

Roundup of 29 BASIC
interpreters and compilersBy Namir Clement Shammas, Steve Leibson,
Jay Halcomb, and Stephen Martin

A famous saying goes like this: "Don't kick a man when he is down, he might get up." After plenty of criticisms, BASIC is "getting up." This review invites you not only to compare BASIC packages but to see how BASIC has evolved.

The BASIC programming language was born over two decades ago at Dartmouth College. It took over a decade for it to find its niche in the world of microprocessors and microcomputers. Today almost every microcomputer comes with an implementation of BASIC. For many who want to learn programming, its ease of use has made it the first language chosen. Yet BASIC has had its share of competitors and critics. Some say that it invites the programmer to write spaghetti code riddled with GOTO patches. Even the creators of BASIC, John Kemeny and Thomas Kurtz, recently commented that BASIC implementations for micros have deviated from the original BASIC. The creators of BASIC followed their words with action: their True BASIC conforms to the proposed ANSI BASIC.

BASIC on micros has come a long way. We still remember the PET Commodore, Radio Shack, and Apple II micros with their primitive BASICs. Since then, Microsoft BASIC has emerged as the industry's de facto standard. It is available for all popular micros. Its superset implementation, IBM PC's BASIC A, is also popular.

Other BASIC implementations have also emerged, some offering more sophisticated language aspects. This is seen by some professionals as a strong influence of the Pascal language on BASIC.

In this review we will be looking at BASIC implementations for microcomputers running MS-DOS/PC-DOS, CP/M 86, CP/M, and the Apple Macintosh Finder, an IBM PC with the 8087 chip was used for the first two. A Kaypro II was used to test the CP/M BASIC.

The five tables in this review include the product's general information, Table

1; data types and variables, Table 2; the program flow control, decision making and error handling, Table 3; functions, subroutines, and overlays, Table 4; and file I/O and graphics, Table 5.

Four benchmark test programs were used to compare the speed of execution (see sidebar). They are:

- The Sieve of Eratosthenes
- A test that reverses the sorted order of an array with 1,000 integers
- The Poisson matrix test
- The sine/cosine test

MS-DOS INTERPRETERS
American Planning Corp.
MEGABASIC

This interpreter has evolved over the past several years. American Planning Corp. originally wrote APC BASIC to use under CP/M on NorthStar computers. With the introduction of 8088-based machines, APC BASIC has become APC MEGA-BASIC. The MEGA refers to the ability of the interpreter to use the full 1MB address space of the processor and to its powerful capabilities.

MEGABASIC is a full-featured interpreter designed to run on a full range of machines. It is not limited to the IBM PC. It offers most of the features of advanced BASIC interpreters, including long variable names (250 characters) and extensive program control structures. MEGA-BASIC is somewhat unusual in that it offers BCD arithmetic instead of binary. The accepted myth is that BCD arithmetic is slower, although that does not show in the benchmarks.

The outstanding features of this interpreter make the product very notable. Starting with version 5.0 to allow speedy coding, especially in loop counters. These are 32-bit integers, a departure from most other BASICs, which offer 16-bit integers.

Memory can be divided into as many as 64 workspaces, each with its own program or function library. Programs can access functions, procedures, and programs in other workspaces through a rigorous access protocol. This allows your program to use as much memory as required. Most BASIC language implementations, even in the 8088 world, limit code to 64K. Each array or string can use

up to 64K, allowing large data structures as well as large program capability.

The workspace concept should not be overlooked. It allows the programmer to create function libraries similar to C libraries to be used in later programs. This is the basis for creating software productivity tools.

One reason BASIC is so popular is that it is simple to try out an idea quickly. In MEGABASIC it is even easier. Whenever you see the interpreter prompt, typing "BASIC" will invoke a nested copy of the interpreter, isolating you from your working program and providing you with a clean slate with which to experiment.

Control-C BASIC Interpreter 286

This implementation from Control-C Software Inc. differs greatly from Microsoft BASIC. New commands have been added—some renamed and implemented differently—while others, like the trigonometric, transcendental and square root functions, are completely eliminated.

The BASIC Interpreter uses a librarian to add a new file, remove an old file, compact dead spaces, display the directory, and write library files to disk. BI has a line editor to alter program lines. It has a range of 1 to 9,999, smaller than what MS-BASIC allows.

BI has two data types: numeric and strings. The PRECISION statement sets the number of decimal places, ranging from zero to seven, for the output, operation, and comparison of numeric expressions and variables. Variable names are limited to one character followed by one digit. This hampers program readability and maintenance.

Dimensioning strings signifies assigning the maximum number of characters per single string. A string may be initialized while being dimensioned. This comes in the form of filling it with a string literal, another string variable or string expression.

The only loop construct is the FOR-NEXT loop, with an EXIT TO < line number > option for exiting the loop to a specific line number. Decision making is carried out with the IF statement. The THEN keyword is dropped. Instead