

DIGITAL STORAGE

In this section, we will examine digital storage systems. We will begin with a brief look at the historical evolution of storage systems for digital computers. We will follow the evolution in an effort to explain the current state of such systems. This section will conclude with some predictions about future directions in digital storage.

Classification of Digital Storage Systems

There are several ways to classify digital storage systems. They may be classified by the format of information storage (i.e., serial or parallel), storage media (for example, magnetic disk, magnetic tape, compact disk, or semiconductor memory), duration of storage (temporary or permanent), proximity to the processing unit, and retrieval method, just to name a few. While we will touch briefly on all of these methods, our main focus will be on current techniques.

Modern digital computers are composed of three basic elements: a central processing unit, memory and input/output devices. The *central processing unit* (CPU) is the “brain” of the computer and performs logical and arithmetic operations on data in an order prescribed by the *program* it is executing. The *input/output* devices form the human interface to the computer. Examples of input/output devices include the familiar computer monitor, printer, mouse, and keyboard. The *memory* holds the set of instructions that form the program to be executed as well as the data to be processed and results. While there are several methods of arranging these components to form a computer, Fig. 1 shows a popular basic configuration. The view of a memory system as shown in Fig. 1, is very coarse. The memory system can be broken down into two basic classes as shown in Fig. 2. Here we differentiate between primary storage and secondary storage. We will consider primary storage to be relatively fast memory placed in close proximity to the central processing unit, while secondary memory is slower and is usually placed farther from the central processing unit. Another method of differentiating the two classes of storage is the data format. Almost all modern CPUs operate on instructions and data in a parallel format, that is, the internal structure of the system retrieves and operates on instruction and information in units of 8, 16, 32, or even 64 bits at a time. Primary storage is usually constructed in a format compatible with the internal *parallel* format or *word size* of the CPU. This allows the CPU to retrieve a complete instruction or datum in a single *read operation* from memory. This provides fast access but requires many physical paths between the CPU and memory (as many paths as the word size).

Primary storage is configured as a series of individually *addressable memory locations*, each location having an unique *memory address*. Each memory address accesses a set of n bits which is determined by the *word size* of the memory. The general structure of primary memory is shown in Fig. 3.

By contrast, secondary storage usually transfers information in *serial* form, that is, transferring one bit at a time

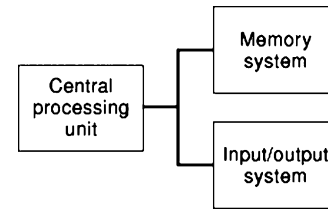


Figure 1. Generalized computer architecture.

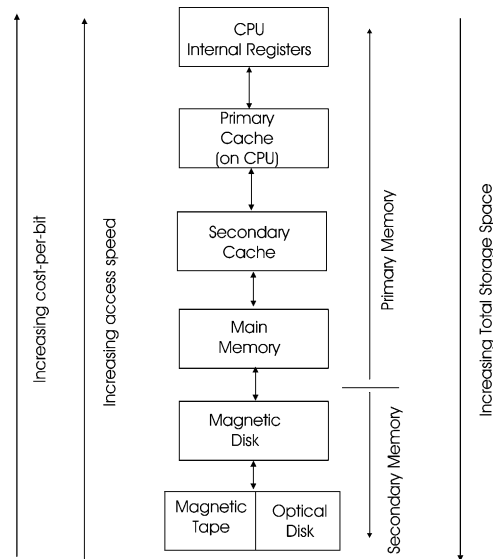


Figure 2. Hierarchy of digital storage systems.

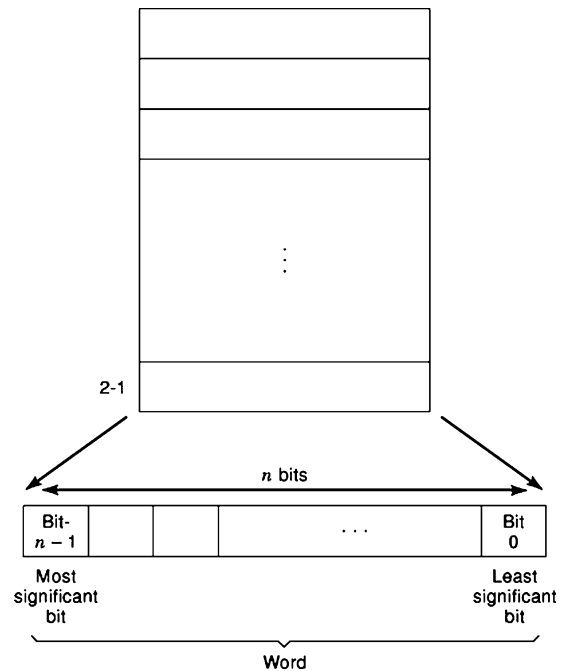


Figure 3. Primary storage layout.

over a single data line. This translates to slower transfer of information but also is less costly as fewer data paths are used.

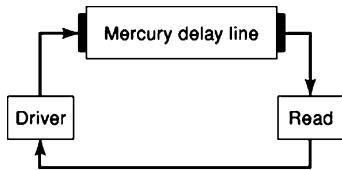


Figure 4. UNIVAC mercury storage tube.

A critical factor in the performance of a computer is the *memory access time* (that is, the delay incurred accessing the memory) and the *transfer rate* of data between the CPU and memory. If the memory access time and transfer rate are significantly slower than the CPU cycle time, the performance will suffer. It is beneficial to use the fastest possible memory in the system. Unfortunately, memory with low access time and high transfer rate is expensive; thus, to provide an affordable system, very fast memory is used in limited quantities. Slower storage is more cost-effective for large capacities. This is usually considered in the *cost per bit* of storage and will be addressed later. As shown in Fig. 2, a memory system is usually constructed as a combination of primary and secondary storage systems.

HISTORY OF DIGITAL STORAGE

To understand the current state and future directions of digital storage systems, one must look at the history of these systems. In this section, we will present a brief and by no means exhaustive history of the development of digital storage systems. For a more complete historical account, the reader is directed to Refs. 1 and 2. Early digital computers required two forms of storage: permanent storage for programs and data, and scratchpad memory for intermediate results. The most popular forms of early permanent storage were *paper tape* and *punch cards*. In both of these systems, holes were punched in paper media to represent instructions and input data. These paper media were fed through a mechanical reading device that was connected to the computer.

Mercury Tubes

Many methods were developed for nonpermanent storage in early computers. One of the more interesting early forms of digital scratchpad memory was used in the 1940's UNIVAC computer shown in Fig. 4 (3). Here mercury tubes were constructed with acoustic transducers at each end. At the transmitting end, data bits were sequentially sent into a horizontal column of mercury. At the opposite end, transducers would convert the acoustic wave back into data bits. The delay between the two ends was regulated in such a way that a fixed number of data bits could be stored in the tube. Each tube was designed to hold ten 91 bit words. A complete system consisted of 100 mercury tubes for a total storage capacity of a whopping 91,000 bits (or approximately 12 kbytes). The mercury tube represents an early form of digital storage with serial access. A picture of an actual mercury memory tube is shown in Figure 5.

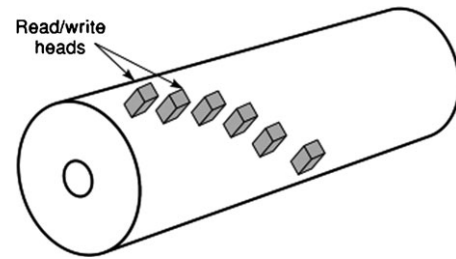


Figure 5. Magnetic drum storage system.

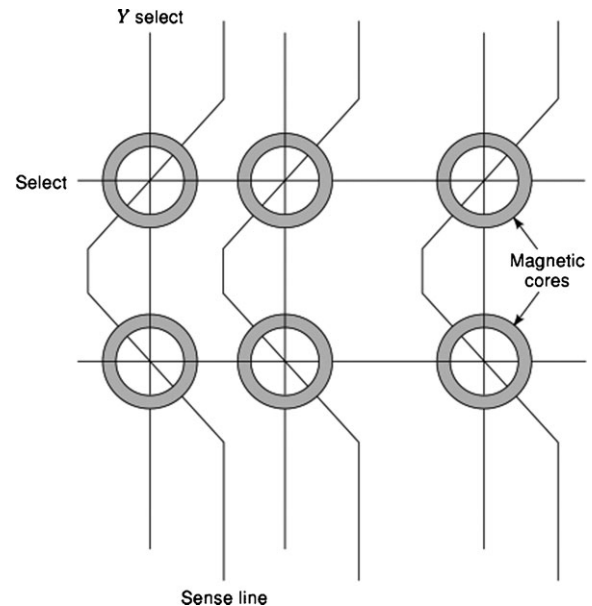


Figure 6. Magnetic core memory.

Magnetic Drums

Over the ensuing years, several other technologies were developed for intermediate storage. An interesting technological development involved the use of a revolving drum coated with a magnetic substance. Several read/write heads were placed around the drum and information could be written to or read from the revolving drum as the appropriate region passed under a head. A simplified version of a magnetic drum is shown in Fig. 6. The major advantage of this type of memory was the nonvolatile nature of the storage, that is, information would remain on the drum even if the power were removed. Extensions of this technology led to tape storage systems and the modern disk storage systems (see section 3.2). A picture of the drum portion of a magnetic drum is shown in Figure 7.

Core

In the mid-1940s, work had begun on using the hysteresis properties of ferromagnetic loops in storage systems. In these systems, round loops of magnetic material were placed at the intersections of a two-dimensional matrix of crossing wires. The loops were arranged so that the *X* wire and *Y* wire passed through the loop (see Fig. 8). The material for the loops was chosen for its square-hysteresis properties, that is, it would remember the direction of magne-

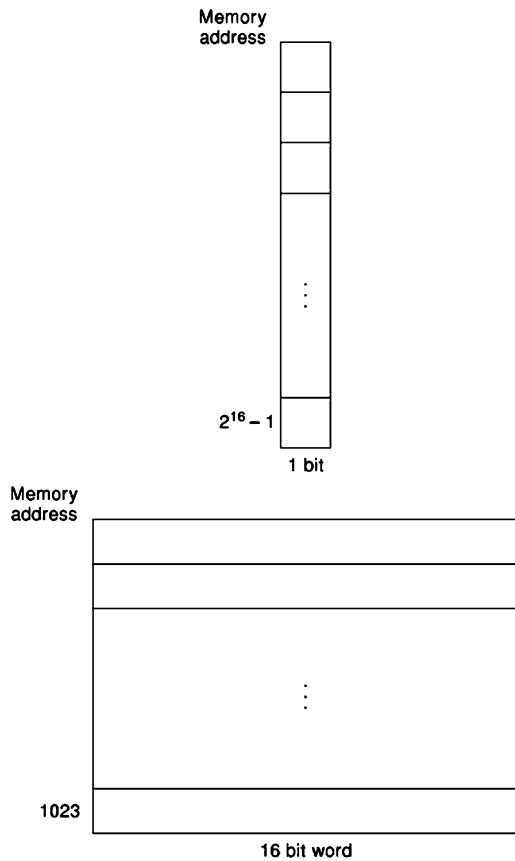


Figure 7. Two formats for a 16 kbit memory device.

tization. If a current were passed through the *X* or *Y* wires separately, the magnetic field created would not be sufficient to change the direction of magnetization of the loop. If, however, both the *X* and *Y* wires were energized in such a way as to create an aiding magnetic field, then the direction of magnetization of the loop would match the field. One of two conditions would occur: if the loop's magnetization matched that of the field's, nothing would happen. If the direction of magnetization was opposite to that of the field's, the loop's magnetization would "flip." Each loop had a third wire (sense wire) passing through it to allow the system to detect whether the loop's magnetic field had flipped or not when energized. In these systems, the energizing of a particular *X* wire and a particular *Y* wire selected one loop in the matrix. If several matrices were used, then multiple bits could be accessed simultaneously. This gave rise to parallel memory access in which the number of matrices used determined the *width* of the memory word. It is also of interest to note that the memory words could be accessed in *random* order, which gives rise to random access memory (*RAM*).

Introduction of Transistors

Late in 1947, the first transistor was developed at Bell Telephone Laboratories (3, 4). Since that time, the transistor has been under almost constant development. The evolution of transistor design and miniaturization has been an important factor in the development of modern digital stor-

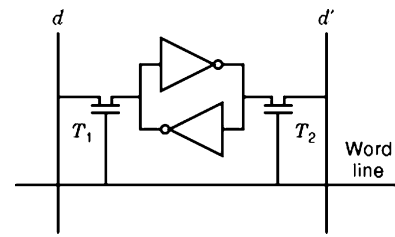


Figure 8. MOS SRAM cell.

age systems. Circuits using transistors as storage elements quickly began to replace memories using magnetic cores. The continued decrease in transistor size has allowed more and more storage elements to be placed on one device.

PRIMARY STORAGE

Semiconductor Memory

Today, the most widely used form of primary storage is readable/writable semiconductor memory. These memory devices are used to provide volatile storage, that is, stored information is lost if the power to the device is removed.

A memory device (chip) is characterized by several factors: fabrication technology, power consumption, volatility, size, and width. The *size* of a memory chip designates how many individual storage cells are present. For example a 16 kbit device would have 2^{14} or (16×1024) individual storage cells (*bits*). The *width* of the device refers to how many bits of information are read or written simultaneously. For example, a $(1k \times 16)$ and a $(16k \times 1)$ device both have the same number of storage cells. In the $(16k \times 1)$ device though, 16 bits of information are retrieved or written per access (see Fig. 7).

Volatile Storage

Two basic fabrication processes are used: bipolar junction transistor technology and metal-oxide semiconductor (*MOS*) technology. The characteristics of bipolar technology include high speed (fast retrieval), high power consumption, and transistors that are relatively large. *MOS* transistors, on the other hand, are very small (and thus many can be placed on a device), have low power, and are slower than bipolar transistors. *MOS* technology has the added advantage that it can operate at low voltages. *MOS* transistors are fabricated using either an *n-channel* or *p-channel* method, depending on whether the impurity used to dope the silicon base material provides an excess or deficiency of electrons. A variant on *MOS* technology is the complementary metal-oxide semiconductor (*CMOS*) technology in which both *n-channel* and *p-channel* transistors are fabricated on the same device.

In some instances, the advantages of both bipolar junction transistor and *MOS* technologies are combined (*Bi-MOS* technology). This process takes advantage of the high packing density and low power consumption of *MOS* transistors for storage cells and the high-speed characteristics of bipolar transistors for transferring data to/from the memory device.

Random access memories fall into two general categories: static random access memories (*SRAMs*) and dynamic random access memories (*DRAMs*). The generalized structure of a MOS SRAM memory cell is shown in Fig. 8. Here a bistable circuit is formed by the feedback of two inverters. The cell (or group of cells forming a particular word) is selected when the word line is set to a logic one (1) and the two transistors (T_1 and T_2) are in the on state. This places the current state of the cell on the data line d and its complement d' . Thus a read operation only involves selecting the appropriate cells and is very fast. For a write operation, the internal state of the cell is set to either a 1 or 0 by placing the desired value and its complement on the data lines and then selecting the cell. We note that if the cell is not selected, the internal state of the cell will not change. In addition, the cell will constantly be using power to maintain its state. The internal structure of a typical memory cell requires six transistors. SRAMs are characterized by being very fast and not very large in terms of total storage capacity.

An important advantage of MOS technology is the ability to fabricate very small and simple capacitors. The capacitor can be used as a storage cell by the representing a 1 or 0 by the presence or absence of a charge. A very simple storage cell can be fabricated with a capacitor and a single transistor as shown in Fig. 9. These simple cells can be made extremely small and many can be packed onto a single chip. To write information to a cell, the word line is set to logic 1 so that the transistor T_1 is in the on state. The bit line is then used to either supply a charge to the capacitor or to discharge it. Once T_1 is in the off state, the charge will remain on the capacitor. The capacitors used are exceedingly small, on the order of femtofarads (10^{-15} F), and natural leakage of charge will cause them to discharge over a period of time. To prevent the loss of information in these types of cells, the contents must be periodically refreshed. This gives rise to the name *dynamic* random access memory. Read operations are somewhat different in DRAMs than SRAMs. The use of capacitors for storing information means that the level of charge must be measured. In a typical DRAM device, the charge on the capacitor of the selected cell is compared to a reference cell (with a charge usually set halfway between 0 and 1) by the *sense amplifiers*. This action discharges the cell's capacitor and, if a 1 was present in the cell, the capacitor must be recharged to a 1 at the end of the read cycle. Thus, access to information in a DRAM will be slower than a SRAM and some time must elapse between successive accesses to a DRAM to allow for refreshing and for restoring values after a read. Despite these limitations, DRAMs provide storage capacity 4 to 6 times greater than SRAMs in the same physical space on an integrated circuit.

Nonvolatile Storage

In the previous section, we examined volatile forms of primary storage. There are several classes of nonvolatile storage in which information is retained even if the power is removed from the device. A read only memory (*ROM*) device has its contents permanently set and can only be read, not written. These devices are used for information that

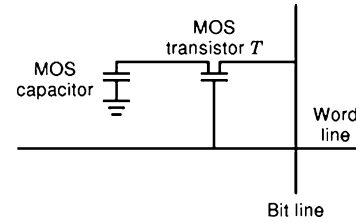


Figure 9. MOS DRAM cell.

does not change or must be present when the system is turned on. One example is the starting process required by personal computers (*PCs*). The contents of ROMs are set during the manufacturing process (mask programmable ROMs). Programmable read only memory (*PROM*) devices can have their contents set by placing them in a special programming station. In these types of devices, the presence or absence of tiny “fuses” determines the contents. The programmer uses an electrical signal to remove the fuses in the appropriate locations.

Another form of nonvolatile storage is the erasable programmable read only memory (*EPROM*). These devices act like ROM devices but their contents may be changed (usually in a special programming device). Most EPROM devices use exposure to intense ultraviolet light to erase the contents before they can be rewritten. Others such as electrically erasable programmable read only memory (*EEPROM*) devices use a special signal and/or voltage to erase the contents. A more recent advance is FLASH memory. The term *flash* was coined by Toshiba to indicate that it could be erased “in a flash.” FLASH memory is a derivation of EEPROM technology and can have its contents changed while it is installed in a system.

Cache Systems

In our discussion of SRAM and DRAM technology, we observed that SRAMs were faster than DRAMs and did not require refresh mechanisms but required significantly smaller storage size. In a computer, the CPU must retrieve information from the memory at least once and usually several times during the execution of an individual instruction. The memory latency (time to access the memory and minimum time between memory accesses) has a direct effect on the overall performance of the system. If we analyze the execution of a typical computer program, we observe that much of the execution time is spent on procedures or routines where a relatively small number of instructions are executed repeatedly. This is known as the *locality of reference*. If the groups of instructions that are currently being executed could be placed in a small amount of very fast memory close to the CPU, then the overall performance of the system can be greatly improved. This small amount of very fast memory is known as *cache*. In Fig. 2, the position of cache in the hierarchy of digital storage systems is shown.

There are many design issues that must be decided when implementing a cache. The total cache memory is usually divided into a number of fixed-sized blocks. The principle of using cache is simple. Consider the case in which the CPU generates a request to read a particular

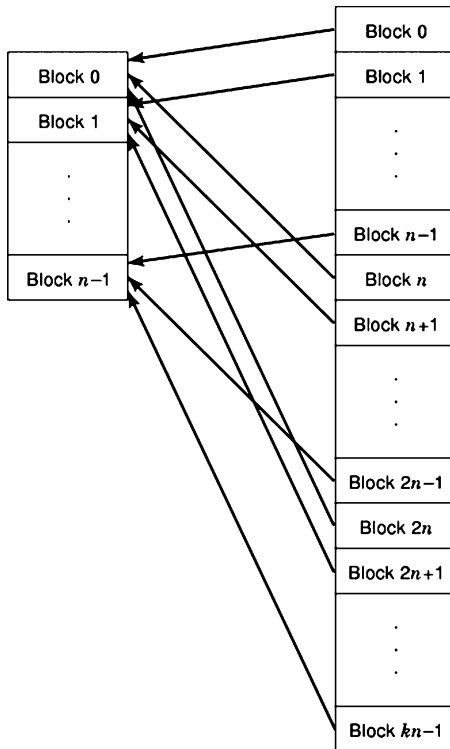


Figure 10. Direct mapped cache system.

location in main memory. The system controlling the cache will determine whether the block containing the target location is in the cache or not. If it is, the contents of the cache are used (this is known as a *cache hit*), if not, the segment containing the target location is transferred from the main memory to the cache (*cache miss*). In some instances, the newly loaded block may replace a block already in the cache. In most systems, the main memory (DRAM storage space) will be many times larger than the cache space. A method to determine how to map the main memory into the cache blocks must be determined. A very simple method is to have memory blocks map to a fixed block in the cache (*direct mapping*). This may be done by simply using the least significant bits of the main memory address as shown in Fig. 10. While this method is simple, it may result in a cache block being replaced when the cache is not full. A better method, known as *fully associative mapping*, is to allow main memory blocks to be mapped to any cache block. This ensures that the cache must be full before replacement begins. Two complications arise from this method, however. First, the complexity of the cache management system is increased, as it must be able to determine quickly if the target location is currently in cache and where it is. Second, an algorithm is required to determine which existing cache block is replaced if the cache is full. Most systems use a compromise between the two systems in which each main memory block can map into a set of cache blocks. This is known as *set-associative mapping*.

Up until this point, we have considered only read operations. In the event of a write operation of a location currently in the cache, a difference will result between the version of that location in main memory and cache. One

solution is to update main memory whenever a write occurs (a *write-through* operation). If a variable is updated frequently, this may cause a significant slowdown in performance. An alternative is to keep track of any writes to cache and mark the location as *dirty*. The main memory version will only be updated when the cache block is replaced.

The size of the block will have an impact on overall performance. If the blocks are too small, a code segment may not fit into one block; if the blocks are too large, then there may not be enough blocks and many replacements will be required. In addition, in the case of a cache miss on a read, large blocks will require more time to transfer from main memory to cache.

The actions of the cache management system on a cache miss can also affect overall performance. If a cache read miss occurs, the system pauses while the main memory block is transferred into the cache. This may cause a significant slow down of the system. Another method is to allow the contents of the target location to *load through* directly to the CPU. A similar situation occurs on a cache write miss. Instead of loading the block containing the target address into the cache, modifying it, and then writing the update, the write operation is passed directly to main memory.

On-Chip Cache. As the speed of processors and memory increases, the physical distance between the CPU and memory has more impact on performance. As shown in Fig. 2, many modern high-speed processors allocate some space on the CPU device itself for level 1 (L1) cache. This cache is very high speed and the close proximity to the CPU improves overall performance. A tradeoff occurs in the design process between minimizing the overall size of the device to reduce cost and improve performance and maximizing the amount of onboard cache.

Secondary Cache. Secondary cache or level 2 (L2) cache, is usually much larger again than the primary cache and physically logically resides between the CPU cache and main memory. The use of two cache stages increases the complexity of the overall system, as cache management systems must be replicated in both. The improvement in performance, however, justifies this additional cost. Further levels of caching are also used in some systems to improve performance.

Separate Data and Instruction Caches. Another method of improving performance in primary storage is to use separate caches for data and instructions. The sizes of total cache and cache blocks can then be tuned individually for better performance. In many systems, physically separate data paths to each cache are provided to allow simultaneous access to data and instructions.

Interleaving. In our discussion of DRAMs, we noted that there is a delay in retrieving the contents of a target location once an access is started and there is a minimum time between sequential accesses to the memory device. In a system employing caching, the primary activity is transferring blocks of information to/from main memory. The

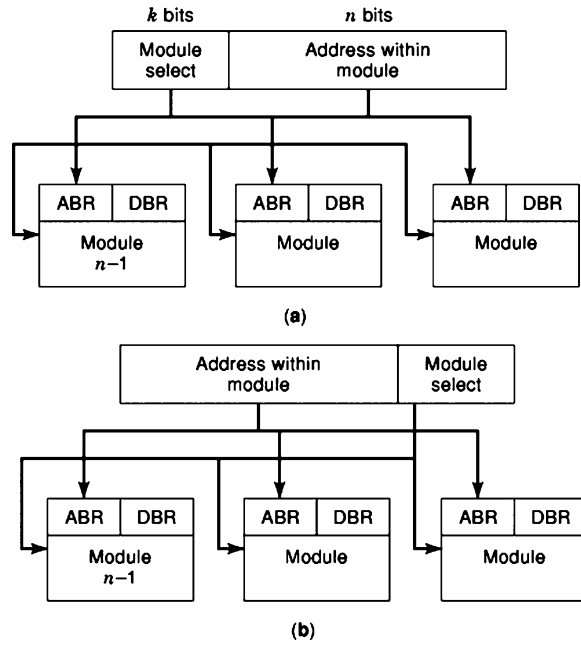


Figure 11. (a) Memory system using consecutive words in each module. (b) Consecutive words in consecutive modules.

blocks transferred are the contents of sequential memory locations. Figure 11(a) shows a main memory system composed of multiple memory devices. The most significant bits of the main memory address are used to select a particular memory chip, while the least significant bits select a location within that chip. In this system, a block transfer from the main memory will require sequential accesses to the same device. An alternative is to use the main memory address lines as shown in Fig. 11(b). Here, the least significant bits of the address select a particular memory chip. If we look at two sequential main memory addresses, they will reside on separate devices. In the case of a main memory formed from n chips, up to n transfers could be started without having to wait for the memory devices to complete their cycles. The drawback with this design is that all of the address space must be filled with memory and may not be practical.

Memory manufacturers also produce DRAMs specifically to work in cached systems. These devices are designed specifically to transfer sequential memory locations quickly. Some devices incorporate serial shift registers to allow “bursts” of data to be transferred. For example, enhanced DRAM (*EDRAM*) or cache DRAM (*DRAM*) incorporates a small SRAM cache into the DRAM memory. The SRAM provides high-speed access to data and can be used in block transfers of information. Double Data Rate SDRAM's (DDR-SDRAM) use two interleaved memory banks and transfer data on both edges of the clock. This effectively doubles the speed of transfer for blocks of sequential memory locations.

In most systems, the CPU and memory work asynchronously. For example, to read a certain location in memory, the CPU places the address of the target location on the address lines and issues a read command. It must then wait for the memory to access the information, place the

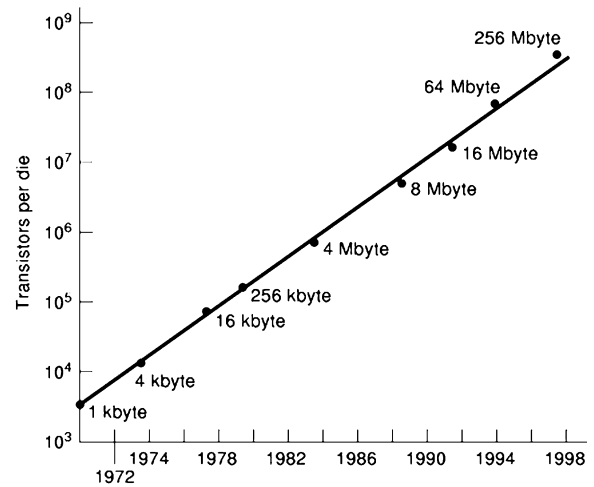


Figure 12. Evolution of DRAM capacity.

data on the data lines, and then signal the CPU that the data are ready. This may require several clock periods during which the CPU must halt processing (*wait states*). An alternative is the use of synchronous DRAM (*SDRAM*). In SDRAM, the memory's operation is controlled by an externally applied clock. This clock is derived from the CPU's clock in such a way that information is exchanged without having to wait for additional memory cycles.

Performance Calculations: Cache Hit Rate and Miss Penalty. Ideally, the active program segment and target data would always be found in cache memory. In this case, the performance of the system would be determined by the speed of fast SRAM. The fraction of time over which this occurs is known as the cache *hit rate*. If a cache miss occurs, the additional time required to fetch or write the required information from outside of the cache is known as the *cache miss penalty* (this is the time that the CPU is unable to continue processing). The average access time for a system with only one level of cache, can be approximated by

$$t_{av} = ht_c + (1 - h)t_m$$

where h is the cache hit rate, t_c is the cache access time, and t_m is the miss penalty time. If two levels of cache are present, the average access time can be calculated as:

$$tav = h1c1 + (1 - h1)(h2tc2) + (1 - h1)(1 - h2)tm$$

where $h1$, and $h2$ are the hit rates for the L1 and L2 caches respectively and $tc1$ and $tc2$ are the access times.

Current Memory Capacities

The number of storage cells per memory chip for DRAM has been used for many years as an indicator of the state of memory evolution (5–8). In Fig. 12, we show the historical evolution and projections for DRAM memory size. It is interesting to note that the size of DRAM has approximately doubled every 18 months (as predicted by Moore's law, discussed later).

There has been a similar trend in SRAM; however, the number of memory cells per chip is significantly lower than that in DRAM chips.

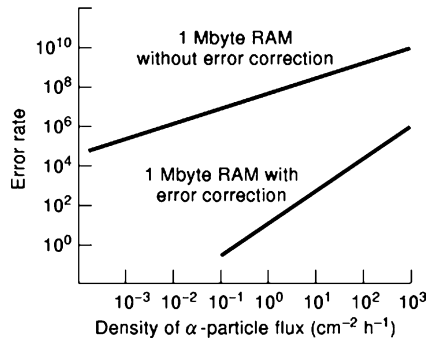


Figure 13. Comparison of memory-chip error rates with and without error-correcting codes.

Error-Correcting Codes

Memory devices are subject to errors that affect the integrity of their contents. Errors are classified as permanent or *hard* errors, for example, a damaged memory cell, or random (*soft*) errors. Soft errors may be the result of electrical noise in the circuit in which the memory device operates or may be caused by various forms of radiation. In particular, naturally occurring α particles can cause a significant number of bit changes in a memory chip. As the density of memory devices increases (that is, the cell size decreases) sensitivity to noise and other naturally occurring faults also increases. To compensate, many memory devices incorporate extra bits in each word to allow for the detection and correction of errors.

The theory of error-correcting codes is very rich, for example, see Ref. 9. In a very simple form, if the system is designed to correct k errors in m bits of data, then at least $2k + 1$ *check bits* must be added to each word. For example, if we wish to detect and correct up to 2 errors in a system with 16 bits of data per word, then each location will have $16 + 2 \times 2 + 1 = 21$ bits. In practice, additional bits are added to permit the detection of more errors. The effectiveness of error-correcting codes in memory devices is shown in Fig. 13 (9). Today, Hamming codes are the most common type of code used for memory error detection and correction.

SECONDARY STORAGE

Secondary storage generally refers to devices that provide large quantities of relatively inexpensive nonvolatile storage. Most secondary storage systems use a serial format for storing data. Figure 14 gives a comparison of the relative costs of storage for various storage methods (10). In addition to cost, speed of access varies among the different methods.

Magnetic Tapes

Magnetic storage systems are all based on the same principle. A coating that is magnetizable is used in all systems. This coating has the property that its magnetic orientation can be set by passing it under a small electromagnetic field generated by the *write head*. Once set, this orientation will remain for a very long time or until it is changed by another

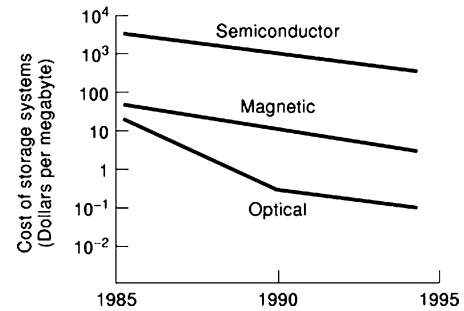


Figure 14. Comparison of cost per unit of storage.

write operation. The orientation of the magnetic field can also be determined by a *read head*. If the magnetized coating is passed under a coil, it will produce a small current in the coil with a polarity relative to the polarity of the magnetic field. This can be done many times without affecting the magnetic properties of the coating.

Magnetic tape systems use a process very similar to sound-recording techniques. A thin flexible tape is coated with an oxide with suitable magnetic properties. This tape is passed under one or more heads that can read from or write to the tape. Older systems used tapes on open reels, while more modern systems use tapes enclosed in special cartridges that minimize damage due to handling. Depending on the width of the tape and the type of material used, the tape may contain one or more parallel *tracks*. Each track will have an associated read/write head. Data on a tape are organized as a series of *records* that are a series of contiguous blocks. Magnetic tape drives provide *sequential access* to information, that is, to access a particular record, the tape must be moved forward or backward until the start of the record is located under a read head. Once the record is located, the data are read from the tape in a continuous fashion. The delay to find the start of a record will be dependent on the maximum physical transport speed of the system, the length of the tape, and the distance from the start of the tape. Magnetic tapes are generally the slowest of the secondary storage systems and are usually used for backup purposes.

Magnetic Disks (Hard Disks)

The principles used in the modern magnetic disk are very similar to those used in the magnetic drum described earlier. In a magnetic disk, a circular platter made from a hard, stable material is coated with a substance with suitable magnetic properties. The disk is spun at a very high speed. Above the disk, one or more read heads are mounted on a rigid arm. The coated surface of the disk is divided into a series of concentric rings or *tracks* as shown in Fig. 15. Each track is subdivided into a series of *sectors*. The size of each sector is generally fixed. There are many different structures that can be used in building hard disks. Disks can contain a single platter or multiple platters. The platters may be coated on one side or both. There may be one moveable read/write head or there may be a fixed head for each track.

To access information on a disk, the proper track and sector must appear under the read/write head. Access time

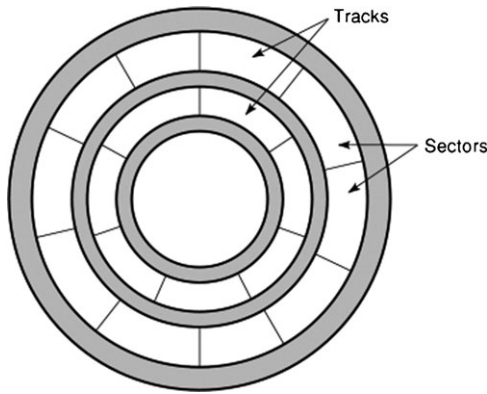


Figure 15. Surface layout of a magnetic disk.

will depend on the type of disk (moveable or fixed head), rotational speed of the disk, and position of the disk when the request was made (i.e., the time required for the target track to get).

The storage capacity of a disk will depend on a number of factors as well. These include diameter of the disk, type of coating used, and track size as determined by the width of the read/write head. The density of storage on a disk is also important. This is determined by the magnetic properties of the material used and the distance between the disk and read/write head. Manufacturers are continually trying to improve the magnetic coatings and to decrease the head to disk distance. Most modern high-density hard disks are sealed to prevent any foreign material from entering. In these devices, the head to disk distances are extremely small. In a *Winchester* disk, the read/write head and slightly flexible arm assembly have an aerodynamic shape. The head actually rests on the disk's surface when it is stopped. When the disk is rotating at high speed, the resulting air movement near the disk's surface creates lift that holds the head a very small distance above the disk. As shown in Fig. 16, hard disks provide the fastest access in secondary storage systems (10).

Floppy Disks

Hard drives are usually mounted within a machine (personal computers) or an enclosure, provide large amounts of storage (tens or hundreds of gigabytes), and are not moveable from machine to machine. Floppy disks, on the other hand, use a flexible material coated with a magnetic material to form the platter. This is enclosed within a plastic shell and is designed to be removable. Another difference is that the read/write heads are actually in contact with the disk when a read or write operation is occurring. To prevent damage, the rotational speed of floppy disks is relatively slow so access time is higher than hard disks. Density and thus storage space are also limited (typically under 2 Mbytes per disk).

Compact Disk–Read Only Memory

The compact disk (*CD*) was introduced in the early 1980s as a method of digital music storage. The disks were relatively inexpensive plastic platters with a reflective metallic coating on one side. Information is stored in binary form as

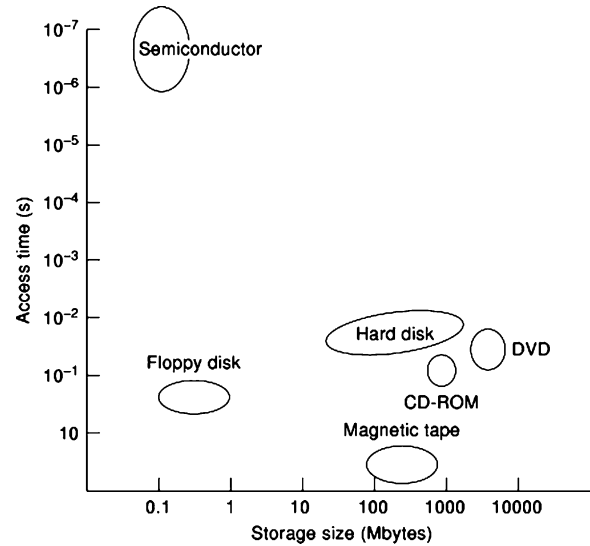


Figure 16. Comparison of access speed and size for digital storage systems.

a series of extremely small undulations (*pits*) in the metallic surface. The pits on the rotating platter form readable interference patterns when a low-power laser is focused on the surface. Like a hard disk, the surface of a CD is divided into tracks and sectors. Hard disks rotate at a constant rate (constant angular velocity). Near the center of a disk, the rate at which the surface coating passes under a read/write head is lower than near the outside of the platter. By contrast, the CD changes its rotational speed depending on which track is being read. Near the center, the rotation is increased so the linear rate of surface passing under the laser is constant (constant linear velocity). This requires a more complex speed control mechanism in the CD reader but the reading mechanism is simplified. There are several forms of CD's in use today. A compact disk–read only memory (*CD-ROM*) is very similar to music CDs. To create a CD-ROM, a special writer is used with which a high-power laser is used to create pits on a mastering disk. This disk is then used to make a die for stamping out multiple copies. The second form are write-once CD's that can be written on the user's computer. These readers/writers have a laser whose output can be varied: low power for reading and a higher power for writing the disk. Once written, the data cannot be erased. The third kind are re-writable CD's which use a special material with a crystalline structure that will melt when heated with the writing intensity of the laser. The contents of the disk can be erased and re-written by the same process.

CD's can store approximately 700Mbytes on a relatively inexpensive disk.

A more recent development is the Digital Versatile Disk (DVD). Like CD's, they were originally used for entertainment purposes (primarily, digital video). Like CD's, they were adapted for data storage. DVD-R is a write once technology where DVD-RW are re-writable. The advantage of DVD technology is the increase in storage capacity (up to 4.7 Gbytes per disk). Also available are DVD's with multiple layers. A double layer DVD can store over 8.5 Gbytes.

CD and DVD readers/writers current use laser in the red region of the light spectrum (640 nm wavelength). New systems are being developed with lasers in the much shorter blue region of the spectrum at 405 nm. This shorter wavelength allows a much higher data density on the same size disk. Capacities of 25, 50 and 100 Gbytes per disk are available for these systems.

Performance Calculations

The performance calculations for disk type systems is determined by several factors:

- Read/write head positioning time to access the target track
- Rotational time to reach the target sector on the track
- Rotational speed and data density (this determines the data transfer rate)

The average access time for a disk device can be calculated as follows:

$$t_{\text{access}} = t_{\text{seek}} + t_{\text{lat}}$$

where t_{seek} is the time required to position the read/write head over the target track and t_{lat} is the rotational time required for the target sector to appear under the read/write head.

COST-per-BIT Comparison

Now that we have explored various forms of primary and secondary storage, the relative cost per bit of storage and the access time or bandwidth of the memory system influences the relative amount of each. In Fig. 14(a) comparison of the relative cost and access times for various forms of storage is presented (10). The overall design of a computer system will be determined by many factors. The memory storage system will be a compromise between performance requirements and budget. For example, today, 1Gbyte of Ram can be purchased for approximately \$100 or \$0.1/Mbyte. A 250 Gbyte Hard drive is about the same cost but only \$0.0004/Mbyte. A package of 100 DVD's can be purchased for about \$25 resulting in a cost of only \$0.00006/Mbyte.

VIRTUAL MEMORY

A requirement of most computer systems is that the instructions and data currently being used must reside in main memory. In many instances, a program is larger than the computer's main memory capacity. In such cases, the active segment of the program will reside in the physical memory while the rest of the program will reside in secondary storage. Most computer systems require an operating system (OS) to work. The operating system is simply a program that is constantly being run and manages the operation of the whole computer system. If programmers were to write a large program for a machine without an OS, they would have to be aware of how much main memory the machine had and ensure that the appropriate segments of code and data were in the memory when

required. The operating system simplifies this task (11, 12). The programmer writes the program (usually starting at the very beginning of memory) as if there were no restrictions on the amount of memory available. The memory used by the program is called *virtual memory* and the addresses used within the program are called *virtual addresses*. When the program is run, the operating system will load the currently needed segments of the program into *real memory* and must translate the virtual addresses into real addresses. Using a method similar to that used in cache systems, the program is broken up into a number of *pages*. Each page contains a fixed number of words. A program, then, will occupy one or more pages. The virtual address is broken into two parts: the most significant bits determine the page number while the least significant bits are used to determine the *offset* or distance from the beginning of the page of a particular address. Real memory is also divided into blocks equal in size to the pages. The operating system maintains a *page table* that is used to keep track of what segments of the program are currently in real memory. When the CPU requests a particular virtual address, the operating system must determine whether or not it is in the real memory. If it is, the virtual address is translated into an address in real memory and the transfer proceeds. If target address is not currently in real memory, then the page containing the required address must be retrieved from secondary storage. This may require the replacement of some other page of the program in real memory. Issues of which page of real memory to replace are similar to the cache block replacement problem.

MOORE'S LAW

In April 1965, Gordon E. Moore (then the head of research at Fairchild Semiconductor), observed that the complexity of integrated circuits as measured by the number of transistors in one device had roughly doubled every year since 1959. This trend in technology is known as *Moore's law* (5, 13). This trend continued until the late 1970s at which point doubling occurred every 18 months. Since that time, the doubling rate has been almost constant. The number of transistors on an integrated circuit is determined by three factors: the linewidths, the size of the die used and the design of the individual transistors. Moore's law is a fairly accurate reflection of the state of integrated-circuit developments.

COMPARISON OF MEMORY SPEED AND PROCESSOR SPEED

One factor not captured by Moore's law is the issue of performance increase. The overall performance of a computer system is determined by a number of factors. One important factor is the rate at which the CPU can step through the operations required to execute an instruction. In most computers this is determined by the *clock rate* of the CPU. In the past few years, clock rates have increased dramatically, from a few million clock cycles per second to almost one-half billion clock cycles per second. A second important factor in determining overall performance is the ac-

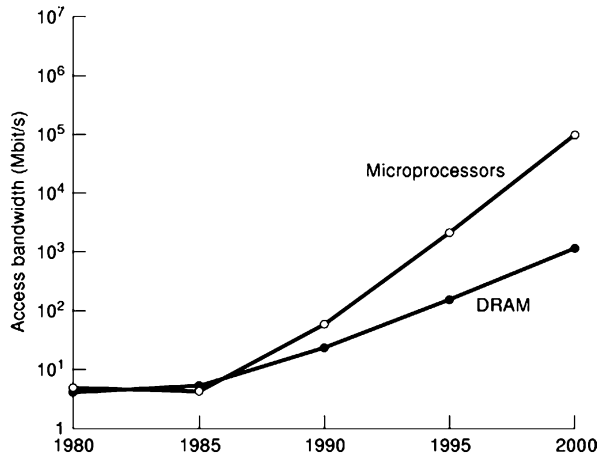


Figure 17. Comparison of memory and microprocessor speeds.

cess time for memory. In the execution of an individual instruction, the memory must be accessed at least once and possibly several times. If memory access times had been decreasing at the same rate as CPU clock speeds were increased, performance measures would have increased at the same rate. This however, has not been the case. Memory access times have decreased somewhat, but the gap between CPU requirements and memory has widened over the years (14). In Fig. 17, we show the trend in memory and CPU developments over the past few years.

DEVELOPING TECHNOLOGIES

Almost all of the main memory systems manufactured today are based on silicon integrated circuits. An inherent trait of silicon is that the higher the switching frequency (switching rate), the higher the power consumption. Gallium arsenide has been used in semiconductor electronics for many years. In the past, silicon was favored due to its relative ease of production and processing. Gallium arsenide, on the other hand, is based on a compound and until recently has only been used in relatively simple semiconductor devices. There are two major advantages to gallium arsenide: switching times are much faster than silicon transistors, and the power used by a transistor is independent of switching frequency. Thus, very-high-speed, low-power devices can be fabricated. The use of gallium arsenide for DRAMs seems impractical due to high internal leakage currents. There is, however, a great deal of research focused on the design and manufacture of gallium arsenide SRAMs (15).

CR-ROMs are currently limited to approximately 650 Mbyte of storage. A recent development has been the introduction of digital versatile disks (DVD). DVD ROM disks are expected to have capacities around 17 Gbyte and to be much faster than current CD-ROM drives.

BIBLIOGRAPHY

1. J. P. Eckert A survey of digital computer memory systems, *Proc. IEEE*, **85**: 184–197, 1997, reprint of Oct. 1953 article.

2. A. Burks Electronic computing circuits of the eniac, *Proc. IEEE*, **85**: 1172–1182, 1997, reprint of August 1947 article.
3. M. Riordan L. Hoddeson The origins of the *pn* junction, *IEEE Spectrum*, **34** (6): 46–51, 1997.
4. M. Riordan L. Hoddeson Birth of an era, *Sci. Amer.*, special issue, Solid State Century, 10–15, 1997.
5. L. Geppert W. Sweet Technology 1998 analysis and forecast, *IEEE Spectrum*, **35** (1): 19–22, 1998.
6. L. Geppert Solid state (development forecast), *IEEE Spectrum*, **34** (1): 55–59, 1997.
7. Y. Patt *et al.* One billion transistors, one uniprocessor, one chip, *IEEE Comput.*, **30** (9): 51–57, 1997.
8. K. Kim C. Hwang J. Lee DRAM technology perspective for gigabit era, *IEEE Trans. Electron Devices*, **45**: 598–608, 1998.
9. T. R. N. Rao E. Fujiwara *Error Control Coding for Computer Systems*, Engelwood Cliffs, NJ: Prentice-Hall 1989.
10. W. Stallings *Computer Organization and Architecture*, 4th ed., Upper Saddle River, NJ: Prentice-Hall, 1996.
11. V. Hamacher Z. Vanesic S. Zaky *Computer Organization*, 4th ed., New York: McGraw-Hill, 1996.
12. J. Hennessy D. Patterson *Computer Architecture: A Quantitative Approach*, San Mateo, CA: Morgan Kaufmann, 1990.
13. R. R. Schaller Moore's law: Past, present and future, *IEEE Spectrum*, **34** (6): 53–59, 1997.
14. B. Prince Memory in the fast lane, *IEEE Spectrum*, **31** (2): 38–41, 1994.
15. I. Deyhimy Gallium arsenide joins the giants, *IEEE Spectrum*, **32** (2): 33–40, 1995.

Reading List

- J. Daniels *Digital Design from Zero to One*, New York: Wiley, 1996.
- W. Stallings *Computer Organization and Architecture*, 4th ed., Upper Saddle River, NJ: Prentice-Hall, 1996.
- Y. Taur *et al.* CMOS scaling into the nanometer regime, *Proc. IEEE*, **85**: 486–504, 1997.
- J. Wakerly *Digital Design Principles and Practices*, 2nd ed., Englewood Cliffs, NJ: Prentice-Hall, 1994.

GORDON B. AGNEW
University of Waterloo,
Waterloo, Ontario, Canada