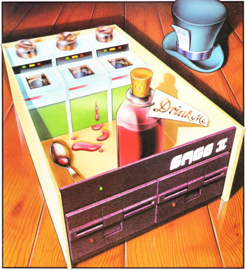
Personal Computer World

BBC GRAPHICS MADE EASY

Canada $2.75/US $2.50/FF 8.80/FL 4.00/SFr 8.00/IR £1.12/
BFr 90.00/Lire 4700/DKr 24.00/NKr 14.55/DM 9.5

FEBRUARY 1983 85p

BRITAIN'S LARGEST SELLING MICRO MAGAZINE



MAINFRAME SPEED IN A MICRO PACKAGE?

PCW Benchtests the super-fast Pascal-based Sage II

BENCHTEST

THE SAGE II

Resident seer Robin Webster prophesies a bright future for this 68000-based system.

There's a lot of revolutionary talk going on right now about the Sage II single-user microsystem.

Listening to the claims made for this Motorola 68000-based machine it seems it could blow the socks off almost everything below the size of a Digital PDP-11: however, the manufacturer's own benchmarks suggest that the lower end PDP-11s have got something to worry about as well.

The Sage II is said to compile Pascal source code at a rate of 1800 lines a minute and load a 20k program from its floppy disk unit in about one second. Indeed, the *PCW* Benchmarks — Pascal and Basic in this case — tend to support the general impression that the Sage makes even tired-out software look good.

But, leaving pure processing speed aside for the moment, the real key to the success of a new computer system is the software it brings within reach of its users. In this sense, the Sage II is somewhat unusual. While Sage Technology sensibly avoided the mistake of producing its very own 'new and better' operating system, it did not follow the crowd by offering the single-user CP/M and MS-DOS operating systems, or the multi-user Unix.

Instead, the primary operating system on the Sage is the portable UCSD p-System marketed by Softech Microsystems. The only other system to be so intimately identified with the p-System is probably Western Digital's Microengine, which has an architecture designed specifically to run p-code.

The notion behind this decision has a lot to do with the fact that there is a large amount of p-System software around. Apple, Texas Instruments, and Digital Equipment users were bitten by the bug a long time ago. Of course, CP/M is still ahead in terms of the sheer number of users and the software packages available, but what they lack in choice p-System users make up for in terms of collective support of the product.

Both in this country and in the US, p-System users have formed a user group called USUS which puts out regular newsletters (at least, as regular as anybody else's) and tend to keep in touch with each other by electronic mail.

But rather than force customers to pick which side of the fence to stand on, Sage Technology is already making arrangements to offer the 68000-based version of CP/M — known as CP/M-68k — and the multi-user Unix operating system within the next few months, as well.

It also appears that Sage Technology intends to go through the whole alphabet of languages (APL, Basic, Cobol, etc) in an attempt to stop its machine from becoming typecast.

The 68000 chip is by no means new — all sorts of companies have been putting their own boxes around it for a couple of years. Its performance has never really been easy to estimate, however, since the majority of machines were designed as multi-user, multi-tasking systems running either Bell Labs' own Unix, or one of the other lookalikes. Good examples would be the Fortune 32:16 and the Wicat.

Sage II designer Rod Coleman told me that, as far as he was concerned, the strengths of his machine are that it had been developed from scratch to be a single-user system based on the Motorola 68000 — a 'chip that will have the biggest share of the 16-bit market'; and that it was somewhat a mixture of Apple Computer technology and Altos packaging, two successful models — to his way of thinking — to follow.

While it might be fairer to compare the Sage's performance with peer machines rather than with the Sirius or IBM Personal Computer, time did not permit a search for a truly comparable machine.

So, before we look at why the Sage is causing quite a stir in the industry, a small qualification on all comparisons. The Sirius and IBM are based on the Intel 8088 chip which has 16-bit internal processing capability. While this may qualify it as a 16-bit chip, all I/O operations must squeeze through the 8088's 8-bit data bus. Therefore, a 16-bit chunk of data has to be taken in or out of the processor in two 8-bit pieces.

The Motorola 68000, however, is a 32-bit internal and 16-bit external chip. This means that it can process twice as much data in one go (32 bits) and can send or receive data in 16-bit lots.

So there is an important distinction to be made between machines like the Sirius and IBM and the Sage — even though they are all referrred to as 16-bit computers in the literature.

Hardware

Having been told about the power of the Sage, it's surprising to see the machine itself, since there's not actually much to look at. In this sense it is a good example to other microcomputer manufacturers who expend more energy on making their machine look substantial or 'good' than they do on making the system provide sensible computing facilities.

The Sage comes in the form of an 18lb unit measuring 5.5in x 12.5in x 16.75in — roughly the same size as an Apple II box with the keyboard missing. But whereas the Apple has only its electronics within the box the Sage comes complete with main memory plus so-called Ramdisk of up to 512k total, two integral disk drives, a cooling fan, and four I/O ports.

The box is a pleasant beige colour, while the front of the disk units are painted black. Above the left-hand drive unit is a multi-colour light-emitting-diode (LED), marked 'processor', which changes colour depending on what the machine is doing: green signifies that the data bus is active; red signifies that it is inactive; other colours mean that the system is processing data. The LED's colour can be controlled by user software.

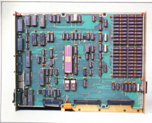
All four I/O ports are arranged at the back of the machine. They are easy to get at since they have been set next to each other at the top edge of the box. From the rear, moving from left to right, there is the reset button, the terminal port, the modem port, the printer port, two blocks of eight microswitches marked 'Group A' and 'Group B', and an IEEE-488 interface. To the left, below the terminal and modem ports, there is the cooling fan exhaust, while to the right under the IEEE-488 interface lurk the on/off switch and the mains power socket.

Also on the back is a printed message informing the user that although the Sage complies with FCC electronic equipment design rules there is a danger that it could cause local electrical disturbances. Further, the message points out that the user will be responsible for the disturbance and might have to take some steps to contain the problem. (There was some radio interference during review sessions.)

The terminal and modem ports are of the usual RS-232C serial type and can be software set to operate at anywhere between 50 and 19,200 baud. The printer port is a Centronics compatible parallel version and in addition to this can be used as a general-purpose I/O or control port.

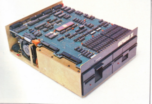
If you're intending to tie up special instrumentation to the Sage, you'll use the IEEE-488 interface which, again, can be

Photography by John Mason



Top view — packed but uncluttered. . .

Ports and microswitches. . .



Main board, sitting neatly above disk drive.

set to do more or less what you want by use of the appropriate software. At the house-keeping level, the interface is controlled by a Texas Instruments TMS9914A chip.

If you're wondering what the Group A and Group B microswitches are for, I'll explain.

The Group A block of eight switches is ostensibly there to enable the system to communicate with terminals having different baud rate capabilities, and the current set-up is always checked out by the system whenever it is switched on. Also, by following the instructions given in the owner's manual, the switches can be set so that rather than booting up in the normal way when switched on, the machine can be made to enter a debugging mode, signified by a '>' prompt.

The Sage Debugging Tool, as it is called, is a powerful program repair utility which offers commands that allow the user to display the contents of all or selected registers, dump the contents of the memory, load in data from floppy disks or other connected devices, set program break points, and modify memory contents. Other than by choice, the SDT is entered whenever errors occur on power-up, or certain system states occur.

Group B switches allow devices to be given a bus address, although this function can be taken over by software if the switches are needed for some other use.

Putting these switches (the only ones I could detect anywhere on the machine) on the outside will do a lot to endear the Sage to those users who have previously had to take the lids off other terminals or computers and search for well-hidden switches to change a baud rate or whatever.

Almost everything about the Sage seems to have been designed so as to make its use and maintenance a simple job — it takes up little space, is easy to move around and requires only the removal of four screws to get inside it. Once the screws are taken out, the sheet metal casing can be lifted off without fear of snagging or stretching any cables, because there just aren't any to be seen hanging out idly.

Inside, all the chips and connecting circuitry are very neatly laid out on one board (measuring about 11.5 x 15.5 inches) which sits atop the disk drives and cooling fan, just under the sheet metal exterior. At first, the fact that the main board was given just enough room to breathe, and no more, made me wonder how extra boards and other devices might be accommodated: was it that customers would have to start hanging things on the outside, a la Tandy?

The answer turned out to be very simple. When extra boards came along, the sheet metal casing would be made a bit taller so that the new boards could stack on top of the one that was already there.

The simplicity and neat design of the Sage's electronics is to be expected, I suppose, since Rod Coleman admits to having built his first arithmetic and logic unit (ALU) out of transistor radio parts when he was in his early teens.

While his company refers to the Sage as being of 'innovative design' or as representing a 'technological and price breakthrough', all technical information, including source software files, is freely published for users wanting to keep it as documentation, or for those more expert who want to pursue their own development ideas. Obviously, this is a great idea from both the company and user point of view, and should shorten the time it will take for a generous amount of useful software (and maybe even hardware) products to appear.

The Sage is based on a 68000 chip set to run at a speed of 8MHz. Compared to this, the Sirius and IBM personals have 8088 chips running at only 5 MHz. This is yet another point in favour of the Sage performance ratings.

The 68000 has a 16-bit external data bus, and, with a 24-bit address bus, can address up to 16 Mbytes, which is nice to know although most users won't require that maximum initially.

Minimum memory on the Sage is 128k, the amount required by the p-System to operate properly. Of this 128k, 64 is allocated to the p-System itself, while the remainder is left as the user area (that's the ideal situation, but the memory areas can be played around with to a certain extent if you know what you're doing). The Sage literature claims that this 128k standard memory really signifies 'The End of the RAM Cram Era' in which 64k was the usual memory allocation.

True, most manufacturers still provide 64k as standard — only the Sirius comes to mind as competitive in that a standard machine comes with 128k memory, two 600k drives, a keyboard, a display, and Microsoft Basic for about £2395. A bottom-end Sage is provided with 128k memory, one 320k disk drive, the p-System and the Pascal, Basic, and Fortran compilers, plus one year's on-site maintenance for around £2870 (VDU and keyboard are extra).

Both machines are expandable: up to 512k of internal RAM for the Sage — more in the case of the Sirius.

Clearly there are pros and cons about buying either machine given these details — but the minimum memory argument is really quite irrelevant. The key thing about the Sage II is that it is quite a different type of machine altogether from the Sirius, offering lots of personal processing power.

And that's the important thing: to decide what level of processing power your applications will need to perform satisfactorily, ations will need to perform satisfactorily, and then to go scouting for the most suitable machine.

One of the key components of the Sage hardware, to my mind, is the combination of the powerful chip with the additional semiconductor memory which can be added, in 128k chunks, and used as Ramdisk — RAM chips configured to appear to the 68000 as if they are in fact a super-fast access disk.

The Benchtest machine came with the full 512k memory (split up into 128k main memory and 384k Ramdisk); one 80-track disk unit capable of storing 640k, and a 320k 40-track disk unit. The terminal used was a Televideo 925, although it could have been a far simpler, and cheaper, type.

The Ramdisk can be used in two ways: you can either boot a system disk directly from the left-hand disk drive into main memory only and treat the Ramdisk as if it were simply another disk drive; or you can boot to the Ramdisk. This second feature can be useful because it allows you to remove the system floppy once the relevant files are loaded. The Ramdisk then acts as if it were a bona fide disk.

This Ramdisk enable/disable ability is handled through a utility program called Sageutil, which is also used to format disks and make copies of the p-System bootstrap loader.

A really useful aspect of the disk configuration element of Sageutil is that it makes it possible for the Sage disk drives to read IBM's 5¼in disk format. Because the machine I was using had only one 80 track drive and one 40 track drive, there were some compatibility problems in copying from one to the other. This was resolved by first loading files into Ramdisk and then down onto the target disk. The only trouble here was that if the Ramdisk was nearly full with files the transient files would be rejected until some file deletions had been made.

The Ramdisk is physically located to the front-right of the circuit board and is made up of four rows of 18 8k chips. Each row of 18 is further sectioned into two lots of nine chips so that eight of them provide 64k of memory while the ninth handles parity checking.

In operation, the Sage is acceptably quiet and consumes only about 70 watts of power — less than a household light-bulb. Therefore it remains cool even when used for long periods. Indeed, the manufacturer claims that the machine could operate without a cooling fan.

UCSD p-System

Although Softech Microsystem's UCSD p-System has been covered in *PCW* before (see three-part article 'P for Perfect'), it's probably worth going over the features of the unusual environment since it currently dictates the way the Sage is used.

The p-System is great for amnesiacs who can never remember what the system command is to copy a file or list a directory. From the moment you boot it up, the p-System always does its best to provide a menu of possible commands that can be used.

The main, or Command level, menu is usually the first thing a user sees and it presents a choice of the following: E(dit, R(un, F(ile, C(omp, L(ink, X(ecute, A(ssem, D(ebug.

You have only to press the first letter of each, so that by typing in 'E' you will be asked for the filename of the file you want to edit and, if it is found on disk or in main memory, another editing-oriented menu will appear. If you want to create a new file, you simply press the return button when asked for a filename and the same editing menu will appear.

'F' for F(ile give you yet another menu which is oriented towards the manipulation of files and file names, as in G(et files from a disk, S(ave files to disk, R(em or delete files, and obtain a file directory L(dir.

Compared to the 'dir', 'pip', and 'era' commands of CP/M, this can be a bit overwhelming at first and there's always the danger of being confused by all the options. Naturally, though, the p-System has its very strong supporters, as do CP/M and Unix.

The p-System editor is very good and can be used as a reliable, if sometimes tricky, word processor when you're not programming with it. It is a full-screen editor in the conventional sense in that it lets you wander wherever you want to on the screen with the cursor. Compared with those on other operating systems, it must be somewhere very near the top of my 'best editors I have known' list.

An interesting thing to note is that Sage Technology was one of the first companies to get its hands on the latest release of the p-System, Version 4.1. The story goes that Rod Coleman contacted Softech about the fact that he was building the Sage and was interested in running the p-System on it. Softech said okay and agreed to give him 'the latest release that was ready' when he had built the hardware. In December last year, the first Sage prototype was up and running and Coleman was lucky enough to get hold of Version 4.1 ahead of virtually everyone else.

Some of the differences between Version 4.1 and the previous p-System releases is that larger files can be created and what are known as subsidiary volumes (a volume is the contents of a device such as floppy disk) became possible.

Not content with just this, Coleman and his associates at Sage Technology are now putting the final touches on what is probably the first multi-user p-System. This is described as being a 'concession' to the concept of multi-user environments and this comment presumably goes for the upcoming Unix system as well.

Having used the p-System on smaller machines like the Apple II and the Sirius, the Sage version is a delight. Because of the 68000/Ramdisk combination, response times for almost everything, except jobs like disk formatting, could be measured in fractions of a second — it's rather like being on one of those powerful timesharing systems.

This kind of performance came in very handy when typing in and compiling the 15

Pascal and eight Basic Benchmarks.

Since a p-code to 68000 native code generator was provided with the Benchtest machine, I've included native code timings as well.

Documentation

All the documentation provided with the Sage was of good quality. The Sage owner's manual had each section clearly labelled, was generally well written and had a comprehensive index at the back — a rare example of a useful piece of documentation.

The p-System documentation is the Softech standard issue, and there's lots of it —system description volumes, compiler volumes, new feature sections, and a general in-depth mass of p-System possibilities. It's all clearly written, though, and is totally necessary if you want to do more than just run off-the-shelf programs.

Potential

Unfortunately, at the time of this review, few software packages were readily available and so I didn't really get a good chance to look at them in detail.

Benchmark timings

Pascal Benchmarks

	p-code 4-word prec	p-code 2-word prec	Native code 2-word prec
magnifier	0.8	0.8	0.1
forloop	10.1	10.2	1.1
whileloop	9.8	9.8	1.0
repeatloop	8.6	8.6	0.9
literalassign	12.1	12.1	1.3
memoryaccess	12.2	22.2	1.3
realarithmetic	24.1	14.9	20.2
realalgebra	22.2	13.7	19.0
vector	21.0	20.9	1.9
equalif	17.2	17.2	1.7
unequalif	16.8	16.8	1.7
noparameters	23.6	24.1	10.1
value	30.9	31.9	23.9
reference	38.3	39.9	37.8
maths	15.1	7.6	7.7

Basic Benchmarks

	p-Code 2-word prec	Native Code
BM1	0.5	0.7
BM2	0.7	1.1
BM3	1.3	2.3
BM4	1.7	3.0
BM5	2.1	8.7
BM6	5.1	8.7
BM7	6.4	11.0
BM8	18.0	—

All timings in seconds. For a full explanation of Benchmark timings, see PCW November 1982.

TDI of Bristol, sole UK distributors of the Sage II (the review machine was borrowed from here) has, however, made great efforts to get UK software up on the machine as quickly as possible. Available right now are the Micromodeller, Microfinesse, and Logicalc financial planning systems; the Aladdin, Logiquest and Beta database systems; the Jarman and Systematics accounting packages; communications software; and word processing software. Also there is an unusual offering from Isis Systems, which has put its Microexpert systems builder onto the Sage. This allows users to create 'intelligent' knowledge bases about some specific field of human expertise which can be used for diagnostic or analysis purposes.

The languages to be made available include Pascal, Basic, Fortran (included in the cost of the machine), microAPL, C, Lisp, Cobol, Forth and the Pascal-like Modula 2.

In terms of operating system choice, CP/M-68K and a Unix system will soon appear, but an event to look forward to is the expected delivery of the Xerox-developed Smalltalk system on the Sage sometime next year. In the meantime, those longing for graphics capabilities will be glad to hear that by the time this issue of *PCW* comes out the Pluto graphics board should have been interfaced as well. With a colour display, this could add another £800 to the cost of the Sage, however.

For those who would rather network than share a Sage, it looks as if the Corvus Omninet local area network system has been adopted by Sage Technology — although this does not rule out other network technologies being offered as well.

Benchmarks

The Benchmarks timings given for native code versus p-code need some explanation, as there are a few different techniques which can be used to produce the native code file.

To produce a native code file, you must use a special software switch — {$N+} — to indicate the point at which native code generation would begin, and a second software switch — {$N−} — to indicate the point at which native code generation should stop.

Typically, the most satisfactory results are obtained if only selected portions of a program (those parts which are processor intensive, for example) are enclosed by these software switches. To reduce the amount of time required to do all the Benchmarks, I simply placed the {$N+} switch at the beginning of all programs to be converted to native code; this had the effect of producing files which were a mixture of native and p-code. In some cases, these files turn out to be larger than the original p-code only file because the native code generator changes only those parts of a program it can make more efficient. So, some native code timings will in fact turn out to be longer than the original p-code!

In the case of the Pascal Benchmarks where this happens, it is because the program is making procedure calls or calls to the floating-point interpreter. In the Basic Benchmarks, the problem is more a case of the Basic language not knowing whether a number is a real or an integer.

The point to bear in mind is that the native code generator is not a Philosopher's Stone to be relied on to magically turn slow, badly written programs into superfast numbercrunchers.

Used wisely, it is extremely useful — used carelessly, it is really not very helpful at all.

Conclusions

It's a little difficult to assess a machine properly without having a good selection of application software to work with, but in the time I've given desk-space to it, the Sage II has really impressed me. Its design is practical, it's easy to use, and it never gave any sort of trouble. With regards to the price, it may be a little expensive when compared with some of the mass-market machines around right now, but it's probably worth it if it's power you need.

I'm also impressed by the way Sage Technology hasn't let any grass grow under its feet — just as I was finishing this review, the Sage IV was announced at the Comdex show in Las Vegas. This new model features half a megabyte of main memory, six serial ports and a winchester controller capable of handling five winchester units. It will probably be smaller in size than the Sage II since it will use half-height floppy or winchester disk units. A graphics board is currently under development, too. This shouldn't affect the life of the Sage II, said a TDI spokesman, because the Sage IV was really just an extension to the family, and not a replacement for the II.

Prices

Smallest Sage II comes with 128k memory, one 40-track drive giving 320k storage, UCSD p-System version 4.1 operating system, Pascal, Basic and Fortran compilers, 68000 Macro Assembler, and one year's onsite service: £2870 (inc VAT).

Top of the line Sage II with 512k memory, two 80 track drives, and software as above: £4594 (inc VAT).

Our thanks to TDI (tel: 0272 742796) for the loan of the test machine.

Technical specifications

Processor	Motorola 68000, 8 MHz
ROM	8k boot ROM, expandable to 64k
RAM	Minimum 128k, expandable up to 512k (including Ramdisk).
I/O ports	Terminal, Modem, Printer ports (RS-232) and IEEE-488 interface.
O/S	UCSD p-System
Languages	Pascal, Basic, Fortran, microAPL, Lisp, Modula 2. Others, including Cobol, C, and Forth, to be added soon.