COMPUTER ARCHITECTURE

A computer accepts digital information, processes it per a set of instructions, and provides the results. Computers found on board aircraft are used to provide vehicle and utility control; flight management; navigation, communication and identification; caution and warning; and other essential functions. The basic structure or architecture of a computer is shown in ***Figure 6-1***. It consists of a Central Processing Unit (CPU), Main Memory, and an Input and Output (I/O) system.

The Address and Data Buses provide a pathway for information to flow between the CPU, Main Memory, and the I/O system. They are referred to as parallel busses because they consist of multiple lines that send related parts of information simultaneously. Information, either data or instructions, contained on each parallel line is sent from multiple ports on the CPU to arrive at multiple ports at either the Main Memory or the I/O system, or vice versa.

The Control Bus, which provides control signals between the CPU, Main Memory, and the I/O system, is a serial bus. Buses that connect the I/O system with external input and output devices, such as displays and storage, are also serial buses. Serial buses differ from parallel buses in that one bit of information is transmitted or received one bit at a time on a single line in a serial fashion, rather than information being sent all at once over multiple lines, as is done with a parallel bus. Parallel buses tend to be short and are internal to the computer, where serial buses are used externally throughout the aircraft, as discussed in *sub-module 04*.

The CPU is the "brain" of the computer. It retrieves and executes instructions (i.e., sequence of steps) stored in memory and coordinates the flow of data throughout the computer in a synchronous manner controlled by the clock timing signals. The clock generates a periodic square-wave pulse train used as timing signals. Following a HIGH pulse from the clock, the CPU retrieves data and instructions from Main Memory or the I/O system on the parallel data bus, processes the data, and writes the result back on to the data bus to send to the Main Memory or to the I/O system if, for example, the result is to be displayed or put in to external storage.

The Main Memory stores the information for later access by the CPU. The Input and Output (I/O) system converts the information to other forms to facilitate communication with other onboard computers and to their operators (e.g., flight crew, technicians, etc. through external serial data buses, such as MIL-STD-1553B, ARINC 429, ARINC 629, etc.

BITS, BYTES, AND WORDS

Computers process information using digital data that is coded in single binary digits, known as "bits". A bit can hold 2^1 values, "0" and "1". Eight bits are grouped to form a "byte". A byte can hold 2^8 or 256 values. One thousand bytes, known as a Kilobyte (KB), is equivalent to the amount of information contained on a page in a book.

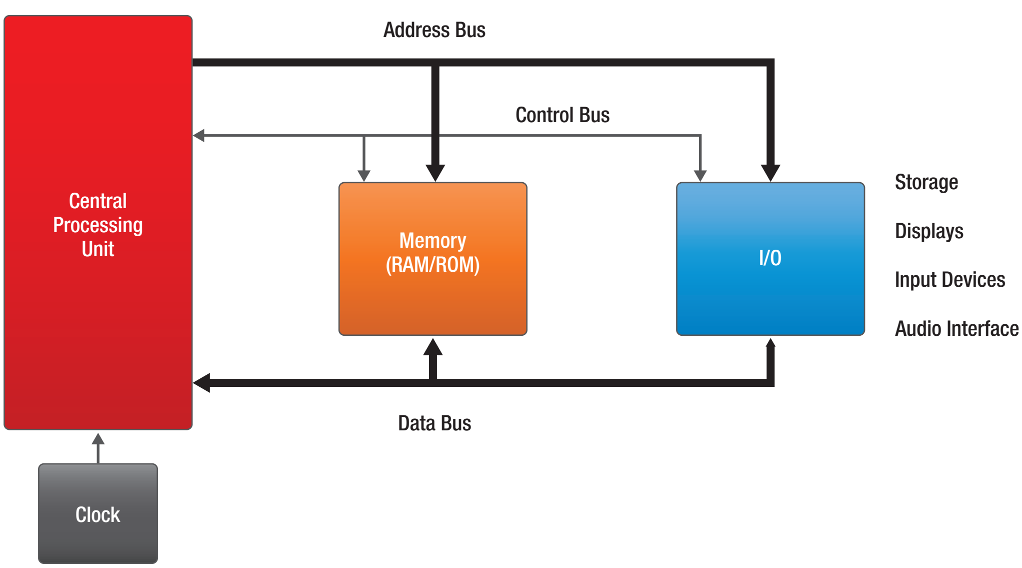

Figure 6-1. Basic computer architecture.

One million bytes, or a Megabyte (MB), is equivalent to the amount of information contained in a large book. A Gigabyte (GB), or one thousand megabytes, is equivalent to the information contained in a two-hour standard-definition movie. A computer's main memory typically stores several GB of information.

1 KB = 1 000 bytes
1 MB = 1 000 KB = 1 000 000 bytes
1 GB = 1 000 MB = 1 000 000 000 bytes

Bits are grouped together to form "words". A word is a basic group of digits treated as a unit by a computer. All modern computers use a multiple of a byte as their word size. The parallel data bus between the CPU and Main Memory is normally the same width as the word size or a multiple of the word size. Thus, a 32-bit computer would typically have 32 data bus lines.

There are two types of words: instructions and data. Instructions retrieved from Main Memory tell the CPU what operation to perform on the data. The instructions are a series of steps, such as "branch" to a new instruction, perform a mathematical operation on the data, write the data back into Main Memory, read from the I/O system, or write to the I/O system. Branch is an instruction that may cause the computer to begin execution of a different instruction sequence.

Figure 6-2, illustrates how a 32-bit instruction word could be formed in a Reduced Instruction Set Computer (RISC). The first 6 bits are reserved for the operations code. With these 6 bits there can be 2^6 or 64 possible instructions. The remaining bits designate the destination register, source register and the immediate value. Registers (R) in the CPU provide a temporary storage area to manipulate the data as per the instruction. An immediate value is stored as part of the instruction which employs it, usually to load into, add to, or subtract from, a number stored in a register.

For example, a typical instruction might be to load the contents of memory at the value in R2, add 8, and place the result into R5. This could be followed by a second instruction to add the contents of R5 and -1, and place the result into R6. A third instruction could be to branch if the previous result is 0.

In addition to instruction words there are data words. The numbers in the data are called operands because they are operated on by the instruction set. Operands are defined as objects of a mathematical operation. There are two basic types of data words: fixed-point and floating-point. Fixed-point data words are most common. Fixed-point is further divided into signed and unsigned data. Unsigned data is the simplest since all the bits in the data word signify the magnitude of the number.

Signed numbers are represented in two's complement form. A two's complement number can be negated (turned negative) by taking the complement of the number and then adding one. For example, bits 111 as an unsigned value would be -1 as a two's-complement value, because 111 would be complemented to 000 and adding one would result in decimal -1 as negated. Likewise, bits 110 as an unsigned value would be -2 as a two's-complement value, bits 101 as an unsigned value would be -3 as a two's-complement value, etc. A 32-bit two's complement signed number can hold values from $-2^{31}-1$ to 2^{31}.

As shown in ***Figure 6-3***, for signed data, the first bit is the sign, whereby logic 0 is plus (+) and logic 1 is minus (-). The remaining bits signify the magnitude of the number. Floating-point data words are used for numbers that need to be expressed in scientific notation. After the sign bit, the 8 bits that follow are used to express the exponent with the remaining 23 bits reserved for the multiplier, as shown in ***Figure 6-4***.

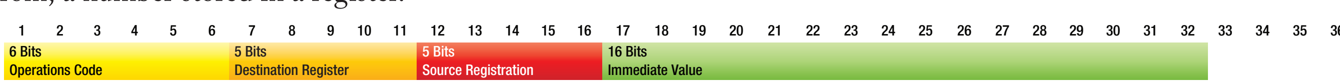

Figure 6-2. 32-bit instruction word format.

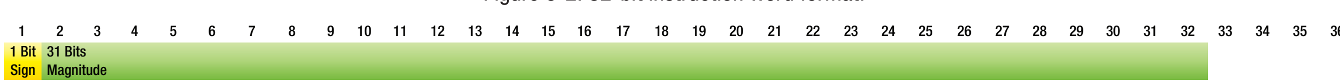

Figure 6-3. 32-bit fixed-point word format.

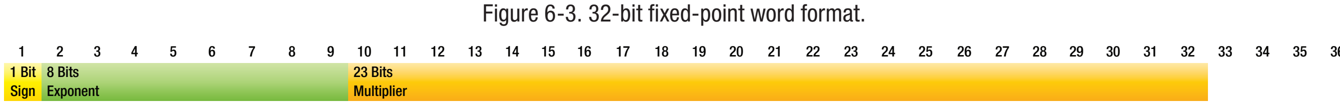

Figure 6-4. 32-bit floating-point word format.

SOFTWARE

Software, unlike hardware, is not a physical entity that can be touched. Instead, software specifies the operations to be performed on data. Data is used as operands by the computer instructions. Instructions can reside in either programs or in algorithms. Think of programs as arithmetic and logic operations described as a sequence of steps that implement an algorithm.

There or two basic types of software, system software and application software. The system software is designed to directly operate the computer hardware. The system software includes the operating system and the I/O device drivers. The operating system instructs the overall computer operation, including start-up, file management, initiating the device drivers, and providing data input and output. It manages resources and provides common services for application software programs that runs on top of the operating system, as shown in ***Figure 6-5***. The application programs provide specific functions (e.g., Microsoft Word on a Personal Computer) to the user under the supervision of the operating system.

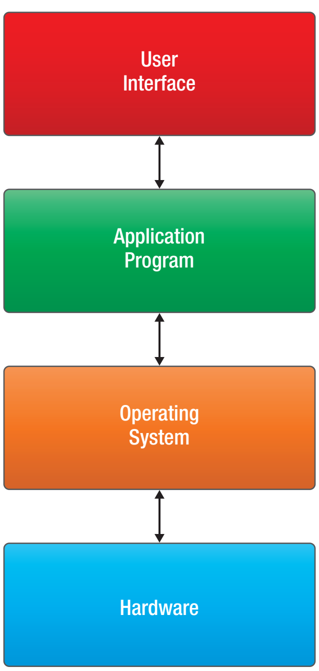

Figure 6-5. Application programs run on top of the operating system.

Software programs are usually written in a high-level programming language, such as C++, which is closer to natural language, and as such, is much easier and more efficient for a programmer to write than machine language. The high-level model-based source code is then translated into executable object code (machine language) via a software program called a compiler or an interpreter. The difference is that a compiler reads the entire source code before it generates the object code, while the interpreter reads one instruction at a time, produces the object code, and executes the instruction before reading the next instruction. Compiled programs execute much faster than interpreted programs, and for this reason, are the only programs used for avionics software development. (***Figure 6-6***)

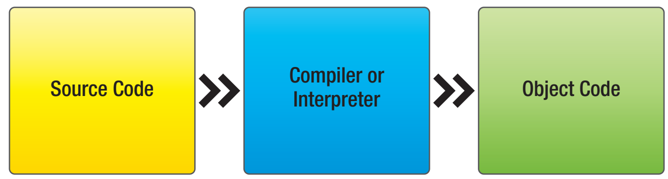

Figure 6-6. High-level source code converted in to object code (machine language).

As an alternative, small compact software programs may be directly written in low-level assembly language, which is a mnemonic representation of machine language (such as using the word "sub" for subtract). Assembly language is translated into machine language using a software program called an assembler. A high-level software command may include several assembly language instructions.

HARDWARE

Figure 6-7, illustrates what is known as the John von Neumann architecture after its discoverer who was the first to present the idea of stored program computers. It consists of a Central Processing Unit (CPU), which contains the Arithmetic Logic Unit (ALU), Control Unit (CU), and Main Memory Unit. It typically interfaces to external secondary memory storage and to input and output devices (called peripherals), such as a keyboard, mouse, display, printer, etc.

CENTRAL PROCESSING UNIT

A CPU microprocessor, such as the Intel Pentium 4 (***Figure 6-8***), retrieves and processes instructions and coordinates the flow of data throughout the computer. It performs math and logic calculations and sends data to and retrieves data from memory and other storage devices. The CPU contains registers, an Arithmetic Logic Unit (ALU), and a Control Unit (CU). The registers provide a temporary storage area to manipulate

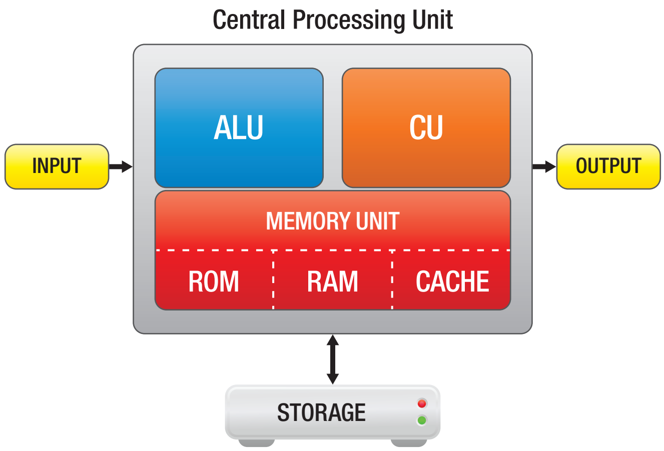

Figure 6-7. Components of the Central Processing Unit.

data during computations. The ALU combines values from the registers, such as adding numbers from different registers, and enters new values in to the registers. Most computer operations are executed in the ALU.

The Control Unit (CU), supervises overall CPU operations, controls the ALU, initiates I/O functions, and decodes instructions to determine whether to add, subtract, multiply, divide, compare, or some other operation. The CU directs the data path between the registers and the ALU to perform a sequence of operations, such as moving an integer from a register to the ALU to execute a given instruction. The CU ensures that an operation is not initiated until the preceding operation is completed.

To execute an instruction, the CPU divides the action to be performed into a sequence of basic steps such that each step will be completed in one clock cycle. The higher the clock speed, the more instructions the computer can execute in any given amount of time. CPU speeds are measured in cycles per second, known as Hertz or Hz. One cycle represents a single task executed, such as adding together two numbers. Computer clock speeds are measured in Gigahertz (GHz), or one billion (10^9) cycles per second. The widely used Intel Pentium 4 runs at clock speeds of 1.3 GHz to 3.8 GHz.

Figure 6-8. Intel Pentium 4 32-bit microprocessor.

MEMORY (RAM, ROM, PROM)

Computer programs and data are stored in memory as coded binary digits (bits). There are two basic types of main memory, Random-Access Memory (RAM) and Read-Only Memory (ROM). The CPU can "randomly" add or remove data from RAM. As such, RAM is typically faster than ROM. The data portion of programs must reside in RAM during their execution. Because of the increased speed of RAM, the instruction portion of most programs is also in RAM. This is different from Read-Only Memory (ROM), which permanently stores data that can't be changed via "random" writes by the CPU. ROM keeps the data stored even after the power has been removed, thus it is termed non-volatile memory. In addition, the CPU has contained within its chip a small RAM cache storage area for frequently used data. The CPU will always access its internal cache memory before retrieving additional data from the main memory or secondary (external storage) memory.

True ROM chips are manufactured with object code stored on the chip. This is known as firmware. A variation of ROM are Programmable Read-Only Memory (PROM) chips that are manufactured blank with no instruction sets. PROMs are programmed after manufacture by plugging them into a PROM programmer where setting of each bit is locked. An ordinary PROM can't be changed once programmed. However, an Erasable PROM (EPROM) can be re-programmed in the field using ultraviolet light, and an Electrically Erasable PROM (EEPROM) can be block erased and byte-written while the computer is running. Flash memory, which is faster than EEPROM, also allows memory to be erased and reprogrammed in to blocks of memory. EEPROM and Flash are used for applications that require periodic updates, such as Operational Flight Programs (OFPs), which control the overall avionic systems operation. Other types of non-volatile memory include secondary data storage, such as rotating hard-disk drives and solid-state flash hard drives.