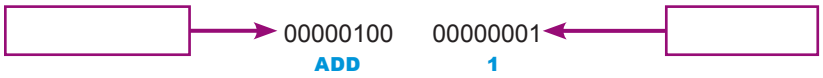
A machine language instruction has two parts: the op code and the operands. An **op code**, which is short for *operation code*, is a command word for an operation such as add, compare, or jump. The **operand** for an instruction specifies the data, or the address of the data, for the operation. In the following instruction, the op code means add and the operand is 1, so the instruction means add 1.



TRY IT!

Label the operand and the op code.

A single high-level instruction very often converts into multiple machine language instructions. Figure 1-36 illustrates the number of machine language instructions that correspond to a simple high-level program.

```
#include <stdio.h>
int main ()
{
int i;

for (i=1; i<=100; i=i+1)
  printf("%d\t",i);
return(0);
}
```

```
00100111101111011111111111100000
10101111101111110000000000010100
10101111101001000000000000100000
10101111101001010000000000100100
10101111101000000000000000011000
10101111101000000000000000011100
10001111101011100000000000011100
10001111101110000000000000011000
00000001110011100000000000011001
00100101110010000000000000000001
00101001000000010000000001100101
10101111101010000000000000011100
00000000000000000111100000010010
00000011000011111100100000100001
00010100001000001111111111110111
10101111101110010000000000011000
00111100000001000001000000000000
10001111101001010000000000011000
00000110000010000000000001101100
00100100100001000000010000110000
```

FIGURE 1-36

The source code program in the left column prints numbers from 1 to 100. This source code is converted to machine language instructions shown in the right column that the computer can directly process.

To summarize what you should now know about programs and instruction sets, a programmer creates human-readable source code using a programming language. A compiler or an interpreter converts source code into machine code. Machine code instructions are a series of 0s and 1s that correspond to a processor's instruction set.

PROCESSOR LOGIC

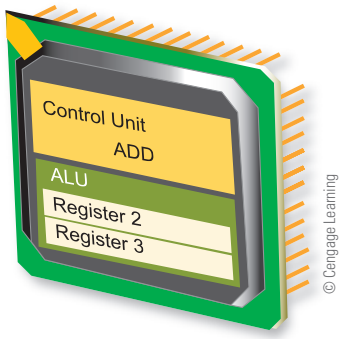
What happens inside a computer chip? A microprocessor contains miles of microscopic circuitry and millions of miniature components divided into different kinds of operational units, such as the ALU and the control unit.

The **ALU** (arithmetic logic unit) is the part of the microprocessor that performs arithmetic operations, such as addition and subtraction. It also performs logical operations, such as comparing two numbers to see if they are the same. The ALU uses **registers** to hold data that is being processed, just as you use a mixing bowl to hold the ingredients for a batch of cookies.

The microprocessor's **control unit** fetches each instruction, just as you get each ingredient out of a cupboard or the refrigerator. Data is loaded into the ALU's registers, just as you add all the ingredients to the mixing bowl. Finally, the control unit gives the ALU the green light to begin processing, just as you flip the switch on your electric mixer to begin blending the cookie ingredients. Figure 1-37 illustrates a microprocessor control unit and an ALU preparing to add 2 + 3.

FIGURE 1-37

The control unit fetches the ADD instruction, then loads data into the ALU's registers where it is processed.



What happens when a computer executes an instruction?

The term **instruction cycle** refers to the process in which a computer executes a single instruction. Some parts of the instruction cycle are performed by the microprocessor's control unit; other parts of the cycle are performed by the ALU. The steps in this cycle are summarized in Figure 1-38.

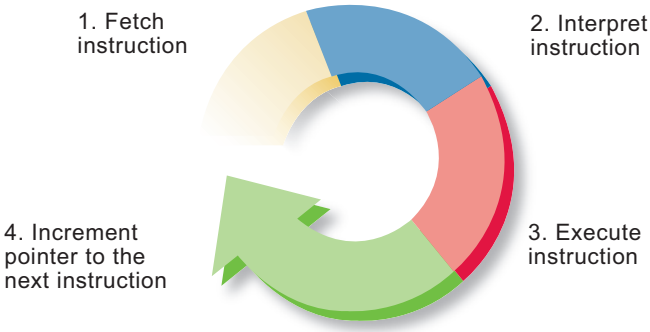


FIGURE 1-38

The instruction cycle includes four activities.

What role does the control unit play?

The instructions that a computer is supposed to process for a particular program are held in memory. When the program begins, the memory address of the first instruction is placed in a part of the microprocessor's control unit called an instruction pointer.

The control unit can then fetch the instruction by copying data from that address into its instruction register. From there, the control unit can interpret the instruction, gather the specified data, or tell the ALU to begin processing. Figure 1-39 helps you visualize the control unit's role in processing an instruction.

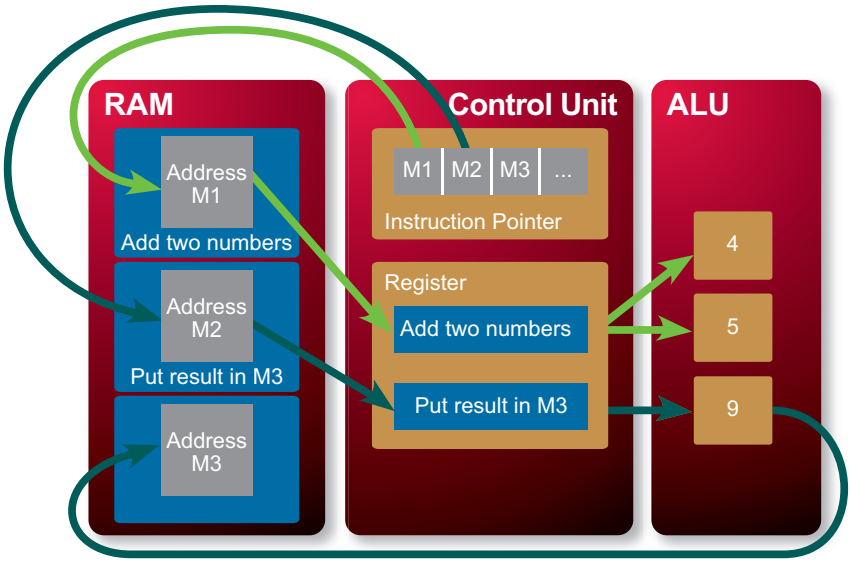
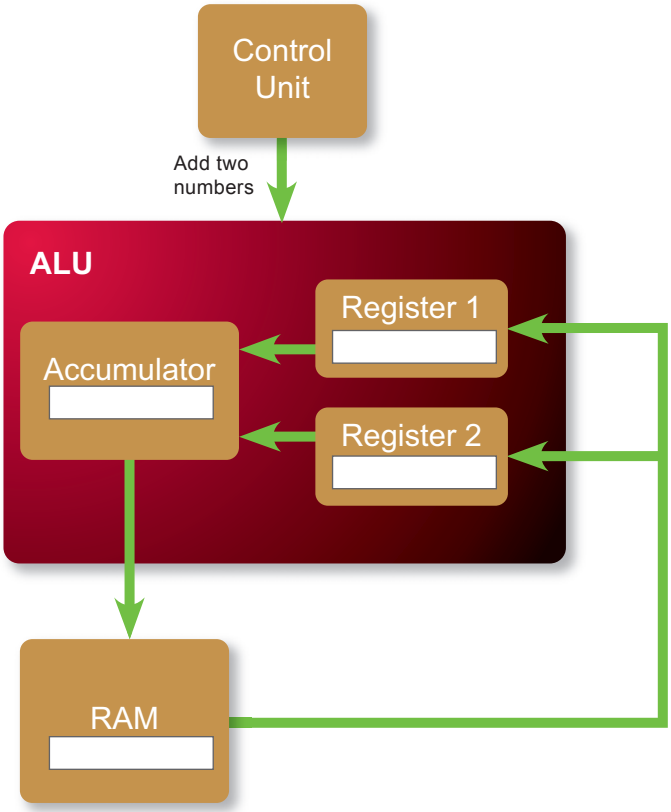


FIGURE 1-39

The control unit's instruction pointer indicates M1, a location in memory. The control unit fetches the "Add two numbers" instruction from M1. This instruction is then sent to the ALU. The instruction pointer then changes to M2. The processor fetches the instruction located in M2, moves it to a register, and executes it.
See how it works.

When does the ALU swing into action?

The ALU is responsible for performing arithmetic and logical operations. It uses registers to hold data ready to be processed. When it gets the go-ahead signal from the control unit, the ALU processes the data and places the result in an accumulator. From the accumulator, the data can be sent to memory or used for further processing. The TRY IT! on the next page helps you visualize what happens in the ALU as the computer processes data.



TRY IT!

Execute the following set of commands using the ALU diagram at left.

Load Register 1 with 4.

Load Register 2 with 5.

Add the two registers.

Move the result to RAM.

What happens after an instruction is executed? When the computer completes an instruction, the control unit increments the instruction pointer to the memory address of the next instruction, and the instruction cycle begins again.

Do I need to know all this detailed stuff? What you should take away from the discussion about programming and instruction sets is the idea that computers and other digital devices accomplish a wide array of complex tasks by performing a very limited set of machine language instructions very fast.

These concepts about how processors work will help you understand the significance of microprocessor performance, such as speed and word size, which you'll learn about in the next chapter.

QuickCheck

SECTION D

1. A(n) _____ converts all of the source code instructions into a new file containing _____ code.
2. A microprocessor is hard-wired to perform a set of activities called a(n) _____ set.
3. A machine language instruction has two parts: a(n) _____ code and an operand.
4. The ALU in your computer's microprocessor holds data in _____ .
5. The microprocessor's control unit contains a(n) _____ pointer that holds the address of the instruction being executed.

CHECK ANSWERS