Personal Computer World Benchmark Tests

At the dawn of the home computer age, in the early 1980s, virtually all computers were supplied with a copy of the BASIC programming language built-in. This was because there was little commercial software available at first and the hobbyists who tended to buy these computers were quite happy to fiddle around with their own programs.

Also at this time there was much greater variation in hardware design than in the current (2004) Wintel PC market which made it difficult to predict the relative operating speeds of the different models. The US magazine Kilobaud published a set of standard test programs in 1977 that could be used on any version of BASIC and would give an idea of how fast the computer ran. With slight modifications and the addition of an eighth test these 'benchmark' programs were adopted by the UK magazine Personal Computer World and all their computer reviews were accompanied by a list of the benchmark timings.

Since any computer carries out its instructions extremely quickly the principle was to have it repeat a simple series of instructions a fixed number of times (normally 1000 times) and time how long this took with a stopwatch. Each benchmark program should be run several times and the average time calculated. The time for each benchmark in seconds and the average for all eight was then listed.

Sometimes slight modifications to the program listings were needed to suit the particular dialect of BASIC in the computer being tested but below are listed the benchmarks in outline, together with notes on just what they are intended to show.

Benchmark 1

```
20 FOR k=1 TO 1000
40 NEXT k
```

FOR-NEXT loops are one of the commonest constructs in BASIC and if they are slow then all programs are likely to be slow.

Benchmark 2

```
20 LET k=0
30 LET k=k+1
50 IF k<1000 THEN GOTO 30
```

This has the same effect as benchmark 1 of having the variable k count from 1 to 1000, but with explicit addition and a jump back. This method is generally much slower than the optimised FOR-NEXT loop. Notice that the benchmarks use line numbers and GOTOs. Most BASICs of the period lacked more sophisticated loop constructs.

Benchmark 3

```
20 LET k=0
30 LET k=k+1
40 LET a=k/k*k+k-k
50 IF k<1000 THEN GOTO 30
```

This time some calculations are done inside the loop. This tests the speed of the arithmetic operators and also of accessing variables. For the benchmarks all variables are assumed to be of floating-point type.

Benchmark 4

```
20 LET k=0
30 LET k=k+1
40 LET a=k/2*3+4-5
50 IF k<1000 THEN GOTO 30
```

The same as benchmark 3 except that constants are used for the arithmetic instead of a variable. If the code for looking up values of variables is inefficient then this test will be faster than the previous one.

Benchmark 5

```
20 LET k=0
30 LET k=k+1
40 LET a=k/2*3+4-5
45 GOSUB 700
```

```
50 IF k<1000 THEN GOTO 30
700 RETURN
```

This introduces a subroutine call. It requires the return address to be put on a stack and then the destination of the call to be found, usually by searching through from the beginning of the program. Long programs in early versions of BASIC would make extensive use of subroutines and thus the efficiency of the calling mechanism was important.

Benchmark 6

```
20 LET k=0
25 DIM m(5)
30 LET k=k+1
40 LET a=k/2*3+4-5
45 GOSUB 700
46 FOR l=1 TO 5
48 NEXT l
50 IF k<1000 THEN GOTO 30
700 RETURN
```

This defines a small array at the start and adds another FOR-NEXT loop inside the main loop. This benchmark has little importance on its own but is used as a baseline for benchmark 7.

Benchmark 7

```
20 LET k=0
25 DIM m(5)
30 LET k=k+1
40 LET a=k/2*3+4-5
45 GOSUB 700
46 FOR l=1 TO 5
47 LET m(l)=a
```

```
48 NEXT l
50 IF k<1000 THEN GOTO 30
700 RETURN
```

This assigns a value to each of the array elements every time through the loop. How much slower this is than benchmark 6 indicates the efficiency of array access.

Benchmark 8

```
20 LET k=0
25 DIM m(5)
30 LET k=k+1
40 LET a=k^2
45 LET b=LN(k)
47 LET c=SIN(k)
50 IF k<1000 THEN GOTO 30
```

The final benchmark uses the more advanced mathematical functions of raising to a power (usually implemented using logarithms), calculating natural logarithms and calculating trigonometric sines.

Since the computer cannot store tables of mathematical functions like sin(x) and tan(x) it has to calculate them for a particular value of x as needed. This is done using a polynomial approximation, often involving Chebyshev polynomials, which requires a long series of arithmetic operations. Hence the higher mathematical functions tend to be much slower than simple addition or multiplication. There are various ways to perform the approximation and some systems trade off accuracy for speed.

Of course there are many criticisms that can be made of these benchmarks, particularly from a modern point of view. They are heavily biased towards arithmetic and internal program structure and make no attempt to measure such things as the speed of the screen display or disc access. But it has to be remembered that at the time they were written many computers had a text only display and no disc drive, and the main use for computers was perceived to be as very fast calculators and data processors.

The fact that the benchmark programs are very short means that number 5, measuring the speed of subroutine calls, is unrealistic. The usual way that the BASIC interpreter would find the destination of a subroutine was to look at all the line numbers in turn, starting at the beginning of the program, until it found the one matching the call. In a program consisting of hundreds of lines this would obviously take longer than if the program only had a few lines. Certainly my experience with a Sinclair Spectrum was that if commonly used subroutines were put near the beginning of the program rather than the traditional position at the end, the program would run very noticeably faster.

The benchmark times obviously depend on the speed of the hardware, mainly the processor, and on the way the BASIC language interpreter is written, so computers with the same processor could often give markedly different timings.

Finally the benchmarks only test the running speed of the built-in BASIC language. Most commercial software (especially games) was either written in or compiled into machine code before running and thus bypassed the BASIC interpreter altogether.

Computer magazines still publish benchmark listings but they are much more complex than the ones listed above. Some may measure a particular aspect of the computer's hardware, such as the speed of the graphical display, whilst others try to simulate the normal use of the machine in running typical programs such as spreadsheets and word processors.

In my opinion the current benchmarks are in their own way just as flawed as the old Personal Computer World ones. The graphics benchmarks may be of interest to someone who mainly uses their PC for playing the latest video games but for more general tasks essentially any modern PC is fast enough. In fact, despite the emphasis that the reviews put on differences in performance, it is surprising just how *little* difference there is between a mid-range and a top-end PC. In group tests of PCs there is typically less than a factor of two difference between the fastest and the slowest. Benchmarks of application software such as word processors are all very well, but in practice the limiting factor is likely to be how fast the user can type rather than how fast the computer is.

Despite these criticisms the old PCW benchmarks probably did serve a purpose to the hobbyist market who were writing their own programs, and did demonstrate, as hardware improved from 8-bit to 16-bit then 32-bit processors and clock speeds increased, just how much faster the machines got. They also have the advantage that they were simple enough and general enough that a version of them could be run on almost any platform, and in programming languages other than BASIC, whereas a test of the speed of Microsoft Excel for instance can only be run on a PC with Windows as the operating system.

Some Typical Results

This first table just gives the average time for the eight benchmark tests on a range of systems. A later table lists the individual results. The figures have been taken from published reviews plus practical tests by the author.

The first column gives the computer make and model, the second column lists the microprocessor type and speed, third is the version of BASIC used, the fourth column is the average of the benchmark timings and the final column shows how many times faster than a Sinclair ZX Spectrum that particular computer is. The Spectrum was one of the best selling computers of its day and is therefore a useful baseline.

Computer/System	**Processor & Speed**	**Language**	**Mean Benchmark Time (seconds)** - Lower is better	**Speed relative to ZX Spectrum** - Higher is better
Sinclair ZX80	Z80A / 3.5MHz	Sinclair BASIC	14.60	3.8
Sinclair ZX81 ('Fast' mode)	Z80A / 3.5MHz	Sinclair BASIC	42.80	1.3
Sinclair Spectrum	Z80A / 3.5MHz	Sinclair BASIC	55.62	1.0
Acorn BBC B	6502 / 2MHz	BBC BASIC	14.33	3.9
Commodore 64	6510 / 1MHz	BASIC	33.11	1.7
Atari 800XL	6502 / 2MHz	BASIC	25.71	2.2
Memotech MTX-500	Z80A / 4MHz	MTX BASIC	19.25	2.9
Hewlett Packard Integral PC	MC68000 / 8MHz	HP BASIC	11.96	4.6
ACT Apricot	8086 / 5MHz	Microsoft BASIC	16.68	3.3
Acorn A3010	ARM250 / 12MHz	BBC BASIC V	0.47	117
Windows PC	Celeron / 2400MHz	Microsoft QBASIC	0.047	1174
Casio CFX-9850G calculator	Not known	BASIC-like	65.30	0.85
Hewlett-Packard 39g+ calculator	ARM 9 / 75MHz emulating HP Saturn	HP-BASIC (compiled)	77.7	0.78
MZX Spectrum emulator	2400MHz Celeron emulating 25MHz ARM3 emulating 3.5MHz	Sinclair BASIC	8.32	6.7

	Z80A			
Virtual A5000 emulator	2400MHz Celeron emulating 25MHz ARM3	BBC BASIC V	0.025	2268
FasterPC emulator	2400MHz Celeron emulating 25MHz ARM3 emulating 8MHz? 80186	Microsoft QBASIC	6.88	8.1
STEEM Atari ST emulator (standard ST speed)	2400MHz Celeron emulating 8MHz 68000	HiSoft FirST BASIC (compiled)	0.51	110
STEEM Atari ST emulator (maximum speed)	2400MHz Celeron emulating 128MHz 68000	HiSoft FirST BASIC (compiled)	0.049	1140

Comments on Benchmark Results

The Sinclair ZX80 is several times faster than the Sinclair Spectrum, even though they share the same processor. This is mainly because the ZX80 only performed integer arithmetic rather than floating point (which meant that BM8 could not be tried) and the Z80A microprocessor could do integer arithmetic directly, whereas floating point calculations had to be built up from a series of integer operations, rather like doing long multiplication by hand. Also the implementation of Chebyshev polynomials in the Spectrum was notoriously slow so that benchmark 8 took a particularly long time.

The Spectrum is in fact the slowest of the true computers tested. Only the programmable calculators are slower. The Memotech MTX-500 which also uses a 4MHz Z80A processor is almost three times faster than a Spectrum. The Sinclair BASIC in the Spectrum was based on that in the ZX81, with extra commands added, which was itself derived from the BASIC in the ZX80.

In the majority of early home computers the BASIC and a rudimentary operating system were stored in read-only memory (ROM) and they also required some random-access memory (RAM) as working space. When the ZX80 was launched in 1980 both ROM and RAM were relatively expensive and the machine had only 4 kilobytes of ROM and 1 kilobyte of RAM. It was a squeeze to fit a reasonable version of BASIC into the 4K ROM and also avoid it using up too much of the RAM, which had to be shared with the user's program and the video display.

As a result the BASIC was written for compactness rather than speed which meant writing general-purpose sections of code which could be called one after another to build up a result, where it might have been possible to write a faster routine if it only had to perform a single task.

This design was carried through into the ZX81 (8K ROM, 1K RAM) and ZX Spectrum (16K ROM, 16K or 48K RAM) even though by this time memory prices had dropped sufficiently that it was probably not necessary.

In practice most Spectrums (Spectra?) ended up selling as games machines and since games were almost invariably written in machine code the slowness of the BASIC did not matter. In fact the Z80A processor was probably faster than the 6502 favoured by many of the Spectrum's competitors.

The Hewlett Packard Integral PC (an early portable) and the ACT Apricot (an IBM PC clone) were some of the first to use the new 16-bit microprocessors, the Motorola 68000 and the Intel 8086 respectively. Since these processors could handle two bytes of data at a time as opposed to the single byte at a time of the 8-bit processors used by the earlier machines in the table, and had higher clock speeds, they should have had a significant speed advantage. The results actually show them to be little better than the best of the 8-bit machines. Perhaps this is because the software writers lacked experience with the new processors and were not able to produce optimised routines.

One thing which stands out from the table is how fast BBC BASIC, as used in Acorn's range of computers, is. Microsoft's QBASIC running on a modern PC with a 2.4GHz Intel Celeron processor is exactly 10 times faster than BBC BASIC running on an Acorn A3010 with a 12MHz ARM250 processor. However the clock speed of the Intel processor is 200 times higher than that of the ARM processor, so for the same clock speed it seems that BBC BASIC is really 20 times faster than QBASIC. (And the Q is for Quick!)

The Casio CFX-9850G is one of the current generation of graphing calculators and uniquely I think it has a four colour display. With its 32K of memory and a speed almost matching that of the Sinclair Spectrum it is therefore nearly the equal of a desktop computer of a little over twenty years ago, in a handheld, battery-powered package and at a fraction of the cost in real terms.

The Hewlett-Packard programmable calculator, model hp-39g+, is remarkably slow considering its hardware. It has a 32-bit ARM 9 microprocessor at 75 MHz, which is at least 100 times faster than the 3.5MHz Z80A microprocessor in the Sinclair ZX Spectrum. Also the BASIC in the calculator is compiled before running whereas the Spectrum's is interpreted (see below for explanation). Compiling typically gives a factor of 5 to 10 increase in speed. Overall then the calculator ought to be several hundred times *faster* than the ZX Spectrum, whereas it is actually slightly *slower*.
One reason is that previous models of the hp-39g calculator used a completely different design of microprocessor (known as Saturn), and when Hewlett-Packard replaced this with an ARM 9, rather than rewriting the operating system to run directly on the ARM processor they created a Saturn *emulator*, so that the ARM 9 has to 'pretend' to be a Saturn, and emulators are always inefficient. Even so it is either a very slow emulator or a very bad compiler for the BASIC to be *so* sluggish.

The last few lines in the table give the results for emulators, all running on a modern PC. An emulator is a program which creates a software model of the hardware of another computer, including all the registers in its processor, and this software model can then run the operating system and programs intended for that computer. Emulators avoid the need to keep old hardware running.

Generally emulation is considered a slow process because of the need to convert each instruction in the emulated machine's instruction set into a series of steps which have the same effect on the modelled hardware, and to convert the screen display into the totally different format needed for current computers. In practice though, such is the rate of increase of microprocessor speeds that emulated versions of computers from 10 or more years ago may well run faster than the genuine machine.

For example the emulation of an Acorn A5000 computer gives a benchmark speed around 20 times quicker than an actual Acorn A3010, whereas in reality it would only have been about 2½ times faster. Again notice that BBC BASIC under emulation is quicker than QBASIC being run directly *on the same processor*.

The result for the MZX emulation of a Sinclair Spectrum is particularly impressive. This was run by using a 2.4GHz Intel processor to emulate an Acorn A5000, then the emulated A5000 is in turn emulating a Sinclair Spectrum. Despite having to go through two levels of emulation this still manages to be 6 times faster than a real Spectrum. (It is also a tribute to the accuracy of the emulators that this even works!)

The Atari ST used the same microprocessor as the Hewlett Packard Integral PC. However in the test where the emulated Atari is running at approximately the same speed as a genuine ST it still manages to be over 20 times faster than the Hewlett Packard computer. This is because the BASIC in the HP was *interpreted* which means that each BASIC command had to be read, analysed and converted into the microprocessor's own language every time through the loop. HiSoft BASIC on the Atari by comparison was a *compiled* language. The BASIC commands were converted into microprocessor instructions just once and stored in this form, then could be executed at full speed. One of the reasons the BASIC programming language has a poor reputation amongst professional programmers is that most dialects of it are interpreted and thus inherently slow.

PCW Benchmark Timings

All timings are in seconds.

Computer/ System	Processor & Speed	Language	Year Introduced	BM1	BM2	BM3	BM4	BM5	BM6	BM7	BM8	Mean	Speed relative to ZX

													Spectrum
Hewlett Packard Integral PC	MC68000 / 8MHz	HP BASIC	1985	1.9	3.5	6.9	7.1	8.8	18.3	27.3	21.9	12.0	4.6
ACT Apricot	8086 / 5MHz	Microsoft BASIC	1983	1.6	5.2	10.6	11	12.4	22.9	35.4	34.4	16.7	3.3
Sinclair ZX80	Z80A / 3.5MHz	Sinclair BASIC	1980	1.5	4.7	9.2	9	12.7	25.9	39.2	Not poss	14.6	3.8
Sinclair ZX81 ('Fast' mode)	Z80A / 3.5MHz	Sinclair BASIC	1981	4.5	6.9	16.4	15.8	18.6	49.7	68.5	162	42.8	1.3
Sinclair Spectrum	Z80A / 3.5MHz	Sinclair BASIC	1982	4.4	8.2	20.0	19.2	23.1	53.4	77.6	239.1	55.6	1.0
Windows PC	Celeron / 2400MHz	Microsoft QBASIC	2003	0.007	0.011	0.024	0.024	0.025	0.072	0.096	0.120	0.047	1174
Acorn BBC B	6502 / 2MHz	BBC BASIC	1982	0.8	3.1	8.1	8.7	9.0	13.9	21.2	49.9	14.3	3.9
Commodore 64	6510 / 1MHz	BASIC	1983	1.2	9.3	17.6	19.5	21.0	29.5	47.5	119.3	33.1	1.7
Atari 800XL	6502 / 2MHz	BASIC	1984	2.2	7.3	19.7	24.1	26.3	40.3	60.1	Not done	25.7	2.2
Memotech MTX-500	Z80A / 4MHz	MTX BASIC	1983	1.6	4.8	11.3	11.0	13.2	23.9	43.3	44.9	19.3	2.9
Acorn A3010	ARM250 / 12MHz	BBC BASIC V	1993	0.04	0.15	0.37	0.33	0.35	0.66	1.04	0.85	0.47	117
Casio CFX-9850G calculator	Not known	BASIC-like	2001	4.4	19.9	57.6	60.7	66.3	91.4	106.4	115.7	65.3	0.85
Hewlett-Packard 39g+ calculator	ARM 9 / 75MHz emulating a HP Saturn processor	HP-BASIC (compiled)	2004	1.5	18.0	38.6	37.2	45.7	58.4	292.7	81.5	71.7	0.78
MZX Spectrum emulator	2400MHz Celeron	Sinclair BASIC	1994	0.67	1.24	2.99	2.94	3.73	9.3	13.4	32.3	8.32	6.7

	emulating 25MHz ARM3 emulating 3.5MHz Z80A												
Virtual A5000 emulator	2400MHz Celeron emulating 25MHz ARM3	BBC BASIC V	2002	0.0006	0.0076	0.02	0.01	0.011	0.038	0.053	0.056	0.025	2268
FasterPC emulator	2400MHz Celeron emulating 25MHz ARM3 emulating 8MHz? 80186	Microsoft QBASIC	1994	0.92	1.43	3.35	3.54	3.4	8.7	11.7	22.0	6.88	8.1
STEEM Atari ST emulator (standard ST speed)	2400MHz Celeron emulating 8MHz 68000	HiSoft FirST BASIC (compiled)	1985	0.06	0.07	0.21	0.27	0.3	0.6	1.12	1.43	0.51	110
STEEM Atari ST emulator (maximum speed)	**2400MHz Celeron emulating 128MHz 68000**	HiSoft FirST BASIC (compiled)	2002	0.0032	0.023	0.033	0.037	0.038	0.057	0.089	0.11	0.049	1140