
CHAPTER 4

LOGICAL OPERATING PRINCIPLES OF COMPUTERS

The most important parts in a computer are its processor and its main memory that can be read and written by the processor. For a computer to do something useful, there must be an executable program in its main memory. An executable program is a sequence of machine instructions which the processor of the computer can read and interpret.

In this chapter, we shall study machine instructions and some general operating principles of computers. First, we will examine the operation of the main memory. Then we will study a processor and programs which are executed by the processor. You will learn how some basic computing activities, like repetition and subroutine calling, are performed at the machine level. This will help you to understand high-level C++ programming in later chapters.

The general operating principles of computers will be explained with the help of an imaginary computer which can be regarded as a logical model of real computers. All computers which are commonly used operate according to the same principles as the imaginary computer, although the imaginary computer is simpler than a real computer. Its simplicity makes the imaginary computer an ideal instrument for learning the basics of computer operation. The computer is imaginary because it has not been built by using electronic or any other physical components. The electronic material of this book contains, however, a simulation program that imitates the imaginary computer on a personal computer.

These are sample pages from Kari Laitinen's book
"A Natural Introduction to Computer Programming with C++".

For more information, please visit

<http://www.naturalprogramming.com/cppbook.html>

4.1 How does the main memory operate?

Computers are able to process information that is stored in their memory in binary form. There are basically two kinds of memory devices in a computer. The main memory is built of RAM memory devices. (RAM is an abbreviation of Random Access Memory.) All other memory devices can be considered auxiliary memory devices. The main memory of a computer is more important than the auxiliary memory devices because programs that are being executed must be kept in the main memory. A computer must have a main memory in order to operate. Although computers like PCs are equipped with auxiliary memory devices (e.g. hard disk), it is possible to build computers without the auxiliary devices.

The main memory of a computer is a device which can be read from and written to by the processor of the computer. Figure 4-1 illustrates a small main memory that is only 16 bytes (16 x 8 bits) in size. Computers generally have a much larger main memory, but we can study the memory operations with just this small main memory. Figure 4-1 shows that four address lines A0, A1, A2, and A3 are needed to select one of the 16 bytes in the memory. The memory addressing control takes care that the right memory location is selected when a certain bit combination is switched to the address lines.

While studying the memory device in Figure 4-1, we suppose that it can be used by switching either zero Volts (0V) or five Volts (5V) to the lines of the device. By switching various voltages to the address lines, it is possible to select certain memory locations, for example in the following way:

A3	A2	A1	A0	MEANING
0V	0V	0V	0V	memory address 0 is selected
0V	0V	0V	5V	memory address 1 is selected
0V	5V	0V	5V	memory address 5 is selected
5V	0V	0V	5V	memory address 9 is selected
5V	5V	5V	5V	memory address 15 is selected

The main memory of a computer has data lines through which information is either moved into the memory (writing of data), or information is moved out of the memory (reading of data). In Figure 4-1 there are 8 data lines D0, D1, D2, D3, D4, D5, D6, and D7 through which one byte of information can either be written to or read from the memory.

Address lines are used to select the desired location in the memory, and data lines are needed to carry information to/from the memory. In addition to address and data lines, there are usually control lines and power supply lines in memory devices. The control lines ensure that the reading and writing operations are carried out in an accurate manner. The memory device in Figure 4-1 has two control lines which have names READ MEMORY and WRITE MEMORY. With these lines (signals) the processor which is using the memory device can perform either a writing operation or a reading operation. Power supply lines are needed to supply electricity for physical memory components, but for simplicity these lines are left out from Figure 4-1.

The control signals of a memory device are activated by the processor that is using the memory. The control signals of the memory device in Figure 4-1 can have the following values and meanings

WRITE MEMORY	READ MEMORY	MEANING
0 (0V)	0 (0V)	No memory operations
0 (0V)	1 (5V)	Read selected memory address
1 (5V)	0 (0V)	Write selected memory address
1 (5V)	1 (5V)	Not allowed combination

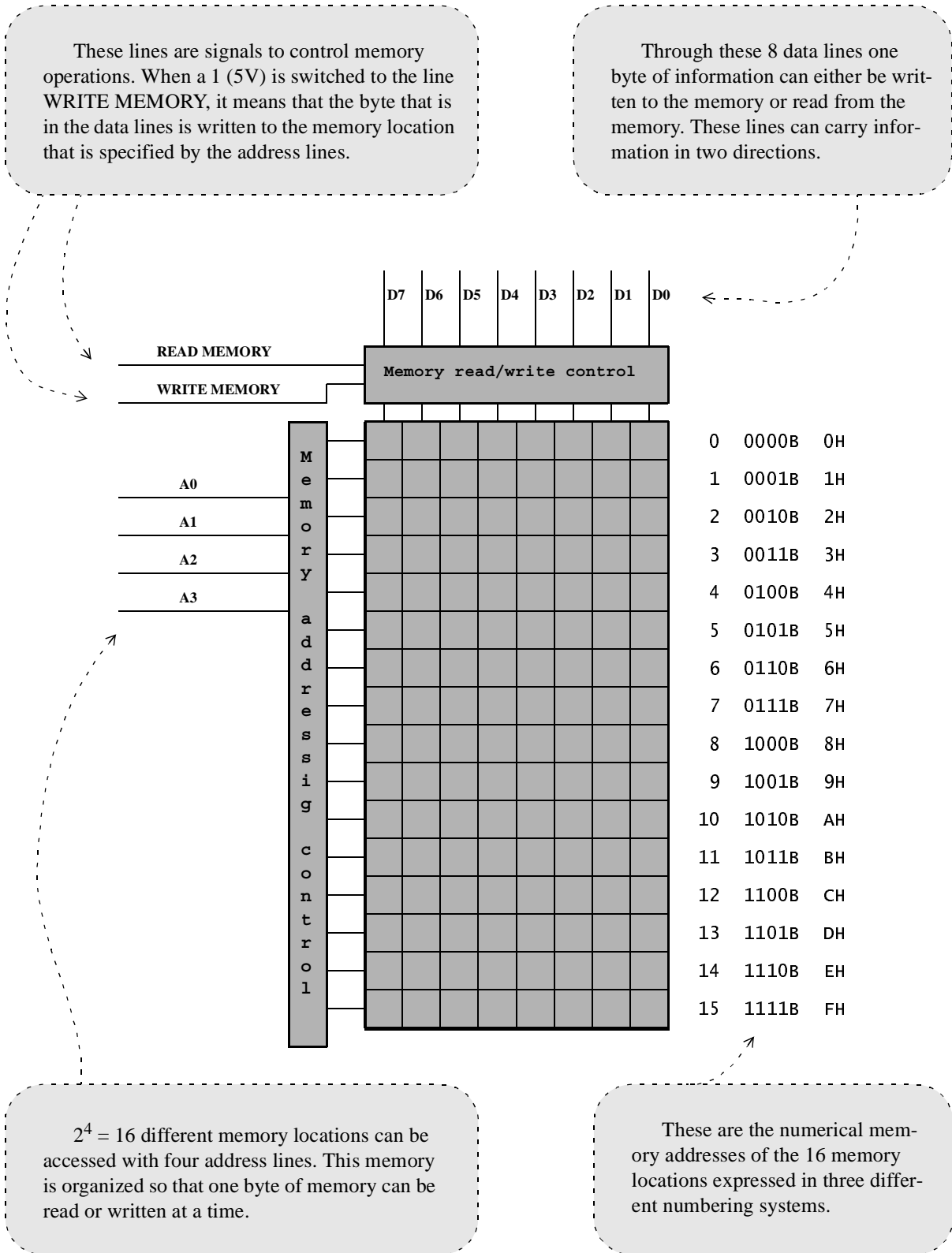


Figure 4-1. A theoretical 16-byte main memory device.

For example, if a processor wanted to write the binary number 01101110B into memory address 5 of the memory device in Figure 4-1, it should switch the following voltages to the various lines:

Address lines				Data lines								Control lines	
A3	A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0	WRITE MEMORY	READ MEMORY
0V	5V	0V	5V	0V	5V	5V	0V	5V	5V	5V	0V	5V	0V

With the above voltages in the address lines, the address 0101B (5 decimal) is selected. When address 0101B is switched to the address lines and the WRITE MEMORY signal is active, the binary number that is switched to the data lines is stored in memory address 5 (0101B) and the information that was previously in that location is written over and lost. This is how writing to the main memory takes place. The WRITE MEMORY signal must be 1 (5V) and the READ MEMORY signal must be 0 (0V) at the moment when data is stored in the memory.

Different memory locations can be written by switching different addresses (different voltage combinations) to the address lines. Figure 4-2 shows a timing diagram that describes four writing operations on the memory device of Figure 4-1. Time goes from left to right in the diagram, and voltages are altered in all of the lines that enter the memory device. Note that the address and input data are always changed before the actual writing takes place. Data is written when the WRITE MEMORY signal goes up. For you to understand Figure 4-2 properly, it may be useful to mark zeroes or ones on those points where input signals change state.

A memory device like the one in Figure 4-1 keeps its contents as long as no memory locations are written over in writing operations. A reading operation cannot change the data stored in a memory device. When a reading operation is activated, the data lines of a memory device work in the opposite direction as for a writing operation. A memory location in the memory device of Figure 4-1 can be read by setting the address of the memory location on the address lines and activating the READ MEMORY signal. For example, if the lines of the memory device would be set to voltages

A3	A2	A1	A0	WRITE MEMORY	READ MEMORY
0V	5V	5V	0V	0V	5V

the data in the memory address 6 (0110B) would be copied to the data lines. The processor which is using the memory device could then read the data from the data lines.

Exercise related to main memory usage

Exercise 4-1. Below, write the voltages that need to be switched to address and data lines in Figure 4-1 in order to write the decimal number 97 into memory address 9 of the memory device. Which voltages must simultaneously be switched to the control lines?

Address lines				Data lines							
A3	A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0

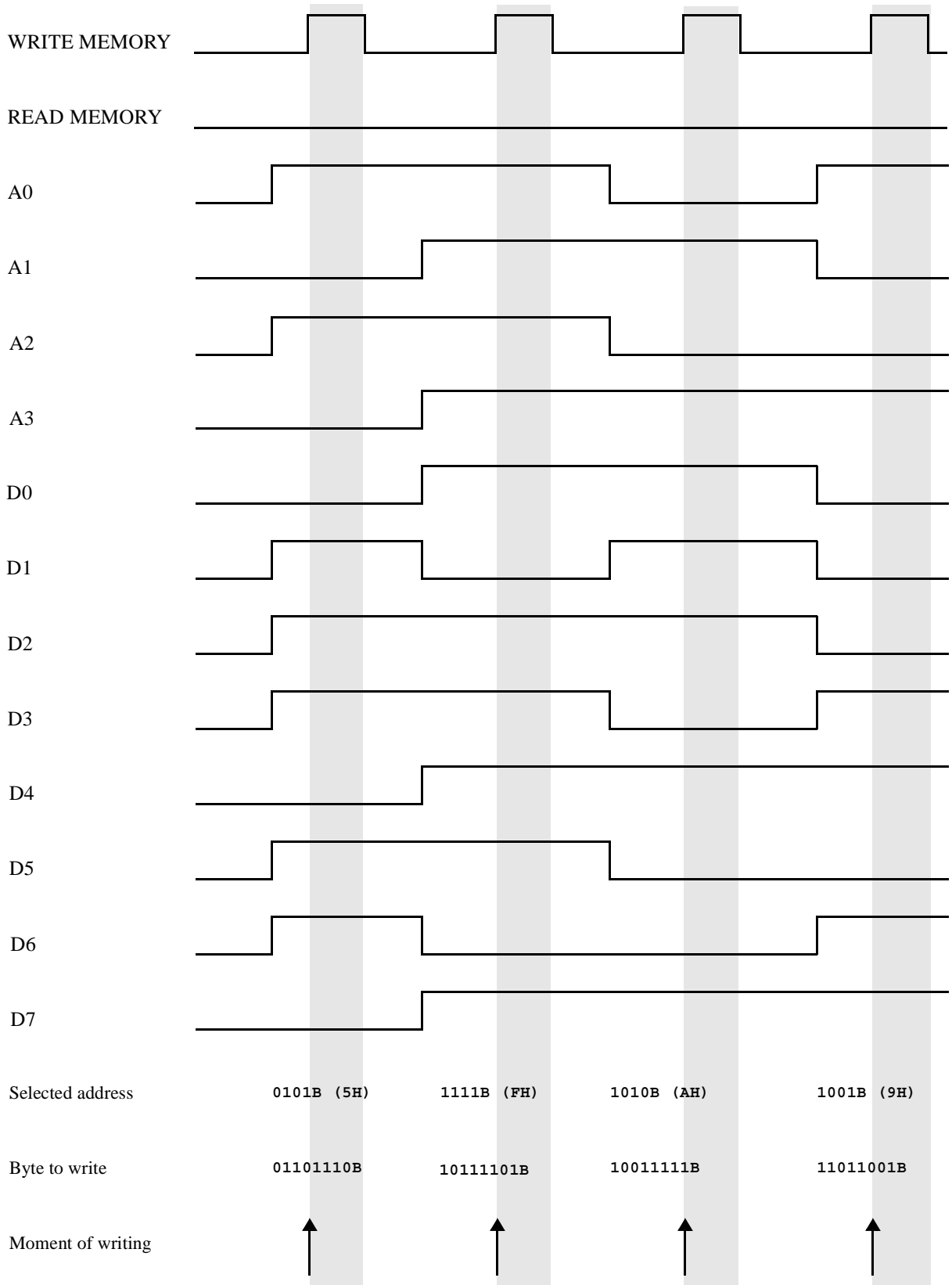


Figure 4-2. Writing four memory locations in the memory device of Figure 4-1.

4.2 The components of an imaginary computer

Now, after having studied how the main memory of a computer can be accessed, we can start investigating the imaginary computer presented in Figure 4-3. This computer is an imaginary one, because it exists only on these pages and in our minds. It would be possible to use electronic components to build this kind of a computer, but because the imaginary computer has been created solely for educational purposes, a "real" imaginary computer would not be as good as the existing commercially available computers. The imaginary computer in Figure 4-3 is a general model of a computer, and it teaches us how real computers work. The electronic material of this book contains a simulation program which shows how the imaginary computer would behave if it were built.

As shown in Figure 4-3, the imaginary computer consists of a main memory, a processor, a keyboard, and a screen. The keyboard and the screen are connected by cables to the processor. The keyboard is an input device through which input data can be supplied to the processor. For example, if the user of the imaginary computer presses key A on the keyboard, the ASCII code of letter A, 41H, is transferred to the processor. Similarly, the processor can output data by sending ASCII codes to the screen. The screen is an output device which shows the ASCII codes as visible characters. For example, if the processor sends code 42H to the screen, it appears as letter B on the screen.

The main memory in Figure 4-3 is similar to the smaller memory that we saw in Figure 4-1. The difference is that the main memory of our imaginary computer is 16 times larger than the memory in Figure 4-1. The imaginary computer has 8 address lines A0, A1, A2, A3, A4, A5, A6, and A7 which make it capable of addressing all 256 bytes of the main memory. Note that not all memory locations are shown in Figure 4-3. The main memory of our imaginary computer is very small when compared to modern commercially available computers, whose main memory consists of millions of bytes.

The imaginary processor uses two control signals, READ MEMORY and WRITE MEMORY, to either read or write data from/to the main memory. The processor can access one byte of memory at a time. For example, if the processor wants to access the memory location A2H (10100010B), it has to set the following voltages to the address lines

A7	A6	A5	A4	A3	A2	A1	A0	address lines
5V	0V	5V	0V	0V	0V	5V	0V	voltages
1	0	1	0	0	0	1	0	binary memory address

When a memory location has been selected by switching appropriate voltages to the address lines, the processor can read the contents of that memory location by setting the READ MEMORY signal to 1 (5V), or it can write the memory location by setting the WRITE MEMORY signal to 1 (5V). In a reading operation, the contents of the selected memory location are copied as 0V and 5V voltages to the data lines D0, D1, D2, D3, D4, D5, D6, and D7. In a writing operation, the voltages on the data lines are copied to the selected memory location, and the old contents of the memory location are lost.

The processor of the imaginary computer is a complex device which is capable of switching 0V or 5V voltages on the address, data, and control lines. In some situations it can read voltages from the data lines. 0V means binary 0 and 5V means binary 1 for the processor. The processor is made active by supplying a clock signal to it. The clock signal is such that its voltage varies constantly between zero and one. If the clock signal goes from zero to one and back to zero 200 times in a second, we say that the processor uses a 200 Hz clock. As we shall learn later, the processor keeps repeating a certain sequence when it is working. The clock signal determines how fast the processor does its job. In our imaginary computer the speed of the processor is not essential, but we must imagine that there is a clock signal which makes things happen.

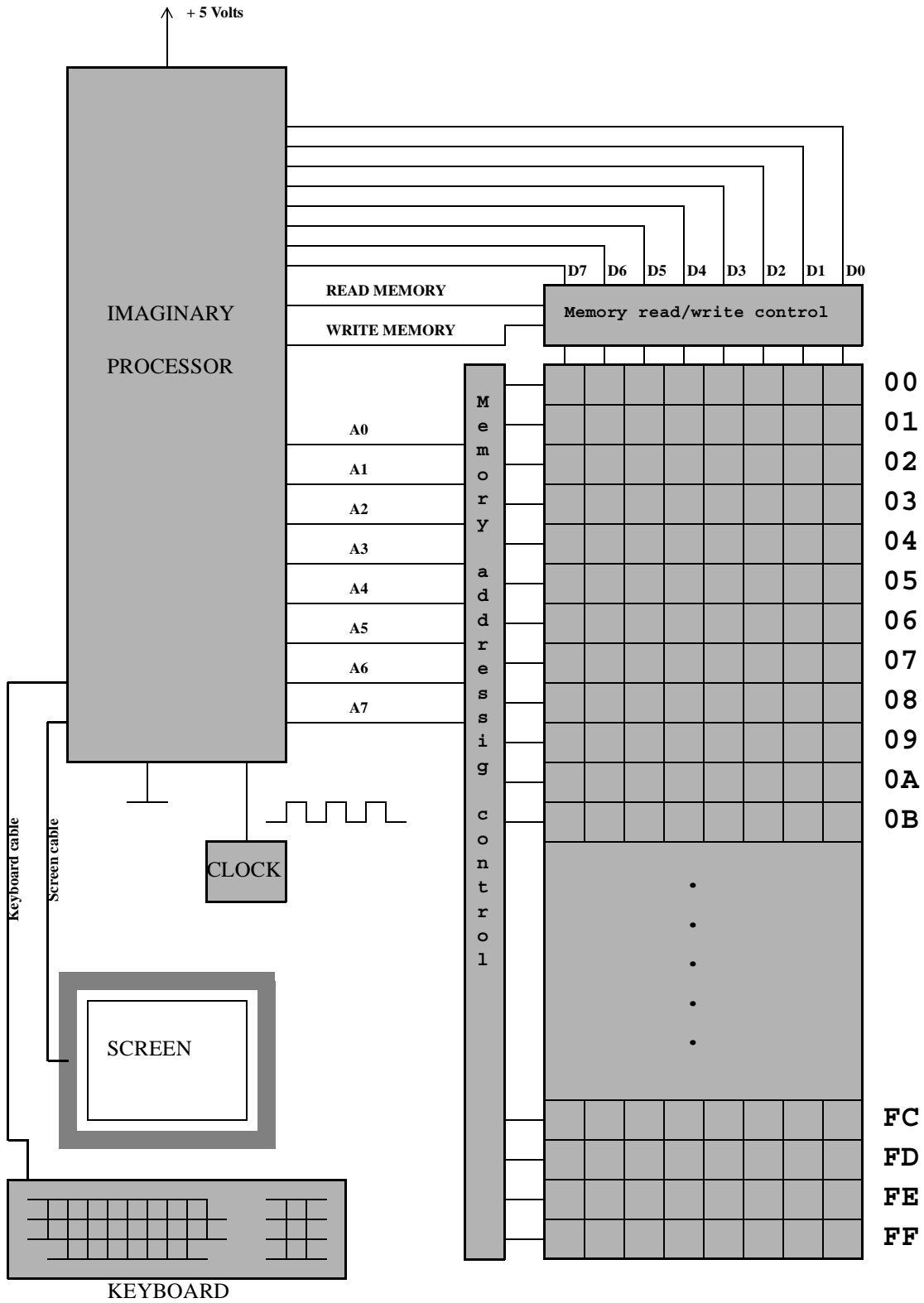


Figure 4-3. An imaginary computer with a 256-byte main memory.