QL corner

Sid Martin and Timothy Green dish up a comprehensive guide to micro-processors and news of a new QDos-compatible operating system

The QL emulator for the Atari ST is to be upgraded with new device drivers and system software, developed by former Sinclair QL designer Tony Tebby.

Tebby has programmed the Futura Datasenter QL emulator for over a year, and has developed a new set of device control software for use within his own company, QJump.

"We are well on the way to having a completely legitimate QDos compatible operating system which will work satisfactorily on the Atári", he told members of the *Quanta* user group. The new system is based on *SMS-2*, the unfinished ROM software developed for Sandy's ill-fated Futura QL clone (not related to the Datasenter).

SMS-2 was announced three years ago, yet it is still incomplete, following the collapse of QL hardware firm Sandy, and the demise of Power Computing's plans to run the system on its 68010/20 PC processor upgrade board (*Shopper* issue 6).

It will not be a 'carbon copy' of QDos and the SuperBasic interpreter, but Tebby feels that there should be few problems running well-written QL software. Work may be "well under way", but Tebby is still cagey about details: "system developers will be able to purchase specifications of the new *SMS-2* utilities and features, but the details of these facilities are likely to change over the course of the development."

QJump has divided the project into seven stages, culminating in a ROM-based emulator that should run on any ST with at least a megabyte of RAM, using ST display modes and colours. This will be a major boost for QL users and enthusiasts, whose allegiance is to Tebby's QDos design rather than David Carlin's quirky QL hardware; but it will be a blow to lose QL features like the network, cartridge and expansion ports.

In future upgrades, leading up to the launch of SMS-2, Tebby intends to boost the emulator with a full network file server running through custom hardware or the ST's MIDI ports – like MGT's SAM network – plus a driver for Atari ACSI hard disks.

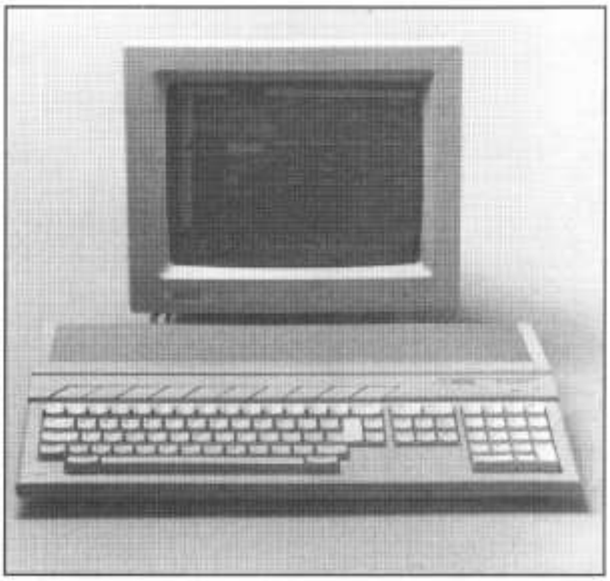
In the short term, QJump is supplying sets of device drivers to those who have already purchased the Norwegian emulator, which normally comes with unlicensed software, including a patched copy of QDos and device drivers from the Sandy SuperQboard. Most of this software was originally written by Tebby, although Amstrad owns the rights to the QL ROM and stubbornly refuses to license anyone else to use it.

The new drivers support the ST keyboard, serial and parallel ports, mouse, and sound chip – via BEEP, with most of the QL parameters, but alas no volume control. QDos-style floppy and RAM disk drivers will be provided, but the screen control code remains unchanged from the original emulator – QL code controlling a ZX-8301 display chip.

The ST's full 68000 processor means that typical QL code runs two to three times faster than on Sinclair hardware. You can speed up the display further with QL software upgrades.

The new ST drivers are 'bundled' with an extended version of Tebby's best-selling *Toolkit 2*, *Hotkey System 2* (from QRAM) and improved directory management, derived from work on the Miracle Systems hard disk.

The current bundle costs £29.90, which entitles the purchaser to up to four intermediate upgrades and a discount of at least £15 on the full *SMS-2* system, when that becomes available. If all goes well, Tebby says, the software-only QL emulator could be with us before the end of the year – but you'd be ill-advised to hold your breath waiting.



ST or QL? Watch out for for the upgraded QL emulator for the ST.

6800X comparisons

Shopper reader J. E Morgan has written in to ask about the speed difference between the Atari's 68000 processor and the QL's 68008. As a rough guide, most QL programs run two to three times faster on an ST – but this is a very vague measure, because the speed varies a great deal depending on the exact instructions being executed.

You need to know three things to work out the speed differential for any instruction. You need the bus bandwidth of the machine – the speed at which bytes or words can be read or written from memory. You must know the clock speed of the processor – 7.5 MHz on the QL, 8 MHz on the ST. (1 MHz means one million timing pulses per second.)

Finally, you need timing data for that particular instruction, which comes from the Motorola processor specification or a comprehensive 68000-programming guide; try *M68000 Programmer's Reference Guide* (Motorola Ltd.), *MC68000 16-bit microprocessor User's Manual* (Prentice-Hall), or *68000 Assembly Language Programming* by Kane, Hawkins and Leventhal (Osborne/McGraw-Hill).

Instruction timings are normally expressed as a number, followed by two other numbers in brackets. For instance, the CLR.W D0 instruction timing is written 8(2/0) in 68008 data, and 4(1/0) for the 68000.

The first number is the number of clock pulses that the processor would need in order to process the instruction, assuming fast RAM. The other figures are only needed if the RAM is slower than normal.

The first number in brackets is the number of memory read cycles performed in the course of the instruction. The second bracketed number is the number of memory write cycles. Thus the QL executes CLR.W D0 by reading two bytes, where the 68000 reads one word.

The key difference between the 68008 and the 68000 is the interface with memory. The 68000 reads two bytes at a time, while the 68008 reads one. In either case, the processor must wait a minimum of four clocks before data is available.

▶103

SuperBasic Explained

The QL's elegant, extensible SuperBasic interpreter has always suffered from a mediocre manual. The Basic sections of Sinclair's *QL User Guide* were written in a rush by Roy Atherton and Stephen Berry, and skimp over or miss out some of the best features of the language.

In 1985 McGraw-Hill UK came close to correcting this problem when it published *QL SuperBasic – The Definitive Handbook*, written by the designer of SuperBasic, Jan Jones. Despite some printing errors, it was required reading for keen QL programmers – clearly written yet detailed, with dozens of examples and a 50-page index.

The handbook is much more than a SuperBasic tutorial – it includes detailed information about defaults, array-handling, syntax analysis and internal data structures. It fills in the gaps in the *QL User Guide* and explains the structure and design philosophy of the language – providing reasons for apparent anomalies.

Unfortunately this book was always hard to find, and has been out of print for several years. Now Quanta has got permission to re-print it, with corrections and a new chapter covering the previously un-documented error- and event-trapping keywords in 'JS' and 'MG' ROMs.

Sinclair Research persuaded Jan to leave this information out of the McGraw-Hill edition, on the grounds that the keywords were bugged on some ROMs and unavailable on others. Since then some of the faults have been fixed by *SuperToolkit 2*, but it has been hard to isolate the others in the absence of proper documentation.

QL SuperBasic – The Definitive Handbook costs £8 to Quanta members, who can place orders with the Secretary, Phil Borman, on (0472) 49850; non-members can order copies from Sector Software on (0772) 454328.

It follows that the 68000 can access memory, at best, twice as fast as a 68008 running at the same clock speed. The ST has 2 * 8 / 7.5 times the bus bandwidth of a fast QL – about 113 per cent more.

Memory access on a standard QL may happen at only half the maximum speed, as the ZX-8301 video chip has priority over the processor and often needs access to internal RAM. Atari runs its RAM at full speed, thanks to a 16MHz memory management unit which interleaves memory accesses from the processor and video chip so both can appear to access RAM at once. So the 520ST can be 327 per cent faster than a QL with slow RAM.

In practice, measurements show a smaller difference. QL ROM runs at full 68008 speed – about 47 per cent of the speed of ST ROM. But many instructions are dominated by the time taken by calculations inside the processor, not memory access, and the 68008 is just as fast as the 68000 once data is inside.

To take an extreme example, the 68000 DIVS instruction takes a maximum of 158 clocks, yet it only involves reading one instruction word. The QL's 68008 can do the same work in 162 clocks – eight clocks to read the instruction word, plus the same 154 while the internal 'arithmetic and logic unit' works out the result. Even in slow RAM, the QL takes only 170 clocks to perform a signed division, making it just 15 per cent slower than an ST.

So the performance difference between QL and ST depends very much on the operations you are performing. An un-expanded QL runs at between 23 and 85 per cent of the speed of an ST whereas an expanded QL runs between 47 and 91 per cent as fast as the Atari.

It's just possible to find a program that the QL will run faster than an Amiga, but it would be unfair to base comparisons on nothing but complex instructions like DIVS! In practice, a real program will use a mixture of simple and complex instructions, and run at a speed somewhere between the two factors.

The QL handles floating-point maths at impressive speed, using ROM code that makes heavy use of the ALU, but is relatively weedy if you just want to copy a block of data from one place to another. It's best to put things where you want them straight away.

There are times when a 16- or even 32-bit processor may be no faster than an 8-bit one, as we shall explain later. For the time being, here's an example of the difference in code style between QL ROM and RAM. You may like to skip the next section if you don't speak assembly language – but, what the hell, you've got this far...

Worked example

It's very important to keep QL code concise, because the time taken to read the instructions is often longer than the time spent handling the data. This example shows the interaction between time spent using the ALU and time spent waiting for RAM. If you're running code in slow RAM, it's faster to multiply by 18 with MULU #18,D0 than to work out with shifts and adds:

```
MOVE.W  D0,D7
ASL.W   #3,D0
ADD.W   D0,D7
ADD.W   D0,D0
```

However, if D7 is spare, the above code works faster than MULU if the instructions are in ROM or fast RAM. At best, the MULU takes 42 clocks plus two for each bit that is '1' in the multiplication factor. #18 is 00010010 in binary, so MULU #18,D0 takes 46 clocks if running in ROM on a 68008.

Four instruction bytes must be read: one word of code, plus the operand word '#18'. If the RAM is half as fast as ROM – about right for a 128K QL – the MULU takes 46+4*4 = 62 clocks in slow RAM.

The four-line version takes 40 clocks in QL ROM, but eight instruction bytes must be read – 16 bits per line. If you run the code in slow RAM this can push the effective execution time up to 72 clocks – eight for internal processing in the LSL.W instruction, plus 64 clocks to read eight bytes at half ROM speed.

On the QL, it often makes sense to use complex instructions, as internal operations run at full 68000 speed. Your aim should be to reduce the amount of time the processor spends reading instructions, as this reduces the load on the bus and increases the chance that the 68008 will be able to get at data in RAM without waiting.

Pipelines and caches

Things get more complicated when branches and jumps are considered, because Motorola processors use a 'pipeline' to read instructions. The 68000 family reads instructions one word ahead of the one that is currently being executed, so it can start on the next instruction as soon as it has with the previous one.

The Z80 and 8088 work by reading an instruction, decoding it, executing it, and going for the next one. The 6502, 6809 and 68000 read the next code word while they are decoding and executing the current one.

In general this 'pipeline' speeds things up, but problems occur if you use a single-word short branch instruction to jump to another routine. The word in the pipeline must be discarded, because it is no longer needed – we're not going to execute the word after the branch, but a word somewhere else.

For this reason a one-word branch instruction takes as much time as a two- word branch. The long branch uses the 'next word' in the pipeline, which the short branch chucks away. This makes it hard to write tight loops on the 68000, because the branch can easily take longer than the instruction it is controlling!

The 68010 processor addresses this problem. It is much like the 68000, except it keeps a small backward pipeline so that it does not need to re-fetch single instructions in tight loops. It executes almost all 68000 instructions, plus a few extras.

The 32-bit 68020 goes one better, with an on-chip 'cache' that stores the last 64 long word instructions fetched. It can access data in the cache 33 per cent faster than external RAM, and is capable of reading cached code and RAM data at the same time. The 68020 can run 68010 programs, but has extra 32- and 64-bit instructions of its own.

Other makes

There is little point in comparing the clock speeds of processors from different families – say, Intel and Motorola – because the amount of work done for each clock pulse varies depending on

Hardware News

Rebel Electronics was still 'finishing' the ROM software for its £200 QL hard-disk interface as this issue went to press; if all goes to plan we should review it here next month.

▶107

the processor design.

For instance, the 8086 and 6809 take a minimum of two clock pulses per instruction, whereas the Z80 takes at least four. The 68000 takes four clocks or more; all its instructions are at least 16 bits long, and it takes four clocks to read a single word from memory. It takes at least eight clocks per instruction, as it must fetch each one as two eight bit bytes.

The deeply unfashionable 6502 would probably have been called a RISC processor if it had been designed 10 years later. Many of its instructions can be executed in a single clock pulse, but it 'cheats' by using a 'two-phase' clock – in other words, it expects a second timing pulse, on another pin, in the middle of each full clock interval.

Of course these figures just compare the speed of simple operations, and take no account of the range and power of more complicated instructions. The 68000 family uses 16-bit instructions, giving 65,536 possibilities – rather than the 256 on processors that use eight-bit instructions. The 68000 can do many things in one step which would require two for a ma byte-addressed code like a Z80 or 8088. That said, the Z80 can do some things in half a step, compared with a 68008.

If your program manipulates byte data, like ASCII text, there may be little advantage in running it on a 16-bit processor. It takes as long to read a byte on a 16-bit processor as it does to read a 16-bit word. The trade-off is even worse on 32-bit microprocessors, which usually read 32 bits – and discard 24 – every time they are asked to fetch a byte.

The wiring inside the processor also affects the speed, with effects that vary depending on the details of the processor design. The 6809 processor contains a hardware multiplier, unlike the other 'eight-bit' 6502 or Z80 chips, so it can multiply in one instruction, rather than a loop of 20 or more.

The 68000 and 8086 families support hardware division as well – but the speed of these complex instructions varies depending on the layout of silicon inside the chip. The same is often true of simpler instructions.

THOR POINTS

The Nordic Thor saga has taken another strange turn, with the news that former CST designer David Oliver is to leave Denmark after less than a year. Work on the Thor XVI computer is intended to continue, but David will no longer be involved in the day-to-day running of the project.

This news reached us indirectly from two channels, shortly before this issue went to press. We have unsuccessfully attempted to contact Thor International and its associate company Dansoft, formerly Sinclair's Danish distributor. Our calls were intercepted by the Danish phone company, which undertook to pass on our enquiries and phone number. A week on – nothing.

We called the published number for Thornton Design Services, the British design partnership in Denmark, but were told that it had left that address, leaving no forwarding number.

The long-awaited order from the Russian government has yet to materialise. We hear that there has been a row over the amount of effort devoted to that potentially large order, at the expense of sales in Western Europe.

In the UK, both PM Engineering and PDQL have chosen to disassociate themselves from Thor International. GAP Software, an early Thor dealer, vanished a year ago.

We put the news to the original UK financier of the Thor, Vic Oliver. He funded the Thor development and provided staff until last August. Since then the elder Mr. Oliver has attempted to recover his money through the courts. The case is proceeding, and several other parties are pursuing David Oliver for non-payment of debts.

The only enthusiastic comment we could muster came from German Stephan Michels, Managing Director of Thornado Systems. His firm is pitching for exclusive Thor distribution rights in West Germany.

Herr. Michels told *Shopper*, "we sold six machines last month, and had no trouble obtaining supplies; the new production quality is very good, and the machines were delivered from Denmark within 14 days." He expected the new Thor ROM (6.4.2) soon, and phoned back later to say that Thor International was still in business, and had simply closed down for a summer holiday.

We hope to report in more detail soon. This may just be a 'silly season' misunderstanding, but we advise readers to exercise caution when dealing with Thor International or its associates.

MEDIA MATTERS

In the last few months there have been several scare stories about the demise of the microdrive cartridge, used in the QL, ICL One Per Desk, BT Tonto and Spectrum microdrive system. The cartridges are mass-produced by Ablex Audio Visual in Telford – the only people with the specialised machinery required.

Each cartridge contains about five metres of 1/16 inch tape, cut out from video tape supplied by BASF in West Germany. Stocks of tape ran out at the beginning of the summer, but new supplies have been obtained and there is no longer any problem maintaining production.

Microdrive cartridges will remain in production "for the next 18 months", according to Ablex Production Manager David MacSorley. Cartridges will continue to be sold in Boots high-street stores, and by specialist QL, Spectrum and ICL suppliers. Most well-heeled users run disks these days, but cartridges are still important, particularly in the QL market where they are the standard medium for software sales.

We contacted Sinclair Research and were told that it would put pressure on Ablex to maintain supplies. It did not anticipate a shortage, and expected stockpiles to be made if demand drops to the point where continuous production is not justified. It takes this support seriously – for instance it still has large stocks of batteries for old Sinclair pocket tellies.

The QL microdrive. Fears over supply shortages curbed.

Most surprisingly, Sinclair Research revealed it was currently researching the possibility of using microdrive cartridges in a new and un-identified product. This seemed rather hard to believe, but it may keep Ablex happy for a while.

The price of cartridges looks set to remain high. Each tape holds about 85K of data on a Spectrum system, or around 110K on QL or ICL drives, making cartridges much more expensive than 3.5" or 5.25" floppy disks, particularly when capacity is taken into account.

The cost of production of each cartridge is reputedly tiny – around 20 pence – but a royalty of 75 pence per unit is believed to be paid to Sinclair Research Ltd, and this accounts for much of the £2 retail price-tag on each cartridge. When Sinclair introduced the microdrive cartridge in 1983 they sold for £4.95 a piece!

Amstrad has acted to protect supplies of 3" disks, which have been scarce in recent months. The 3" Hitachi-format disks are made by two firms: Matsushita and Maxell. In July, Amstrad ordered six million disks, phased for delivery over the next few months.

The 3" disks are mainly used in Amstrad's CPC, PCW and Plus Three machines, but they also crop up in some MSX and QL systems. Amstrad used to sell large numbers of CF-2 disks, but recent 'Amsoft' labelled media were imported by a separate firm, Disk Xpress.

Now Xpress has lost the concession, after supply hickups. Amstrad manager Peter Roback told Computer Trade Weekly, "The marketplace demands that somebody is in it who can commit in big enough quantities to keep supplies healthy. We are ensuring that sufficient disks are available for users."