Chapter 11
Physical Memory And Physical Addressing

Comer, D. (2017). *Essentials of Computer Architecture* (2nd ed.). CRC Press.

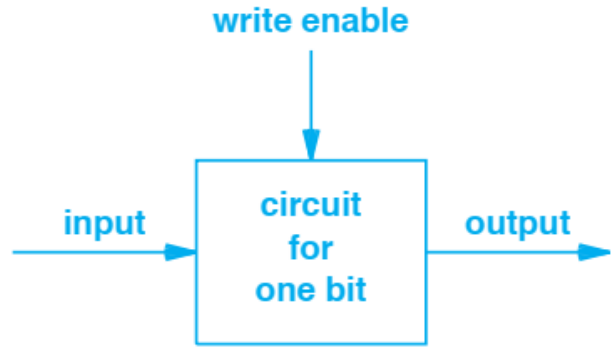
Mano, M. M., Kim, C. R. & Martin, T. (2015). *Logic and Computer Design Fundamentals* (5th ed.). Pearson.

Contents

Contents

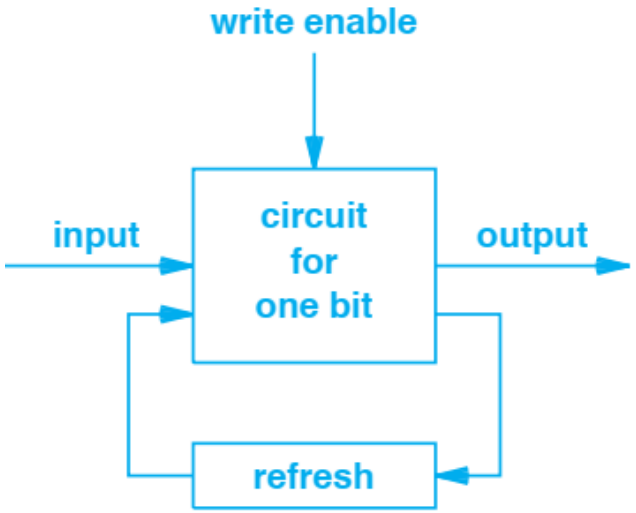
1. Static And Dynamic RAM Technologies

- The technologies used to implement Random Access Memory can be divided into two broad categories.
  - Static RAM (SRAM) is the easiest type for programmers to understand because it is a straightforward extension of digital logic.
    - Conceptually, SRAM stores each data bit in a latch, a miniature digital circuit composed of multiple transistors similar to the latch.
    - Although it performs at high speed, SRAM has a significant disadvantage: high power consumption (which generates heat).



1. Static And Dynamic RAM Technologies

- The alternative to static RAM, which is known as *Dynamic RAM (DRAM)*, consumes less power.
- DRAM uses a circuit that acts like a capacitor, a device that stores electrical charge.
  - When a value is written to DRAM, the hardware charges or discharges the capacitor to store a 1 or 0.
  - Later, when a value is read from DRAM, the hardware examines the charge on the capacitor and generates the appropriate digital value.
  - DRAM loses its charge in a short time (e.g., in some cases, under a second).
    - In practice, computers that use DRAM contain an extra hardware circuit, known as a *refresh* circuit, that performs **the task of reading and then writing a bit.**
- SRAM is faster than DRAM, but costs more, so the system had a smaller amount of SRAM (intended for items that were accessed frequently) and a large amount of DRAM (intended for items that were not accessed frequently).



2. The Two Primary Measures Of Memory Technology

- Two primary measures
  - Density
  - Latency and cycle times
- Density
  - The term density refers to the number of memory cells per square area of silicon.
  - In practice, however, density often refers to the number of bits that can be represented on a standard size chip or plug-in module.
    - For example, a *Dual In-line Memory Module (DIMM)* might contain a set of chips that offer 128 million locations of 64 bits per location, which equals 8.192 billion bits or one Gigabyte. Informally, it is known as a 1 gig module.
  - Higher density has the disadvantages of increased power utilization and increased heat generation.
  - The density of memory chips is related to the size of transistors in the underlying silicon technology, which has followed Moore's Law. Thus, memory density tends to double approximately every eighteen months.

2. The Two Primary Measures Of Memory Technology

- Separation Of Read And Write Performance
  - In many memory technologies, the time required to fetch information from memory differs from the time required to store information in memory, and the difference can be dramatic.
    - Some memory technologies take much longer to write values than to read them.
  - Therefore, any measure of memory performance must give two values: the performance of read operations and the performance of write operations.

2. The Two Primary Measures Of Memory Technology

- Latency and memory controllers
  - Latency: The time that elapses between the start of an operation and the completion of the operation.
  - To access memory, a device (typically a processor) presents a read or write request to the *memory controller*.
  - Memory controller translates the request into signals appropriate for the underlying memory, and passes the signals to the memory chips.

To minimize latency, the controller returns an answer as quickly as possible. However, after it responds to a device, a controller may need additional clock cycle(s) to reset hardware circuits and prepare for the next operation.

Because a memory system may need extra time between operations, latency is an insufficient measure of performance; a performance measure needs to measure the time required for successive operations.

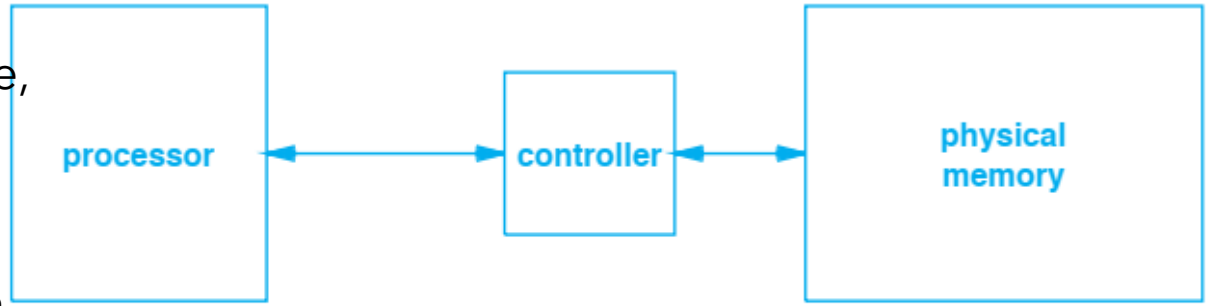


Figure 11.3 Illustration of the hardware used for memory access. A controller sits between the processor and physical memory.

3. Synchronous And Multiple Data Rate Technologies

- What happens if the processor's clock differs from the clock used for memory?
  - The system still works because the controller can hold a request from the processor or a response from the memory until the other side is ready.
  - Unfortunately, the difference in clock rates can impact performance – although the delay is small.
- To eliminate the delay, some memory systems use a *synchronous* clock system.
  - The clock pulses used with the memory system are aligned with the clock pulses used to run the processor.
    - A processor does not need to wait for memory references to complete.
- Synchronization can be used with DRAM or SRAM
  - SDRAM– Synchronous Dynamic Random Access Memory
  - SSRAM– Synchronous Static Random Access Memory

3. Synchronous And Multiple Data Rate Technologies

- In many computer systems, memory is the bottleneck – increasing memory performance improves overall performance.
  - Finding memory technologies with lower cycle times.
  - One approach uses a technique that runs the memory system at a multiple of the normal clock rate (e.g., double or quadruple).
    - Because the clock runs faster, the memory can deliver data faster. The technologies are sometimes called fast data rate memories, typically double data rate or quadruple data rate. Fast data rate memories have been successful, and are now standard on most computer systems, including consumer systems such as laptops.

3. Synchronous And Multiple Data Rate Technologies

Technology	Description
DDR-DRAM	Double Data Rate Dynamic RAM
DDR-SDRAM	Double Data Rate Synchronous Dynamic RAM
FCRAM	Fast Cycle RAM
FPM-DRAM	Fast Page Mode Dynamic RAM
QDR-DRAM	Quad Data Rate Dynamic RAM
QDR-SRAM	Quad Data Rate Static RAM
SDRAM	Synchronous Dynamic RAM
SSRAM	Synchronous Static RAM
ZBT-SRAM	Zero Bus Turnaround Static RAM
RDRAM	Rambus Dynamic RAM
RLDRAM	Reduced Latency Dynamic RAM

Figure 11.4 Examples of commercially available RAM technologies. Many other technologies exist.

4. Memory Access and Memory Bus

- To achieve high performance, memory systems use parallelism:
  - The connection between the processor and controller consists of many wires that are used simultaneously.
  - Each wire can transfer one data bit at any time.

The technical name for the hardware connection between a processor and memory is *bus* (more specifically, *memory bus*).

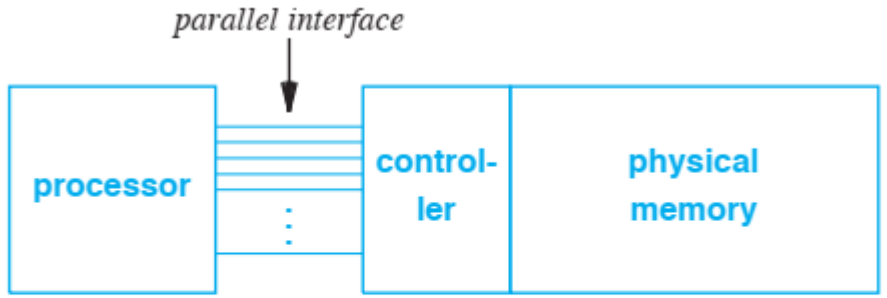


Figure 11.5 The parallel connection between a processor and memory. A connection that contains N wires allows N bits of data to be transferred simultaneously.

5. Words, Physical Addresses, and Memory Transfers

Physical memory is organized into words, where a word is equal to the memory transfer size. Each read or write operation applies to an entire word.

- From an architectural standpoint, using parallel connections can improve performance.
- From a programming point of view, the parallel connections define a *memory transfer size* (i.e., the amount of data that can be read or written to memory in a single operation).
- N is the memory transfer size. A block of N bits is sometimes called a *word*, and the transfer size is called the *word size* or the width of a word.
- Memory is organized into an array.
  - Each entry in the array is assigned a unique index known as a *physical memory address*; the approach is known as *word addressing*.

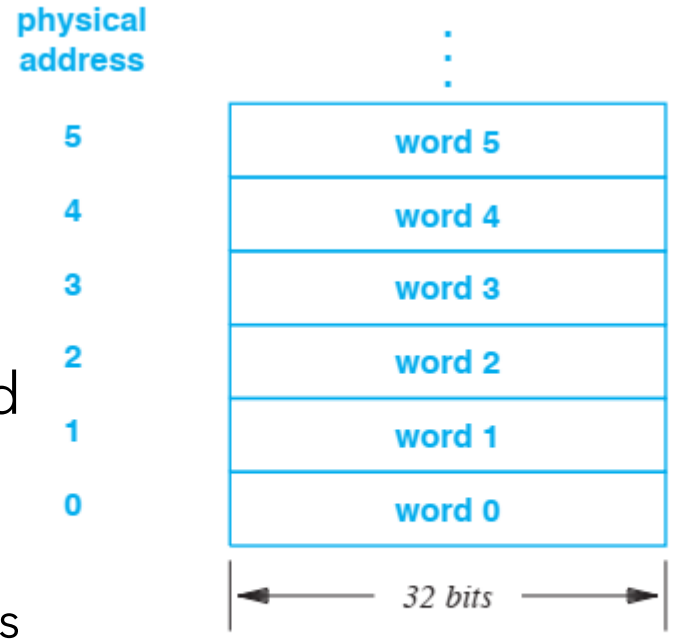


Figure 11.6 Physical memory addressing on a computer where a word is thirty-two bits. We think of the memory as an array of words.

Memory Word Size and Data Types

- In theory, performance can be increased by adding more parallel wires.
  - For example, an interface that has 128 wires can transfer data at twice the rate of an interface that has 64 wires.
  - The question arises: how many wires should an architect choose? That is, what word size is optimal?
- A word size of thirty-two bits is popular, especially for low-power systems; many high-performance systems use a sixty-four-bit word size.

6. Byte Addressing and Mapping Bytes To Words

- Physical memory is organized into words, where a word is equal to the memory transfer size. The approach is known as *word addressing*.
- *Byte addressing* is especially convenient for programming because it gives a programmer an easy way to access small data items such as characters.

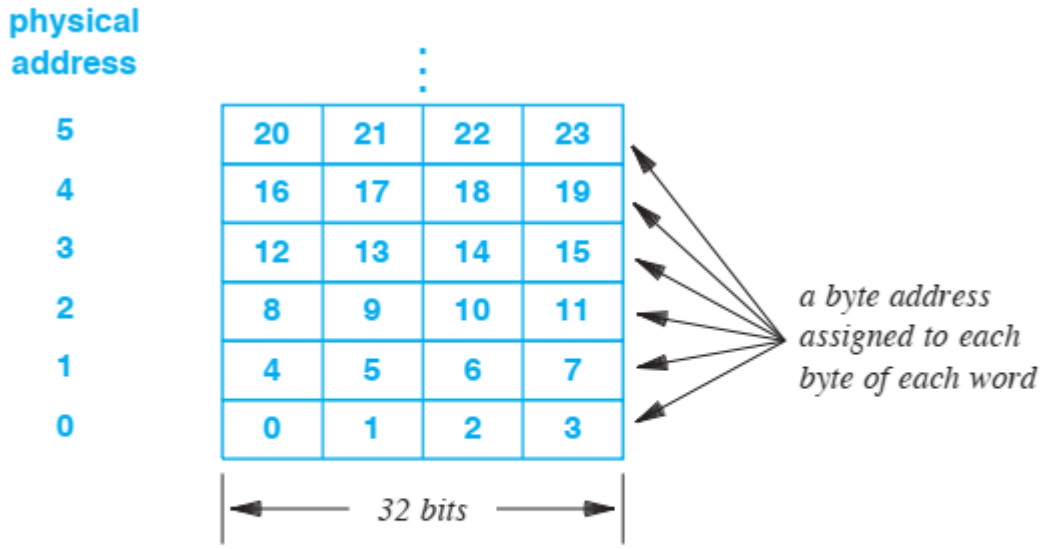


Figure 11.7 Example of a byte address assigned to each byte of memory even though the underlying hardware uses word addressing and a thirty-two-bit word size.

6. Byte Addressing and Mapping Bytes to Words

- Memory controllers must convert byte addresses issued by the processor to word addresses used by the memory system.
- If the processor requests a read operation for byte address 17, the controller must issue a read request for word 4, and then extract the second byte from the word.
- Because the memory can only transfer an entire word at a time, a byte write operation is expensive.
  - For example, if a processor writes byte 11, the controller must read word 2 from memory, replace the rightmost byte, and then write the entire word back to memory.

7. Byte Alignment and Programming

- Knowing how the underlying hardware works helps explain a concept that programmers encounter: *byte alignment*.
  - We say that an integer value is *aligned* if the bytes of the integer correspond to a word in the underlying physical memory.
  - In Figure 11.7, for example, an integer composed of bytes 12, 13, 14, and 15 is aligned, but an integer composed of bytes 6, 7, 8, and 9 is not.

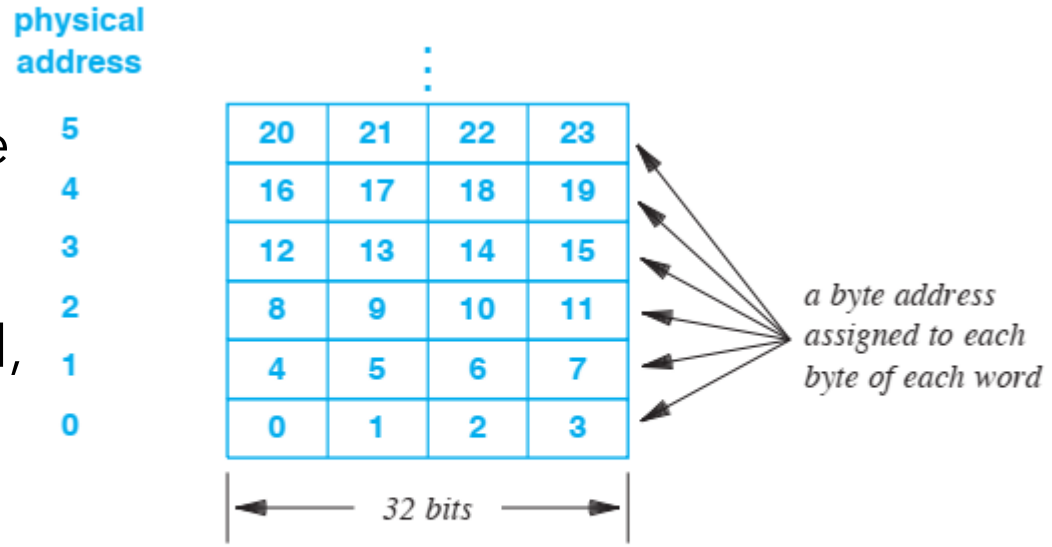


Figure 11.7 Example of a byte address assigned to each byte of memory even though the underlying hardware uses word addressing and

7. Byte Alignment and Programming

- On some architectures, byte alignment is required – the processor raises an error if a program attempts an integer access using an unaligned address.
- On other processors, arbitrary alignment is allowed, but unaligned accesses result in lower performance than aligned accesses.

8. Multiple Memories with Separate Controllers

- Some architectures contain multiple physical memories.
  - Hardware parallelism can be employed to provide higher memory performance.
  - Instead of a single memory and a single controller, the memory system can have multiple controllers that operat in parallel.
- Interface hardware receives requests from the processor.
- The interface uses the address in the request to decide which memory should be used, and passes the request to the appropriate memory controller
- Because memory controllers can operate in parallel using two memory controllers allows more memory accesses to occur per unit time.

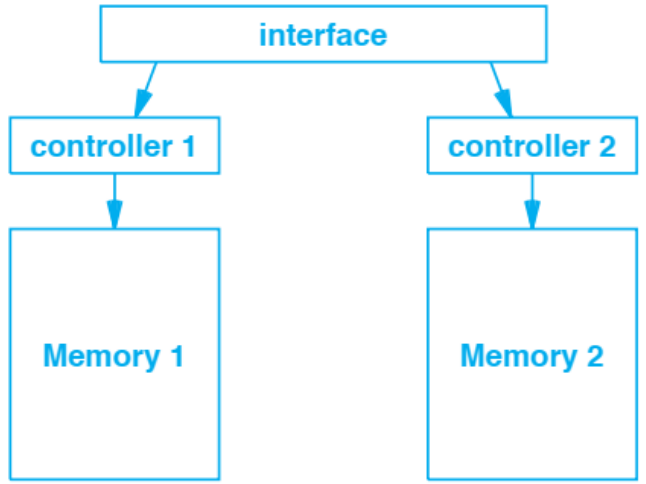


Figure 11.11 Illustration of connections for two memory modules with separate controllers.

9. Memory Banks

- Suppose two identical memory modules are each designed to have physical addresses 0 through M – 1.
- The interface can arrange to treat them as two *banks* that form a contiguous large memory with twice the addresses.
- Addresses 0 through M–1 are assigned to one bank and addresses from M to 2M – 1 are assigned to the second bank.

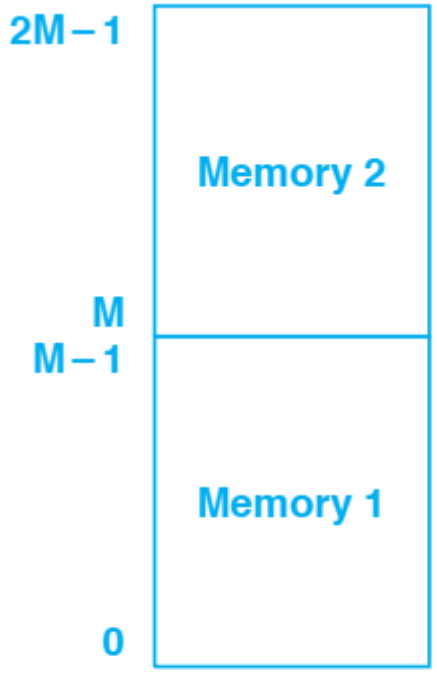


Figure 11.12 The logical arrangement of two identical memory banks to form a single memory that is twice the size.

10. Interleaving

- Observe that many programs access data from **sequential memory locations**.
  - In a banked memory, sequential locations lie in the same memory bank, which means that successive accesses must wait for the controller to reset.
- Interleaving uses the idea of separate controllers, but instead of organizing memories into banks, interleaving places consecutive words of memory in separate physical memory modules.
  - Interleaving achieves high performance during sequential memory accesses because a word can be fetched while the memory for the previous word resets.
  - We use the terminology *N-way interleaving* to describe the number of underlying memory modules (to make the scheme efficient, *N* is chosen to be a power of two)

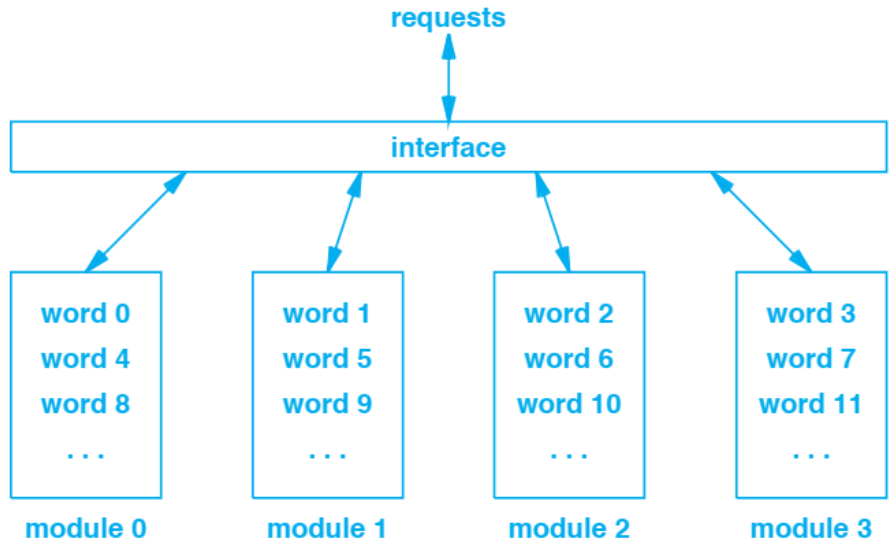


Figure 11.13 Illustration of 4-way interleaving that illustrates how successive words of memory are placed into memory modules to optimize performance.