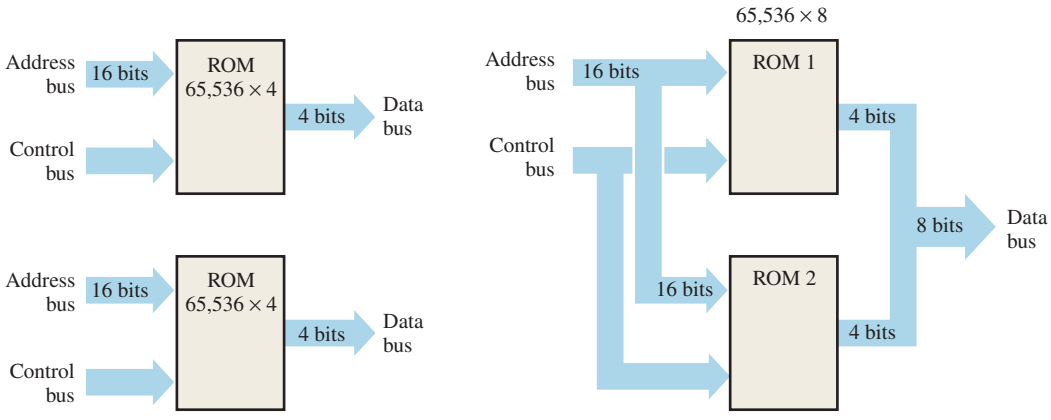
Available memory can be expanded to increase the word length (number of bits in each address) or the word capacity (number of different addresses) or both. Memory expansion is accomplished by adding an appropriate number of memory chips to the address, data, and control buses. SIMMs and DIMMs, which are types of memory expansion modules, are introduced.

After completing this section, you should be able to

- Define *word-length expansion*
- Show how to expand the word length of a memory
- Define *word-capacity expansion*
- Show how to expand the word capacity of a memory
- Discuss types of memory modules

Word-Length Expansion

To increase the **word length** of a memory, the number of bits in the data bus must be increased. For example, an 8-bit word length can be achieved by using two memories, each with 4-bit words as illustrated in Figure 11–40(a). As you can see in part (b), the 16-bit address bus is commonly connected to both memories so that the combination memory still has the same number of addresses ($2^{16} = 65{,}536$) as each individual memory. The 4-bit data buses from the two memories are combined to form an 8-bit data bus. Now when an address is selected, eight bits are produced on the data bus—four from each memory. Example 11–2 shows the details of 65,536 × 4 to 65,536 × 8 expansion.



(a) Two separate 65,536 × 4 ROMs

(b) One 65,536 × 8 ROM from two 65,536 × 4 ROMs

FIGURE 11–40 Expansion of two 65,536 × 4 ROMs into a 65,536 × 8 ROM to illustrate word-length expansion.

EXAMPLE 11–2

Expand the 65,536 × 4 ROM (64k × 4) in Figure 11–41 to form a 64k × 8 ROM. Note that "64k" is the accepted shorthand for 65,536. Why not "65k"? Maybe it's because 64 is also a power-of-two.

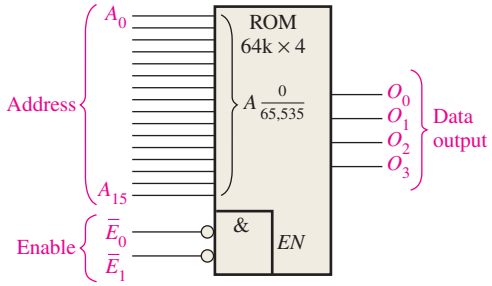


FIGURE 11–41 A 64k × 4 ROM.

Solution

Two 64k × 4 ROMs are connected as shown in Figure 11–42. Notice that a specific address is accessed in ROM 1 and ROM 2 at the same time. The four bits from a selected address in ROM 1 and the four bits from the corresponding address in ROM 2 go out in parallel to form an 8-bit word on the data bus. Also notice that a LOW on the enable line, $\overline{E}$, which forms a simple control bus, enables *both* memories.

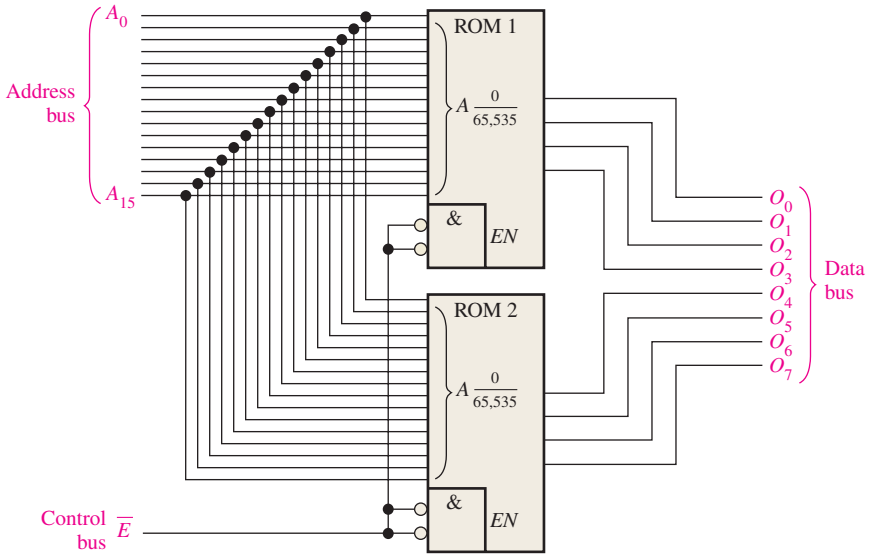


FIGURE 11–42

Related Problem

Describe how you would expand a 64k × 1 ROM to a 64k × 8 ROM.

EXAMPLE 11–3

Use the memories in Example 11–2 to form a 64k × 16 ROM.

Solution

In this case you need a memory that stores 65,536 16-bit words. Four 64k × 4 ROMs are required to do the job, as shown in Figure 11–43.

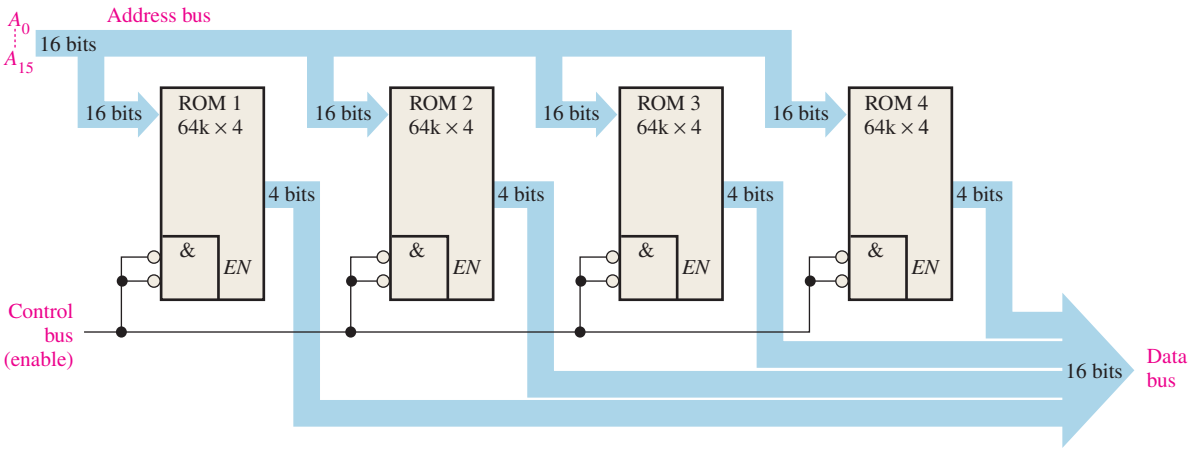


FIGURE 11–43

Related Problem

How many 64k × 1 ROMs would be required to implement the memory shown in Figure 11–43?

A ROM has only data outputs, but a RAM has both data inputs and data outputs. For word-length expansion in a RAM (SRAM or DRAM), the data inputs *and* data outputs form the data bus. Because the same lines are used for data input and data output, tri-state buffers are required. Most RAMs provide internal tri-state circuitry. Figure 11–44 illustrates RAM expansion to increase the data word length.

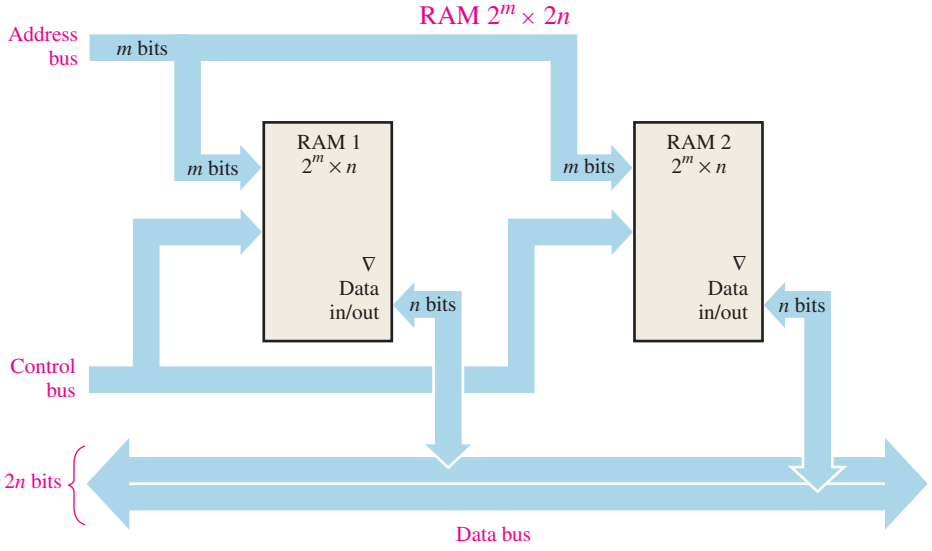


FIGURE 11–44 Illustration of word-length expansion with two $2^m \times n$ RAMs forming a $2^m \times 2n$ RAM.

EXAMPLE 11–4

Use 1M × 4 SRAMs to create a 1M × 8 SRAM.

Solution

Two 1M × 4 SRAMs are connected as shown in the simplified block diagram of Figure 11–45.

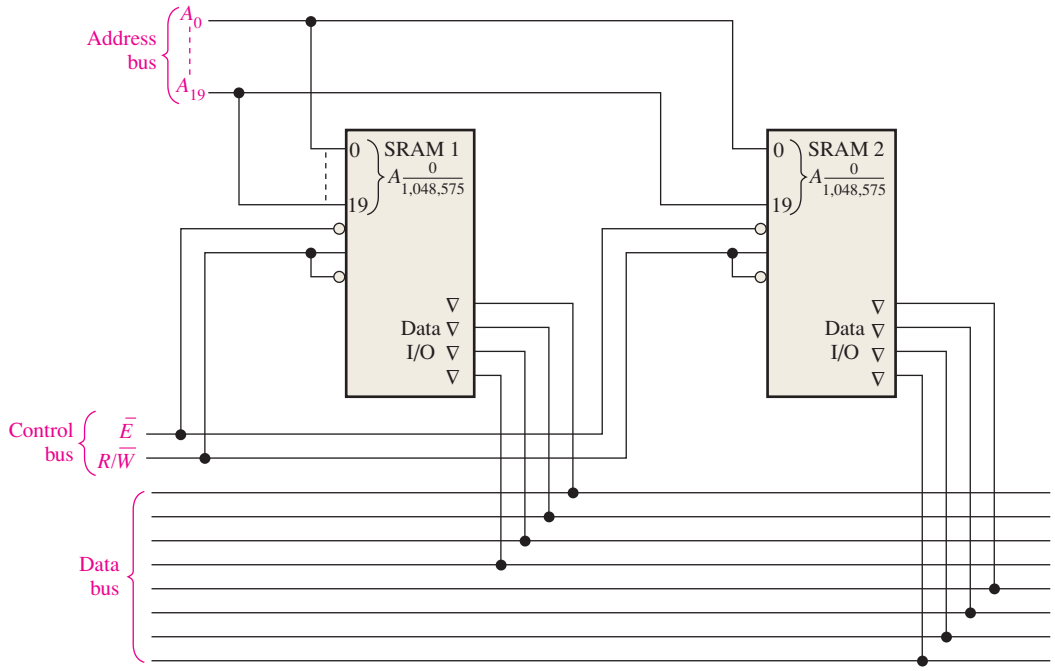


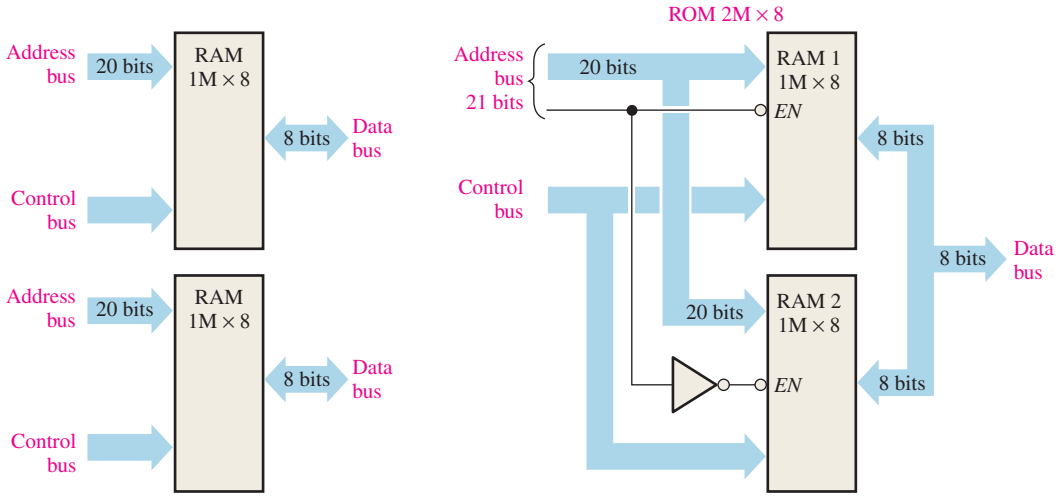
FIGURE 11–45

Related Problem

Use 1M × 8 SRAMs to create a 1M × 16 SRAM.

Word-Capacity Expansion

When memories are expanded to increase the **word capacity**, the *number of addresses is increased.* To achieve this increase, the number of address bits must be increased, as illustrated in Figure 11–46, (where two 1M × 8 RAMs are expanded to form a 2M × 8 memory).



(a) Individual memories each store 1,048,576 8-bit words

(b) Memories expanded to form a 2M × 8 RAM requiring a 21-bit address bus

FIGURE 11–46 Illustration of word-capacity expansion.

Each individual memory has 20 address bits to select its 1,048,576 addresses, as shown in part (a). The expanded memory has 2,097,152 addresses and therefore requires 21 address bits, as shown in part (b). The twenty-first address bit is used to enable the appropriate memory chip. The data bus for the expanded memory remains eight bits wide. Details of this expansion are illustrated in Example 11–5.

EXAMPLE 11–5

Use 512k × 4 RAMs to implement a 1M × 4 memory.

Solution

The expanded addressing is achieved by connecting the enable ($\overline{E}_0$) input to the twentieth address bit (A_{19}), as shown in Figure 11–47. Input $\overline{E}_1$ is used as an enable input common to both memories. When the twentieth address bit (A_{19}) is LOW, RAM 1 is selected (RAM 2 is disabled), and the nineteen lower-order address bits (A_0–A_{18}) access each of the addresses in RAM 1. When the twentieth address bit (A_{19}) is HIGH, RAM 2 is enabled by a LOW on the inverter output (RAM 1 is disabled), and the nineteen lower-order address bits (A_0 – A_{18}) access each of the RAM 2 addresses.

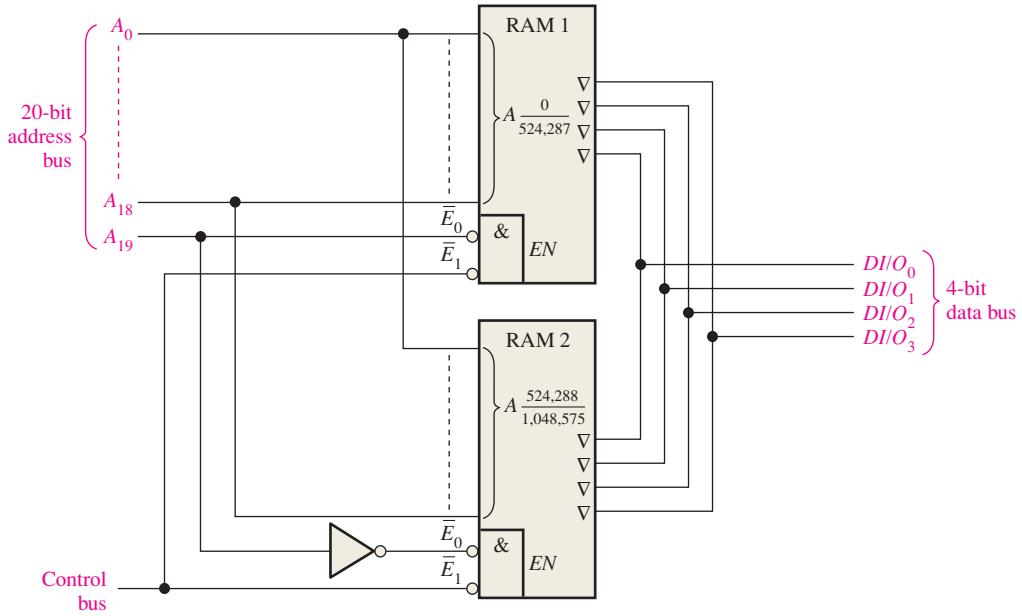


FIGURE 11–47

Related Problem

What are the ranges of addresses in RAM 1 and in RAM 2 in Figure 11–47?

Memory Modules

SDRAMs are available in modules consisting of multiple memory ICs arranged on a printed circuit board (PCB). The most common type of SDRAM memory module is called a **DIMM** (dual in-line memory module). Another version of the DIMM is the SODIMM (small-outline DIMM). A type of memory module, generally found in older equipment and essentially obsolete, is the **SIMM** (single in-line memory module). The SIMM has connection pins on one side of a PCB where the DIMM uses both sides of the board. DIMMs plug into a socket on the system mother board for memory expansion. A generic representation of a memory module is shown in Figure 11–48 with the system board connectors into which the modules are inserted.

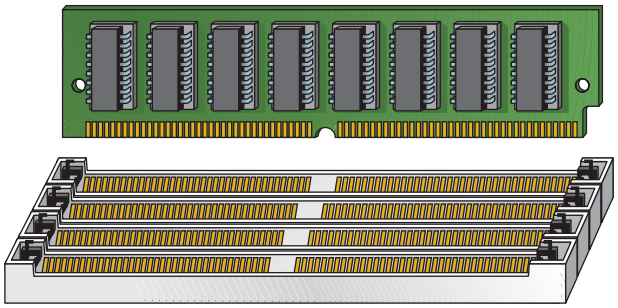


FIGURE 11–48 A memory module with connectors.

DIMMs generally contain DDR SDRAM memory chips. DDR means double data rate, so a DDR SDRAM transfers two blocks of data for each clock cycle rather than one like a standard SDRAM. Three basic types of modules are DDR, DDR2, and DDR3.

- DDR modules have 184 pins and require a 2.5 voltage source.
- DDR2 modules have 240 pins and require a 1.8 voltage source.
- DDR3 modules have 240 pins and require a 1.5 voltage source.

The DDR, DDR2, and DDR3 have transfer data rates of 1600 MB/s, 3200 MB/s, and 6400 MB/s respectively.

*** find info for DDR 4 and DDR5**
***# of pins**
***voltage source required**
***data transfer rate**

Memory components are extremely sensitive to static electricity. Use the following precautions when handling memory chips or modules such as DIMMs:

- Before handling, discharge your body's static charge by touching a grounded surface or wear a grounding wrist strap containing a high-value resistor if available. A convenient, reliable ground is the ac outlet ground.
- Do not remove components from their antistatic bags until you are ready to install them.
- Do not lay parts on the antistatic bags because only the inside is antistatic.
- When handling DIMMs, hold by the edges or the metal mounting bracket. Do not touch components on the boards or the edge connector pins.
- Never slide any part over any type of surface.
- Avoid plastic, vinyl, styrofoam, and nylon in the work area.

When installing DIMMs, follow these steps:

1. Line up the notches on the DIMM board with the notches in the memory socket.
2. Push firmly on the module until it is securely seated in the socket.
3. Generally, the latches on both sides of the socket will snap into place when the module is completely inserted. These latches also release the module, so it can be removed from the socket.

SECTION 11–6 CHECKUP

1. How many 16k × 1 RAMs are required to achieve a memory with a word capacity of 16k and a word length of eight bits?
2. To expand the 16k × 8 memory in question 1 to a 32k × 8 organization, how many more 16k × 1 RAMs are required?
3. What does DIMM stand for?

11–7 Special Types of Memories

In this section, the first in–first out (FIFO) memory, the last in–first out (LIFO) memory, the memory stack, and the charge-coupled device memory are covered.

After completing this section, you should be able to

- Describe a FIFO memory
- Describe a LIFO memory
- Discuss memory stacks
- Explain how to use a portion of RAM as a memory stack
- Describe a basic CCD memory

First In–First Out (FIFO) Memories

This type of memory is formed by an arrangement of shift registers. The term **FIFO** refers to the basic operation of this type of memory, in which the first data bit written into the memory is the first to be read out.

One important difference between a conventional shift register and a FIFO register is illustrated in Figure 11–49. In a conventional register, a data bit moves through the register only as new data bits are entered; in a FIFO register, a data bit immediately goes through the register to the right-most bit location that is empty.

Conventional Shift Register

Input	X	X	X	X	Output
0	0	*X*	*X*	*X*	→
1	1	0	*X*	*X*	→
1	1	1	0	*X*	→
0	0	1	1	1	→

X = unknown data bits.
In a conventional shift register, data stay to the left until "forced" through by additional data.

FIFO Shift Register

Input	—	—	—	—	Output
0	—	—	—	0	→
1	—	—	1	0	→
1	—	1	1	0	→
0	0	1	1	0	→

— = empty positions.
In a FIFO shift register, data "fall" through (go right).

FIGURE 11–49 Comparison of conventional and FIFO register operation.

Figure 11–50 is a block diagram of a FIFO serial memory. This particular memory has four serial 64-bit data registers and a 64-bit control register (marker register). When data are entered by a shift-in pulse, they move automatically under control of the marker register to the empty location closest to the output. Data cannot advance into occupied positions. However, when a data bit is shifted out by a shift-out pulse, the data bits remaining in the registers automatically move to the next position toward the output. In an asynchronous FIFO, data are shifted out independent of data entry, with the use of two separate clocks.

FIFO Applications

One important application area for the FIFO register is the case in which two systems of differing data rates must communicate. Data can be entered into a FIFO register at one rate and taken out at another rate. Figure 11–51 illustrates how a FIFO register might be used in these situations.

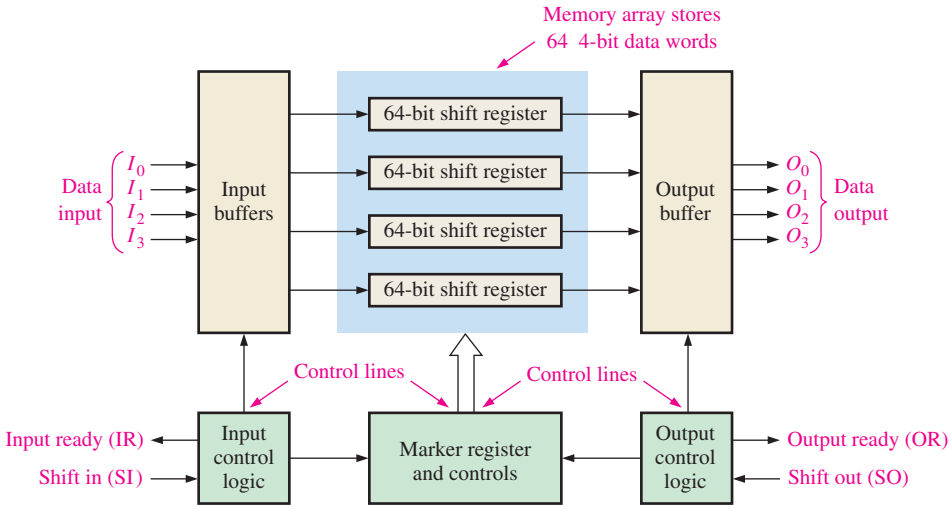


FIGURE 11–50 Block diagram of a typical FIFO serial memory.

(a) Irregular telemetry data can be stored and retransmitted at a constant rate.

(b) Data input at a slow keyboard rate can be stored and then transferred at a higher rate for processing.

(c) Data input at a constant rate can be stored and then output in bursts.

(d) Data in bursts can be stored and reformatted into a constant-rate output.

FIGURE 11–51 Examples of the FIFO register in data-rate buffering applications.

Last In–First Out (LIFO) Memories

The **LIFO** (last in–first out) memory is found in applications involving microprocessors and other computing systems. It allows data to be stored and then recalled in reverse order; that is, the last data byte to be stored is the first data byte to be retrieved.

Register Stacks

A LIFO memory is commonly referred to as a push-down stack. In some systems, it is implemented with a group of registers as shown in Figure 11–52. A stack can consist of any number of registers, but the register at the top is called the *top-of-stack*.

To illustrate the principle, a byte of data is loaded in parallel onto the top of the stack. Each successive byte pushes the previous one down into the next register. This process is illustrated in Figure 11–53. Notice that the new data byte is always loaded into the top register and the previously stored bytes are pushed deeper into the stack. The name *push-down stack* comes from this characteristic.

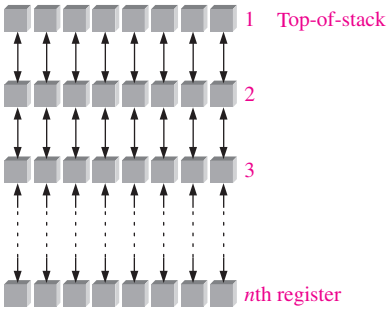


FIGURE 11–52 Register stack.

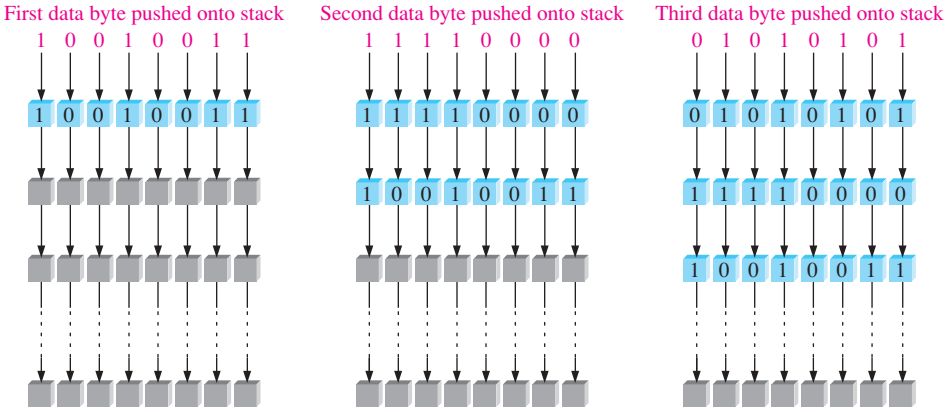


FIGURE 11–53 Simplified illustration of pushing data onto the stack.

Data bytes are retrieved in the reverse order. The last byte entered is always at the top of the stack, so when it is pulled from the stack, the other bytes pop up into the next higher locations. This process is illustrated in Figure 11–54.

RAM Stack

Another approach to LIFO memory used in microprocessor-based systems is the allocation of a section of RAM as the stack rather than the use of a dedicated set of registers. As you have seen, for a register stack the data move up or down from one location to the next. In

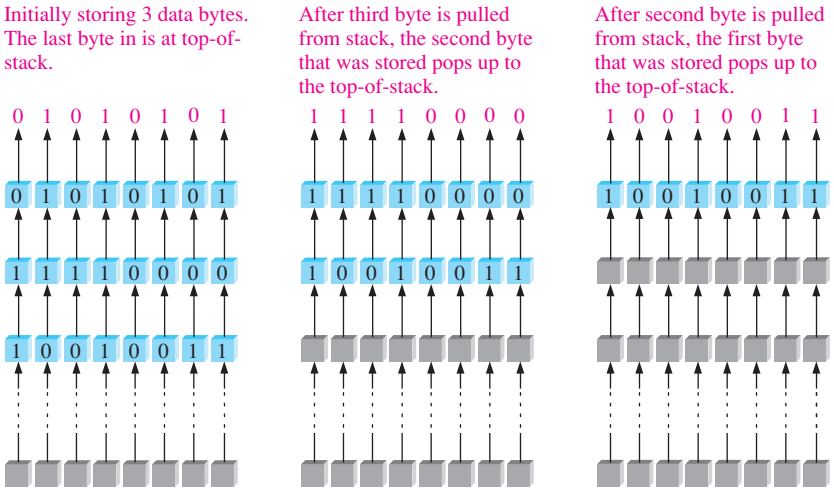


FIGURE 11–54 Simplified illustration of pulling data from the stack.

a RAM stack, the data do not move but the top-of-stack moves under control of a register called the stack pointer.

Consider a random-access memory that is byte organized—that is, one in which each address contains eight bits—as illustrated in Figure 11–55. The binary address 0000000000001111, for example, can be written as 000F in hexadecimal. A 16-bit address can have a *minimum* hexadecimal value of 0000_{16} and a *maximum* value of $FFFF_{16}$. With this notation, a 64 kB memory array can be represented as shown in Figure 11–55. The lowest memory address is 0000_{16} and the highest memory address is $FFFF_{16}$.

Now, consider a section of RAM set aside for use as a stack. A special separate register, the stack pointer, contains the address of the top of the stack, as illustrated in Figure 11–56. A 4-digit hexadecimal representation is used for the binary addresses. In the figure, the addresses are chosen for purposes of illustration.

Now let's see how data are pushed onto the stack. The stack pointer is initially at address $FFEE_{16}$, which is the top of the stack as shown in Figure 11–56(a). The stack pointer is then decremented (decreased) by two to $FFEC_{16}$. This moves the top of the stack to a lower memory address, as shown in Figure 11–56(b). Notice that the top of the stack is not stationary as in the fixed register stack but moves downward (to lower addresses) in the RAM as data words are stored. Figure 11–56(b) shows that two bytes (one data word) are then pushed onto the stack. After the data word is stored, the top of the stack is at $FFEC_{16}$.

Figure 11–57 illustrates the POP operation for the RAM stack. The last data word stored in the stack is read first. The stack pointer that is at FFEC is incremented (increased) by two to address $FFEE_{16}$ and a POP operation is performed as shown in part (b). Keep in mind that RAMs are nondestructive when read, so the data word still remains in the memory after a POP operation. A data word is destroyed only when a new word is written over it.

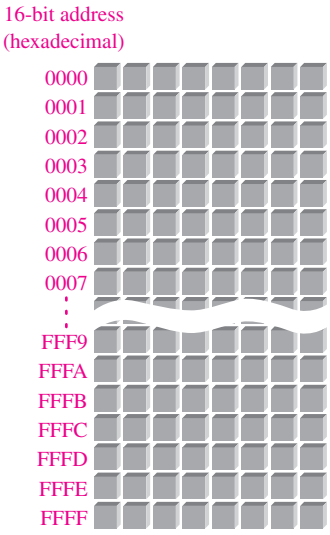
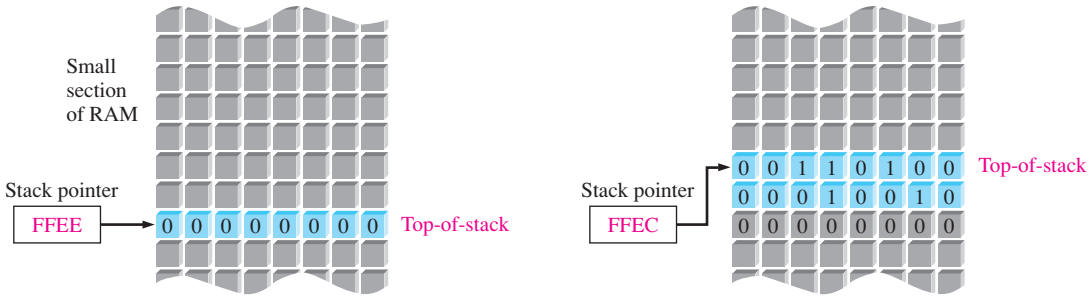


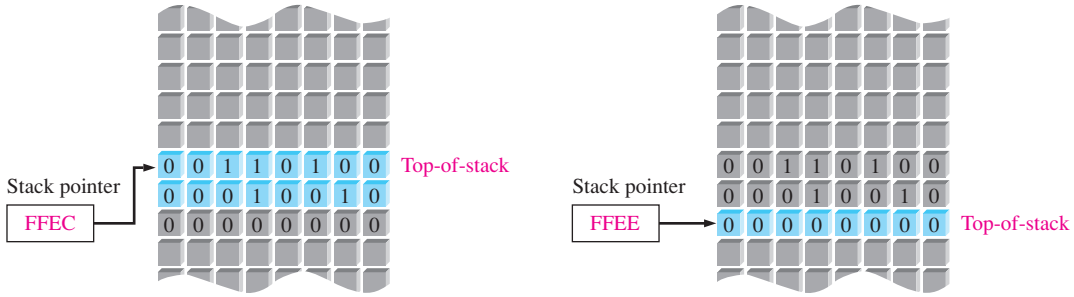
FIGURE 11–55 Representation of a 64 kB memory with the 16-bit addresses expressed in hexadecimal.



(a) The stack pointer is initially at FFEE before the data word 0001001000110100 (1234) is pushed onto the stack.

(b) The stack pointer is decremented by two and the data word 0001001000110100 is placed in the two locations prior to the original stack pointer location.

FIGURE 11–56 Illustration of the PUSH operation for a RAM stack.



(a) The stack pointer is at FFEC before the data word is copied (popped) from the stack.

(b) The stack pointer is incremented by two and the last data word stored is copied (popped) from the stack.

FIGURE 11–57 Illustration of the POP operation for the RAM stack.

A RAM stack can be of any depth, depending on the number of continuous memory addresses assigned for that purpose.

CCD Memories

The **CCD** (charge-coupled device) memory stores data as charges on capacitors and has the ability to convert optical images to electrical signals. Unlike the DRAM, however, the storage cell does not include a transistor. High density is the main advantage of CCDs, and these devices are widely used in digital imaging.

The CCD memory consists of long rows of semiconductor capacitors, called *channels.* Data are entered into a channel serially by depositing a small charge for a 0 and a large charge for a 1 on the capacitors. These charge packets are then shifted along the channel by clock signals as more data are entered.

As with the DRAM, the charges must be refreshed periodically. This process is done by shifting the charge packets serially through a refresh circuit. Figure 11–58 shows the basic concept of a CCD channel. Because data are shifted serially through the channels, the CCD memory has a relatively long access time. CCD arrays are used in many modern cameras to capture video images in the form of light-induced charge.

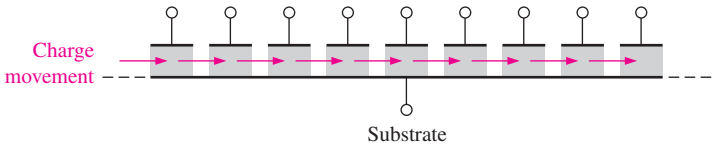


FIGURE 11–58 A CCD (charge-coupled device) channel.

SECTION 11–7 CHECKUP

1. What is a FIFO memory?
2. What is a LIFO memory?
3. Explain the PUSH operation in a memory stack.
4. Explain the POP operation in a memory stack.
5. What does the term *CCD* stand for?

11–8 Magnetic and Optical Storage

In this section, the basics of magnetic disks, magnetic tape, magneto-optical disks, and optical disks are introduced. These storage media are important, particularly in computer applications, where they are used for mass nonvolatile storage of data and programs.

After completing this section, you should be able to

- Describe a magnetic hard disk
- Discuss magnetic tape
- Discuss removable hard disks
- Explain the principle of magneto-optical disks
- Discuss the CD-ROM, CD-R, and CD-RW disks
- Describe the WORM
- Discuss the DVD-ROM

Magnetic Storage

Magnetic Hard Disks

Computers use hard disks as the internal mass storage media. **Hard disks** are rigid "platters" made of aluminum alloy or a mixture of glass and ceramic covered with a magnetic coating. Hard disk drives mainly come in three diameter sizes, 3.5 in., 2.5 in., and 1.8 in. Older formats of 8 in. and 5.25 in. are considered obsolete. A hard disk drive is hermetically sealed to keep the disks dust-free.

Typically, two or more disks are stacked on top of each other on a common shaft or spindle that turns the assembly at several thousand rpm. A separation between each disk allows for a magnetic read/write head that is mounted on the end of an actuator arm, as shown in Figure 11–59. There is a read/write head for both sides of each disk since data are recorded on both sides of the disk surface. The drive actuator arm synchronizes all the read/write heads to keep them in perfect alignment as they "fly" across the disk surface with a separation of only a fraction of a millimeter from the disk. A small dust particle could cause a head to "crash," causing damage to the disk surface.

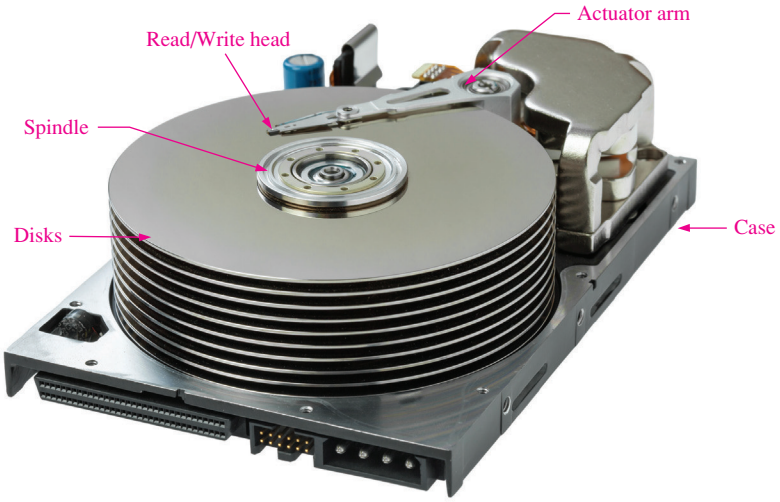


FIGURE 11–59 A hard disk drive. FrameAngel/Shutterstock

Basic Read/Write Head Principles

The hard drive is a random-access device because it can retrieve stored data anywhere on the disk in any order. A simplified diagram of the magnetic surface read/write operation is shown in Figure 11–60. The direction or polarization of the magnetic domains on the disk surface is controlled by the direction of the magnetic flux lines (magnetic field) produced

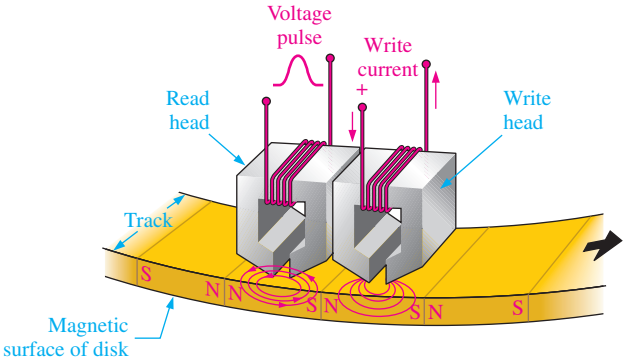


FIGURE 11–60 Simplified read/write head operation.

InfoNote

Data are stored on a hard drive in the form of files. Keeping track of the location of files is the job of the device driver that manages the hard drive (sometimes referred to as hard drive BIOS). The device driver and the computer's operating system can access two tables to keep track of files and file names. The first table is called the FAT (File Allocation Table). The FAT shows what is assigned to specific files and keeps a record of open sectors and bad sectors. The second table is the Root Directory which has file names, type of file, time and date of creation, starting cluster number, and other information about the file.

by the write head according to the direction of a current pulse in the winding. This magnetic flux magnetizes a small spot on the disk surface in the direction of the magnetic field. A magnetized spot of one polarity represents a binary 1, and one of the opposite polarity represents a binary 0. Once a spot on the disk surface is magnetized, it remains until written over with an opposite magnetic field.

When the magnetic surface passes a read head, the magnetized spots produce magnetic fields in the read head, which induce voltage pulses in the winding. The polarity of these pulses depends on the direction of the magnetized spot and indicates whether the stored bit is a 1 or a 0. The read and write heads are usually combined in a single unit.

Hard Disk Format

A hard disk is organized or formatted into tracks and sectors, as shown in Figure 11–61(a). Each track is divided into a number of sectors, and each track and sector has a physical address that is used by the operating system to locate a particular data record. Hard disks typically have from a few hundred to thousands of tracks and are available with storage capacities of up to 1 TB or more. As you can see in the figure, there is a constant number of tracks/sector, with outer sectors using more surface area than the inner sectors. The arrangement of tracks and sectors on a disk is known as the *format*.

A hard disk stack is illustrated in Figure 11–61(b). Hard disk drives differ in the number of disks in a stack, but there is always a minimum of two. All of the same corresponding tracks on each disk are collectively known as a cylinder, as indicated.

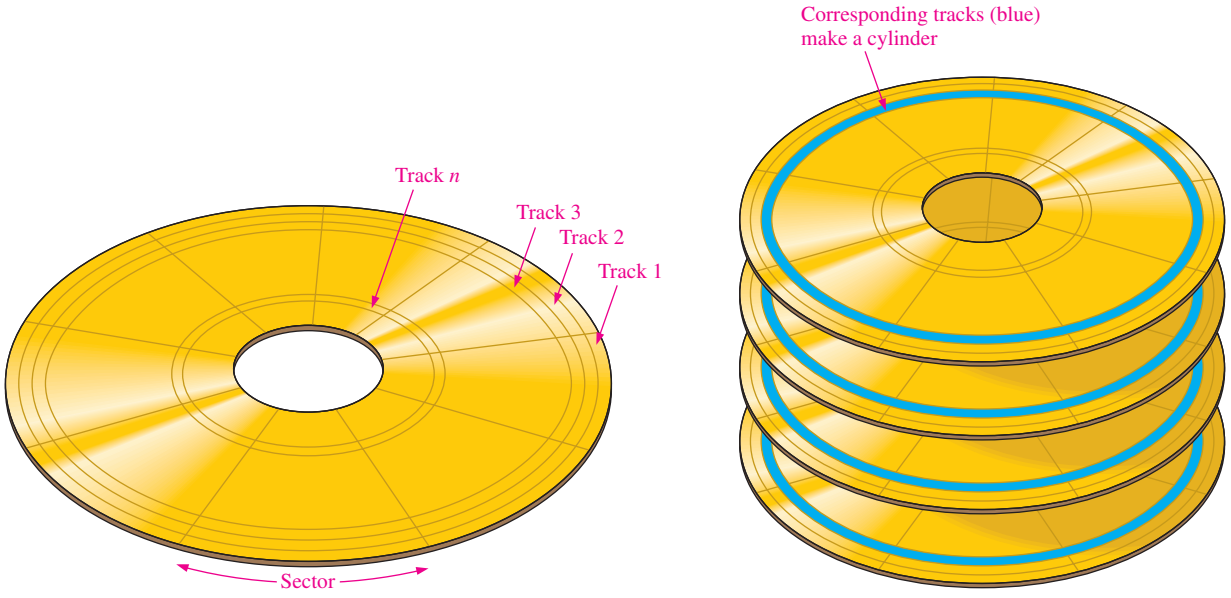


FIGURE 11–61 Hard disk organization and formatting.

Hard Disk Performance

Several basic parameters determine the performance of a given hard disk drive. A *seek* operation is the movement of the read/write head to the desired track. The **seek time** is the average time for this operation to be performed. Typically, hard disk drives have an average seek time of several milliseconds, depending on the particular drive.

The **latency period** is the time it takes for the desired sector to spin under the head once the head is positioned over the desired track. A worst case is when the desired sector is just past the head position and spinning away from it. The sector must rotate almost a full

revolution back to the head position. *Average latency period* assumes that the disk must make half of a revolution. Obviously, the latency period depends on the constant rotational speed of the disk. Disk rotation speeds are different for different disk drives but typically are from 4200 rpm to 15,000 rpm.

The sum of the average seek time and the average latency period is the *access time* for the disk drive.

Removable Hard Disk

A removable hard disk drive with a capacity of 1 TB is available. Keep in mind that the technology is changing so rapidly that there most likely will be further advancements at the time you are reading this.

Magnetic Tape

Tape is used for backup data from mass storage devices and typically is slower than disks because data on tape is accessed serially rather than randomly. There are several types that are available, including QIC, 8 mm, and DLT.

QIC is an abbreviation for quarter-inch cartridge and looks much like audio tape cassettes with two reels inside. Various QIC standards have from 28 to 108 tracks that can store from 80 MB to 1.6 GB. More recent innovations under the Travan standard have lengthened the tape and increased its width allowing storage capacities up to 10 GB. QIC tape drives use read/write heads that have a single write head with a read head on each side. This allows the tape drive to verify data just written when the tape is running in either direction. In the record mode, the tape moves past the read/write heads at approximately 100 inches/second, as indicated in Figure 11–62.

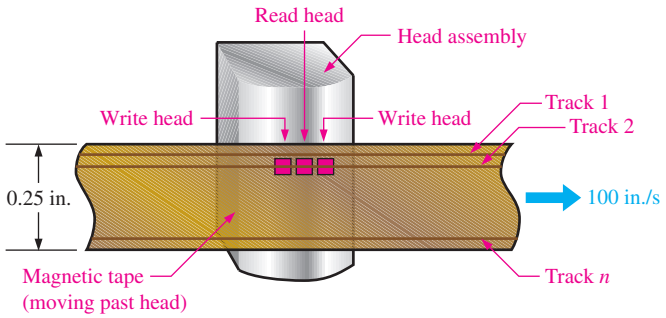


FIGURE 11–62 QIC tape.

8 mm tape was originally designed for the video industry but has been adopted by the computer industry as a reliable way to store large amounts of computer data.

DLT is an abbreviation for digital linear tape. DLT is a half-inch wide tape, which is 60% wider than 8 mm and, of course, twice as wide as standard QIC. Basically, DLT differs in the way the tape-drive mechanism works to minimize tape wear compared to other systems. DLT offers the highest storage capacity of all the tape formats with capacities ranging up to 800 GB.

Magneto-Optical Storage

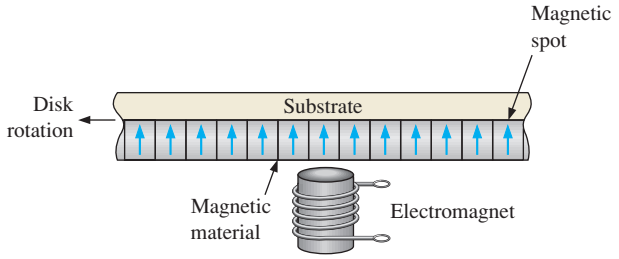
As the name implies, magneto-optical (MO) storage devices use a combination of magnetic and optical (laser) technologies. A **magneto-optical disk** is formatted into tracks and sectors similar to magnetic disks.

The basic difference between a purely magnetic disk and an MO disk is that the magnetic coating used on the MO disk requires heat to alter the magnetic polarization. Therefore, the MO is extremely stable at ambient temperature, making data unchangeable. To write a data bit, a high-power laser beam is focused on a tiny spot on the disk, and the

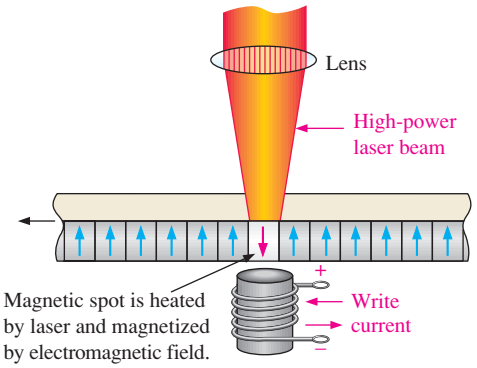
InfoNote

Tape is a viable alternative to disk due to its lower cost per bit. Though the density is lower than for a disk drive, the available surface on a tape is far greater. The highest-capacity tape media are generally on the same order as the largest available disk drive (about 1 TB—a terabyte is one trillion bytes.) Tape has historically offered enough advantage in cost over disk storage to make it a viable product, particularly for backup, where media removability is also important.

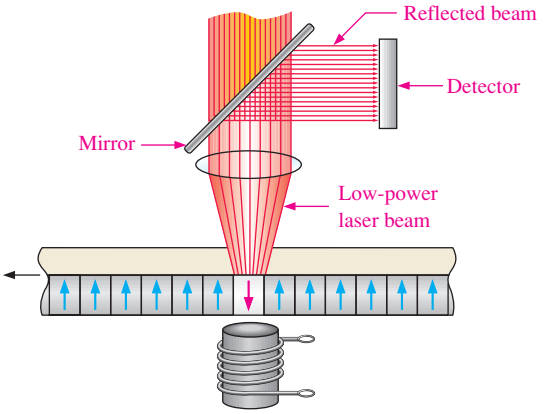
temperature of that tiny spot is raised above a temperature level called the Curie point (about 200°C). Once heated, the magnetic particles at that spot can easily have their direction (polarization) changed by a magnetic field generated by the write head. Information is read from the disk with a less-powerful laser than used for writing, making use of the Kerr effect where the polarity of the reflected laser light is altered depending on the orientation of the magnetic particles. Magnetic spots of one polarity represent 0s and magnetic spots of the opposite polarity represent 1s. Basic MO operation is shown in Figure 11–63, which represents a small cross-sectional area of a disk.



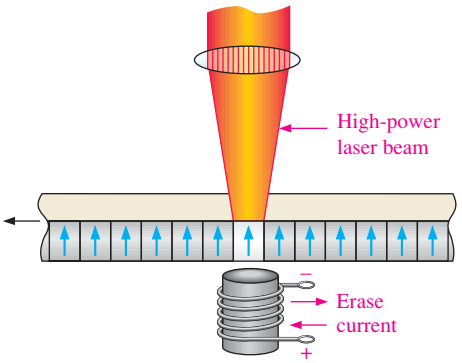
(a) Unrecorded disk



(b) Writing: A high-power laser beam heats the spot, causing the magnetic particles to align with the electromagnetic field.



(c) Reading: A low-power laser beam reflects off of the reversed-polarity magnetic particles and its polarization shifts. If the particles are not reversed, the polarization of the reflected beam is unchanged.



(d) Erasing: The electromagnetic field is reversed as the high-power laser beam heats the spot, causing the magnetic particles to be restored to the original polarity.

FIGURE 11–63 Basic principle of a magneto-optical disk.

Optical Storage

CD-ROM

The most common Compact Disk–Read-Only Memory is a 120 mm diameter disk with a sandwich of three coatings: a polycarbonate plastic on the bottom, a thin aluminum sheet for reflectivity, and a top coating of lacquer for protection. The **CD-ROM** disk is formatted in a single spiral track with sequential 2 kB sectors and has a capacity of 680 MB. Data are prerecorded at the factory in the form of minute indentations called *pits* and the flat area surrounding the pits called *lands*. The pits are stamped into the plastic layer and cannot be erased.

A CD player reads data from the spiral track with a low-power infrared laser, as illustrated in Figure 11–64. The data are in the form of pits and lands as shown. Laser light

reflected from a pit is 180° out-of-phase with the light reflected from the lands. As the disk rotates, the narrow laser beam strikes the series of pits and lands of varying lengths, and a photodiode detects the difference in the reflected light. The result is a series of 1s and 0s corresponding to the configuration of pits and lands along the track.

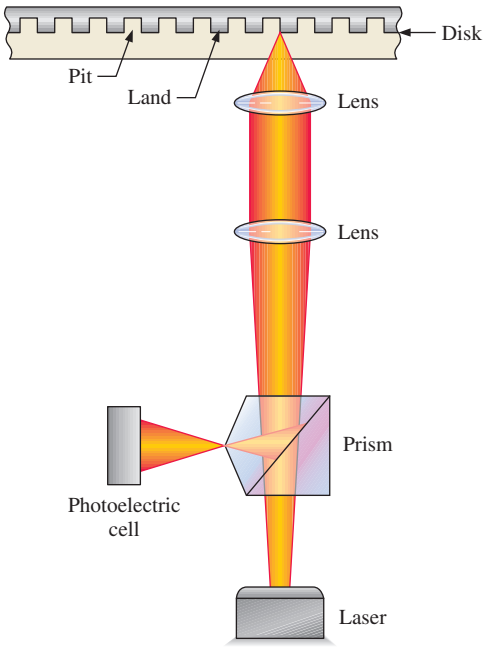


FIGURE 11–64 Basic operation of reading data from a CD-ROM.

WORM

Write Once/Read Many (**WORM**) is a type of optical storage that can be written onto one time after which the data cannot be erased but can be read many times. To write data, a low-power laser is used to burn microscopic pits on the disk surface. 1s and 0s are represented by the burned and nonburned areas.

CD-R

This is essentially a type of WORM. The difference is that the CD-Recordable allows multiple write sessions to different areas of the disk. The **CD-R** disk has a spiral track like the CD-ROM, but instead of mechanically pressing indentations on the disk to represent data, the CD-R uses a laser to burn microscopic spots into an organic dye surface. When heated beyond a critical temperature with a laser during read, the burned spots change color and reflect less light than the nonburned areas. Therefore, 1s and 0s are represented on a CD-R by burned and nonburned areas, whereas on a CD-ROM they are represented by pits and lands. Like the CD-ROM, the data cannot be erased once it is written.

CD-RW

The CD-Rewritable disk can be used to read and write data. Instead of the dye-based recording layer in the CD-R, the **CD-RW** commonly uses a crystalline compound with a special property. When it is heated to a certain temperature, it becomes crystalline when it cools; but if it is heated to a certain higher temperature, it melts and becomes amorphous when it cools. To write data, the focused laser beam heats the material to the melting temperature resulting in an amorphous state. The resulting amorphous areas reflect less light than the crystalline areas, allowing the read operation to detect 1s and 0s. The data can be erased or overwritten by heating the amorphous areas to a temperature above the crystallization

temperature but lower than the melting temperature that causes the amorphous material to revert to a crystalline state.

DVD-ROM

Originally DVD was an abbreviation for Digital Video Disk but eventually came to represent *Digital Versatile Disk.* Like the CD-ROM, **DVD-ROM** data are prestored on the disk. However, the pit size is smaller than for the CD-ROM, allowing more data to be stored on a track. The major difference between CD-ROM and DVD-ROM is that the CD is single-sided, while the DVD has data on both sides. Also, in addition to double-sided DVD disks, there are also multiple-layer disks that use semitransparent data layers placed over the main data layers, providing storage capacities of tens of gigabytes. To access all the layers, the laser beam requires refocusing going from one layer to the other.

Blu-Ray

The **Blu-ray** Disc (BD) is designed to eventually replace the DVD. The BD is the same size as DVDs and CDs. The name *Blu-ray* refers to the blue laser used to read the disc. DVDs use a red laser that has a longer wavelength. Information can be stored on a BD at a greater density and video definition than is possible with a DVD. The smaller Blu-ray laser beam can read recorded data in pits that are less than half the size of the pits on a DVD. A Blu-ray Disc can store about five times more data than a DVD. Typical storage capacities for conventional Blu-ray dual-layer discs are 50 GB, which is the industry standard for feature-length video. Triple layer and quadruple layer discs (BD-XL) can store 100 GB and 128 GB, respectively. Storage capacities up to 1 TB are currently under development.

SECTION 11–8 CHECKUP

1. List the major types of magnetic storage.
2. Generally, how is a magnetic disk organized?
3. How are data written on and read from a magneto-optical disk?
4. List the types of optical storage.

11–9 Memory Hierarchy

A memory system performs the data storage function in a computer. The memory system holds data temporarily during processing and also stores data and programs on a long-term basis. A computer has several types of memory, such as registers, cache, main, and hard disk. Other types of storage can also be used, such as magnetic tape, optical disk, and magnetic disk. Memory hierarchy as well as the system processor determines the processing speed of a computer.

After completing this section, you should be able to

- Discuss several types of memory
- Define memory hierarchy
- Describe key elements in a memory hierarchy

Three key characteristics of memory are cost, capacity, and access time. Memory cost is usually specified in cost per bit. The capacity of a memory is measured in the amount of data (bits or bytes) it can store. The access time is the time it takes to acquire a specified unit of data from the memory. The greater the capacity, the smaller the cost and the greater the access time. The smaller the access time, the greater the cost. The goal of using

a memory hierarchy is to obtain the shortest possible average access time while minimizing the cost.

The speed with which data can be processed depends both on the processor speed and on the time it takes to access stored data. **Memory hierarchy** is the arrangement of various memory elements within the computer architecture to maximize processing speed and minimize cost. Memory can be classified according to its "distance" from the processor in terms of the number of machine cycles or access time required to get data for processing. Distance is measured in time, not in physical location. Faster memory elements are considered closer to the processor compared to slower types of memory elements. Also, the cost per bit is much greater for the memory close to the processor than for the memory that is further from the processor. Figure 11–65 illustrates the arrangement of elements in a typical memory hierarchy.

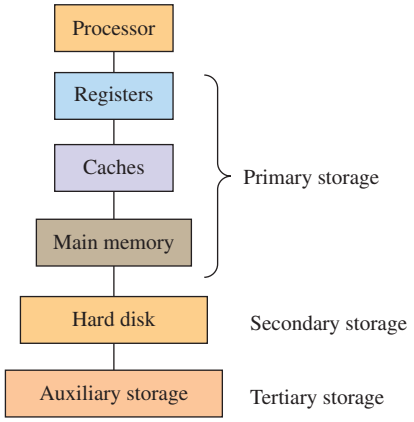


FIGURE 11–65 Typical memory hierarchy.

A primary distinction between the storage elements in Figure 11–65 is the time required for the processor to access data and programs. This access time is known as **memory latency**. The greater the latency, the further from the processor a storage element is considered to be. For example, typical register latency can be up to 1 or 2 ns, cache latency can be up to about 50 ns, main memory latency can be up to about 90 ns, and hard disk latency can be up to about 20 ms. Auxiliary memory latency can range up to several seconds.

Registers

Registers are memory elements that are located within the processor. They have a very small latency as well as a low capacity (number of bits that can be stored). One goal of programming is to keep as much frequently used data in the registers as possible. The number of registers in a processor can vary from the tens to hundreds.

Caches

The next level in the hierarchy is the memory cache, which provides temporary storage. The L1 cache is located in the processor, and the L2 cache is outside of the processor. A programming goal is to keep as much of a program as possible in the cache, especially the parts of a program that are most extensively used. There can be more than two caches in a memory system.

Main Memory

Main memory generally consists of two elements: RAM (random-access memory) and ROM (read-only memory). The RAM is a working memory that temporarily stores less

frequently used data and program instructions. The RAM is volatile, which means that the stored contents are lost when the power is turned off. The ROM is for permanent storage of frequently used programs and data; ROM is nonvolatile. Registers, caches, and main memory are considered primary storage.

Hard Disk

The hard disk has a very high latency and is used for mass storage of data and programs on a permanent basis. The hard disk is also used for virtual memory, space allocated for data when the primary memory fills up. In effect, virtual memory simulates primary memory with the disadvantage of high latency. Capacities range up to about 1 terabyte (TB).

$$1\text{ TB} = 1{,}000{,}000{,}000{,}000\text{ B} = 10^{12}\text{ B}$$

In addition to the internal hard disk, secondary storage can also include off-line storage. Off-line storage includes DVDs, CD-ROM, CD-RW, and USB flash drive. Off-line storage is removable storage.

Auxiliary Storage

Auxiliary storage, also called *tertiary storage,* includes magnetic tape libraries and optical jukeboxes. A **tape library** can store immense amounts of data (up to hundreds of petabytes). A petabyte (PB) is

$$1\text{ PB} = 1{,}000{,}000{,}000{,}000{,}000\text{ B} = 10^{15}\text{ B}$$

An **optical jukebox** is a robotic storage device that automatically loads and unloads optical disks. It may have as many as 2,000 slots for disks and can store hundreds of petabytes.

Relationship of Cost, Capacity, and Access Time

Figure 11–66 shows how capacity (the amount of data a memory can store) and cost per unit of storage varies as the distance from the processor, in terms of access time or latency, increases. The capacity increases and the cost decreases as access time increases.

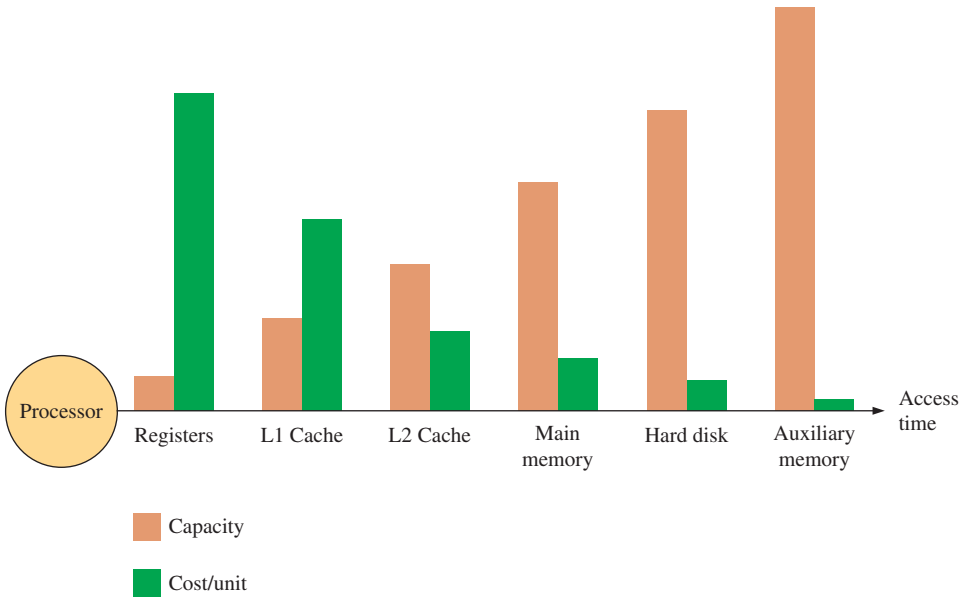


FIGURE 11–66 Changes in memory capacity and cost per unit of data as latency (access time) increases.

Memory Hierarchy Performance

In a computer system, the overall processing speed is usually limited by the memory, not the processor. Programming determines how well a particular memory hierarchy is utilized. The goal is to process data at the fastest rate possible. Two key factors in establishing maximum processor performance are locality and hit rate.

If a block of data is referenced, it will tend to be either referenced again soon or a nearby data block will be referenced soon. Frequent referencing of the same data block is known as *temporal locality*, and the program code should be arranged so that the piece of the data in the cache is reused frequently. Referencing an adjacent data block is known as *spatial locality*, and the program code should be arranged to use consecutive pieces of data on a frequent basis.

A **miss** is a failed attempt by the processor to read or write a block of data in a given level of memory (such as the cache). A miss causes the processor to have to go to a lower level of memory (such as main memory), which has a longer latency. The three types of misses are instruction read miss, data read miss, and data write miss. A successful attempt to read or write a block of data in a given level of memory is called a *hit*. Hits and misses are illustrated in Figure 11–67, where the processor is requesting data from the cache.

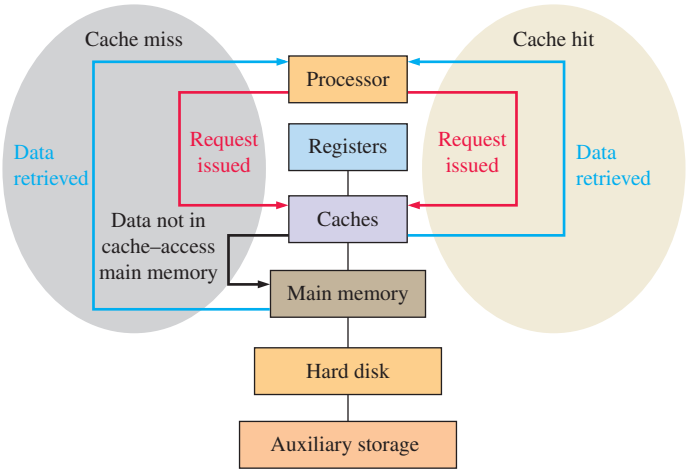


FIGURE 11–67 Illustration of a cache hit and a miss.

The **hit rate** is the percentage of memory accesses that find the requested data in the given level of memory. The *miss rate* is the percentage of memory accesses that fail to find the requested data in the given level of memory and is equal to 1– hit rate. The time required to access the requested information in a given level of memory is called the *hit time*. The higher the hit rate (hit to miss ratio), the more efficient the memory hierarchy is.

SECTION 11–9 CHECKUP

1. State the purpose of memory hierarchy.
2. What is access time?
3. How does memory capacity affect the cost per bit?
4. Does higher level memory generally have lower capacity than lower level memory?
5. What is a hit? A miss?
6. What determines the efficiency of the memory hierarchy?

11–10 Cloud Storage

Cloud storage is a system, usually maintained by a third party, for securely storing data in a remote location that can be conveniently accessed through the Internet. A file on a computer can be stored on secure remote servers and accessed by various user devices such as computers, smart phones, and tablets. Cloud storage eliminates the need for local backup storage such as external hard drives or CDs. When you use cloud storage, you are essentially storing your files or documents on Internet servers instead of or in addition to a computer. The term *cloud* may have originated from the use of a symbol that resembled a cloud on early network diagrams.

After completing this section, you should be able to

- Describe cloud storage
- Explain what a server is
- State the advantages of cloud storage
- Describe several properties of cloud storage

The Cloud Storage System

A **cloud storage** system consists of a remote network of servers (also called *nodes*) that are connected to a user device through the Internet, as shown in Figure 11–68. Some cloud storage systems accommodate only certain types of data such as e-mail or digital pictures, while others store all types of data and range in size from small operations with a few servers to very large operations that utilize hundreds of servers. A facility that houses cloud storage systems is called a **data center**. A typical storage cloud system can serve multiple users.

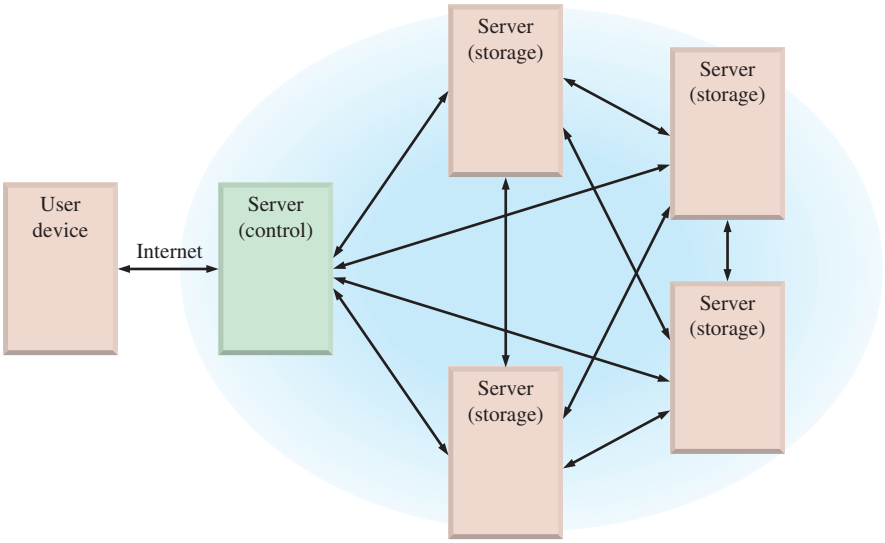


FIGURE 11–68 A typical cloud storage system architecture consists of a master control server and several storage servers that can be accessed by a user device over the Internet.

Servers typically operate within a client-server architecture, where the client is the user that is subscribing to the cloud storage. Theoretically, a **server** is any computerized process that shares a resource with one or more clients. More practically, a storage server is a computer and software with a large memory capacity that responds to requests across a network to provide file storage and access as well as services such as file sharing. The control server

(a) A typical rack of servers (b) A typical server room in a data center

FIGURE 11–69 Cloud servers. (a) Jojje/Shutterstock (b) Oleksiy Mark/Shutterstock

coordinates the activities within the storage cloud network among other servers and manages user access. A server rack and data center are shown in Figure 11–69.

At its simplest level, a cloud storage system needs just one storage server connected to the Internet. When copies of a file are sent by a client to the server over the Internet, the data are stored. When the client wishes to retrieve the data, the storage server (node) sends it back through a Web-based interface or allows the client to manipulate the file on the server itself.

Most cloud storage systems have many storage servers (hundreds in some cases) to provide both capacity and redundancy. A grouping of servers is sometimes called a *cluster*. Depending on the system architecture, a given system may have multiple clusters. A simple system with four storage servers illustrating file storage redundancy is shown in Figure 11–70. When a client sends data to the cloud, it is stored in multiple servers. This redundancy guarantees availability of data at any time to the client and makes the system highly reliable. Redundancy is necessary because a server requires periodic maintenance or may break down and need repairs. In addition to storage server redundancy, most cloud storage systems use power supply redundancy so that all servers are not operating from the same power source.

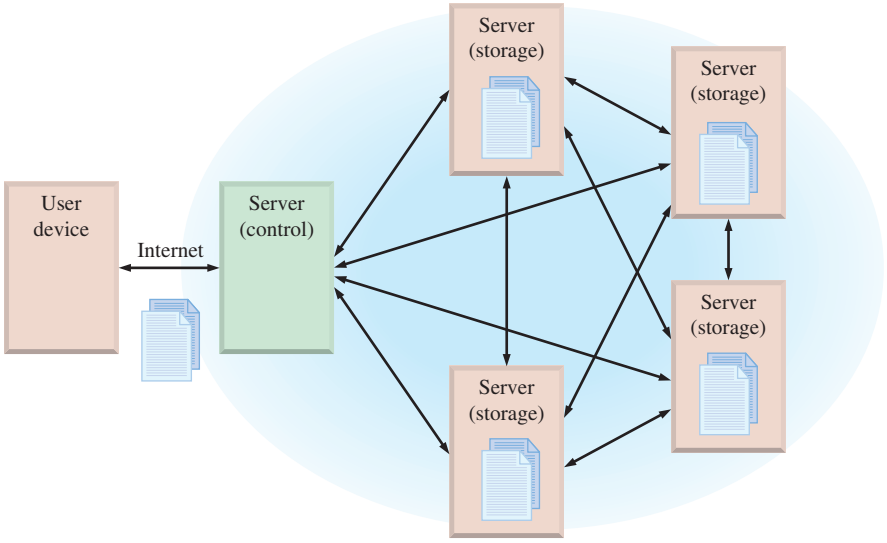


FIGURE 11–70 A simple cloud storage system with storage redundancy. In this case, the data are stored on four different servers.

In addition to reliability that provides assurance that a client's data are accurately stored and can be retrieved at any time, a second major factor for cloud storage is security that the data cannot be compromised. Generally, three methods are used to provide data security:

- *Encryption* or encoding, which prevents the data from being read or interpreted without proper decryption tools
- *Authentication*, which requires a name and password for access
- *Authorization*, which requires a list of only those people who can have access to the data

Cloud storage has certain advantages over traditional data storage in a computer. One advantage is that you can store and retrieve data from any physical location that has Internet access. A second advantage is that you don't have to use the same computer to store and retrieve data or carry a physical storage device for data backup around with you. Also, the user does not have to maintain the storage components. Another advantage of cloud storage is that other people can access your data (data sharing).

Architecture

The term *architecture* relates to how a cloud storage system is structured and organized. The primary purpose of cloud storage architecture is to deliver the service for data storage in a specific way. Architectures vary but generically most consist of a front end, a control, and a back end, as depicted in Figure 11–71.

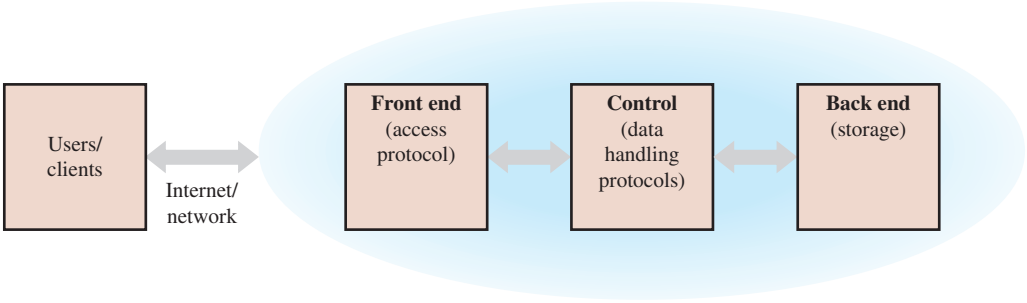


FIGURE 11–71 Generic architecture of a cloud storage system.

A cloud storage system uses various protocols within the architecture that determine how the data are accessed and handled. A **protocol** is a standardized set of software regulations, requirements, and procedures that control and regulate the transmission, processing, and exchange of data among devices. For example, common Internet protocols are HTTP (Hypertext Transfer Protocol), FTP (File Transfer Protocol), TCP/IP (Transfer Control Protocol/Internet Protocol), and SMTP (Simple Mail Transfer Protocol).

An API is an Application Programming Interface, which is essentially a protocol for access and utilization of a cloud storage system. There are many types of APIs. For example, a commonly used one is the REST API. REST stands for Representational State Transfer. An API is a software-to-software interface, not a user interface. With APIs, applications talk to each other "behind the scene" without user knowledge.

Cloud Storage Properties

The following cloud storage properties determine the performance of the system.

- *Latency*. The time between a request for data and the delivery of the data to the user is the **latency** of a system. Delay is due to the time for each component of the cloud storage system to respond to a request and to the time for data to be transferred to the user.

- *Bandwidth*. Bandwidth is a measure of the range of frequencies that can be simultaneously transferred to the cloud and is defined as a range of frequencies that can be handled by the system. Generally, the wider the bandwidth, the shorter the latency and vice versa.
- *Scalability*. The **scalability** property indicates the ability of a cloud storage system to handle increasing amounts of data in a smooth and easy manner; or it is the cloud's ability to improve movement of data through the system (throughput) when additional resources (typically hardware) are added. When the performance of a system improves proportionally to the storage capacity added, the system is said to be scalable. Scaling vertically (scale up) occurs when resources (hardware and memory) are added to a single server (node). Scaling horizontally (scale out) occurs when more servers (nodes) are added to a system.
- *Elasticity*. **Elasticity** is a cloud's ability to deal with variations in the amount of data (load) being transferred in and out of the storage system without service interrupts. There is a subtle difference between scalability and elasticity when describing a system's behavior. Essentially, *scalability* is a static parameter that indicates how much the system can be expanded, and *elasticity* is a dynamic parameter that refers to the implementation of scalability. For example, a storage system may be scalable from one to 100 servers. If the system is currently operating with 20 servers (nodes) and the data load doubles, its elasticity allows 20 more nodes to be added for a total of 40. Likewise, if the data load decreases by half, the elasticity allows 10 nodes to be removed. A server can be added or removed by powering it up or down in a proper manner without disrupting service to the user. Elasticity results in cost efficiency because only the number of servers required for the data load at any given time are consuming power.
- *Multitenancy*. The **multitenancy** property of a cloud storage system allows multiple users to share the same software applications and hardware and the same data storage mechanism but not to see each other's data.

SECTION 11–10 CHECKUP

1. What is a cloud storage system?
2. What is a server?
3. How does a user connect to a cloud storage system?
4. Name three advantages of a cloud system.

11–11 Troubleshooting

Because memories can contain large numbers of storage cells, testing each cell can be a lengthy and frustrating process. Fortunately, memory testing is usually an automated process performed with a programmable test instrument or with the aid of software for in-system testing. Most microprocessor-based systems provide automatic memory testing as part of their system software.

After completing this section, you should be able to

- Discuss the checksum method of testing ROMs
- Discuss the checkerboard pattern method of testing RAMs

ROM Testing

Since ROMs contain known data, they can be checked for the correctness of the stored data by reading each data word from the memory and comparing it with a data word that

is known to be correct. One way of doing this is illustrated in Figure 11–72. This process requires a reference ROM that contains the same data as the ROM to be tested. A special test instrument is programmed to read each address in both ROMs simultaneously and to compare the contents. A flowchart in Figure 11–73 illustrates the basic sequence.

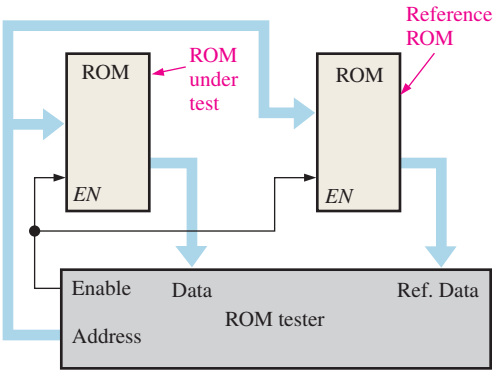
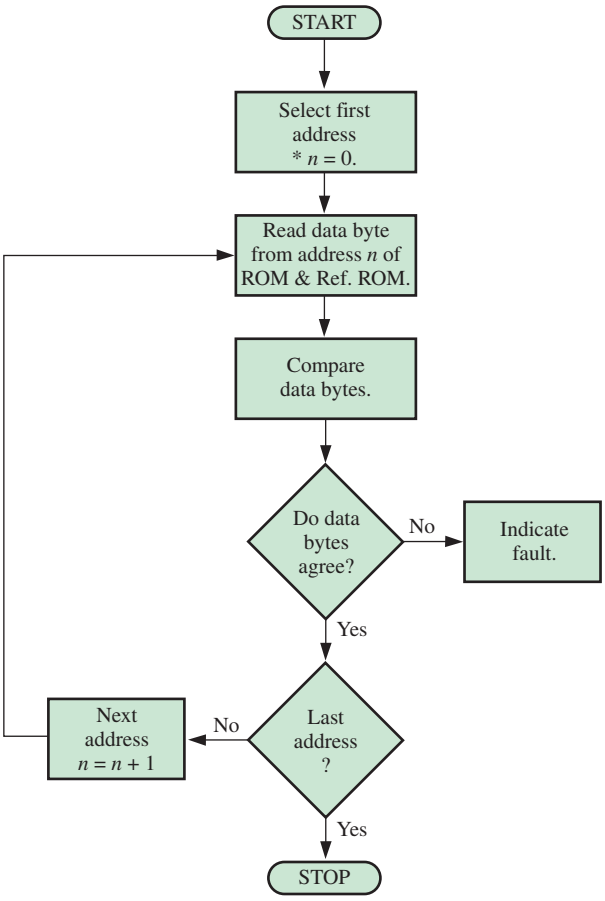


FIGURE 11–72 Block diagram for a complete contents check of a ROM.



* n is the address number.

FIGURE 11–73 Flowchart for a complete contents check of a ROM.

Checksum Method

Although the previous method checks each ROM address for correct data, it has the disadvantage of requiring a reference ROM for each different ROM to be tested. Also, a failure in the reference ROM can produce a fault indication.

In the checksum method a number, the sum of the contents of all the ROM addresses, is stored in a designated ROM address when the ROM is programmed. To test the ROM, the contents of all the addresses except the checksum are added, and the result is compared with the checksum stored in the ROM. If there is a difference, there is definitely a fault. If the checksums agree, the ROM is most likely good. However, there is a remote possibility that a combination of bad memory cells could cause the checksums to agree.

This process is illustrated in Figure 11–74 with a simple example. The checksum in this case is produced by taking the sum of each column of data bits and discarding the carries. This is actually an XOR sum of each column. The flowchart in Figure 11–75 illustrates the basic checksum test.

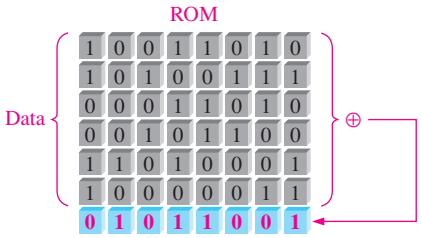


FIGURE 11–74 Simplified illustration of a programmed ROM with the checksum stored at a designated address.

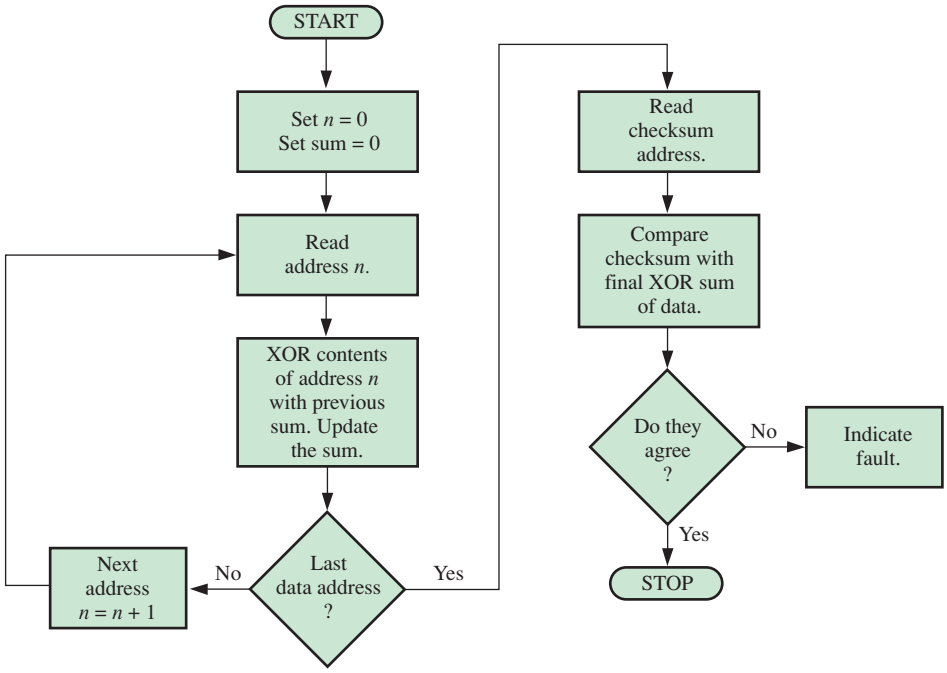


FIGURE 11–75 Flowchart for a basic checksum test.

The checksum test can be implemented with a special test instrument, or it can be incorporated as a test routine in the built-in (system) software or microprocessor-based systems. In that case, the ROM test routine is automatically run on system start-up.

RAM Testing

To test a RAM for its ability to store both 0s and 1s in each cell, first 0s are written into all the cells in each address and then read out and checked. Next, 1s are written into all the cells in each address and then read out and checked. This basic test will detect a cell that is stuck in either a 1 state or a 0 state.

Some memory faults cannot be detected with the all-0s–all-1s test. For example, if two adjacent memory cells are shorted, they will always be in the same state, both 0s or both 1s. Also, the all-0s–all-1s test is ineffective if there are internal noise problems such that the contents of one or more addresses are altered by a change in the contents of another address.

The Checkerboard Pattern Test

One way to more fully test a RAM is by using a checkerboard pattern of 1s and 0s, as illustrated in Figure 11–76. Notice that all adjacent cells have opposite bits. This pattern checks for a short between two adjacent cells; if there is a short, both cells will be in the same state.

After the RAM is checked with the pattern in Figure 11–76(a), the pattern is reversed, as shown in part (b). This reversal checks the ability of all cells to store both 1s and 0s.

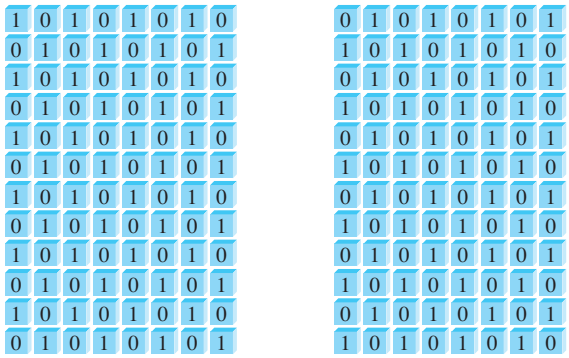


FIGURE 11–76 The RAM checkerboard test pattern.

A further test is to alternate the pattern one address at a time and check all the other addresses for the proper pattern. This test will catch a problem in which the contents of an address are dynamically altered when the contents of another address change.

A basic procedure for the checkerboard test is illustrated by the flowchart in Figure 11–77. The procedure can be implemented with the system software in microprocessor-based

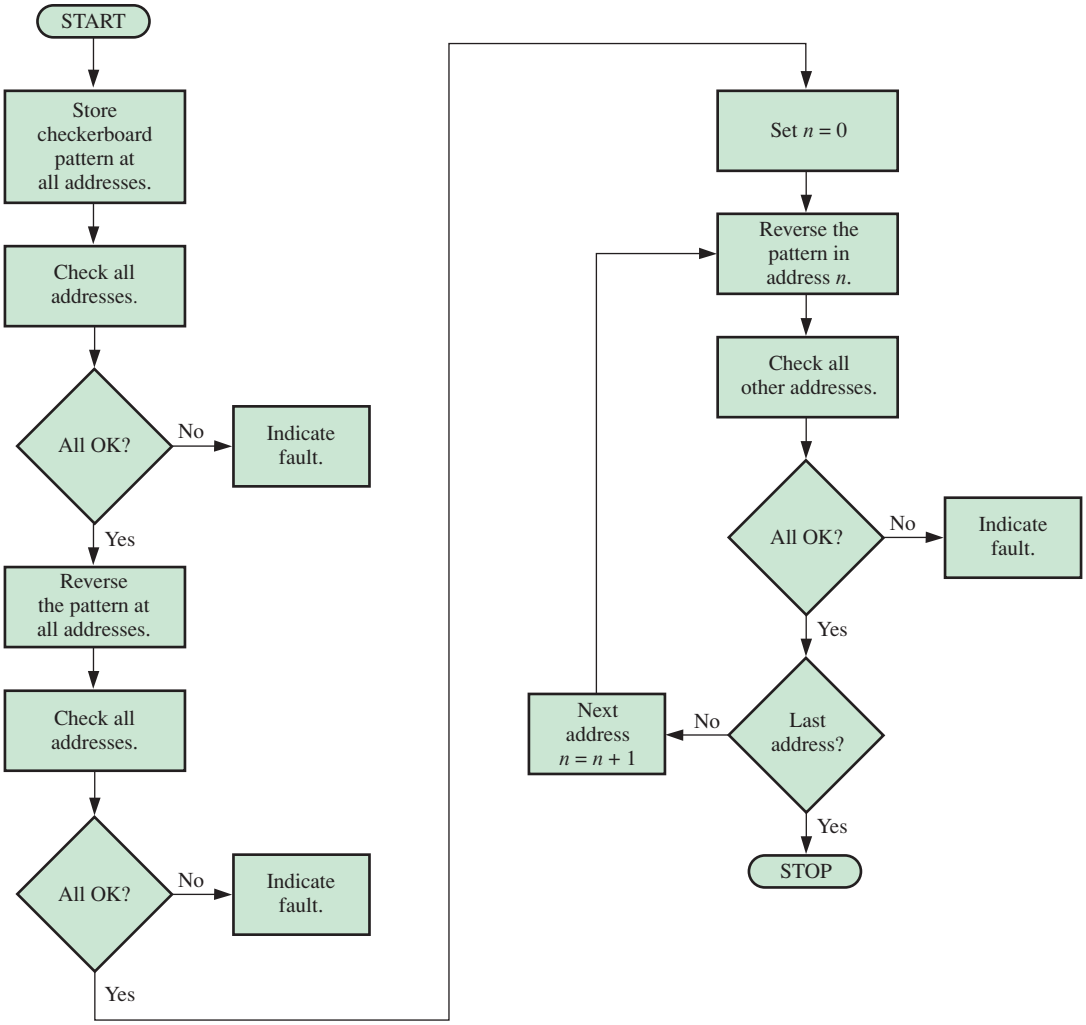


FIGURE 11–77 Flowchart for basic RAM checkerboard test.

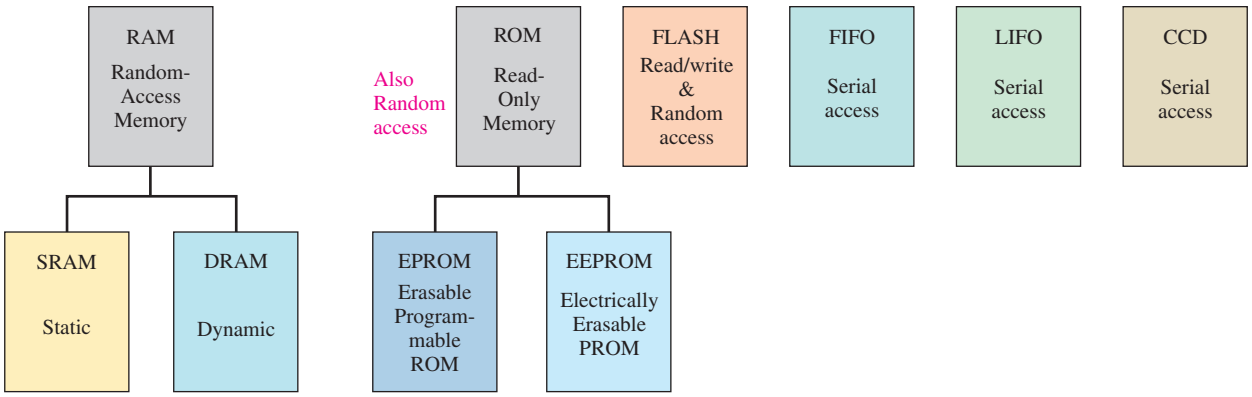
systems so that either the tests are automatic when the system is powered up or they can be initiated from the keyboard.

SECTION 11–11 CHECKUP

1. Describe the checksum method of ROM testing.
2. Why can the checksum method not be applied to RAM testing?
3. List the three basic faults that the checkerboard pattern test can detect in a RAM.

SUMMARY

- Types of semiconductor memories:



- Types of SRAMs (Static RAMs) and DRAMs (Dynamic RAMs):