cuit boards, with extras bolted on to improve the printed specification. The machines looked neat in their new boxes, but internally they were little different from expanded QLs.

Both firms tried to buy the rights to the QL design from Amstrad, but failed. Then Sandy announced the Futura, a SuperQL with a real 68000 processor, rather than the slightly slower 68008 of a normal QL, and an amazing new video display that promised to out-Amiga the Amiga.

The Futura was never finished – the display circuit was just too fast for production. The rest of the design may surface soon, although it WON'T be as a QL clone – watch this space!

What has materialised, courtesy of CST (0438 352150), is a machine called the Thor-16. This looks much like the old Thor-1 which had a QL inside, but in fact it's a totally new design, which runs all the old QL programs, much faster.

The processor is a 68000, rather than a 68008, which makes many operations twice as fast. Memory access is also much faster, because the video circuit only stops the processor a quarter as often as it did on the QL. The bus bandwidth of a standard QL was about 1 megabyte per second; at best, the Thor 1 could go twice as fast. The Thor 16 is at least twice as fast again.

The circuits run slightly faster, at 8 million pulses a second rather than 7.5, and this has many ramifications – in particular it makes display lines a little shorter, so you no longer find that the leftmost columns of the display are off the margin on some monitors.

The display format is similar to that of the QL, but with the option to forget about flashing in MODE 8 and use 16 colours instead. There's no palette, so you can't mix your own colours as you can on the Amiga or ST.

The standard 512K machine is not cheap – it costs between £700 and £800, depending upon the number of 720K floppy drives you want. Expansion memory runs even faster, and plugs into 3 slots already on the board – 512K costs £100 and 2 megabytes will set you back £300.

CST have had to re-write the entire QL operating system, to avoid infringing Amstrad's rights.

**Superclone 2**

The Spectrum is not the only Sinclair design that has attracted attention from new manufacturers since the Amstrad putsch. So far two 'SuperQLs' have been announced, and one of them should be in production by the time you read this.

Just after the Amstrad takeover, in the summer of 1986, both Sandy and CST announced QL follow-ups, called the QXT-640 and Thor-1 respectively. In fact both boxes just contained standard Sinclair cir-

In the process they've made improvements wherever they could retain compatability with the old QL. In particular you can have several displays in memory at once, choosing between them instantly, and the text printing routines have been totally re-designed to make output 20 or more times faster than a normal QL.

Sound is NOT compatible with the QL. The old BEEP command will still exist, but sound will be handled by a fast (200 KHz) 8 bit digital to analogue converter. This is similar to the arrangement in the Amiga, but the Thor-16 is mono rather than stereo.

The ports have been re-hashed. There's a standard QL expansion connector for eight-bit add-ons, which should copé with all QL add-ons other than memory boards. The Thor-16 has proper serial ports – you can select any speed up to 38,400 baud independently for transmit and receive. The network is compatible with the QL, but can go eleven times faster, communicating through cheap bell-wire instead of audio co-axial cable. Again, there will be more news in next month's Sinclair Scene.

**History**

Curiously, the QL market is busier now than it has ever been. Programs and add-ons were slow to arrive after the QL launch, not least because Sinclair didn't ship any machines until half way through 1984.

After that it took a while for programmers and users to get used to the new way of doing things – the QL was the first home machine to use Motorola 16 bit chips, and a Z80 or 6502 programmer needed a while to come to terms with the new bells and whistles, and then to learn how to take advantage of them.

There was a delay of about a year, during which time the press forgot about the QL completely. Then the good programs began to come through, and since then the QL community has grown by recommendation, rather than through hype. The Amstrad takeover was worrying, but it turned out to be a boon – in the closing years, Sinclair held the market back, by being there but doing nothing. Now there's lots of new hardware and software about, and the range is growing all the time.

**QDOS And CP/M 68K**

The QL operating system was originally planned to be CP/M-68K – a rather useless variation that accepts the same device requests as old CP/M, but needs completely new programs.

CP/M-68K is the bottom level of the Atari ST's 'TOS', which we're meant to think stands for 'Tramiel Operating System'. Luckily, it wouldn't fit into a 128K machine with Psion's packages, so QL users ended up with QDOS – a device-independent, multi-tasking operating system in ROM.

QDOS is very concise – about 7K of 'kernel' to handle tasks, plus 11K of 'device drivers'. These are the routines that let the processor communicate with the outside world – screen windows, the keyboard, network, serial ports, and the microdrives.

GST's CP/M-68K did come out in 1985, but it was obvious why Sinclair had rejected it. It was slow, large and clumsy. Atari make the best of a bad job on the ST, adding GEM and spreading the result over 200K of ROM and 500K of RAM – but CP/M 68K was never viable on the QL.

In general QDOS works very well, and although mainframe users might blanch at the thought of multi-tasking without memory-management hardware to shuffle things around, the QL can certainly do things that single-tasking machines cannot, while avoiding the 'overhead' of other multi-tasking operating systems like Amigados.

**Speedscreen**

In the past, one of the main bugbears of QL use has been the slow text display. Compared with other micros, text rippled across and down the screen – you could almost see the characters being drawn. The QL prints text on a high resolution display, so that a screenful of text involves writing 131,000-odd dots in four colours, but even so the 68008 should be able to print much faster than the QL did.

In fact the original GST display was even slower – about a third the standard speed. Sinclair staff adapted that code to make text output faster, but they didn't have time to re-write it completely, which was what it needed.

Three years on, Creative CodeWorks (021-426 5199) have done the necessary work, and changed the QL operating system to make the display much faster. 'Speedscreen' speeds up text and user-defined graphics, as well as scrolling, cursor actions and block clearing. It works with all MODE 4 (small character) programs – you just load it before your main software and proceed as usual – except more quickly!

The original QL ROM uses the same routine to clear a whole window as it does to print a small red cursor. Speedscreen replaces such generalised routines with specialised code for the most common and time-consuming cases: Speedscreen includes twenty different routines to print a character, depending on colour, position and size.

The exact speed-up factor depends on what you're doing, but scrolling is between two and eight times faster. Text output usually goes five to twelve times more quickly, depending on the colours and character-size you're using.

Of course, Speedscreen only speeds up display output, and that's only part of the job done by most programs – the other parts run at the usual speed, so it's no help with programs that spend all their time crunching numbers.

We've had Speedscreen in our systems for a few months, and we soon notice when it's not there, but it's a great shame Sinclair didn't get the QL screen routines right to start with.

**QL CP/M**

Sandy (0234 219814) impressed the world by producing the first CP/M system for the QL. The QL hasn't got anything much like a Z80 in it, with the possible exception of the 8049 second processor, which is crashable – try CTRL-ALT-7 – but not programmable in any case.

The 68008 is much cleverer than the Z80, but it works in a very different way. It cannot recognise any Z80 instruction without reading it, looking it up in a table, then executing an appropriate routine, and this involves many instructions where the Z80 only executed one. If you're very, very clever you can get the 68008 somewhere near the speed of the Z80. But it's an art!

The Sandy CP/Mulator is good, and very compatible with CP/M programs. Compatibility is always a difficult thing to judge, but in general, if a program is not fussy which 'real' CP/M computer you use, it will work with the CP/Mulator.

***Sinclair staff adapted GST code to make text output faster, but they didn't have time to re-write it completely, which was what it needed***

The main snag of the CP/Mulator is its very slow display output. You can accelerate this to some extent with Speedscreen, but most of the delay is not the fault of the QL but a consequence of the way the emulator works.

When the CP/M program wants to print a line of text the emulator looks at each character individually, checks to see whether or not it is a 'special case', and then passes it on to the QL for printing. This means that the QL system is called once for every character, rather than once for the whole line, so the system spends most of its time passing things around rather than actually printing.

Creative CodeWorks has offered a free Speedscreen package to any hacker who can come up with a simple 'patch' to make CP/Mulator print with IO.SSTRG rather than repeated IO.SBYTEs. This should speed up operation noticeably in all cases, and it would give Speedscreen the chance to go really fast. Meanwhile, Digital Precision recently announced a competitor to CP/Mulator, in the form of a new product called 'Success'. We are expected to believe that this stands for 'SUpreme Cp/m Code Emulation SyStem'!

The price is £49.95, as for Sandy's product, but Success has no need of a plug-in cartridge. This may be an advantage in some applications – for instance, if you want to use DP's 'Gigamouse' or the Supertoolkit cartridge network driver – but it may reduce the speed of the code, especially if you're using an unexpanded 128K QL.

The QL design slows access to the memory inside the machine, so programs often run faster if expansion RAM is fitted. Digital Precision claim that Success runs as quickly as a 2MHz CP/M system – in other words, about half as fast as a Plus 3 – but they don't say whether or not you need extra memory to reach that speed. We plan to compare Success with CP/Mulator in a future issue of Sinclair Scene.