The World's Simplest Computer Table of Contents

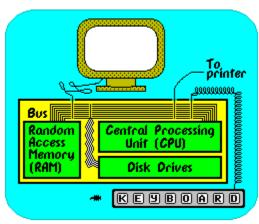
Return to SunShine Table of Contents

Page 1:	The World's Simplest Computer
Page 2:	A Bit Is Like a Bulb
Page 3:	Bits Can Be Many Things
Page 4:	Eight Bits Make a Byte
Page 5:	Half of a Byte Is a Nibble
Page 6:	A Word Is One or More Bytes
Page 7:	Review
Page 8:	Bytes Are Like Characters
Page 9:	Bytes Are Also Numbers
Page 10:	One Byte, Two Meanings
Page 11:	A Primitive Computer
Page 12:	A Five-Million-Bit Computer
Page 13:	256 Variations of Byte
Page 14:	A Byte Can Be Many Things
<u>Page 15:</u>	ASCII ("As-Key") to the Rescue!
<u>Page 16:</u>	Is it ASCII?
Page 17:	(ASCII Illustration)
Page 18:	Pixels, Light Bulbs, and Bits
<u>Page 19:</u>	Pixels Copy Bits
<u>Page 20:</u>	From Characters to Pixels
Page 21:	Bytes Have Addresses
<u>Page 22:</u>	The Byte Post Office
Page 23:	The CPU Is the Handling Area
Page 24:	Byte Addition
<u>Page 25:</u>	Adding Bytes from Memory
<u>Page 26:</u>	Variables Are Assigned Addresses
<u>Page 27:</u>	Bytes Take the Bus
Page 28:	The Real Bus
Page 29:	Memory Maps
Page 30:	Interrupts Are Traffic Controllers
Page 31:	RAM Has Many Uses
Page 32:	In the Chips
	The CPU Chip
	Putting it all Together
Page 35:	<u>Disk Drives</u>
Page 36:	The Keyboard
Page 37:	The Video Display
Page 38:	
Page 39:	Circuit Boards, Part 1
Page 40:	Circuit Boards, Part 2
Page 41:	Beware the DIP Switches
Page 42:	<u>Bugs</u>
Page 43:	Everything Taken Apart
Page 44:	<u>Like Many, Like None</u>
Page 45:	
Page 46:	What's Going on in There?

Page 47:	The CPU Governs Everything
Page 48:	Loading the Program
Page 49:	Following the Program Code
Page 50:	Typing the Document
Page 51:	Printing the Document
Page 52:	Saving the Document
Page 53:	Other Types of Programs
Page 54:	Load, Load, Change, and Save
Page 55:	<u>Test</u>
Page 56:	Answer: Yes
Page 57:	It Takes a Program to Write a Program
Page 58:	Loading the Language
Page 59:	Typing the New Program
Page 60:	Translating to Program Code
Page 61:	Back to Bytes
Page 62:	The Lowly Bit
Page 63:	Are Computers Intelligent?
Page 64:	The World's Simplest Computer (Revisited)



Shrinking the System



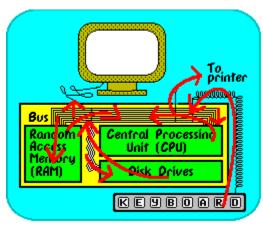
The illustration is being shrunk down to size to make further explanations easier.

Note that the CPU is not to scale even in this smaller illustration. It is actually on a single chip inside the computer cabinet.

Some things have been left out: The power supply, the speaker, and the cooling fan, for example. They do not contribute much towards understanding how a computer works.



What's Going on in There?



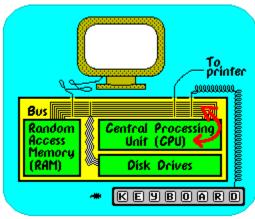
With so much going on inside the computer, many users do not understand where the data is at in there.

Sometimes their data disappears and they do not know why.

Over the next few pages, the illustrations will show what happens inside the computer during popular types of programs.



The CPU Governs Everything



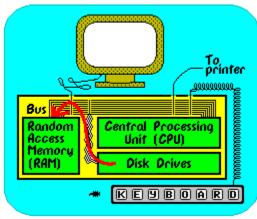
The CPU rules all of the other parts of the computer.

It does this by sending codes, in the form of bytes, through the bus to the other components.

The CPU gets feedback in the same way on such matters as the values as variables, the status of the program, and whether the printer is ready.



Loading the Program



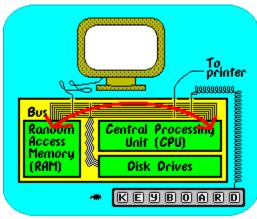
A word processing program is used as the first example.

The program, which has been stored on a disk, is sent through the bus and is put into RAM.

The program is stored on the disk in the form of bytes. It is also sent through the bus and saved in RAM as bytes.



Following the Program Code



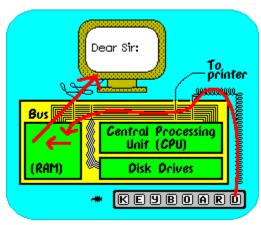
Once the program code is in RAM, it is read by the CPU.

The CPU will obtain an instruction from the program code and then direct the appropriate part of the computer to perform that instruction.

The CPU will then get the next instruction, and the next, going on down the sequence of program code, performing each one in order.



Typing the Document



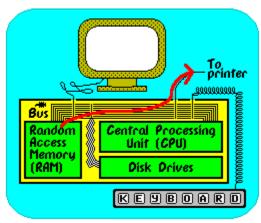
In the word processing program, as the user types the document, it is placed into variables in RAM.

Following the directions of the program, the document is also transferred from the variables to the part of RAM which designates the video display.

It is actually in two locations in RAM. First, in the variables location; and, second, in the video display location. Of course, all of this is in the form of bytes.



Printing the Document



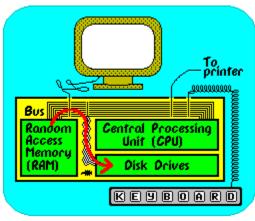
When a copy of the document is printed, it is sent from the variables in RAM to the printer.

This transfer is done through the bus in the form of bytes.

Do not forget that the CPU is directing these activities. Arrows to and from the CPU have been omitted for the sake of simplicity.



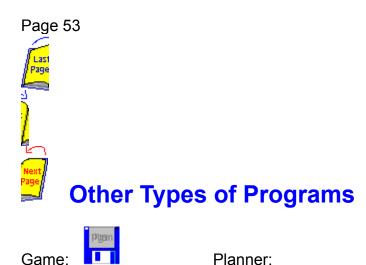
Saving the Document



Once the document is finished, it can be saved to the disk for future use.

It is transferred from the variables in RAM through the bus to the disk drive. Of course, all of this is done in the form of bytes and is controlled by the CPU.

Even though the document has been put on the disk, it is still also in RAM until it is cleared, or overwritten by other data, or until the computer is turned off.



Tutor:

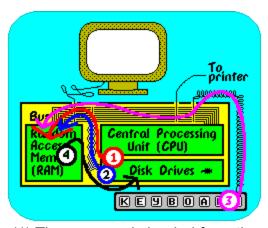
Other types of programs work in a similar way to a word processing program. First, the program, itself, is loaded from the disk drive into RAM. Second, if any data is being used from a previous session, it is loaded into RAM from the disk drive. Third, the data is increased, changed, or decreased by the user. Finally, the data is transferred from RAM back to the disk drive and onto a disk.





Load, Load, Change, and Save

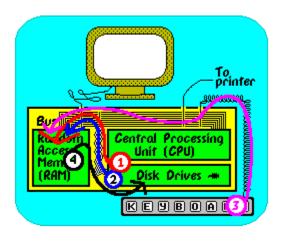
Most programs are similar in this respect: You load the program, then load the data, then change the data, and then save the data.



- (1) The program is loaded from the disk into RAM.
- (2) Previous data, if any, is loaded from the disk into RAM.
- (3) The data is changed from the keyboard.
- (4) The new or changed data is saved from RAM to the disk.



Test



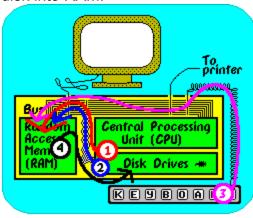
- (1) Load.
- (2) Load.
- (3) Change.
- (4) Save.

A word processing program has been loaded into the computer along with a document for a long-term project. The user accidentally ruins the document. Can the document be salvaged?



Answer: Yes

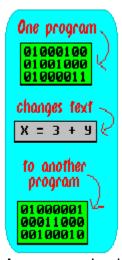
In the test, only the first three steps had been done. Therefore, the original document is still on the disk. To salvage it, go back to Step 2 and load it, again, from disk into RAM.



- (1) Load.
- (2) Load.
- (3) Change.
- (4) Save.



It Takes a Program to Write a Program



A programming language is needed in order to program a computer, and a programming language is a program, itself.

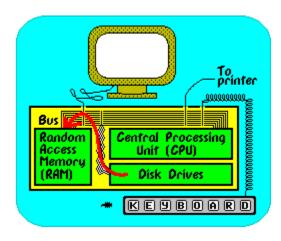
Many new computer users try the BASIC programming language. This stands for Beginner's All-purpose Symbolic Instruction Code.

Many other types of programming languages are available, such as C, Pascal, Fortran, Cobol, and others.

Sometimes a computer is distributed with a programming language, such as BASIC, included. More often, a programming language has to be purchased separately.



Loading the Language

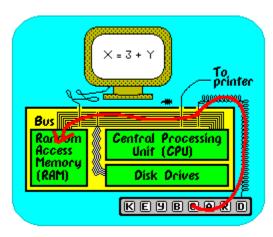


In order to begin programming, the language must first be loaded into RAM.

It is probably loaded from a disk. However, with BASIC, it is possible that it may be loaded from a read-only memory (ROM) chip.



Typing the New Program

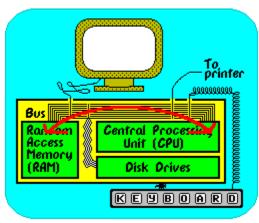


As the program is typed, in BASIC or whatever other programming language is being used, it is saved as ASCII text in variables in RAM.

The newly written program can be saved to a disk in this text format, to be recalled and worked on, again, later.



Translating to Program Code



Before the newly written program can be performed by the computer, it has to be translated to machine language.

This translation is done internally by the CPU, using RAM and following the directions of the programming language.

Once the newly written program has been changed to machine language, most programming languages allow it to be saved to a disk as a program.



Back to Bytes

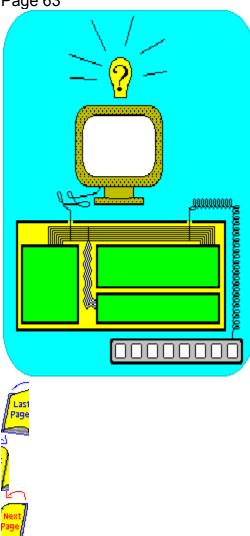
01000001 10011000 00100010 01000000 0100 01000001 911000 10011 100010 00100 Bytes represent numbers, characters, 01000100 1100 programming code, graphics, and anything else 00100 01000 0¢ 01 1001 00100001 .000 01001000 00001 00100 01000 01001 a programmer wants them to represent. 1001 01000001 01 00001 01000 11000 10011 000 10011000 No matter what a computer is doing, it is 1010 00100010 96 00010 00100 using bytes. 3100 01000100 01 00100 01000 1001 00100001 00001 00100 000 01001000 01000 01001 The keyboard, RAM, CPU, disk drive, and 1001 01000001 00001 01000 printer all work in bytes. Adding, subtracting, 000 10011000 11000 10011 9010 00100010 99919 00100 multivplying, dividing and everything else a 1100 01000100 00100 01000 computer does is done with bytes. 00001 01000 00001 00100 1001 00100001 96 .000 1001 01001000 01000001 Word processors, spreadsheets, databases, 000 10011000 11000 10011 and all other programs use bytes. 9010 00100010 9100 01000100 96 00010 00100 010€ 000100 01000 1001 00100001 00100001 00100 100 01001000 01000001 01000001 01000001 01000001 01000001



A bit is nothing but 1/8 of a byte. It is the smallest unit of information which a computer can handle.

A bit is similar to a light bulb in that it can be either on or off.

However, everything that a computer does, it does with bytes. Since bytes are made of bits, this means that all that a computer can do is done with little pieces of information that can be either on or off. **Bits!**



Are Computers Intelligent?

Humans have developed a revolutionary machine where millions of tiny switches can represent numbers, characters, and, even, complex ideas. The switches, themselves, are no smarter than light bulbs which can be turned on and off. But, humans have devised ways to combine these switches to represent ideas.

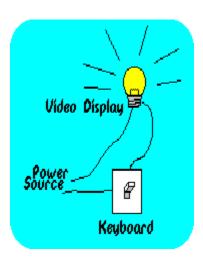
It is not the switches which are intelligent.

It is the humans.





The World's Simplest Computer (Revisited)



Just as humans use switches to turn room lights on and off, they can use millions of switches in computers to represent complex ideas.

Reduced to its simplest form, all a computer is...

...is a switch.