The RISC Workstation

Dominic Sweetman
Systems Architect
Whitechapel Workstations
75 Whitechapel Road
London E1 1DU
ENGLAND.

For several years it has been fashionable for computer science pundits to predict the end of the road for computers as we have known them; only non-Von Neumann machines offered radical improvements in the performance of VLSI based systems. Now that RISC is the buzz-word of 1987, we can return the treatises on parallel processing to the shelves and enjoy the consequences of the 20-fold increase in the performance of Von Neumann microprocessors between 1983 and 1988.

The new RISC micros are in some ways like the writable microcode machines which were popular ten years ago, prompting the rediscovery of some old insights about the architecture of interactive computers. This talk celebrates the revival of an architecture in which a splendidly fast CPU does (nearly) everything. It worked in the early '70s: this talk describes how it works now.

Dominic Sweetman has been with Whitechapel since its founding in 1983, and is the specifier and architect of their new RISC workstation. His career has led him from support programming through communications, local area networks, operating systems, window managers and hardware design.

RISC - a new generation of microprocessors

RISC is becoming one of the most abused acronyms in the computing industry, but that should not be allowed to obscure the most significant set of ideas in computer architecture since the development of microprocessors. The word comes from ''Reduced Instruction Set Computer'', stressing that RISC machines gain performance by having a simple set of instructions which they execute particularly fast. RISCs are already showing 10 times the performance of the classic DEC VAX11/780 minicomputer, and it seems likely that the performance which can be squeezed from a single chip will continue to double every 12 to 18 months for quite a while yet.

Getting more power

There are two ways to get more power out of a processor; make it do higher-level operations in its machine instructions, or make it do simple things faster and let the software handle the complications. Even in the pioneering days of computing the UK workers (notably Alan Turing at Manchester) were designing simple, streamlined hardware while the US teams built circuits to handle sophisticated operations at each bite. There have always been successes from both camps. Through the late 60s and 70s enormously successful machines such as IBM 370 mainframes and DEC VAX minicomputers offered powerful and complex instructions; more recently systems such as the Cray supercomputers or the more modest Pyramid minis achieved unexpectedly high performance by simplifying.

Every microprocessor architecture from the 8086 on has been hailed (at least by its manufacturer) as ''designed to support high-level languages'', but researchers have known for a long time that language compilers are happier working with simple, regular instruction sets. It is the compiler argument which probably gave the initial impetus to the industry's most recent flirtation with simplified computers.

The new RISC chips are not characterised by a single identifiable architectural feature - they do not even have particularly small instruction sets - but they have a lot in common. The term has come to be associated with a particular style of processor architecture and a whole collection of related innovations. Together these features seem to result in about 4-5 times the performance which can be squeezed from an older instruction set. At present there is no chip which benefits from both RISC architecture and really state-of-the-art silicon fabrication; a RISC made as cleverly as an 80386 would probably be twice as fast as the current 7-10 Mip RISC products made by MIPS and Sun.

Why are RISCS so much faster?

The point of this paper will be lost on you if you find it hard to believe these extravagant performance claims. Why are RISCs so much faster? Two important reasons are:

- *The older micros are slower than they need to be.*

  Established micros are much slower than they need to be because they are *microcoded.* Microcoded CPUs execute their (often complex) instructions by an internal microprogram which runs on a very fast and simple processor and co-ordinates the functions of the various bits of the chip.

  The individual elements of the chip are small enough to design correctly, and the microcode can be debugged. The original VAX, 68000 and 8086 designs were microcoded because it was the only technique available at the time to make the design task manageable.

  On average the latest 32-bit micros take 4 or 5 internal micro-instructions to complete each machine code instruction.

- *RISCs are pipelined.*

  A car factory may turn out 500 cars a day, but no-one should expect it to be able to make one car from start to finish in three minutes. A pipelined processor contains an assembly line for performing instructions; a modern 10 Mip RISC will have four or five instructions somewhere between started and completed. Just as in the factory raw materials from suppliers need to be ordered in advance, a pipelined processor needs to be kept fed with instructions and data. A smart compiler which schedules instructions to avoid hold-ups is important.

  It is possible to pipeline non-RISC designs, but it appears to be very difficult to pipeline their crucial arithmetic operations and register transfers. Maybe in a few years time further improvements in chip design tools will enable pipelined complex-instruction set computers to make a comeback, but for the moment the big gains of pipelining are for RISCs only.

The RISC's pipeline and its lack of microprogramming enable it to move the production line along one place with every tick of the clock, and therefore to deliver 4-5 times the performance of a similar-speed CISC.

An enthusiast's view of Workstations

I share with most of those in the workstation business the view that interaction between human beings and computer systems is best carried out by a fast general-purpose single-user system located on its user's desk, connected by a high-speed network to shared data and resources. The essence of the scheme is that users want to share data, but want their own processor to run interactive applications.

RISC technology in workstations

Interactive high-resolution graphics and the use of very high level languages to implement the user interface require computing power which is beyond the reach of 68020s and 80386s, so the workstation market is reacting to RISC technology with great enthusiasm. IBM and HP were in there very early, recently joined by Silicon Graphics and the market leaders Sun. Most of these products are firmly aimed at the most profitable high-end sector, but Whitechapel Workstations have announced a 10 Mip desktop targetted at the high-volume OEM market. Our product will not be unique for long, and as high-end performance doubles and redoubles RISCs will find their way onto every desk.

Up to now the workstation builders have mostly used 68000 family processors (though DEC used microVAXes). The use of a common instruction set has never been as important in the workstation market as it is for PCs, but it was still comforting to be able to offer binary compatibility across the range.

An open environment

The price exacted for the performance of RISC is an instruction set which must be changed as performance increases, and this is pushing this young market into a second generation. The new machines will operate in a new and much more open environment:

- *Independence from CPU architecture.*

  Since instruction sets will be peculiar, different and unfriendly, the new machines will always be programmed in high-level languages, with systems software written in that ubiquitous portable assembler, C. Neither workstation system manufacturers nor applications developers will be tied to the CPU architecture.

- *Evolution of standards*

  Networking, graphics and resource sharing systems will be provided by a number of components which are or will become de-facto industry standards. Favourites are Unix, the Arpanet TCP/IP protocols, Sun's NFS shared file system and the X window system.

- *Mixed-vendor networks*

  Heterogenous networks of workstation and shared-resource systems will be made possible, and will become increasingly common.

- *Application portability*

  Portability between the systems will taken for granted; it will be possible to take an application, re-compile it, and run it with the same level of confidence that you now have that a PC clone will run your favourite PC application. There is already a significant subculture of large pieces of software, in the public domain as source code, which are running on many different CPUs. Programs such as the interactive adventure game *hack* would not survive the disapprobation of system managers if they took many days to port! What games programmers can do now, applications specialists can and will do tomorrow.

- *Special-purpose systems*

  Systems will be developed to fill specialist requirements on the network; file servers, print servers, array processors, CPUs with large physical memories, and gateways are examples.

- *Manufacturers will support standards*

  The construction of standard interfaces is in the interest of most of the current market leaders. All the workstation vendors are vulnerable to the entry of the big computer companies into the market, and setting standards quickly makes them much safer. The availability of well supported *de-facto* standards will free applications from becoming attached to particular hardware by their software interfaces. Since any application will run on any workstation, differentiation will be more by performance than capability. Workstations are likely to become more universal, even interchangeable.

For the next few years these factors will make workstation manufacturers concentrate on improving CPU performance to a greater extent than would be necessary or possible for DOS-compatible PCs or superminicomputers. The result will be a rapid improvement of the price/performance ratio, rather like the development from the first PCs to the modern PC/AT clone; and that has seen performance increase ten times while prices have dropped.

The advantages of uniprocessors

It is more fashionable to be a multiprocessor than a uniprocessor. Every systems architect should know that a diagram with boxes labelled ''intelligent I/O subsystem'', each containing its own microprocessor, is a *sine qua non*. Why should we resist this trend?

The Bugatti of the workstation world was surely the Xerox Alto, built in the early 70's for the researchers at the Palo Alto Research Center. It was a most elegant design which did much to stimulate the seminal work carried out at PARC, delivering minicomputer performance and unmatched interactive graphics in a package which could be replicated for each researcher.

The Alto design was simplified by a radical principle; a single high-speed microcontroller did everything. It had a writable control store, so the instruction set could be altered dynamically to support different programming environments. The bit-slice microcontroller was at that time able to absorb all the bandwidth of an affordable memory system, so the controller performed all i/o; an independent DMA channel or graphics processor would not have speeded the system up much because of memory contention.

The RISC in a uniprocessor system

RISC processors today do a similar job to the bit-slice controllers of a dozen years ago and like all good radical ideas, there is life in the Alto concept yet. Now that personal computer peripheral controllers are so readily available it doesn't make much sense to fight through a forest of microcode to run the discs, but the microprocessor has big advantages for controlling the screen and input devices.

Graphics without accelerators

A straightforward frame buffer with all graphics functions, written in a portable high-level language, running on the main processor has some nice attributes:

- The rendering speed is now competitive with last year's dedicated graphics accelerators; as CPU performance doubles and redoubles we expect such systems to outpace all except the most extravagant graphics hardware.
- Provides better than 100,000 co-ordinate transformations per second.
- Supports many different graphics models; which is just as well since standardisation is not quite complete. The two favourites are X windows (just being frozen, but with discussions still at an early stage for the 3D standard) and the extended PostScript of NeWS. PHIGS 3D is still under discussion.

  When the main CPU provides all the graphics, the software runs in the same rich environment as all applications, and the discipline of using a general-purpose instruction set makes it unlikely that evolution of computer graphics techniques will cause trouble. By contrast, when existing workstations run standard graphics software such as X Windows

developments will leave it useless. It is surprising how many workstation software/hardware combinations leave an expensive and complex graphics accelerator by-passed.

The resulting workstation is cheap enough to be dedicated to one person, and consumes little enough power to go on his or her desk.

Putting it all together

The new Whitechapel product is (as far as we know) the first desktop, mid-range workstation to exploit a state-of-the-art RISC processor. It uses a very high performance microprocessor unit, the MIPS R2000. Running at 16 MHz with 24K of 25nS cache memory and a fast tightly-coupled main memory system, the integer performance is roughly 10 Mips, using the DEC VAX11/780 as a 1 Mip yardstick. Performance is gained by pruning unnecessary functionality rather than by the use of exotic components, with the whole system tuned for interaction.

Perhaps by the next generation it will make good sense to connect all peripherals via the network; meanwhile this workstation provides local disc storage and interfaces other devices by using VLSI peripheral controllers developed for the PC market. These components are cheap, small and don't get too hot.

You should be able to see the result soon.

Crystal ball gazing

There seems little doubt that careful application of now well-known RISC principles and relatively unproblematic developments in CMOS silicon technology will carry the performance of microprocessors to between 50 and 100 Mips. What on earth can we do with all that power?

There are a number of ways to exploit raw CPU performance to make the whole system work better:

- *Increase user-visible performance*

  Large interactive applications are rarely fast enough.
- *Run more complicated applications.*

  There are many common CAD tasks, such as the automated routing of printed circuit board tracks, for which software exists but whose complexity defeats current workstation and minicomputer speeds.
- *Trade speed for simplicity*

Burn performance in the interest of making software easier to write and debug. The user benefits from better software available sooner. A good example of this is how Xerox used exotic workstations to permit software prototyping in Smalltalk and other very high level languages; a 100 Mip workstation will allow the construction of production user interfaces in languages like Smalltalk.

Anyone who has ever wrestled with making a large application user-friendly knows how desirable this is.

- *Simplify the hardware*

  Use the faster processor to take on a larger range of system functionp, thus keeping total system costs down. The Apple Macintosh is a beautiful example of how this can be done.

The immediate future

When, as now, processor performance is increasing really fast you can do all these things at once. Over the next couple of years you will see the following trends in the workstation market:

- Users will come to expect better interactive performance from large software systems.
- The migration of large processor-intensive applications from minicomputers to desktop workstations or specialised low-cost networked computing engines will continue.
- Software developers will expect to be able to use flexible high-level languages for interactive application development, and have the hardware so fast that the result does not have to be re-coded to make a product.
- Only very special-purpose workstations will retain complex hardware accelerators; a fast CPU cluster and simple peripherals will suffice for the great majority of applications. Animation and solid modelling will be usable on general-purpose computers.

In the near future, the workstation market will continue to develop with extraordinary speed. It is well worth watching; what is being prototyped here is the interface between computers and their users, and by the mid 90s it will have as profound an effect on our use of computers as the rise of the PC.

Acknowledgements

The thoughts in this paper have mostly been fermented by talking and working with colleagues, not by reading. The greatest debt is to Bob Newman, founder of Whitechapel and the architect of its first product, the MG-1 workstation. Bob proved that it was possible to build a workstation east of California and his rate of innovation gave us something to live up to. There are many others, but not enough space to list them.