RETROGRADE

THE ULTIMATE GUIDE TO PRE-MILLENNIAL PC HARDWARE

IS YOUR FLOPPY DISK KNOWLEDGE MISSING? FIND OUT HOW THEY WORK INSIDE!

IBM AND 100% COMPATIBLE PC SYSTEMS

286, 386, 486 and Pentium
CGA, EGA and VGA
3.5in high-density floppy drive
Voodoo, PowerVR and GeForce
Sound Blaster, Roland MT-32 and PC speaker
MS-DOS and Microsoft Windows 1.0

REQUIRED: PDF reader and fondness for old computers

HARDWARE ANALYSIS | INTERVIEWS | **BUILD A RETRO GAMING PC** | LOADS MORE

CUSTOM PC

THE BEST-SELLING MAG FOR PC HARDWARE, OVERCLOCKING, GAMING & MODDING

THE MAGAZINE FOR
PC HARDWARE ENTHUSIASTS

ISSUE 228 OUT NOW

VISIT **CUSTOMPC.CO.UK** TO LEARN MORE

WELCOME

BEYOND NOSTALGIA

On the face of it, the first PC we had in my family home bears little resemblance to the multi-core beasts of today with at least 16GB of RAM and often terabytes of storage. Forget running Crysis – it couldn't even run Eye of the Beholder (it needed 640KB of RAM). It had one floppy drive (no hard drive), it could only play most games in four (hideous) colours at 320 x 200, and the only sound came from a bleeptastic internal speaker.

Even then, there was very clearly room for a lot of improvement, but one of the things I love about the PC is that there's been a continual line of progress from the 1980s to the PCs we have now. Apple has flipped between different instruction sets, console makers have come and gone, and the Amiga and Atari ST have disappeared, but the PC standard has stuck with Intel's x86 instruction set and continued to grow throughout that time.

If you like, you can attach a USB floppy drive to your new PC, stick in a 3.5in boot disk for MS-DOS 3.3 from 1988, set the EFI to boot from the floppy drive, and your brand-new PC will still boot with that archaic operating system, running it natively.

I look back on those neolithic PC days with fondness. Double-checking the system requirements stickers on the corners of game boxes; making multiple boot disks for different upper memory configurations; poring over the multi-page retailer price lists in bumper PC mags; gawping at 256-colour VGA after upgrading from CGA; first hearing synth music in games with a Sound Blaster; and being astounded by the first 3D accelerator cards.

But at **Custom PC** we like to delve into PC hardware to find what makes it tick, and the same goes for the vintage gear in our Retro tech section. We don't just want to say 'hey, remember this!' in the hope of triggering a bout of fuzzy, rose-tinted nostalgia – we want to get our hands on the old hardware, show you how it worked and tell you the story of its development.

This free digital book contains nearly 100 pages of content from that section, not only covering the inner workings of loads of PC hardware and software from the 1980s and 1990s, but also speaking to some of the people involved and even showing you how to build your own retro PC. If you like what's in this book, and you like PCs as much as we do, there's a good chance you'll also like our magazine. Go and have a look at **custompc.co.uk** CPC

EDITOR
Ben Hardwidge

✉ ben.hardwidge@raspberrypi.com

🐦 @custompcmag

Raspberry Pi PRESS

CONTENTS

PROCESSORS

INTEL 286

K.G. Orphanides takes a technical look
at Intel's 16-bit swansong

We're now so used to tiny transistors that the 7nm process used to fabricate AMD's latest Zen 3 CPUs hardly seems worth mentioning now – it's hard to keep track of the numbers of transistors when they get into billions. However, you only have to look at early PC CPUs to see just how far silicon manufacturing has come. Intel's 80286 processor was released in 1982, and fabricated on a 1.5μ (1,500nm) manufacturing process, compared to the 3μ (3,000nm) process used by its predecessor, the 8086. It packed in 134,000 transistors: 4.6 times as many as the 8086. By comparison, AMD's 7nm Zen 2 processors contain up to 9.8 billion transistors.

The 80286 was introduced with an entry-level-model clock speed of just 6MHz. This figure would go as high as 12.5MHz for the popular Intel 80286-12, and up to 25MHz for late-era takes on the CPU by other manufacturers, such as AMD and Harris. It would be the last, fastest 16-bit PC processor Intel made.

THE 80287 COPROCESSOR

Since the 8086, floating-point coprocessor chips – popularly known as maths coprocessors – had been made available as optional additions via a motherboard socket. They allow addition, subtraction, multiplication, division and square root calculations on numbers with decimal points to be carried out more quickly than on a standard integer unit, improving performance in arithmetic-intensive applications.

Originally, that was mostly accounting and computer-aided design (CAD) software, but later games were also able to take advantage of the hardware, notably including 1989's SimCity and flight sims such as Falcon 3 in 1991. The 486SX series was the last range of Intel CPUs to be released without a built-in maths coprocessor – its sibling 486DX integrated a floating point unit into the CPU.
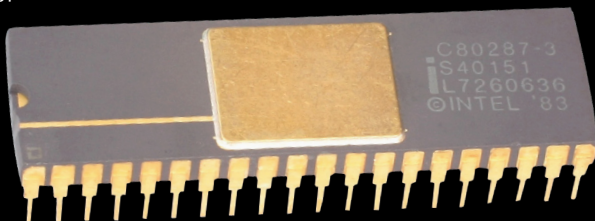
An optional 80287 coprocessor provided the 286 with a floating point unit

Its successor, the 80386, was a true 32-bit processor, with a 32-bit data bus and memory addressing to match. But even as its technology was superseded, the 286 was just hitting its stride in the home PC market, which it would dominate until 386 and 486-based PCs started to become vaguely affordable in the early 1990s.

A VISION FOR THE FUTURE

When development began on the 80286 in 1979, Intel's product requirements document envisioned that the powerful new processor would be primarily used in industrial applications, from telecoms to manufacturing automation and medical instruments. It was explicitly designed to be compatible with the 8086, ensuring that software for the older processor would run without modification on the new device. But unlike the 80186 (see opposite), PCs weren't on the 286's original roadmap.

In Intel's 1984 annual report, which details the 286's development, release and nascent domination of the industry, the company admits that in hundreds of pages of planning materials 'the personal computer – which would eventually become its biggest user – wasn't mentioned *once*'.

The 80286 was announced in February 1982, and the designers had a working prototype to show industry partners that spring, promising 'about three times the performance of any other 16-bit microprocessor'. However, after initial testing of the first 286 wafers, 'progress just seemed to drop to a snail's pace', according to logic design supervisor Jim Slager, again quoted in Intel's 1984 annual report. The processor wasn't yet running fast enough, and the testing programme for CPUs that would come off the manufacturing line was running late.

But in June 1982, IBM – then the world's largest maker of computers – came calling. IBM had been using Intel's 8088 since 1979 and it was looking to give a power boost to its next generation of PCs: the IBM model 5170, better known at the IBM PC/AT.
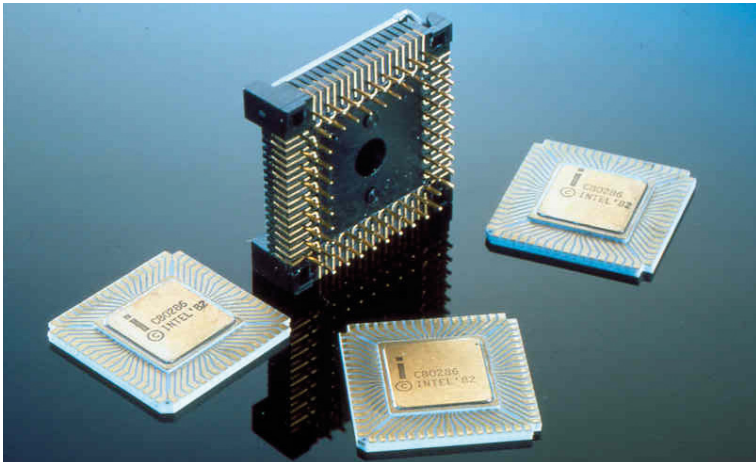
Intel pulled together a cross-disciplinary task force to complete the testing tools, address bugs and complete the parallel development of motherboard components. Marketing focused on a new public presentation of the 80286, highlighting its superiority to Motorola's popular 68000 processor and emphasising that it was far more than a minor update to the 8086.

Intel emphasised the 286's multi-user and multi-tasking capabilities, including variable privilege levels to restrict access to specific parts of memory, as well as an instruction set designed to rapidly switch between programs, providing support for Unix as well as DOS.

The marketing push – and especially IBM's adoption of the processor – worked. Chip samples were delivered to customers later the same year and, in 1983, volume production of the 80286 began. The IBM PC/AT launched in August 1984, prompting a wave of AT-compatible computers from companies including Compaq and NEC. By the end of 1988, Intel estimates, there were around 15 million 286-based PCs in use worldwide.

Unlike the 80186, PCs weren't on the 286's original roadmap

A DIFFERENT MODE

Changes to memory handling were a headline feature of the 286, but software support was slow to emerge. The processor introduced protected mode memory addressing and retained real mode addressing to ensure compatibility with applications designed for the 80186, 8088 and 8086.

In real mode, like the 8086, the 286 can address up to 1MB of memory via a 16-bit address bus. In protected mode, it can address up to 16MB of memory using a 24-bit bus. This approach has security and stability benefits in that, in protected mode, different programs and users can't access memory segments in use by others.

Protected mode made the 286-compatible with Unix-based operating systems such as Microsoft Xenix, and its secure memory handling made it possible for up to eight users on terminals to be connected to a 286-based Xenix server.

THE OBSCURE, WILDLY SUCCESSFUL 80186

Released at around the same time as the 286, the 80186 was fully software-compatible with the 8086, with an emphasis on increased performance at the lowest possible cost. It was an instant success, and Intel produced 30 times as many 80186s as 8086s in the new processor's first year of release.

Although Intel at one point envisioned the 186 being used in workstations, word processors and PCs, it was the 286 that ultimately came to dominate the desktop market. Unlike the 286, the 186 had its clock generator, timer and interrupt controller – previously motherboard components – built into the CPU.

However, these integrated components weren't compatible with the hardware used in the IBM PC, leading IBM to select the 286 for its PC/AT range of computers.

The 186 was nonetheless massively successful, due to its speed and ease of integration into other systems, appearing in coprocessors, communications controllers, flight management computers and general-purpose microcontrollers.

It did appear as the main CPU of a few PCs, including the 1986 Sega AI in Japan, the Tandy 2000 in the USA and the frankly inexplicable RM Nimbus schools PC in the UK. Intel ended production of the 186 in 2007, although fully compatible third-party clones are still available.

The 186 was hugely successful outside of the desktop PC world. Image credit: Konstantin Lanzet

To ensure backwards compatibility, the system has to boot in real mode and then be switched into protected mode by setting a status register bit. To get out of protected mode, you have to reset the CPU. This switching process was crash-prone in some versions of IBM's OS/2 operating system, where it was used to provide an MS-DOS compatibility mode. Some manufacturers put out specialised motherboards, which integrated additional 'warm reset' capabilities.

However, protected mode simply wasn't used by MS-DOS, the most popular operating system used with the processor. Instead, an undocumented instruction, LOADALL, allowed the CPU to access all memory from real mode. It was critical to the HIMEM.SYS file used to manage memory, and allowed real-mode processes to access up to 16MB of RAM by updating the segment-descriptor cache to point at an extended memory address.

Sierra's King's Quest II: Romancing the Throne explicitly supported the 286-based IBM PC/AT

The 80287 add-on maths coprocessor could be used to improve performance in some games, including SimCity

Protected mode would evolve with the later adoption of 32-bit addressing in the 80386. By 1988, Windows 3.0 was able to take advantage of a 16-bit protected mode environment, compatible with both 286 and 386 processors, and Microsoft released compilers and SDKs for third-party developers.

Windows 3.0's use of 16-bit, rather than 32-bit, protected mode memory addressing ensured backwards compatibility with the 286, but this would be abandoned with the release of Windows for Workgroups 3.11, which requires the 32-bit protected mode introduced with the 386.

The 80286 die was built on a 1.5-micron (1,500nm) process. Image credit: Pauli Rautakorpi / CC BY creativecommons. org/licenses/ by/3.0
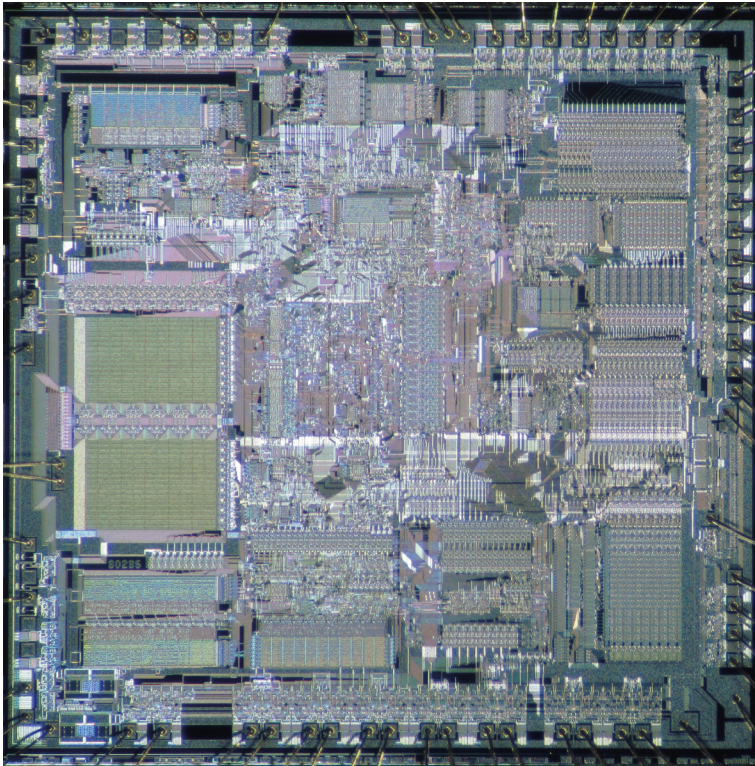
NEW INSTRUCTIONS

Developed simultaneously, the 286 and 186 shared a number of new additions to their instruction set architecture, above and beyond those of the original 8086. Like its predecessor, the 286 instruction set has a 16-bit word size – the number of bits (binary on/off switches) on which it can operate with one instruction.

Shared with the 80186 are the ENTER, LEAVE, BOUND, INS, OUTS, PUSHA, POPA, PUSH immediate and IMUL immediate

A 10MHz 286 could execute programs up to six times faster than a 5MHz 8086

instructions, and a range of immediate shifts and rotates. These include both mathematical operations, such as the signed integer multiplication of IMUL, and data handling operations. An example of the latter is PUSHA (push all registers), which saves the contents of all eight general registers, used to temporarily store data, to the stack, to and from which instructions can store or retrieve data.

The 80286 additionally added ARPL, CLTS, LAR, LGDT, LIDT, LLDT, LMSW, LSL, LTR, SGDT, SIDT, SLDT, SMSW, STR, VERR and VERW. Most of these instructions are used for protected mode memory handling, but a few, such as SMSW (store machine status word) and LMSW (load machine status word) are used in real mode.

By 1990, memory-hungry games, such as the US release of Sorcerian, advertised their need for 'AT-compatible' PCs
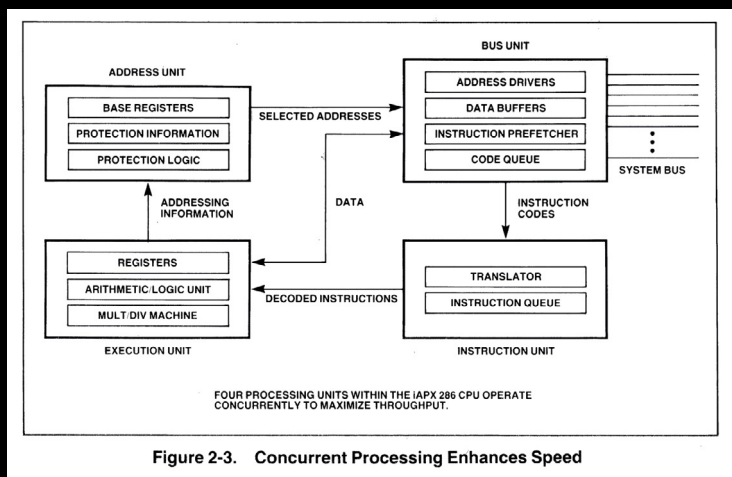
Figure 2-3.   Concurrent Processing Enhances Speed

The 80286 has a dedicated address unit, bus unit, instruction unit and execution unit

The 286's machine word status is used to indicate the presence of features such as an 80287 maths coprocessor (see p106), and whether the CPU is supposed to be running in protected or real mode. The introduction of instructions to efficiently end the execution of a task, save its state and switch to another, loading its last state, significantly improved multitasking performance.

PERFORMANCE

The 286 provided a marked performance boost over the 8086 and 8088. This was in part down to faster clock speeds, particularly when 12.5MHz, 16MHz and even faster 286 CPUs became popular. The CPU also benefited from significant architectural redesigns, enabling a 10MHz 286 to execute programs up to six times faster than a 5MHz 8086, according to Intel's Introduction to the iAPX 286 document.

A 12MHz 286 can calculate between 1.28 and 2.66 million instructions per second (MIPS), compared to 0.330 MIPS for a 5MHz 8086 and 0.750 MIPS for a 10MHz 8088. The 286's instructions per clock (IPC) count works out at 0.21 MIPS per megahertz. To help achieve this, the 80286 CPU comprises four independent processing units: address unit, bus unit, instruction unit and execution unit, compared with the two-unit execution and bus organisation of the 8086. It has demultiplexed address and data buses to improve bus efficiency, particularly in protected mode.

The instruction unit can decode and hold a queue of three prefetched instructions, which it sequentially feeds to the execution unit. Meanwhile, the presence of a dedicated address unit, which calculated the physical addresses in memory of the instruction and data being called upon, offered a key performance improvement over previous systems.

GAMING

The 286's extra power meant that more was possible for game developers. New instructions for moving data between stacks and registers benefited those working in high-level languages such as C. Although the increasing multimedia capabilities of PC systems through the 1980s also played a significant role, the PC's processor power was becoming apparent. That said, in the 1980s, 286 systems were still prohibitively expensive compared with more family-orientated microcomputers, as well as low-end 8086-based PC-compatible machines.

Despite this, the second instalment in Sierra's King's Quest series, 1985's Romancing the Throne, explicitly supported the 286-based IBM PC/AT, booting directly from a floppy disk. By 1990, popular series, such as Ultima and Wizardry, which had once been developed for rival systems, such as the Apple II and IIGS, were receiving MS-DOS first releases.

It wasn't all positive. Some older games whose performance was fixed to clock cycles became unplayably fast, which led to the widespread use of 'Turbo buttons', which would slow the system down to clock speeds comparable with 8086 and 8088 CPUs. Other 286 PCs had a BIOS option to do the same, and utilities such as Mo'Slo were developed in the 1990s to slow down overspeed games. CPC

INTEL 386

Ben Hardwidge looks back at the PC's first 32-bit CPU

We often complain about the over-inflated price of graphics cards these days, but the prices of today's PC components are extraordinarily generous in comparison with the early days. If you want the latest top-end Threadripper CPU, the fastest gaming GPU and an enormous amount of storage, a machine such as Chillblast's Fusion Conqueror (see p32) will deliver all of it in a well-built machine for £5,999 inc VAT.

Now, I'm not going to pretend that's a small amount of money – it's unaffordable for most of us. But, to get some perspective, let's take the TARDIS back to September 1986, when Compaq released the Deskpro 386, marketed as the first 'true' 32-bit computer. This was a good 11 months after Intel first launched the first 12MHz 386 CPUs, but seven months before IBM's first 386 machine got out of the doors, marking a new era where 'clone' PCs were becoming dominant.

Adjust that figure for 34 years of inflation, and the price is £19,890

The top-end launch model of the Deskpro came with a 16MHz Intel 80386 CPU, 1MB of RAM, a 130MB hard drive and a 1.2MB 5.25in floppy drive. It cost $8,799 US, which works out at around £6,737 (the exchange rate in 1986 was very similar to now). Adjust that figure for 34 years of inflation, and the price is £19,890, and that doesn't even include VAT.

If you couldn't afford superfluous luxuries such as a 130MB hard drive, you could alternatively plump for the cheaper model with a 40MB drive – a bargain at $6,499 US (£14,694 ex VAT, adjusted for inflation). This is why most PC users at this time used machines with much older CPUs, often with no hard drive and small amounts of memory, for many years – I was still using an 8MHz 8086 a good 12 years after 1978 when that CPU was first launched.

Just like the prices, the numbers involved with the manufacturing process of the 386 are staggering compared with today's CPUs. The first 386 chips contained 275,000 transistors, which made them a marvel of miniaturisation at the time, but that's a piddly number compared with over 9 billion transistors, which you'll find in the Ryzen 9 3950X across all its dies. In terms of raw transistor numbers, a Ryzen 9 3950X is like 35,000 386 CPUs.

Those transistors were massively bigger as well, produced on a 1,000-1,500nm node, compared to 7nm in AMD's latest CPU dies. The very first CPUs off the production

Inside a 386 die, with 275,000 transistors
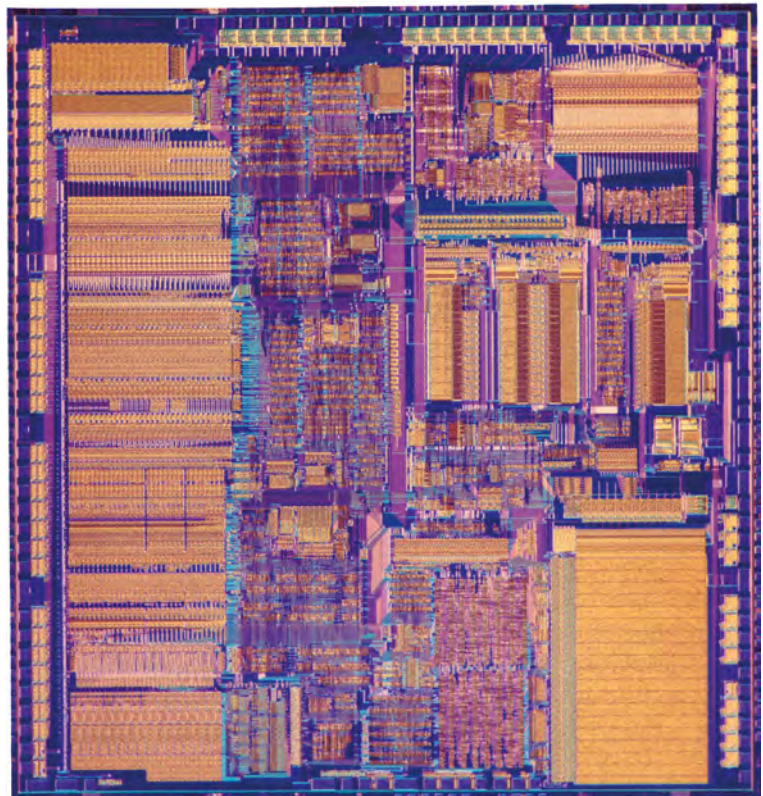
line were clocked at 12MHz, then 16MHz, with 20MHz, 25MHz and 33MHz flavours launching later – even the latter is around 1 per cent of the clock speed we see on today's CPUs. Pin-compatible CPUs were also made by AMD, as well as other manufacturers, including Cyrix.

MEMORY MANAGEMENT

The first 32-bit x86 CPU was big news in the computing world though. While Motorola's 68000 (used in the Atari ST and Commodore Amiga, among others) had introduced us to an internal 32-bit CISC CPU architecture back in 1979, it also used a 16-bit external data bus and a 24-bit address bus. Intel's first 80386 CPUs were 32-bit internally and across external buses, offering a huge advance over the previous 16-bit 8088, 8086 and 80286 processors.

An Intel marketing shot for the 386 shows a 16MHz 386 CPU, as well as an 80387 CPU and some 1.2MB 5.25in floppy disks
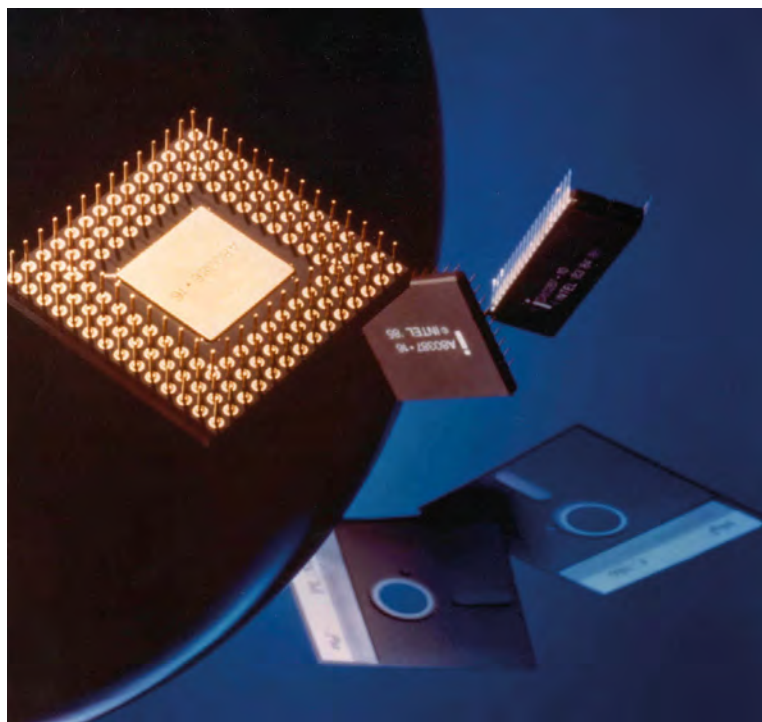
In theory, this meant a PC could now address 4GB of RAM (a limit that would only become seriously challenged 20 years later), although realistically the limits of technology at the time meant that most 386 PCs could only address up to 32MB, and even that was considered overkill. For reference, my 386 PC in the 1990s came with 4MB of RAM, but I upgraded it to 8MB using 30-pin SIMMS and it felt decadent.

More importantly for the time, the 386's memory system was designed to be easily extended well beyond the 640KB base memory limit of MS-DOS. The ins and outs of archaic memory systems are well beyond the scope of a two-page nostalgia piece, but the basic gist is that a 16-bit x86 CPU could only address 64KB of memory, so any memory on top of this figure had to be divided into 'segments' that it could address separately.

In order to maintain backwards compatibility, the 386 still retained this segmenting approach in 'real mode', but it also offered a new form of 'protected mode'. This mode was first introduced with the 286 to allow the use of virtual memory (effectively paging to a hard drive). However, the 386 added an on-board paging translation unit to mediate between the segments and the physical address bus, which effectively enabled the computer to present all these segments as one big sea of memory, even though it was technically still segmented.

It made for a much friendlier memory system for software developers, particularly for memory-hungry graphical user interfaces, and it paved the way for PCs with ever larger memory allocations.

THE JOY OF SX

The ability to address so much memory was overkill for the home market, though, and the prices of original 386 machines put them well out of the reach of this market anyway. To get the 386 into home machines, Intel introduced a cut-down version called the 386SX, with the original design now getting the 'DX' suffix.

This isn't to be confused with the 'SX' and 'DX' suffixes used on the later 486 chips though. When it came to 486 CPUs, the DX versions had a built-in floating-point unit, called a math coprocessor at the time, while the SX chips only had an integer unit, although you could add an 80487 math coprocessor to most 486SX machines separately.

Conversely, neither the 386SX or DX had a built-in floating point unit – you needed a separate 80387 coprocessor if you wanted that. The difference between the 386SX and DX was that the former had a 16-bit data bus, although it kept the CPU's internal 32-bit architecture. The idea was that having a 16-bit data bus would cut down on the need for highly intricate PCBs with loads of traces, reducing the cost of manufacturing. The other knock-on effect of fewer connections was that a 386SX could only address up to 16MB of RAM. However, as we've already covered, this was still way more than enough for the home market at the time.

SOFTWARE

The big problem for the 386 for most of its useful lifespan was mainstream software support. An executable file called

A special 386 version of Links gave you gorgeous SVGA graphics for the time

Technically, you could play Doom on a 386, but only on a tiny screen surrounded by a big frame

Master II, Myst The Elder Scrolls: Arena, Sim City 2000 and UFO: Enemy Unknown (otherwise known as XCOM) all required a 386 CPU as the bare minimum. There was also a special 386 version of the Golf game Links, giving you superior graphics at 800 x 600.

That said, I ran many of these games on my 20MHz 386SX in the early 1990s, and while they technically worked, I usually had to run them at extremely low detail, and even then the frame rate would have been unacceptable by today's standards. Running Doom required me to have big bars around a tiny screen in order to make the game playable.

THE 386'S LEGACY
There was clearly room for improvement where gaming was concerned, but the 386 laid the foundation for what was to come, being the binary blueprint for many of its successors. It introduced us to the IA-32 (sometimes called i386 or x86) standard that's still used by some software today – any Windows software in the 'Program Files (x86)' folder on your C drive will be fundamentally based on this instruction set, and Intel continued to develop new IA-32-only CPUs well into the Pentium 4 era. It was only when AMD launched its first 64-bit AMD64 CPUs in 2003, and PCs started bumping up against that 4GB memory limit, that mainstream CPUs started to push into the 64-bit era.

It was the 386SX that got these powerful PCs into our homes for gaming

In many ways, my old HP Vectra 386 is the PC for which I hold the most affection from the past. I was still using my 20MHz 386SX up until 1997, a good 12 years after the first 386 chips came off the production line, and I'd pushed my machine as far as it could go. Every 30-pin SIMM slot was filled; all the IDE channels were occupied by hard drives and a quad-speed CD-ROM drive; most of the 16-bit ISA slots were taken up by a 1MB SVGA card, a 14.4K modem and a 16-bit sound card (with a wavetable daughterboard).

It's so different to my PC now, which only has one of its expansion slots filled. My 386 might have struggled with Windows 95 and Doom, but it ran Windows 3.1 well, and it made for an awesome setup for playing Dune, Civilization and the LucasArts adventures.

Plus, while the 386SX was considered to be limited in comparison with the 386DX at the time, it was the 386SX that got these powerful PCs into our homes where we could use them for gaming. Once the 386 started getting into homes, the PC started to take off as the leading games machine that we know today. It was the 386SX that that first properly put the PC in front of the Amiga and Atari ST when it came to gaming power, leading to PC exclusives such as X-Wing and Myst, and the PC has never looked back since. **CPC**

EMM386 was made a part of several variations of DOS to allow these primitive operating systems to access a 386's extended memory, but Microsoft's Windows operating system was still stuck in the 16-bit era at this time. There were nods to the 386's capabilities in Windows 3, including a 386 Enhanced Mode (if you had 2MB of RAM) that let you run DOS and Windows software at the same time, but there was no mainstream 32-bit operating system.

It wasn't until Microsoft introduced Windows NT 3.1 in 1993 that 32-bit Windows became a reality, but even then there was little supporting 32-bit software, and it also ran slowly on most machines at the time. It wasn't until Windows 95 came out, ten years after Intel made the first 386 CPUs, that the 386's internal 32-bit architecture was properly used in everyday software. It introduced the Win32 API, giving you proper 32-bit computing abilities, and it enabled filenames longer than eight characters.

I was thrilled at the time. I was still using a 20MHz 386SX with 8MB of RAM as my main PC, which *just* satisfied Windows 95's system requirements. It was dog-slow, of course. As a reference point, in the morning I would switch on the PC. Then I would go downstairs, eat a bowl of Weetabix, then make and drink a cup of tea. By the time I got back to my PC, Windows 95 would have just about finished loading. Windows 95 was really designed for 486 and Pentium machines, but you could still run it on a 386, finally fulfilling its 32-bit promise.

That doesn't mean the 386 was useless for all this time though. It still had loads of power when acting as a 16-bit processor – upgrading from a 16MHz 286 to a 33MHz 386 made a huge difference to the performance of Windows 3.1, and for gamers, the 386 was the holy grail. This was before the days of GPUs and 3D accelerators, so every aspect of number crunching for games was performed on the CPU, which meant you needed all the CPU power you could get.

By the early 1990s, PC gaming had started to progress from basic EGA graphical adventures and platform games, and were starting to see games that really took advantage of processing power. If you wanted to play Wing Commander II or Strike Commander, you really needed to have a 386, and preferably a 486. Meanwhile, X-Wing, TIE-Fighter, Doom, The 7th Guest, Dungeon

INTEL 486

Stuart Andrews recalls the mighty CPU that made the PC the ultimate powerhouse

The 486 went into development at an interesting time for Intel. The Intel 386 line had seen Intel snatch a victory from the jaws of a disaster, making up for the failure of Intel's new-fangled iAPX432 architecture with a mix of strong compatibility and great performance. Its design team, led by chief architect John Crawford, had dragged the 16-bit x86 architecture into the 32-bit era and kept Intel ahead of the pack.

But other manufacturers were moving fast. Arch-rival AMD was already developing its own 386 CPUs and only Intel's litigation was delaying their release. Cyrix was already producing Intel-compatible maths co-processors and was threatening to move into CPUs. Intel needed an awesome new product.

It combined a tighter, more streamlined pipeline with an integrated L1 cache

On the other hand, there was a lot of conflict within Intel – and within the computing community at large – over the future direction of processor architecture. Many felt that CISC (Complex Instruction Set Computer) architecture, as used in the x86 line, was a technological cul-de-sac; that performance would flatten out within a few years as Moore's law met fundamental barriers of computing.

They saw RISC (Reduced Instruction Set Computer) architecture as the future, using a smaller number of more versatile instructions and optimising the hell out of the architecture to drive performance. While one team at Intel worked on a successor to the 386, another was working on a new RISC processor that eventually became the i860. You probably haven't heard of the i860, which tells you a lot about how this situation played out.

In theory, the i860 should have trumped any 386 successor, but in 1985, Intel's CEO, Andy Grove, put John Crawford and hotshot architect Pat Gelsinger in charge of the design. Crawford and Gelsinger had already worked together on the 386 and shared a strong belief in the potential of the x86 and CISC architecture. Both felt that, while RISC had its advantages, a redesigned x86 chip could keep up.

What's more, it could do it without forcing big software publishers to redevelop their applications, rebuild operating

The 1st-generation 486 was twice as fast as a 386 with the equivalent clock speed. Image credit: Andrzej W K, own work, CC BY–SA 3.0

systems and optimise compilers. When you threw more transistors at the problem and increased their frequency, there was no reason why a CISC chip couldn't compete with a RISC CPU. Apply Moore's Law and keep increasing speeds, and a CISC chip might even crush it.

OPTIMISE THE PIPELINES!

Gelsinger and Crawford focused on delivering a processor that was fully 386-compatible and would build on the existing 32-bit architecture but would give you a massive increase in performance – at least double, clock for clock. They took inspiration from what was going on with the new RISC CPUs, paying particular attention to how instructions were loaded, organised, decoded and executed on the CPU.

The 2nd-generation DX2 chips doubled their predecessors' clock speed, a feat never replicated by any subsequent Intel CPU. Image credit: Henry Mühlpfordt, own work, CC BY-SA 3.0

THE RIVALS

If Intel's processor design teams put the 486 far ahead of the pack in terms of performance, its legal teams did a cracking job of suppressing any competition. However, eventually Cyrix and AMD won their legal fights, and 486 competitors began to appear. Cyrix's 486SLC and DLC processors, released in 1992, were particularly interesting.

Effectively a 386DX with a 486 instruction set and just 1KB of L1 cache, they still used a 32-bit bus and gave users a cheap halfway house – a 486DLC33 could run software at roughly the same speed as a 25MHz 486-SX. Not only were the processors more affordable, but they plugged into existing 386 motherboards, meaning the platform as a whole was cheaper.

I had one of these beauties in my first PC, and while it was noticeably less capable than my friend Brian's mighty 33MHz 486-DX, it could still run X-Wing, Ultimate Underworld II, Alone in the Dark and – eventually – Doom. Ultima VIII: Pagan? A bit more of a slideshow, but then it wasn't a great Ultima, so who cares?

AMD released its own 486 chips in 1993, and while they were late to the party, AMD made up for it with a repeat of a classic 386 performance trick. AMD's CPUs ran on a 40MHz bus, meaning that the, SX-40 and Am486DX/2-80 were slightly faster than the equivalent Intel CPUs.

Meanwhile, AMD's straight Am486 DX-25 and 33 and SX-33 gave you the same performance as Intel's equivalents at lower prices. AMD even released what it called the AM5x86-133 in 1995, which competed with the low-end Pentium 75 but was actually a 486 running on a 4x multiplier with a 33MHz clock.

The big innovation was to combine a tighter, more streamlined pipeline with an integrated L1 cache – a first in a mainstream CPU. With 8KB of high-speed SRAM as a store for recently used instructions and data on the same silicon, the instruction pipeline could be fed with a consistent flow, enabling it to execute the simplest and most commonly used instructions at a sustained rate of one per clock cycle – an achievement that RISC devotees believed was beyond a CISC processor.

The new pipeline had five stages, although the first – the Fetch stage – wasn't strictly necessary for each instruction, as the CPU could fetch about five instructions with every 16-byte access to the cache. Once fetched, instructions went through two decoding stages, where they were organised and fed into the execution units. Here they were executed, and the results written back to registers or memory in a final write back stage.

The cache minimised any delay in loading data and instructions, and did such an effective job of caching data and instructions that the processor only had to go to system memory on roughly 5 to 10 per cent of memory reads. What's more, many 486 motherboards incorporated a secondary cache with 16KB or more of high-speed RAM, reducing latency even further. Meanwhile the two decoder stages enabled those instructions to be pipelined and processed more efficiently – with five instructions running through the pipeline, one would normally be processed with every clock cycle.

The result was a spectacular improvement in performance. On integer instructions – very much the meat and potatoes of computing at the time – the 486 was at least twice as fast as a 386 running at the same clock speed, and sometimes 2.5 times as fast. This meant the CISC-based 486 could hit similar levels of performance to the RISC-based i860, while still being compatible with all the existing x86 software. There was no need to rebuild or recompile – code developed for the 286 and 386 just worked.

With enough processing power to run it full-screen at a full VGA resolution, Doom became the 486-DX2's killer app

With over 1.2 million transistors, the 0.8micron 486-DX2 relied on Moore's Law and higher clock speeds to trash the theoretically superior RISC competition. Image credit: by Matt Britt, own work, CC BY-SA 3.0
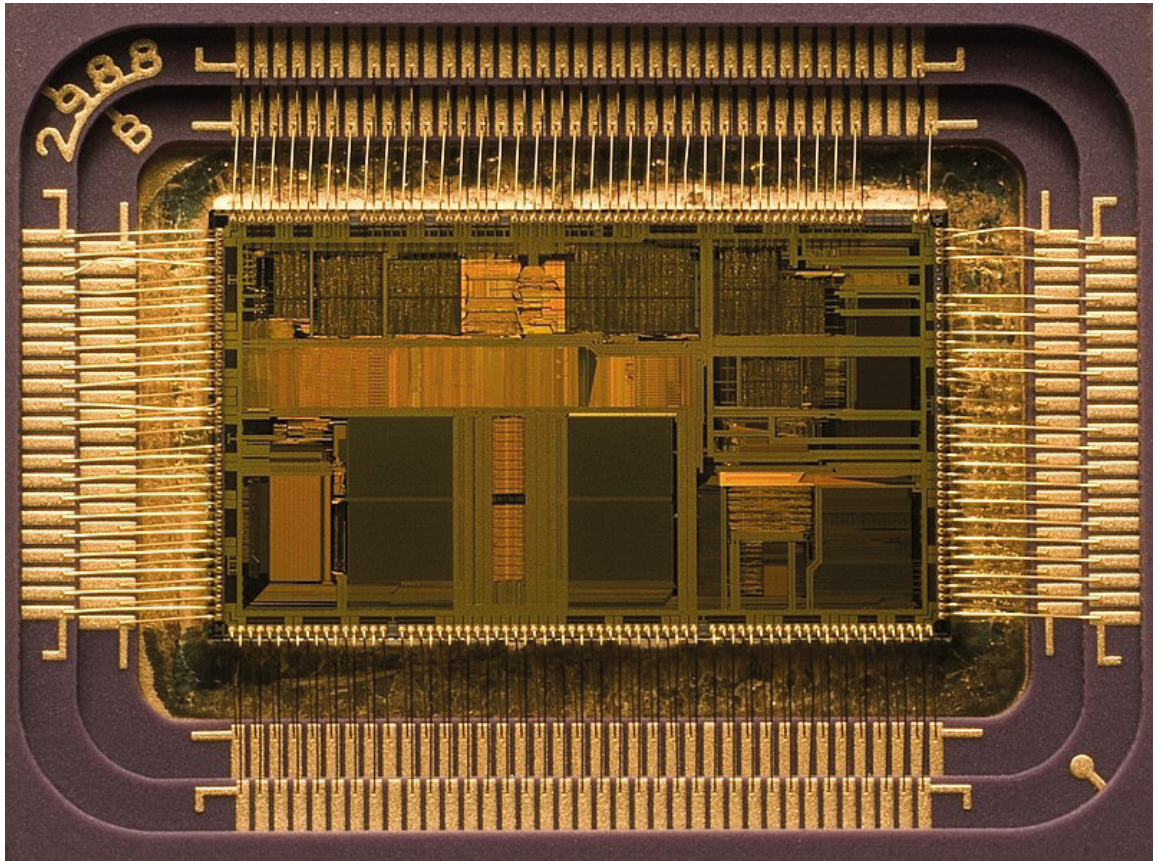
At this point, floating point instructions weren't so commonly used, but here the news was just as good. Previous Intel processors had worked with optional, discrete maths co-processors, which handled all the floating point logic. These were expensive and not popular outside of business, as only a few major business applications, such as dBase, Borland Quattro Pro and Lotus 1-2-3, actually used a Floating-Point Unit (FPU). The 486-DX, however, integrated one directly onto the processor die, connected to the CPU by its own dedicated local bus.

This meant there was less overhead in shifting data between CPU and FPU; this, combined with other optimisations, resulted in a significant improvement in floating point performance. Fast forward a few years, and Quake would require a CPU with a floating point unit, with the system requirements citing a 486-DX4 as the minimum. Today, it's impossible to imagine a CPU without an FPU, and that's thanks to the mighty 486.

Beyond this, differences from the 386 were relatively small. The 486 had a few extra 'atomic' instructions that sped up some basic operations, but nothing compared with the instructions added with the 80286 or 386. The 486 also didn't mess with the 386's memory model; it could still address 4GB of RAM across a 32-bit bus, with a protected mode that presented both real and virtual memory as one big pool. However, its improved Memory Management Unit performance meant it was much more efficient at shifting data between the system RAM, the CPU and the cache.

DOUBLE THE CLOCKS!

There was one final architectural change that was to have a major impact, even on today's PCs. Intel CPUs from the 8086 to the 1st-generation 486 ran at the same frequency as the external bus that connected all the core components together. This meant that the initial 486-DX processors, introduced in 1989-1990, ran at the same 20, 25 and 33MHz speeds as the I/O bus. Intel pushed speeds higher, releasing

OVERDRIVEN

The 486 marked another shift in Intel's tech and marketing strategy by embracing the whole idea of PC upgrades. As Intel released its clock-doubling DX2 and DX4 processors, it also released Overdrive versions designed to boost existing PCs. Some 486 Overdrive processors were simply replacement CPUs, plugging into the existing 168-pin socket and replacing, say, your 25MHz 486-SX with what was effectively a 50MHz 486-DX – albeit at an eye-watering cost of $549 to $699 US.

At this point, not every CPU could be removed from its socket, but luckily many 486 motherboards shipped with their own 169-pin upgrade socket, originally designed to fit a 487-SX maths coprocessor for 486-SX machines. Sneakily, the 487-SX was actually a fully functional 486-DX with an extra pin that told the motherboard to ignore the existing CPU, and the OverDrive chips just repeated the trick with some extra control circuitry with 50MHz and 66MHz 486-DX2 CPUs.

Doubling your speed was definitely tempting, and SX owners got a maths co-processor in the mix as well. And while Intel pushed the benefits with AutoCAD, WordPerfect and Corel Draw, the biggest sellers for OverDrive chips were undoubtedly games such as Strike Commander, Falcon 3.0 and Doom.

Games such as Strike Commander pushed the 486 architecture to its limits with advanced 3D texture mapping and Gouraud shading

Cyrix's low-cost 486 alternatives would work inside a 386 motherboard, making them the bargain Intel alternative of the day

a 50MHz 486-DX, but the 50MHz bus speed began to cause problems for components elsewhere on the bus.

Luckily, the 486 design team had an ace to play: it decoupled the CPU clock speed from the motherboard clock speed and enabled the CPU to run at double the system clock. This fired up the 486-DX2, launched in 1992, to run at internal speeds of 40MHz, 50MHz and even a staggering 66MHz, making the 66MHz 486-DX2 the RTX 3080 of its day in terms of its impact on gaming performance.

The 486-DX4, introduced two years later, went even further, tripling the bus speed to hit 75MHz and 100MHz; a staggering level of performance that trashed the available RISC competition. The team's confidence in the x86 architecture no longer looked misplaced.

WALLET-WHACKING POWER

So, the 486 launched with an undeniable advantage in performance in a market where – thanks to Intel's ace legal department –other x86 chip vendors had practically nothing. There was just one problem. While Intel had moved production down to a 1-micron process, it still had over 1.2 million transistors – a big step from the 275,00 in the original 1.5-micron 386. This made it a comparatively big chip and, partly thanks to its $250 million US R&D costs, also an expensive one.

At launch, the 33MHz 486DX alone cost around $950 US (nearly $1,900 in today's money), which was roughly three times the cost of the equivalent (and still pretty speedy) 386. A 486 PC cost users somewhere north of £2,000 (roughly £4,500 today). Intel's response – you guessed it – was to put out a cut-down, cost-conscious alternative, and 1991's 486-SX wasn't actually such a bad deal.

When Intel tried the same trick with the 386, it released a hobbled version with a 16-bit data bus and slower clock speeds, but the 486-SX was basically a 486-DX with the FPU disabled. At the time, with so little software that supported the FPU, this wasn't much of an issue, and by the time the 486-SX was released, it only cost around $250 to $300 US.

THE 486 EFFECT

The power of the 486 was transformative at a time when the CPU was the biggest star of the PC show. Sure, it was supported by a platform where VGA and SVGA graphics cards were growing more powerful, and where standardisation around the VESA local bus and, later, PCI standards was opening up the PC for more powerful add-on cards. However, the 486's advances in integer and floating point performance arrived just at a point where advances in gaming graphics needed them most.

In the early 1990s, as prices dropped to more affordable levels, the 486 hit its peak. Just check out the games that emerged. Ultima Underworld and its sequel, Strike Commander, Wing Commander III, X-Wing, Ultima VIII: Pagan, IndyCar Racing and Alone in the Dark all launched between March 1993 and December 1994, and with their texture-mapped, Gouraud-shaded 3D graphics, these PC showcases needed all the processing grunt that they could get.

A few simulations, such as Spectrum Holobyte's Falcon 3 and Digital Image Design's TFX, even used the FPU. And then, of course, came Doom; a game that you could just about run on a 386 in a stamp-sized patch in the middle of the screen, but looked amazing running full-screen at the full VGA resolution on a 66MHz 486-DX2.

If all those other games had pushed the PC as the high-end gaming platform of the early 1990s, Doom confirmed it. Even when the PlayStation and Saturn consoles launched a few years later with their fancy-pants, hardware-accelerated 3D tricks, they still struggled to run Doom classic smoothly in full screen. The 66MHz 486-DX2 could do it on its own, simply using sheer number-crunching power. People saw it, liked it and pulled out their wallets. The idea of the PC as the real gaming powerhouse was born. **CPC**
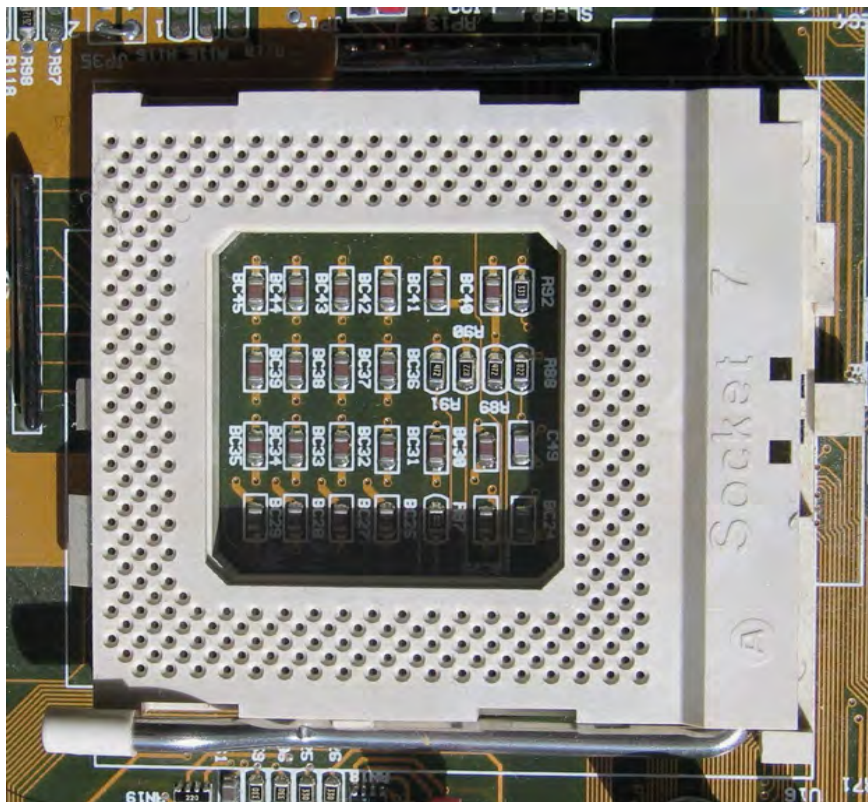
SOCKET 7

Ben Hardwidge recalls the strange pocket of time
in the 1990s when one motherboard could support
CPUs from multiple manufacturers

Just imagine if you could pick up any one of the motherboards in this month's Z490 Labs test (see p44), stick a Ryzen chip in it and know it would not only fit, but also work fine. In fact, imagine you're not just limited to Intel and AMD CPUs, but you could put a CPU from all sorts of other chip manufacturers in your brand new motherboard. Not only that, but there's a choice of chipsets all designed to work with all these CPUs as well.

We're so used to exclusive socket and chipset designs now that the idea seems like commercial suicide, but this was the situation in the Socket 7 era of the 1990s. This period is a strange little oasis in the time between times, where CPU and chipset manufacturers just assumed their parts needed to be compatible with each other.

Until this time, Intel had completely governed the design of x86 CPUs, bringing us the 8088, 8086, 286, 386 and 486 (and others) in various guises, and drafted in third parties such as AMD and Cyrix to make clone chips to fill out the supply and meet demand. That all changed when Intel introduced the first Pentium-branded CPUs.

From this point, third-party companies weren't allowed to reproduce Intel's flagship desktop CPU microarchitecture, or use the Pentium brand. Instead, the old clone chip suppliers, which still had an x86 licence from the cloning days, had to design their own CPUs.

The first Pentium CPUs were launched on the 5V Socket 4. In this era, Cyrix and AMD instead focused on launching '5x86' CPUs designed as upgrades for existing Socket 3 486 motherboards, as did Intel's Pentium Overdrive CPUs. However, it was the later 3.3V Socket 5 and Socket 7 platforms that saw Intel, Cyrix and AMD targeting the same CPU socket.

The only difference between Socket 5 and 7 was that the latter upped the total pin count from 320 to 321, and

Socket 7 was found on both AT and ATX motherboards, with chipsets from multiple chip makers. Photo credit: Konstantin Lanzet

Socket 7 could provide dual voltage to the CPU via a split rail. The sockets are otherwise basically the same, to the point where you could put a Socket 5 CPU in a Socket 7 motherboard and it would run fine.

CHIPSET CHOICES

This was before AMD made its own chipsets, but there were still plenty of options. If you wanted the best compatibility, your best Socket 7 option was Intel's Triton series, which peaked with the Triton 430TX in 1997. The 430TX supported either a 60MHz or 66MHz front side bus, and also gave you the option of three types of memory – fast page non-parity, EDO and SDRAM, with the latter two options coming in the brand new DIMM form factor. This led to many motherboards coming with both DIMM slots and the older 72-pin SIMM slots.

> # The K6–III even pushed the clock speed up to 550MHz

However, your choice wasn't limited to Intel chipsets. Plenty of third-party chip makers, including VIA, ALi, SiS and Opti had their own Socket 7 chipset options. For the most part, they held up pretty well, and they were usually cheaper than genuine Intel boards, but there were also sometimes compatibility problems. As an example, when I worked in a computer shop in the late 1990s, we often had problems with ALi-based motherboards not working with the 32x Samsung CD-ROM drive we stocked.

IS IT A BIRD, IS IT A PLANE? NO, IT'S SUPER SOCKET 7!

Intel pulled the plug on Socket 7 after the Pentium MMX, and instead moved its Pentium II CPUs to the new Slot 1 format (see Issue 200, p107). In the meantime, it settled on ATX as the motherboard and PSU standard for Pentium II.

There were some ATX Socket 7 motherboards, but most of them used the older AT form factor, which split the main power socket into two parts and only had a (large DIN socket) keyboard output fixed to the board as standard – the rest of the ports all connected to the motherboard with ribbon cables.

If you used an AT power supply, you also had to physically switch off the PC after use, as it couldn't be shut down with software. Again, though, this was a strange crossover period, and there were motherboards that conformed to the AT form factor, but which also had both AT and ATX power sockets.

While Intel was busying itself with ATX and Slot 1, though, AMD and its chipset partners went all out on Socket 7. The result was Super Socket 7, which maintained compatibility with older Socket 7 CPUs, but also supported AGP graphics cards and could clock the front side bus at up to 100MHz. There was also a range of Super Socket 7 motherboards in both ATX and AT form factors.

Super Socket 7 was great for cash-strapped enthusiasts, as it meant you could keep most of your old PC – the PSU, case, hard drive and even the memory in many cases; you just needed a new motherboard and CPU if you wanted a decent upgrade. It was massively cheaper than upgrading to Pentium II.

You could run a Pentium CPU in a Super Socket 7 motherboard too, or a Cyrix M-II or IDT WinChip 2, but what you really wanted was an AMD K6-II or K6-III. AMD's last Super Socket 7 CPUs really pushed the limits of this old socket and the AT era, with the 100MHz front side bus often making these systems faster than the 1st-generation 66MHz Pentium II CPUs, while costing much less money. The K6-III even pushed the clock speed up to 550MHz, and integrated 256KB of L2 cache onto the die.

END OF AN ERA

AMD finally moved to its own Slot A platform with the first Athlon CPUs, as well as introducing the Ironbridge chipset under its own brand, before it discontinued the K6-III at the end of 2003, eight years after Intel first launched Socket 7. Meanwhile, Cyrix was bought by VIA, which later produced a few CPUs for Intel's Socket 370 platform, as well as its own embedded EPIA platform. But the days of multiple CPUs being supported by one socket are now over – the mainstream desktop PC market has since been mainly dominated by just Intel and AMD using their own CPU dedicated sockets. **CPC**

AMD ATHLON

Stuart Andrews recalls the first AMD x86 CPU that properly put the wind up Intel

The summer of 1999 wasn't a great time for Intel, and it really should have been. In February it had launched the Pentium III, a supercharged upgrade of the P6 microarchitecture. Cyrix, whose 6x86 processors had embarrassed some 1st-generation Pentiums, was effectively finished, its tech now in the hands of VIA Technologies. That just left AMD, whose K6 line of processors had captured some of the budget PC market, but didn't have the optimised pipelines, cache or floating point performance to give Intel any serious competition.

But when AMD released its first K7 Athlon processors to reviewers in June, something unexpected happened. Sure, there was already some buzz about the new 'K7' CPU, thanks to intriguing early demos and briefings, but a Pentium III killer? Not likely. Yet when the final production samples hit magazine labs and website testbenches, it became clear that the new Athlon was pretty special

AMD's chip wasn't just matching Pentium III, clock speed for clock speed, but beating it. Worse, it was beating it in the kind of floating point intensive apps that Intel considered home territory, including 3D games. Athlon was kicking Intel right where it hurt, and that eye-watering discomfort wasn't going to let up any time soon.

K7 COMES TOGETHER

How exactly did AMD manage this feat? Well, as with so many standout products in the hardware space, the answer involves several developments all coming together at the same time. On the one hand, the success of the K6 II and III had left AMD in a surprisingly strong position.

The K6 architecture had made the most of technology bought in with the company's 1996 acquisition of NexGen and had pumped money into AMD's war chest. It had also cemented AMD's position as Intel's most credible rival.

What's more, AMD also had new CPU and bus technology developed by the Digital Equipment Corporation (DEC) for its Alpha RISC processors. It had even taken on most of DEC's RISC CPU design team, including key architects, Dirk Meyer and Jim Keller.

Thanks to a patent cross-licensing deal with Motorola, AMD also had a head start on new copper-based die manufacturing technologies, not to mention a new chip fab in Dresden on its way to use them. This would become important later on.

All this helped lead to a revolutionary design – the first 7th-generation x86 processor.

The original 0.25-micron (250nm) Athlon had a die with over 22 million transistors – the highest transistor count of any x86 processor to date. It also had an ingenious split cache system, with 128KB of on-chip L1 cache operating at clock speed, plus another 512KB of L2 cache included in the processor module.

This L2 cache operated at a fraction of the clock speed – half-speed on the initial models – but with breathing room to scale to cover higher and slower speeds later on. This arrangement gave Athlon a performance advantage over the

Shipping in 550, 600 and 650MHz versions, the original K7 Athlon took the benchmark battle to Intel – and won. Image credit: Maddmaxstar CC BY-SA 3.0

earlier K6 processors, even before you factored any other architectural improvements into the equation.

But these improvements were just as significant. Meyer, Keller and their team designed an architecture that was capable of decoding three x86 instructions simultaneously and – crucially – symmetrically, unlike the Pentium III.

True, the Pentium III's instruction pipeline could handle three simple instructions at once, but feed it more than one lengthy, complex instruction and it choked, as only one pipeline could manage the workload. The Athlon, by contrast, could chew through three complex instructions without any trouble. You got three instructions at a time, every time.

The final kicker was that AMD was no longer second rate on floating point operations

What's more, the design featured a new level of optimised branch prediction, which was not only more accurate in guessing what the next operation would be, but faster to recover when it got that guess wrong.

Like the team brought in from NexGen, the team brought in from DEC had serious skills and experience in RISC chip design, and AMD put this to good use. The Athlon architecture converted x86 instructions into more efficient 'macro ops' and then those 'MOPS' into RISC operations, which the CPU's execution units could work on, nine to a clock.

This design was incredibly efficient by the standards of the day, but it was also conducive to scaling upwards. Where the K6-III had been stuck at 500MHz, the Athlon launched at 500, 550 and 600MHz speeds, matching the 600MHz of Intel's top-end Pentium III. As if that wasn't enough, AMD added a 650MHz version in fewer than six weeks after

launch. The final kicker was that AMD was no longer second rate on floating point operations.

Not only were the Athlon's floating point units (FPUs) much faster than the weedy FPUs of the K6 line, but AMD built on the SIMD instructions of its 3DNow! Technology, with 24 new instructions on top of the original 21. Most mimicked the cache and streaming controls seen in Intel's mighty SSE tech, but AMD also bundled in new DSP and complex maths extensions, plus MP3 and Dolby Digital decoding tools. This chip was built to game and entertain.

There was one final way that AMD now matched Intel – the Athlon was AMD's first chip to abandon sockets and embrace the slot. AMD's Slot-A connector harnessed DEC's EV6 bus and bus protocol, which allowed for burst data transfers at double the rate of Intel's equivalent GTL+, giving you a whopping 1.6GB/sec of bandwidth between the CPU and the motherboard chipset.

The Athlon's front side bus operated at double the 100MHz speed of the memory bus, and as faster RAM became available, this gave AMD scope to up the FSB speed even further, to 266MHz or even 400MHz. What's more, with a slot design, AMD could combine its CPU die and L2 cache in the one package, and that package was a whole lot easier to fit. And to make sure dozy upgraders didn't try to stuff AMD CPUs into Intel slots or vice versa, it cleverly reversed the physical design.

AWESOME ATHLON

Talk about architectures and specs was all very well, of course, but nothing really prepared those of us benchmarking PCs in the late 1990s for the sheer undeniable awesomeness of Athlon. The results of benchmarks wouldn't have made comfortable reading for Intel, especially once the Athlon 650 rolled out in August. Both the Athlon 600 and Athlon 650 were faster than the Pentium III 600 in Quake III: Arena, whether paired with the hero graphics chip of the day – Nvidia's Riva TNT2 – or with 3dfx's still speedy Voodoo 3.

The Athlon was around 10 per cent faster in standard Windows applications, and up to 20 per cent faster in gaming benchmarks. The Athlon 600 was 10fps faster than the Pentium III 600 in the fiendishly demanding Quake II Crusher benchmark. As further tests from the likes of AnandTech proved, even a Pentium III overclocked to 650MHz couldn't keep up.

Inside the Athlon cartridge. Check out the Slot-A connector, the CPU core and the two modules of L2 cache. Image credit: Tullius CC BY-SA 3.0

AMD-K7™ Processor Block Diagram
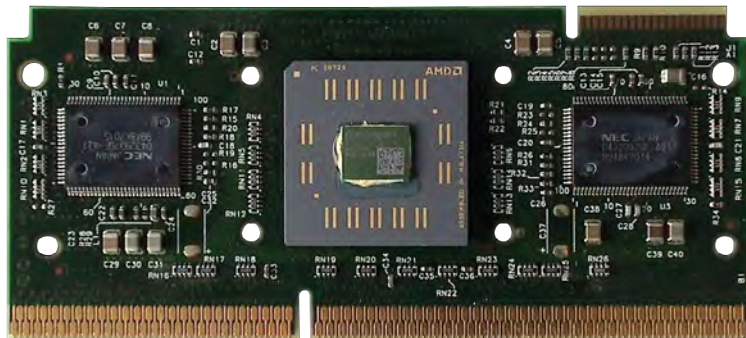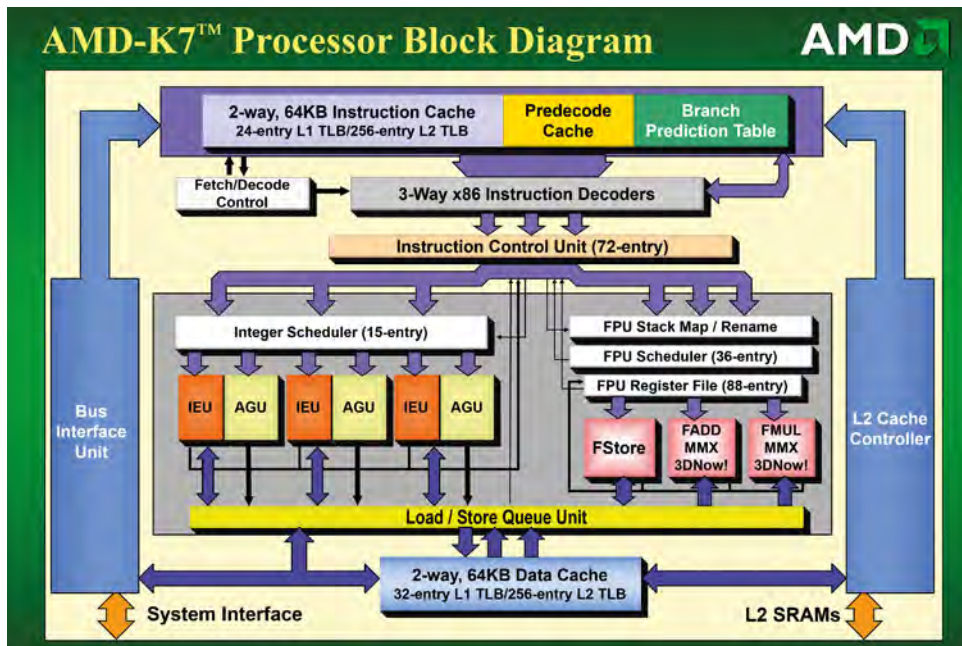
ISSUES AND OVERCLOCKS

Of course, no new CPU comes without teething troubles. Early buyers found a range of compatibility issues with specific hardware, partly because Athlon was a complex, power-hungry CPU, and partly because of AGP slot power issues affecting many motherboards and driver issues with the latest Nvidia cards. With certain VIA chipsets and less consistent power supplies, you could find yourself in a world of instability. Nvidia even released a driver update for its graphics chips that disabled the high-performance 2x mode on the AGP slot when Athlon was detected.

Some enthusiasts were also disappointed with the Athlon's limited overclocking potential. The K6 line had been a treat for overclockers – gamers upped 450MHz CPUs to 600MHz routinely, and there was much debate in PC magazines about whether we should allow manufacturers to send in pre-overclocked systems.

The Athlon wasn't having any of that. The only ways to overclock the original CPUs were to crack open the modules and interfere manually with the resistors, or to purchase a third-party 'Goldfingers' device which did it all for you. Through either method you could increase your multiplier and give your Athlon a healthy speed boost, although it meant invalidating your warranty along the way.

THE AGE OF ATHLON

The Athlon set the stage for a golden age of PC CPUs. Intel struck back with Coppermine, then AMD replaced the K7's old aluminium interconnects with copper, and ran the L2 cache at the full speed of the CPU. The 2nd-generation Athlon 'Thunderbird' processors could match and even beat the Coppermine Intel Pentium IIIs, causing Intel to push even further with its Coppermine T CPUs.

Before we knew it, 1GHz was starting to look like old hat. 1333MHz and 1400MHz were the new targets. Meanwhile, the K7 architecture was making waves at the budget end. Where Intel's cheap, cache-less Celeron processors couldn't handle Deus Ex, Half-Life or Unreal Tournament, AMD's K7 Duron CPUs were storming through them.

There's still a lot of affection for the Athlon in the PC enthusiast community. At a time when Intel seemed unassailable, it was the first chip to really knock it off its feet. This made Intel try a little harder, and the result was that everybody won. In fact, it's a mark of that affection that when AMD created a new budget Zen-based processor line-up in 2018, it still used the Athlon brand, with the Athlon 3000G coming out in November 2019. It's a sign of a classic brand when it's still being used 20 years later. CPC

The Athlon's microarchitecture was a revolutionary leap from the relatively simple K6, with more advanced pipelines, faster FPUs, more cache and a three-wide instruction decoder

And this was just the beginning. In September, Intel launched the Pentium III 600B – a variant of the 'Katmai' Pentium III with a 133MHz front side bus. It couldn't match the Athlon 550 in many benchmarks, let alone the 600 and 650MHz versions, and still lagged behind the Athlon in when it came to gaming performance

In October, AMD responded with a 700MHz Athlon that pulled even further ahead. AnandTech's benchmarks of the time put it 20 per cent faster than the Pentium III 600B in the Quake II Crusher benchmark. It was nearly 27 per cent ahead in Quake III.

It was only with the launch of its Coppermine Pentium III processors in October 1999 that Intel could claw back the lead. Yet while the Pentium III 733EB was now king of the hill, an Athlon 700 could still benchmark faster in many tests than Intel's 700MHz Coppermine Pentium III.

As the clock speeds rose, the competition just grew hotter. In November 1999, AMD launched a new series of Athlons with a 0.18-micron (180nm) K75 core, taking the top speed up to 750MHz. In January and February, these were followed with 800 and 850MHz CPUs. Then just as Intel geared up to launch a (gasp!) 1GHz Coppermine Pentium III in March 2000, AMD stole its thunder by launching the Athlon 1000. To really take the proverbial, it did it two days earlier, giving AMD the first 1000MHz x86 CPU.

Athlon was first, but it wasn't fastest. The Pentium III 1000EB was actually ahead of the Athlon 1000 in many tests, partly due to superior SSE support in many popular benchmark games.

Yet there was only a few frames per second in it, and Athlon systems often had the edge on price. What's more, the Pentium III 1000 was only available to system builders at the time of launch. Anyone could get their hands on the 1GHz Athlon at the time.

INTEL SLOT 1

Holograms, black boxes and mountains of cache. **Ben Hardwidge** recalls the weird moment in time when Intel's CPUs came in Slot format

Packaging your CPU inside a big box before slotting it into your motherboard seems like a recipe for a thermal catastrophe now, but for a brief period around the turn of the Millennium, Intel (and later AMD) mounted their CPUs on circuitboards inside sleek black packages. They looked great too. There was now room for proper logos and a flashy physical design. Instead of dropping a nondescript-looking ceramic square into your motherboard, you had a fancy black box with a hologram on the front.

With their slick packaging, the first Pentium II CPUs looked great in the TV adverts and promo shots. I remember wanting one just because they looked so good – a small hologram sticker clearly goes a long way towards manipulating people like me! To the uninformed, it looked like these attractive slot-based CPUs were the way of the future, but if you took a peek inside the box, they were clearly a result of technological limitations at the time.

An original SECC Pentium II – look at the size of the cache chips on either side of the CPU area

CACHE FOR QUESTIONS

To understand the need for slot processors, we need to start by going back a bit further in time. Before the Pentium II, Intel had two major CPU designs. It had the Socket 7 Pentium MMX for consumer PCs, which was the last gasp for the first Pentium design, now running at up to 233MHz. MMX stands for multimedia extensions, and it effectively enabled a lot more functions to be handled in software on the CPU, rather than in hardware. For example, an MMX CPU enabled you to properly use a software PCI modem, rather than a full hardware one, saving you some money.

For servers and workstations, Intel had also introduced the Pentium Pro, a massive chip that was heavily geared towards pure 32-bit computing. It lacked consumer frills such as MMX instructions, but you could run more than one Pentium Pro in parallel on a multi-socket board.

The Pentium Pro also had a massive L2 cache that ranged between 256KB and 1MB, depending on the model. At this time, there was no way to integrate this cache directly into the CPU die, but the Pentium Pro did incorporate its huge L2 cache in the same Socket 8 package as the CPU die, and it also ran the cache at the same speed as the CPU. There was a big problem with this approach at the time though – making these Socket 8 packages with full-speed cache was an expensive process, and there were low yields.

Intel wanted to combine the two ideas, making a desktop CPU with loads of L2 cache, as well as consumer features such as MMX. It also needed to be better at executing 16-bit code (which was still used by some software at the time) than the Pentium Pro and, most importantly, it needed to be affordable to manufacture on a large scale.

This meant compromising, as Intel knew it couldn't practically equip a mainstream desktop CPU with loads of full-speed cache in a socket

Inside a Pentium II die – that's 7.5 million transistors produced on a 350nm manufacturing process, and with no integrated L2 cache
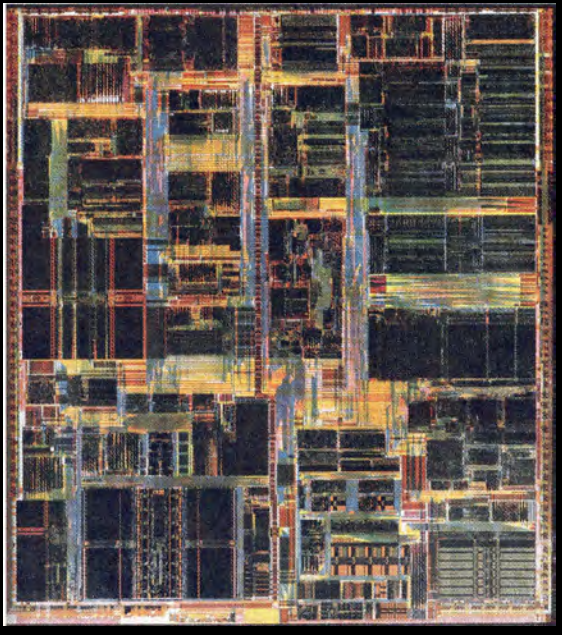
package. The answer was to manufacture the CPU package on a usual square format without the L2 cache, and to then mount that package on a circuitboard that contained the cache, resulting in the Pentium II in 1997. It had 7.5 million transistors, produced on a 350nm manufacturing process.

Like the Pentium Pro, the CPU used a separate 'back-side bus' to communicate with the cache, but unlike the Pentium Pro, the Pentium II could only run the L2 cache at half the speed of the CPU. Intel attempted to counter the performance of the cache by first doubling the amount of L1 cache, from the Pentium Pro's 16KB to 32KB on the Pentium II. The Pentium II's L2 cache also had a 16-way associativity, compared with 8-way on the Pentium Pro. A higher associativity means the CPU has a greater chance of finding the data it needs in that cache, but that it can take longer to search for it than a cache with lower associativity.

The other way Intel bumped up the Pentium II's performance was by simply equipping it with *a lot* of this L2 cache. All the first models of Pentium II came with a pair of

large 256KB cache chips, giving you 512KB in total – more than you found on some Pentium Pro CPUs.

By the end of it, you had a circuitboard containing a full CPU package in the middle, with two large cache chips next to it. This was then encased in a box with a thermally conductive metal back. The whole package was called a single edged contact cartridge, or SECC, and you would then attach a heatsink and fan arrangement to the metal back, and slot the whole setup into your motherboard.

The SECC package looked good on the surface, but if you took one apart, you could see that it was a bit of a bodge job. I was working in a computer shop at the time, and we joked that the Pentium II was a 'Socket 7 on a circuitboard' – you could even see the solder points where the socket pins could have been located on the CPU package. It was still a normal square CPU package – it was just mounted on a board instead.

Performance was mixed. If you were running full 32-bit software in Windows 95, then the Pentium II was generally faster than the Pentium MMX, but the latter still had the edge in some 16-bit software, such as MS-DOS games. It also didn't help that the first Pentium II CPUs used the same 66MHz front side bus as the final Pentium MMX chips, with the first Pentium IIs running at 233MHz, 266MHz and 300MHz, and a 333MHz variant arriving later, following a die shrink to 250nm. This meant that, in some cases, the top-end 233MHz Pentium MMX was faster than the low-end 233MHz Pentium II.

THINK OUTSIDE THE BOX

The processor's new clothes came well and truly off in 1998 when Intel introduced its budget range of Slot 1 CPUs, with the still ridiculous name of Celeron. The first generation of Celeron CPUs, codenamed Covington, removed all of the L2 cache from the circuitboard, as well as all the fancy, hologram-clad packaging. This left you with a peculiar-looking green circuitboard with a square CPU clearly soldered into the middle of it - Intel called this non-cartridge arrangement a SEPP format.

The lack of cache meant these Celerons performed poorly at the time, pushing people looking for a budget CPU towards AMD's K6 line-up, which still used the aging Socket 7 form

Slot 1 Celerons didn't come in a fancy chassis, and the first models didn't come with any L2 cache either. Photo by Qurren
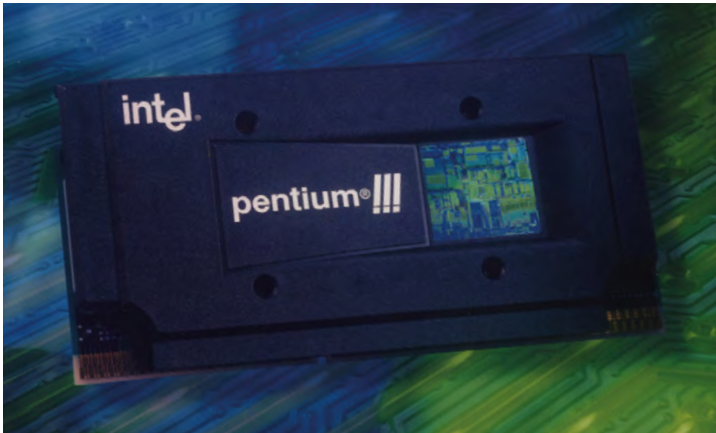
The Pentium III maintained the front with the hologram, but used the new SECC2 packaging, which left the circuitboard bare at the back

factor that Intel had deserted. However, the next generation of Celerons in 1999, codenamed Mendocino, overturned this part of CPU market. They were still mounted on circuitboards at first, but Intel had now nailed a method to produce a small amount of L2 cache on the same die as the CPU, running at full speed.

These new Celerons came with 128KB of full-speed on-die cache, meaning they were quicker than the 1st-gen (and much more expensive) Pentium II CPUs in some applications. By this time, Intel's next generation of Pentium II CPUs used a 100MHz front side bus, rather than 66MHz, which provided a significant performance boost over their predecessors.

Accompanied by the new Intel 440BX chipset, the new CPUs ran at 350MHz, 400MHz and 450MHz, and Intel clearly hoped that this FSB tweak would help distinguish the Pentium II line-up from the new Celeron lineup, despite the latter's faster cache.

if you took one apart, you could see that it was a bit of a bodge job

Unfortunately for Intel, overclockers had started discovering that there was plenty of headroom for some of the Mendocino Celerons to go much faster, despite Intel locking down the multipliers in an attempt to prevent it. If you put a new Celeron in an Intel 440BX board, or a board with VIA's competing 100MHz FSB Apollo Pro chipset, you could try moving the 66MHz FSB jumper to the 100MHz setting. If you were lucky, and you had a decent heatsink and fan on your CPU, your 300MHz Celeron would suddenly be running at 450MHz, thanks to its 4.5x multiplier. Combine the clock speed with the full-speed cache and your £60 processor could potentially outperform a £400 one.

I remember this well, and bought a 333MHz Mendocino Celeron with a 5x multiplier, in the hope of running it at 500MHz. It booted, but soon fell over once you got into Windows. Thankfully, my VIA Apollo Pro board also gave me the option to run the FSB at 75MHz or 83MHz if you tweaked the jumper switches right, and the latter setting stably ran my budget CPU at 415MHz. I had no need to buy a Pentium II now.

'Slotket' adaptors enabled you to plug a Socket 370 CPU into a Slot 1 motherboard. Photo by Konstantin Lanzet

FINAL SLOTS

While the first Mendocino Celerons were still mounted on Slot 1 circuitboards in order to maintain motherboard compatibility, their L2 integrated cache design meant they no longer technically needed the rest of the circuitboard. A few months later, the first Socket 370 Celerons started appearing, with 'Slotket' adaptors required in order to plug them into Slot 1 motherboards. It was a bizarre setup that persisted for an unusual length of time.

Intel wasn't quite ready to give up Slot 1 yet. Intel started by tweaking the design of the CPU chassis, removing the metal plate at the back. The final arrangement, called SECC2, retained the plastic front cover with the hologram, but left the circuitboard and CPU die bare at the back, in order to facilitate better thermal transfer to the cooler.

Next came the Pentium III, codenamed Katmai, which added SSE instructions, but was still fundamentally based on the same P6 core as the Pentium II. It also still had an external half-speed L2 cache setup, with both the CPU and cache mounted on a circuitboard. It wasn't until the Coppermine (don't be fooled by the name – all the interconnects were aluminium, rather than copper) revision of Pentium III, with a die shrink to 180m, that Intel finally integrated 256KB of full-speed L2 cache into a CPU die containing 29 million transistors.

Later came a 133MHz front side bus and Intel's 820 chipset, accompanied by high-bandwidth but expensive RDRAM. However, the Slot 1 design still persisted. Even the first Pentium III to break the 1GHz barrier was based on a slot design. Intel needed to maintain compatibility, which was handy for many of us enthusiasts who had worked out that you could still run the latest CPUs on some old 440BX boards by overclocking the front side bus to 133MHz. There was also no shortage of Slotket adaptors at this time, enabling you to install Socket 370 CPUs into Slot 1 motherboards.

The final Slot 1 CPU I saw was an engineering sample of a 1.13GHz CPU that Intel sent to PC Pro magazine, but the chip was recalled due to stability problems. The slot era was now over, and motherboards based on Intel's later SDRAM-based 815 chipset only came in Socket 370 format. The Pentium III carried in on socket format, as did the later Pentium 4, and the CPU industry hasn't looked back since. Slot processors might have looked good, and a part of me misses the fancy casing with the holograms, but there's no doubt that integrating cache directly onto the die is a much faster and more efficient way of doing it. **CPC**
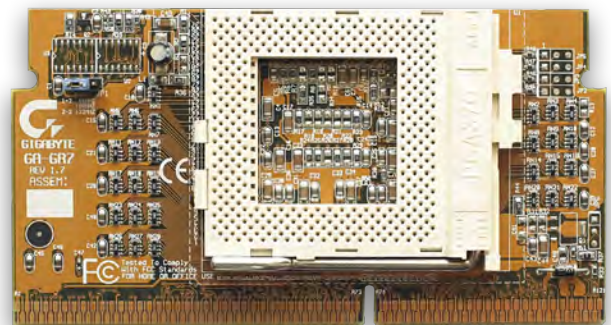
GRAPHICS

CGA

Ben Hardwidge delves into the workings of the PC's very first colour graphics adaptor

Some 16-colour ASCII art by Ben Hardwidge, aged 12

P eople often nostalgically reminisce about archaic technology, laughing about the frustrations and limitations of cassette tapes or floppy disks, before adding 'but they were amazing at the time!' You simply can't hide the horror of the PC's first colour graphics adaptor (CGA) behind such rose-tinted glasses (although orange-tinted glasses might help – more on that later). Nobody, absolutely nobody, thought CGA was amazing at the time.

I had a CGA PC in the 1980s, and but even then you felt disappointed when you fired up a PC game to be greeted by a mess of purple and black on the screen. At the time, we joked that CGA stood for 'crap graphics adaptor'. Nobody thought of IBM computers as games machines then, of course – CGA was the product of IBM trying to make a graphics standard that could display bar charts properly. It wasn't meant to compete with the Commodore 64.

Better graphics came to the PC later, of course, but CGA was supported for a long time. The later EGA (enhanced graphics

adaptor) and VGA (video graphics array) cards were very expensive at first, so CGA still had a home in cheap IBM PC compatible machines, such as Amstrad's PC1512. CGA first appeared in 1981, but new software was still supporting it well into the early 1990s – you can even run Windows 3.0 on it.

TEXT MODE

At its basic level, a standard 16KB (yes, KB) CGA card can access a palette of 16 colours, or rather eight colours at two intensities. It's basically 4-bit colour, with three bits allocated to red, green and blue (RGB), and the fourth bit enabling you to change the 'intensity' of the colour (RGBI).

At the first level of intensity, you get black, blue, green, cyan, red, magenta, brown and light grey. The second level of intensity basically gives you the same colours but with an extra level of intensity, which turns the brown into a yellow, the light grey into a white and the black into a dark grey, while creating light versions of the other colours.

Now, you might think 16 colours sounds okay for 1981, but you can only display all these 16 colours on the screen at once in text mode – the mode you used to see on BIOS screens before we had fancy EFI systems. On a CGA card, the text display has an effective resolution of 640 x 200, but it can only display text characters on it, with 80 characters on the X axis, and 25 on the Y axis.

As a kid, I used to play around with this mode quite a lot, as it was the only way to get a lot of colours on the screen. If you knew your ASCII codes, you could display various lines and blocks as text characters and make a picture. You effectively have to 'type' a picture, rather than drawing it – I used to spend hours doing it. To type ASCII codes, you hold down Alt and type a three-digit number – 176, 177, 178 and 256 give you three blocks of variable shading and a solid block, for example – it still works in Windows. In this text mode, you could assign each character a foreground and a background colour.

Game developers used this mode too – I had a clone of Ms Pac-Man that used to run in text mode rather than graphics mode, as well as a clone of Breakout called Bricks. On a standard CGA card, it was the only way to get access to lots of colours. There was a trick to enable you to display all 16 colours at an effective graphical resolution of 160 x 100, by changing the number of lines of each text character to display. However, it was rarely used. If you wanted graphics rather than text, you usually either had four colours on the screen at 320 x 200, or one colour at 640 x 200.

TRY CGA FOR YOURSELF

In the unlikely event that you want to try out the shocking disgrace that is CGA graphics for yourself, you can do it in DOSBox (**dosbox.com**). This handy software creates a virtual machine designed to recreate a high-spec PC from the 1990s. It loads a sound card and MIDI drivers automatically, and gets you set up with a mouse too. It's great if you want to play a round of Doom or X-Wing.

However, later VGA cards didn't support CGA palette-switching as standard. They could run CGA software, but usually in the default black, white, magenta and cyan palette, even if they used a different palette on a CGA machine. DOSBox runs in VGA mode by default, which results in the same problem.

To get around it, you'll need to open Options in your Start menu's DOSBox folder, which takes you into the config file. Scroll down to the '[dosbox]' section, and type 'cga' after 'machine='. After that, scroll down to the '[render]' section, and type 'true' after 'aspect='.

On some monitors you may find that you still don't get the correct 4:3 aspect ratio, even after changing the aspect setting to true. If that happens, we found that setting 'fullresolution=' to '1366x768' fixed it on our 4K monitor. We have no idea why, but it seems to work.

If you want to run a really old game, it may also only be optimised for early processors, and will run too fast on DOSBox's standard settings. If you want to emulate an XT-era 8086 PC, scroll down to [cpu] and type 'simple' after 'core=' and change the number of cycles to 530 (this isn't exact, but it was near enough in our tests).

COLOUR GRAPHICS

Let's start with the former, as that was the one that enabled you to get actual colour graphics on your PC. Generally, black was the background colour, and you then had three other colours. As standard, most games used CGA in BIOS mode 4 (the default BIOS mode for graphics), with the high-intensity version of palette 1, which gave you black, white, light cyan light and magenta. It enabled you to make clearly defined shapes with black on white, gave you cyan for skies and water and then everything else would have to be filled in with magenta. It generally looked hideous, although it was sometimes better for space games – Captain Blood looked surprisingly good in this mode.

You could get other palettes too. Palette 0 was also available in BIOS mode 4, and gave you red, green, black and brown as standard, or light red, light green, black and yellow in high-intensity mode. The latter mode generally looked better in games to me. It meant you couldn't get blue for skies, but you could do pretty sunsets and dark dungeons well. One of my favourite games to use this palette was a fantasy barbarian game called Targhan, which genuinely did look amazing considering the technology it was using.

As a kid, I also discovered a trick while playing with the night vision filters for my Dad's binoculars. If you look at the cyan, magenta, black and white palette through an orange filter, it becomes the light yellow, light red, light green and black palette. I bought some orange acetate from the local art shop and stapled it to a cardboard frame with Blu-Tack in each corner – I could then swap between palettes at will!

The low-intensity version of this palette was also used in games occasionally. One example is Pharaoh's Tomb, an early work by George Broussard at Apogee, who later went on to work on the Duke Nukem games.

Another trick often used by game developers was to switch the CGA card to BIOS mode 5, which in high-

you fired up a PC game to be greeted by a mess of purple and black on the screen

This breakout clone, called Bricks, was effectively built in text mode so it could access all 16 colours
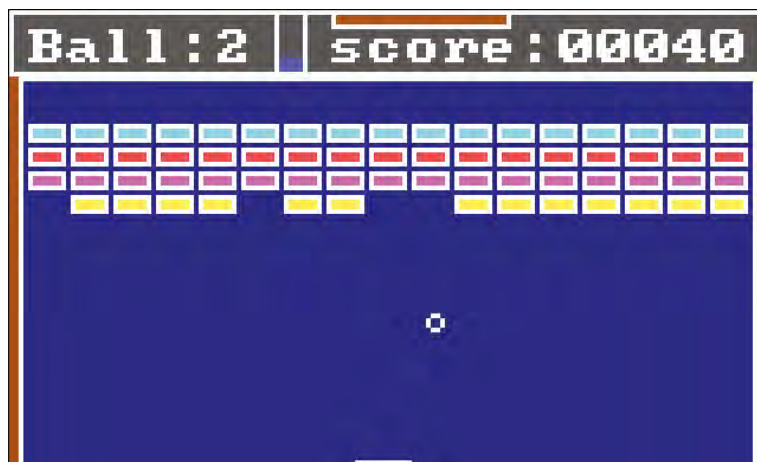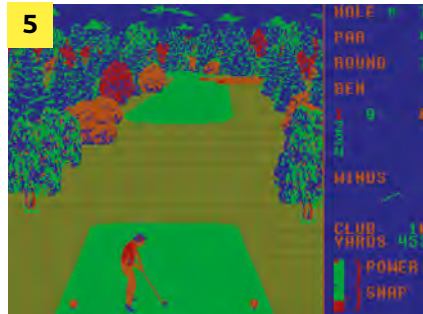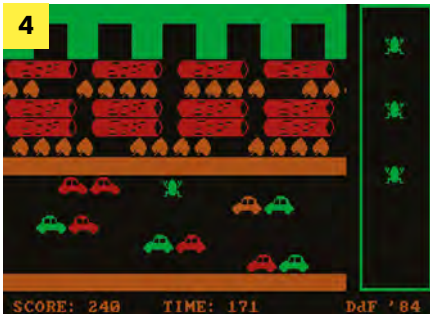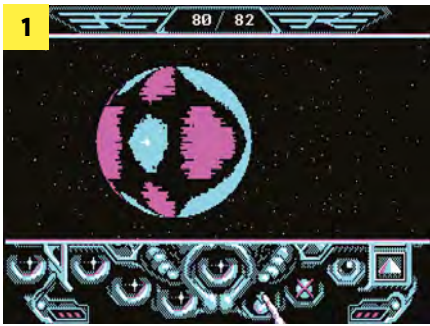
1. Captain Blood
BIOS mode 4, palette 1 high intensity

2. Formula 1 Grand Prix Circuit
BIOS mode 5

3. Targhan
BIOS mode 4, palette 0 high intensity

4. Ribit (a Frogger clone)
BIOS mode 4, palette 0 low intensity

5. World Class Leaderboard
BIOS mode 4, palette 0 low intensity, black background replaced with blue

intensity mode gave you access to a black, white, light red and light cyan palette. It had the same limitations as the default cyan, magenta, black and white palette, but to my eyes, the red looked less garish than magenta .

A few games also ventured outside these palettes with some tricks, which usually involve replacing black as the background colour. Sierra's Leisure Suit Larry in the Land of the Lounge Lizards, for example, used palette 0 at low intensity, but replaced the black background colour with blue (it looks hideous). This palette worked well in golf game World Class Leaderboard, though, with green and brown trees, red leaves, green grass and blue skies and water – colours you should be able to take for granted.

Sierra used the same trick in King's Quest IV: The Perils of Rosella, but using the BIOS mode 5 palette, again replacing the black with blue. The result was a blue, cyan, red and white palette, which worked well with blue sea against cyan sky, but meant the grass and trees looked very odd.

MONO GRAPHICS
The other main graphical option available to standard CGA cards was the 'high-resolution' 640 x 200 monochrome mode. It was used in games that had a fair amount of detail in the graphics, such as Sim City, Death Track and Xenon II: Megablast, among others. It was also used for early GUI operating systems, such as Gem and Windows 3.0.

However, only the horizontal resolution was higher than the colour graphics resolution – the vertical resolution was the

Xenon II: Megablast 640 x 200 'high-resolution' monochrome mode

same. The result was double-height, rectangular pixels, rather than square ones. This mode also produced a hideous moiré effect on lots of CGA monitors, making it difficult to look at the screen.

COMPOSITE MODE
There was one more trick to getting a standard CGA card to display more characters, and it involved cleverly using the composite output, rather than the 9-pin RGB monitor output. Irritatingly, most PAL TVs in the UK weren't able to handle this mode, as it's dependent on the NTSC chroma decoder mistakenly seeing some luminance signals as colour.

As a result, you could effectively make new colours by lining up pixels in certain patterns on an NTSC display, and again by using different intensities. By placing one colour pixel next to another one, you could make an entirely new colour, and it looked solid rather than a messy mix of pixels. The result is astonishing, enabling you to create a much wider colour palette.
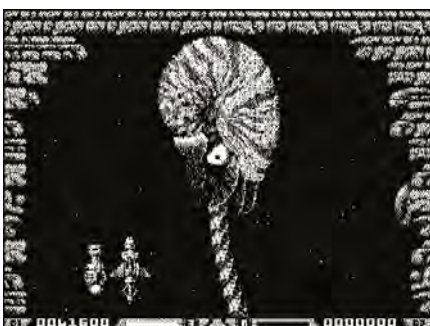
The disadvantage, of course, is that the effect can only be achieved by placing pixels next to each other, which effectively reduces the horizontal resolution from 320 to 160. Some games supported this mode, though, including Sierra's original King's Quest game. CPC

King's Quest in composite CGA **King's Quest in RGB CGA**

EGA

Stuart Andrews recalls how 16 colours changed the PC world

Pity the poor PC of 1983-1984. It wasn't the graphics powerhouse we know today. IBM's machines and their clones might have been the talk of the business world, but they were stuck with text-only displays or low-definition bitmap graphics. The maximum colour graphics resolution was 320 x 200, with colours limited to four from a hard-wired palette of 16. Worse, three of those colours were cyan, brown and magenta, and half of them were just lighter variations of the other half.
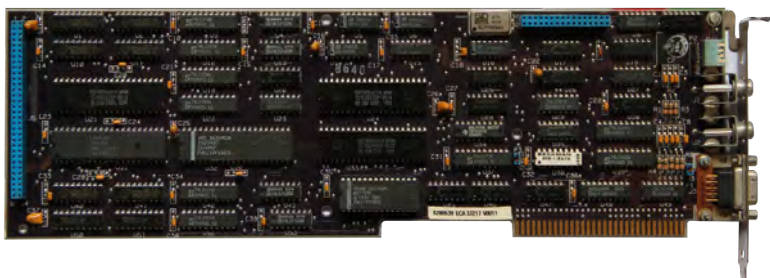
By this point, IBM's Color Graphics Adaptor (CGA) standard was looking embarrassing. Even home computers such as the Commodore 64 could display 16-colour graphics, and Apple was about to launch the Apple IIc, which could hit 560 x 192 with 16 colours. IBM had introduced the Monochrome Display Adaptor (MDA) standard, but this couldn't dish out more pixels, only higher-resolution mono text.

Meanwhile, add-in-cards, such as the Hercules or Plantronics Colorplus, introduced higher resolutions, but did nothing for colour depth. The PC needed more, which IBM delivered with its updated 286 PC/AT system and the Enhanced Graphics Adaptor (EGA).

THE NEW STATE OF THE ART

The original Enhanced Graphics Adaptor was a hefty optional add-in-card for the IBM PC/AT, using the standard 8-bit ISA bus and with support built into the new model's motherboard. Previous IBM PCs required a ROM upgrade in order to support it.

Using Chips and Technologies' EGA chipset, early graphics card manufacturers such as ATi could produce smaller, cheaper boards. Credit: Vlask, CC BY-SA 3.0

It was massive, measuring over 13in long and containing dozens of specialist large scale integration (LSI chips), memory controllers, memory chips and crystal timers to keep it all running in sync. It came with 64KB of RAM on-board but could be upgraded through a Graphics Memory Expansion Card and an additional Memory Module Kit to up to 192KB. Crucially, these first EGA cards were designed to work with IBM's 5154 Enhanced Color Display Monitor, while still being compatible with existing CGA and MDA displays. IBM managed this by using the same 9-pin D-Sub connector, and by fitting four DIP switches to the back of the card to select your monitor type.

EGA was a significant upgrade from low-res, four-colour CGA. With EGA, you could go up to 640 x 200 or even (gasp) 640 x 350. You could have 16 colours on the screen at once from a palette of 64. Where once even owners of 8-bit home computers would have laughed at the PC's graphics capabilities, EGA and the 286 processor put the PC/AT back in the game.

The original IBM EGA card was a whopper, even without the additional daughtercard and memory module kit. Credit: Vlask, CC BY-SA 3.0

BIRTH OF AN INDUSTRY

However, EGA had one big problem; it was prohibitively expensive, even in an era when PCs were already astronomically expensive. The basic card cost over $500 US, and the Memory Expansion Card a further $199. Go for the full 192KB of RAM and you were looking at a total of nearly $1,000 (approximately £2,600 inc VAT in today's money), making the EGA card the RTX 3090 of its day, and only slightly more readily available. What's more, the monitor you needed to make the most of it cost a further $850 US. EGA was a rich enthusiast's toy.

However, while the initial card was big and hideously complex, the basic design and all the tricky I/O stuff were relatively easy to work out. Within a year, a smaller company, Chips and Technologies of Milpitas, California, had designed an EGA-compatible graphics chipset. It consolidated and shrunk IBM's extensive line-up of chips into a smaller number, which could fit on a smaller, cheaper board. The first C&T chipset launched in September 1985, and within a further two months, half a dozen companies had introduced EGA-compatible cards.

Other chip manufacturers developed their own clone chipsets and add-in-cards too, and by 1986, over two dozen manufacturers were selling EGA clone cards, claiming over 40 per cent of the early graphics add-in-card market. One, Array Technology Inc, would become better known as ATI, and later swallowed up by AMD. If you're on the red team in the ongoing GPU war, that story starts here.

CHANGING GAMES

EGA also had a profound impact on PC gaming. Of course, there were PC games before EGA, but many were text-based or built to work around the severe limitations of CGA. With EGA, there was scope to create striking and even beautiful PC games.

This didn't happen overnight. The cost of 286 PCs, EGA cards and monitors meant that it was 1987 before EGA support became common, and 1990 before it hit its stride. Yet EGA helped to spur on the rise and development of the

Forgive the blocky pixels and 16-colour palette. In Catacombs 3-D and Catacombs: Abyss lay the seeds of Wolfenstein and Doom

PC RPG, including the legendary SSI 'Gold Box' series of Advanced Dungeons and Dragons titles, Wizardry VI: Bane of the Cosmic Forge, Might and Magic II and Ultima II to Ultima V.

It also powered a new wave of better-looking graphical adventures, such as Roberta Williams' Kings Quest II and III, plus The Colonel's Bequest. EGA helped LucasArts to bring us pioneering point-and-click classics such as Maniac Mansion and Loom in 16 colours. And while most games stuck to a 320 x 200 resolution, some, such as SimCity, would make the most of the higher 640 x 350 option.

With EGA, there was scope to create striking and even beautiful PC games

What's more, EGA made real action games on the PC a realistic proposition. The likes of the Commander Keen games proved the PC could run scrolling 2D platformers properly. You could port over Apple II games such as Prince of Persia, and they wouldn't be a hideous, four-colour mess.

And when the coder behind Commander Keen – a certain John Carmack – started work on a new 3D sequel to the Catacomb series of dungeon crawlers, he created something genuinely transformative. Catacomb 3-D and Catacomb: Abyss gave Carmack his first crack at a texture-mapped 3D engine, and arguably started the FPS genre.

Sure, EGA had its limitations – looking back, there's an awful lot of green and purple – but with care and creativity, an artist could do a lot with 16 colours and begin creating more immersive game worlds.

A SLOW DECLINE

EGA's time at the top of the graphics tech tree was short. Home computers kept evolving, and in 1985, Commodore launched the Amiga, supporting 64 colours in games and up to 4,096 in its special HAM mode. Even as it launched EGA, IBM was talking about a new, high-end board, the Professional Graphics Controller (PGC), which could run screens at 640 x 480 with 256 colours from a total of 4,096.

PGC was priced high and aimed at the professional CAD market, but it helped to pave the way for the later VGA standard, introduced with the IBM PS/2 in 1987. VGA supported the same maximum resolution and up to 256 colours at 320 x 200. This turned out to be exactly what was needed for a new generation of operating systems, applications and PC games.

What extended EGA's lifespan was the fact that VGA remained expensive until the early 1990s, while EGA had developed a reasonable install base. Even once VGA hit the mainstream, many games remained playable in slightly gruesome 16-colour EGA. Much like the 286 processor and the Ad-Lib sound card, EGA came before the golden age of PC gaming, but this standard paved the way for the good stuff that came next. **CPC**

VGA

Stuart Andrews looks at the tech that transformed the PC into a gaming and graphics powerhouse, 256 colours at a time

The technology that put PC graphics firmly on the map arrived in April 1987 as part of IBM's PS/2 line of PCs. IBM saw the PS/2 as the answer to its biggest problems, putting Big Blue (as we all used to call it) back in control of the PC architecture and one step ahead of the clone manufacturers.

To do so, it had Intel's latest processors, cutting-edge connection options and the fastest floppy disk storage, not to mention a revolutionary new high-bandwidth system bus. But what turned out to be the PS/2's most important

In 1987 the PC wasn't exactly considered a graphics powerhouse

feature was its new graphics hardware – the Video Graphics Array, or VGA.

In 1987 the PC wasn't exactly considered a graphics powerhouse. Apple's Mac II, launched in March the same year, had a graphics card that could support up to 256 colours at 512 x 384. The Commodore Amiga could display full-screen animated graphics with up to 64 colours at 320 x 240, or 4,096 colours in still images using its legendary, flicker-tastic HAM mode.

The best the PC had to offer was the EGA (Enhanced Graphics Adapter) standard, covering resolutions of up to 640 x 350, but with only 16 simultaneous colours from a fixed palette of 64. If you wanted to create graphics or play games on your PC, you needed to really like basic colours with a strange preponderance of green and purple. Graphics enthusiasts were rendering ray-traced 3D graphics on their Amigas, albeit very slowly, but nobody sensible would even think of doing so on a PC.

VGA didn't put the PC at the graphics cutting edge, but it did put it back in the race. The new hardware supported resolutions of up to 640 x 480 with 16 colours, or 320 x
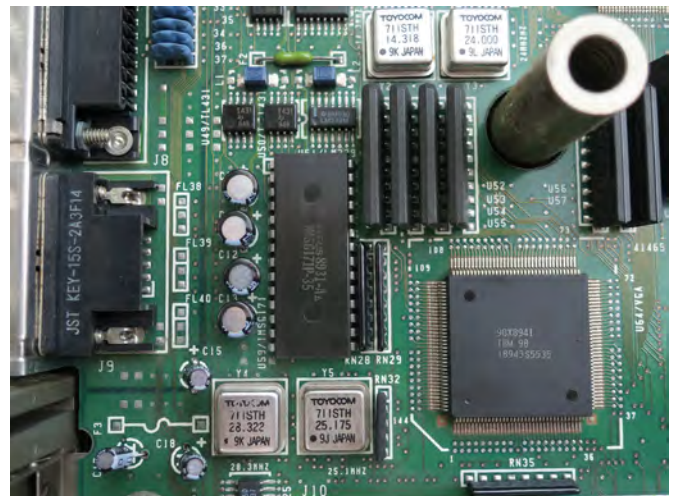
The VGA section of a 1988 IBM PS/55 model 5550-T. You can see the VGA chip, the INMOS RAMDAC, the two timing crystals and 256KB of video RAM. Credit: Darklanlan, CC 4.0 custompc.co.uk/CC4

200 with up to 256. What's more, those 256 colours could be redefined at any time, from an 18-bit palette of 262,144 colours. With VGA, you could put a photo on the screen and it kind of looked like a photo. Artists could create 2D images with sophisticated colour and shading effects. PC games went from looking shocking to looking seriously awesome. VGA was literally a game changer.

PC GRAPHICS GET THE WOW FACTOR

Weirdly, VGA didn't arrive as a new standard, or even as an add-in graphics card. On the first PS/2 PCs, it came in the form of a chip containing the display controller, along with 256KB of dedicated RAM, a pair of timing crystals and an external RAMDAC.

This already made it a much more integrated technology than the original EGA chipsets, which contained dozens of processors, and put it more in line with the integrated chips coming from third parties. What's more, it was the higher-

ATi was one of many graphics chip and card manufacturers to first clone VGA, then enhance it. Credit: Samuel Demeulemeester, CC 4.0
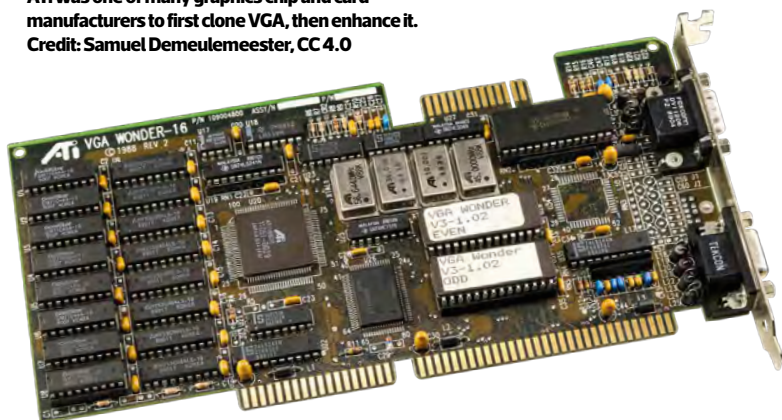
end option of two new graphics standards. The cheaper PS/2 models were stuck with the Multi Colour Graphics Adapter (MCGA) which had the same 256-colour mode but lacked VGA's higher resolutions.

Like IBM's new MCA bus architecture, MCGA didn't last long beyond the PS/2, but VGA developed a life of its own. Beyond hardware-level support for smooth scrolling, and a barrel shifter designed to shift incoming data from the CPU to the display at seven bits at a time, it didn't actually do much in the way of graphics acceleration.

However, it did set a new baseline standard for PC graphics, and for hardware and software support. Crucially, through its RAMDAC and 15-pin D-Sub connector, it established how the PC could convert digital instructions into a 256-colour analogue video signal, setting the stage for the 16-bit and 24-bit colour standards to come.

Instead of sending six colour signals from the graphics card to the monitor, like the older EGA chipsets, the VGA chipset and its RAMDAC sent only three signals – red, green and blue, with a potential 64 different levels for each. For VGA, this resulted in an 18-bit palette of up to 262,144 colours, 256 of which could appear simultaneously in Mode 13h. Once adopted, this same core technology gave scope for 16-bit and 24-bit colour in later graphics chips, with up to 65,536 colours or 16.7 million colours on the screen at once.

Resolution wasn't the base level VGA spec's strength. In fact, PC journalists of the time pondered why it was

Eye of the Beholder – VGA's larger colour palette (right) gave artists the chance to use more realistic shading compared with EGA (left)

stuck at 320 x 200 in Mode 13h. However, programmers found workarounds. A handful of games, such as the legendary horror game Dark Seed, opted to work with a reduced 16-colour palette in order to use the full 640 x 480 resolution. Meanwhile, Michael Abrash, who would later work with id Software on Quake, worked out an approach that enabled programmers to use 256 colours at a slightly higher resolution of 640 x 240, which he dubbed Mode X.

Meanwhile, Windows 2.0 moved to adopt the 640 x 480 mode with 16 colours, bringing the interface closer to what we expect from a GUI today. However, many of the applications and games we think of as belonging to the VGA era stuck to Mode 13h and its 320 x 200 resolution. What's more, with the CPU performing most of what we'd now call the GPU's legwork, this was arguably for the best – until the Intel 486 appeared in 1989, there wasn't any really CPU powerful enough to handle gaming at higher resolutions.

THE IMPACT OF VGA

Luckily, those colours alone had a huge impact. The ZSoft Corporation's PC Paintbrush and Electronic Arts' Deluxe Paint II revolutionised professional graphics and computer art on the PC, thanks to 256-colour support. VGA also made CorelDRAW, launched in January 1989, a realistic alternative to the digital design packages appearing on Apple's computers.

Meanwhile, for PC games, VGA was nothing short of transformative. Sure, the 64,000 pixels on your monitor looked a little chunky; however, with 256 colours, the artists working at leading developers, such as LucasArts, Sierra Online, Microprose, Electronic Arts and Origin Systems, were able to produce sprites that looked more like recognisably human (or inhuman) characters, and background scenery that could bring their game worlds to life. Plus, while the PC couldn't pull off the same smooth scrolling, sprite-scaling tricks as the Commodore Amiga or 8-bit consoles, its best games were developing a visual richness of their own. As the PC moved into the 386 era, it was beginning to be taken seriously as a gaming machine.

Taken on its own, the first VGA chipset wouldn't have made such an impact. After all, you only got it to use it if you bought a pricey IBM PS/2 machine. Instead, it really only gained momentum once it began to appear in add-in cards.

The Secret of Monkey Island – moving to VGA (right) enabled PC developers to create more human-like characters compared with EGA (left)

IBM was first out of the gate with its PS/2 Display Adapter, a card that gave any reasonably modern IBM-compatible PC with ISA slots a VGA chipset for the princely sum of $599 US (about £420 inc VAT then and £1,200 inc VAT in today's money).

Yet by this point, the older EGA standard had spawned a growing industry of third-party manufacturers, adept at mimicking or reverse-engineering IBM's technology and spawning their own versions. What's more, these guys didn't stop at simply replicating IBM's latest standards; they wanted to add a little extra sauce to their cards by actively enhancing them.

As a result, October 1987 saw the launch of the first VGA-compatible third-party graphics card, the STB VGA Extra. It

Developers were able to produce sprites that looked more recognisably human

did everything VGA did, albeit with a few foibles here and there, with some optimisations that made it slightly faster. By mid-1988 to 1989, the likes of Tseng Labs, Cirrus Logic, Chips and Technologies and ATi were entering the fray, and not only were they driving prices down to $339 US, but they were also adding new capabilities. These enhanced VGA cards added features to accelerate video, or increased the RAM to 512KB, and tinkered with the BIOS to cover more

Dark Seed sacrificed colour depth for resolution, in order to do justice to H.R. Giger's artwork

advanced resolutions, such as 800 x 600 in 16 colours or 640 x 480 with 256 colours.

This in turn put pressure on the system bus. The original VGA controllers were so undemanding that they couldn't exhaust the miserable bandwidth of the 8-bit ISA bus, but as these new chipsets emerged, they required more bandwidth and a spot on the wider 16-bit ISA bus.

As time went on and Intel's CPUs grew faster, demands would grow accordingly, resulting in the development of the Extended ISA (EISA) bus and VESA Local Bus. However, this complicated the situation further, with the fastest enhanced VGA cards, based on Tseng Labs or Cirrus Logic tech, performing best in 16-bit versions running on the 16-bit ISA bus, although this wasn't always the case with every chipset.

By 1989, NEC would lead the early graphics chipset manufacturers in the creation of the Video Electronics Standards Association and the Super VGA BIOS, opening up support for higher resolutions and colour depths across the PC industry. Windows acceleration became the new battleground and video acceleration became the next cutting-edge technology.

Yet all these new cards and advanced feature sets still had the VGA standard at their core. VGA became the base requirement for new PCs running later versions of Windows or IBM's OS/2. In many respects, IBM had built the foundation of PC graphics for the next ten to 15 years. In fact, you could argue that VGA is still the foundation.

If so, it probably wasn't a whole lot of comfort to IBM. While VGA was the last graphics standard IBM managed to establish, it wasn't for the want of trying. Even as it launched VGA, it was preparing its 8514 graphics adaptor, with fixed functions to accelerate common 2D drawing processes, such as drawing lines or filling shapes with colour. In 1990, it hoped to supersede VGA with its new 1,024 x 768, 256-colour standard, XGA.

Both these new standards floundered because they were designed to run on IBM's MCA bus, while IBM's clone-making rivals focused on getting the most out of the existing 16-bit ISA bus, before working on the proposed EISA replacement. The result? Super VGA became the new de facto standard, while IBM lost its domination of the PC industry. Bad news for Big Blue, but good news for those of us who enjoyed the more cost-conscious, game-focused machines in the years that followed.

THE VERY BEST OF EARLY VGA

Coming at you in 256 glorious colours at 320 x 200

WING COMMANDER

By 1990, VGA was well established, the 386 had become the mainstream PC CPU and the 486 had just appeared. All three technologies found their perfect showcase in Wing Commander. Chris Roberts' dazzling space combat game offered stunning scaling and rotating sprite spaceships, Star Wars-inspired cinematic cutscenes and thrilling mission design. It paved the way for LucasArts's X-Wing games and the Elite revival, while showing the way forward for a new breed of Hollywood-influenced, story-driven games.

WOLFENSTEIN 3D

This pioneering Nazi-blasting FPS from id Software was originally designed to run with EGA graphics, but that became unthinkable once John Carmack and his crew were unleashed on VGA. The texture-mapped walls made the most of simple bitmapped textures, but the sprites for the Nazis, zombies and Hitler-loving hounds looked dazzling in 256 colours, as id pushed the 386 CPUs of the era to their limits. Doom would push 3D realism further still, but even that relied on the limited capabilities of the humble VGA card.

EYE OF THE BEHOLDER

Before The Elder Scrolls, Lands of Lore and Ultima Underworld redefined the RPG genre, Eye of the Beholder set a new benchmark for the expected standard of graphics.

With its graphics built in Deluxe Paint II and powered by 256-colour VGA, it updated the 'Dungeon Master' tile-based dungeon crawler genre, adding customisable characters, an engaging story, and the kind of D&D lore we've come to know and love. People tend to remember the excellent Amiga port of this game, but the PC version was the original, and in many ways the best.
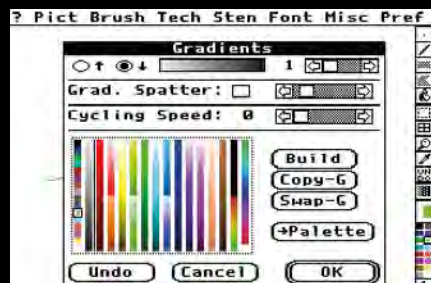

THE SECRET OF MONKEY ISLAND

It's a toss-up which was more influential – the first of Ron Gilbert's beloved pirate series or the awesome Indiana Jones and the Fate of Atlantis. Either way, these two titles used VGA's capabilities to full effect, with impressive sprite characters and glorious backdrops that made the most of the larger colour palette. You no longer had to use your imagination to visualise locations, because the artists had done the hard work for you. The Secret of Monkey Island II: Le Chuck's Revenge went even further, with graphics that embraced a stunning, hand-painted look.

DELUXE PAINT

By 1987, PC users were already getting sick of Commodore Amiga users rubbing their face in the dirt with Deluxe Paint. With advanced drawing tools, fills and scaling capabilities, these smug gits could create images as cool-looking as the legendary King Tut mask or Birth of Venus. When Deluxe Paint II was ported to the PC in 1988, PC users were invited to the party, helping to establish the PC as the graphics powerhouse it would become with the arrival of Adobe Photoshop, Paint Shop Pro and CorelDRAW.

DARK SEED

Combining a point-and-click adventure with psychological horror and the art of Alien maestro, H.R. Giger, Dark Seed used VGA in an unusual way, dropping down to 16 colours in order to hit the maximum 640 x 480 resolution (apparently, Giger made this a condition of the team using his art). In any case, it worked, mixing sequences set in an American town setting with scenes straight from one of Giger's dark sci-fi body horror netherworlds. Even now, it's one weird-looking game. **CPC**

3DFX VOODOO 3D

Reflective surfaces, smooth frame rates and the pure awesomeness of GLQuake. **Stuart Andrews** recalls the truly transformative effect of 3Dfx's Voodoo chipset on PC gaming

I t's a classic case of being the right company with the right tech at the right time. 3Dfx launched its revolutionary Voodoo Graphics chipset just as fully polygonal 3D graphics hit the mainstream and PC gamers wanted an easy and accessible way to get them.

In late 1996, Quake and Tomb Raider had just been released, the Nintendo 64 was out in Japan and North America, and the Sony PlayStation and Sega Saturn were still in their first year. Reliant purely on CPU horsepower, and with no dedicated 3D hardware to back it up, the PC was beginning to lose its place as the king of gaming platforms.

Sure, it had a bunch of 2D/3D accelerator cards, but they were too damn slow to make any difference. With the Voodoo Graphics chipset, 3Dfx played a bigger role than any other graphics hardware manufacturer in turning around that situation. In doing so, it made 3D acceleration an absolute, cast-iron must-have feature.

THE BIRTH OF VOODOO

3Dfx was founded in San Jose, California in 1994, by a trio of ex-Silicon Graphics (SGI) employees, Ross Smith, Scott Sellers and Gary Tarolli. At the time, SGI was by far the biggest name in 3D graphics, with its enormously expensive workstations used to create the pioneering CGI effects in Terminator 2 and Jurassic Park.

What's more, SGI was already involved in 3D gaming hardware, developing the core components for what would eventually become the Nintendo 64. At this time, however, some of SGI's engineers were thinking that there were serious opportunities being overlooked in developing 3D hardware for PCs.

One group would eventually leave to found a company called ArtX, which would later get bought by ATI. Meanwhile, Smith, Sellers and Tarolli founded a new startup, Pellucid, in 1992, with the intention of bringing affordable 3D hardware to the PC.

In 1993, Pellucid was bought by Media Vision, a company that had grown rich from selling multimedia kits for PCs during the CD-ROM revolution. Pellucid had proposed the design and manufacture of a PC 3D gaming chip, and Media Vision wanted some of that action.

Unfortunately, Media Vision had its own (mostly legal) issues, and went out of business. However, just when the situation looked bleak, Scott Sellers met Gordon Campbell, founder of the pioneering graphics chip manufacturer, Chips & Technologies. Campbell asked the trio what they wanted to do, and helped them to find the venture capital to do it.

With Smith working as vice president of sales and marketing, Sellers and Tarolli used all the know-how they'd built up at SGI and Pellucid to design a cost-efficient 3D architecture built specifically to handle the polygonal rendering pipeline used in 3D games.

If 3Dfx needed a killer app, GLQuake delivered. You could play id's cutting-edge 3D title at 640 x 480 in 16-bit colour at a smooth 30fps

Tomb Raider was a 3Dfx showcase, smoothing out the blocky textures, improving frame rates and adding transparent water to the mix

With Sellers working on the hardware and Tarolli on the core algorithms, the 3Dfx team came up with the idea of an add-in card that only accelerated 3D, and left 2D graphics and Windows acceleration to a separate graphics card. At first, all they had working was a software simulation built in C and running on a Pentium 90 processor, but this evolved into a card based on two heavily optimised processors.

The first, the Frame Buffer Interface, took polygon scene data from the CPU and applied Z-buffering and Gouraud shading, tracking which polygons were visible, and ensuring that only those were drawn and filled, then applying shading to provide an impression of simulated light and colour.

Each frame of the image would then be converted into scan lines from top to bottom, then sent on to the second chip. The Texture Mapping Unit, or T-Rex as it was known, applied perspective-correct textures, complete with mipmapping (the process of using smaller, less-detailed textures as an object gets further away) and bilinear or trilinear filtering (smoothing out blocky textures when displayed at their largest size close to the viewpoint).

What's more, the T-Rex supported alpha blending, for convincing transparency effects. No other consumer-grade graphics hardware was able to handle this at the time. Each chip worked with its own frame buffer or texture memory
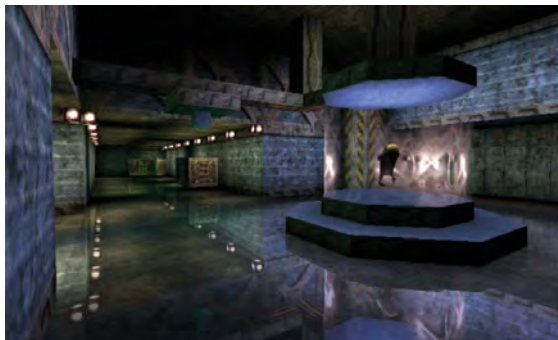
By the time Unreal hit the market, 3Dfx was established as the best tech to run it. Check out those shiny surfaces and lavish textures

– a bank of 2MB of high-bandwidth (for the time) EDO RAM, and the resulting scanlines were fed out to a DAC, which output to a good, old-fashioned analogue VGA output.

THE FIRST CARDS

The fact that the Voodoo Graphics chipset was 3D-only helped to keep down the price, but it did make using the card a little strange. While the card itself could talk to the CPU and system RAM through the PCI bus, it worked in tandem with an existing 2D graphics card for 2D DOS and Windows acceleration, only taking over when there were 3D graphics to be rendered.

This happened through a D-Sub pass-through cable running from the output of the 2D card to an input on the Voodoo Graphics card. While some 3Dfx cards handled the switching electronically, others actually had a mechanical switch. On these, you could literally hear when the Voodoo Graphics card kicked into action.

3Dfx never manufactured its own 1st-generation cards. Instead, the designs and chips were sold and licensed to third-party manufacturers, with Diamond and Orchid first out of the gate with the Monster 3D and Righteous 3D in late 1996. These first cards sold for approximately £300, which was a lot but not exorbitant for a PC graphics card at the time.

What's more, these beauties could perform amazing feats with even fairly modest PC configurations. At a time when even Intel's Pentium 133 processors were struggling to deliver consistently good frame rates with the standard software renderer in some demanding games, you could slot a Monster 3D into your Pentium 90 system and see great-looking, silky-smooth visuals.

You could now run Quake at 640 x 480 in glorious 16-bit colour and still hit 30fps

Yet 3Dfx's work went beyond designing the architecture to creating an API that enables game developers to support the card. At the time, there were no 3D engines that supported 3D hardware and no standard APIs for developing 3D games. OpenGL was focused mainly on CAD and workstation graphics, while Intel was unwilling to release its new 3DR rendering library for use on hardware that would run DOS games. Microsoft had yet to develop what became Direct3D.

As a result, 3Dfx developed its own API, GLide. This was based on OpenGL, so it wasn't unfamiliar to experienced 3D developers, but it pared back the calls and instructions to focus on those used in real-time 3D games.

To show off Glide's capabilities, 3Dfx didn't just have its own internal demos, but a range of Atari and Midway arcade games, including the racer, San Francisco Rush, and the beat-'em-up, Mace: The Dark Age. These ably demonstrated what the new hardware could do. All that was needed were some suitably awesome PC games.
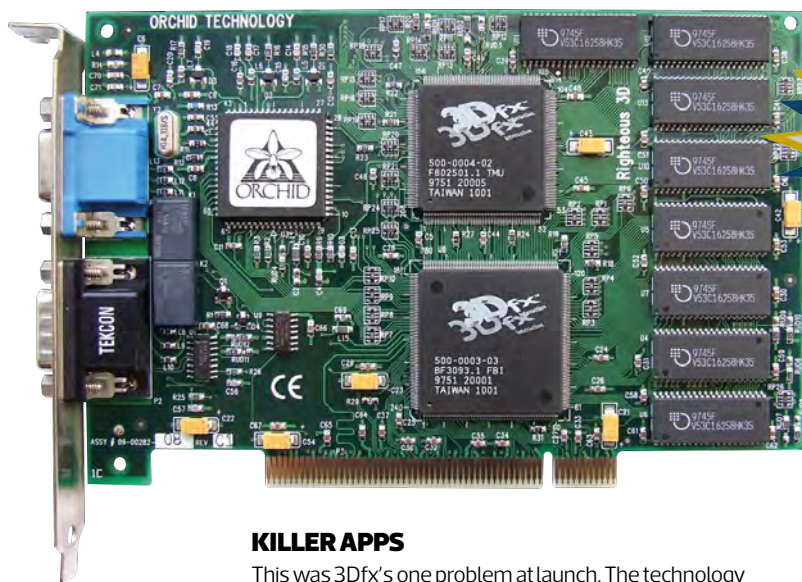
The Orchid Righteous 3D was one of the first Voodoo boards to hit the market, along with Diamond's mighty Monster 3D

KILLER APPS

This was 3Dfx's one problem at launch. The technology itself was impressive, and the cards came with some decent demos, including a slick 3D combat demo, Valley of Ra, which featured amazing reflective surfaces and gouraud shaded characters, and a stunning dolphin sim, Grand Bleu. Orchid and Diamond took them around to show to eager PC journalists, and jaws consistently hit the floor, but there still wasn't a killer app.

At this point, the early 2D/3D graphics cards all tended to support the same games, and we'd got used to seeing the likes of Descent 2, Actua Soccer, Terminal Velocity and MechWarrior 2 with only mildly improved, filtered 3D textures running at frame rates that barely climbed above what you could get with a software renderer. The Voodoo 3D ran these games faster at higher resolutions, but nobody was going to pay £300 for that.

Luckily, 3Dfx soon had two absolute bangers. The first was Tomb Raider. Lara Croft's debut was already one of the most stunning-looking games around on the Sega Saturn, Sony PlayStation and PC, but the pixelated, low-resolution graphics meant that you weren't seeing it at its best.

However, just a few months after launch, the publisher, Eidos, released a patch that allowed you to run Tomb Raider under GLide. The effect was amazing, not only smoothing out the blocky textures and adding transparent water, but allowing

With its high-res models and reflective surfaces, 3Dfx's lead tech demo was a jaw-dropper. Nobody had seen anything like this outside of Sega's Virtua Fighter arcade games

you to play the game at a 640 x 480 resolution at close to 30fps. You saw it and you wanted Voodoo in your life.

An even more impressive transformation awaited us with id Software's Quake. I first played Quake on a Pentium 133 laptop with 16MB of RAM, and the game was only just playable at a 360 x 240 (or half SVGA). And when I say playable, I mean the right side of 20fps.

Then 3Dfx released MiniGL, a cut-down version of OpenGL designed to handle just the functions used in Quake – id responded with a port of the game, GLQuake, which could take advantage of the MiniGL wrapper. The port had its problems, including gloomy brightness levels, but the bilinear filtered textures went from looking slightly rough to looking awesome, and you could now run the game at 640 x 480 in glorious 16-bit colour and still hit 30fps.

Serious PC gamers saw Quake running unaccelerated and then accelerated, then voted with their wallets. Sure, the new Pentium MMX CPUs released in 1997 could run the game at a decent lick, but did it look as good as Voodoo? Not even close.

GLQuake sold 3Dfx cards, and a growing user base boosted game support. True, 3Dfx had rivals. Videologic's PowerVR tech was affordable and efficient, but it also used an unconventional tile-based rendering pipeline and needed a faster CPU to get the best out of it. Rendition's Verite chipsets looked promising, but were too pricey and struggled with their 2D performance.

3Dfx grew to become a kind of de facto standard just as the next wave of 3D games started taking off. From Need for Speed II SE to Myth: The Fallen Lords, Shogo: Armor Division and Unreal, Voodoo Graphics made the best-looking games of the era look even better and run at what seemed incredible speeds. The PC was back on top as the most technologically advanced gaming platform of the era.

3Dfx continued through a glorious period. Its 1997 Voodoo Rush 2D/3D graphics chipset was admittedly a dud, suffering from a lack of memory bandwidth and sync issues with the on-board 2D graphics chip. However, 1998's Voodoo 2 was a worthy successor, arriving just a few months after another iD showcase, Quake II.

This purple period wasn't to last, as GLide fell out of favour and ATi and Nvidia delivered high-performance all-in-one graphics chips, but we owe 3Dfx a huge amount for bringing 3D power to the PC when it needed it most – and helping to show the world the full potential of hardware-accelerated 3D graphics. CPC

POWER VR

Ben Hardwidge catches up with the PowerVR folks from Imagination Technologies (formerly VideoLogic), to discuss early PC 3D accelerators

Back when PCs were still in horrible beige boxes, John Major was nasally shouting over the despatch box and Nvidia was just a glint in Jensen Huang's eye, VideoLogic (now Imagination Technologies, the firm also behind PURE radios) started work on the PowerVR project. It resulted in some of the first PC 3D accelerators and, since then, PowerVR has become a mobile GPU system of choice, found in the iPhone 7 and numerous Android phones.

I headed up to Imagination Technology's HQ in Kings Langley to chat with some of the folks who worked on the original PC PowerVR cards. I'm taken to a meeting room, where a spread of PC relics from the early 1990s to the 2000s is laid out. They include never-released products, including the Kyro 3 and various pre-release boards, as well as some classics.

WHERE DID IT START?

Simon Fenny, PowerVR Research Fellow, picks up the first one – an enormous PCB with a 16-bit ISA interface. 'The whole PowerVR project started in July 1992,' he says, 'and in about early

VideoLogic's Apocalypse 3DX was the first mainstream PowerVR PC 3D accelerator

1993, this first card first came out – it would have been in a 486 PC, so not very good floating point performance. We had a Texas Instruments DSP on there to do all the transform and lighting. This board would later do tile-based deferred rendering, with real-time shadows, and proper 3D volume shadows, but it didn't have texturing, because it was hard enough to fit that all onto one chip.'

Tile-based deferred rendering is the key to PowerVR. 'Tile-based rendering and deferred rendering are two separate things,' explains Kristof Beets, Senior Director, Product Management & Technology Marketing, PowerVR. 'Most of our competitors today have some form of tile-based rendering. Fundamentally, that means you bucket your geometry, so instead of rendering triangle by triangle, you first sort your triangles and then render each tile.

'The key benefit is local processing. The further your data goes, the more power it uses. If you keep it very tight, it's much more efficient. Memory loves big transactions, so blasting a tile and loading the texture data for a tile is really effective.'

'The reason why we're still so good at tiling is because of all the clever algorithms and data structures that go behind it, which Simon and those guys came up with in the 1990s – it's how you sort triangles effectively into those buckets.' Basically, the work done on the early PC 3D accelerators is still useful in smartphones today.

The next part is deferred rendering, a benefit of which is that you can identify objects that are hidden behind other objects before shading them, so you only shade the objects you can see. 'It's like painting by numbers,' says Fenny. 'Imagine you're drawing your triangles, and instead of filling in colours you say, ''This is triangle 1, that's triangle 5 and that's triangle 6.'' You then say, ''Okay, send those off and fill in all the 1s. Oh what's the next one? 3 is the next one – do those'', within each tile.
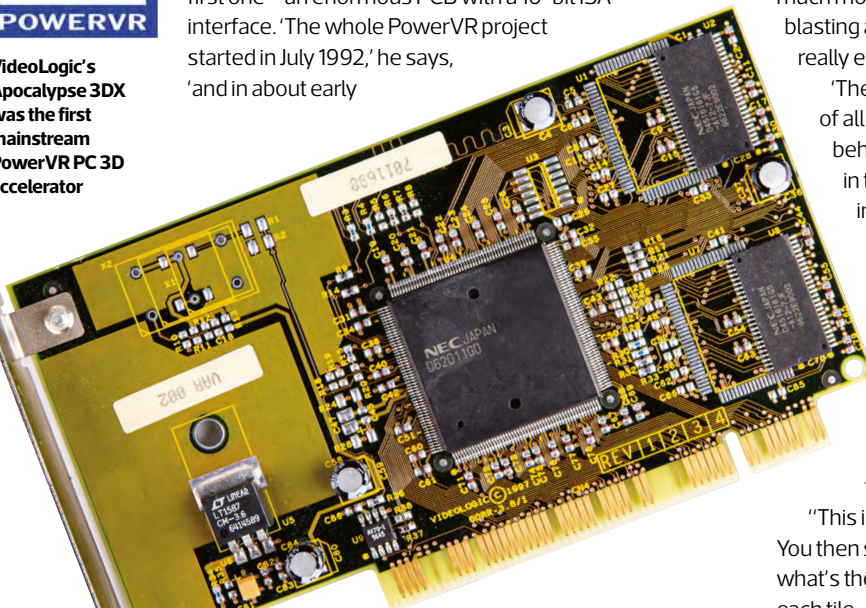
'If something else is behind you, don't bother even shading that. If you just have a normal tile renderer, it might be local, but you still end up drawing a car behind something else, and then a wall over the top. Why would you bother spending all that effort? Some other people will sort things so that it works properly, but it's expensive to sort things.'

THE FIRST PC CARDS

VideoLogic was initially targeting the arcade market with PowerVR, but as PC tech progressed, the team soon turned to looking at the PC. 'Thankfully, the Pentium had come along with the PCI bus,' says Fenny, 'so we were able to do the transform and lighting on the Pentium. We'd send the models over the PCI bus into the chip, which would then render it. These cards would basically mix the signal coming in from the VGA card.'

The first mainstream product based on this tech was the VideoLogic Apocalypse 3DX. This mixing of the signal was a key part of the PowerVR formula at the time. The first 3dfx Voodoo and PowerVR cards were dedicated 3D accelerators, meaning you needed a second '2D' graphics card to output a display to your monitor.

Voodoo cards needed a VGA analogue loopback cable between your 3D card and 2D card. PowerVR cards did it much more cleanly (at least from a hardware perspective), mixing the signal over the PCI bus. 'We realised that, with

the PCI bus, you could not only write things in, but you could burst things right out,' says Fenny. 'Because it was tile-based rendering, if you finished your tile completely you could do that and be really efficient on the bus.'

3dfx wasn't using tile-based rendering, and its Voodoo cards used a Z-buffer to solve the visibility problem. It's a situation that not only meant Voodoo cards had to use loop-back cables, but they also had to allocate some of their

An early ISA Rapier 24 card – the big gold Texas Instruments chip handles transform and lighting

frame buffer memory to the Z-buffer. That's why 1st-gen Voodoo cards are limited to 16-bit colour at 640 x 480, while PowerVR cards could go higher.

'If you turn off Z-buffering, which means a lot of messing around in software, 3dfx could get at 800 x 600 in 16-bit,' says Fenny, 'but we were streaming at 24 bits per pixel.' One area where 3dfx had the upper hand was system requirements. You could get decent performance from a Voodoo card with a Pentium 90, but a PowerVR card needed a beefier CPU to get the most out of it.

THE DREAMCAST

PowerVR was on a roll, and it had caught the eye of Sega while it was developing the Dreamcast. 'I remember being in a couple of meetings, saying it does this and this, and they just looked at us thinking, "That's not possible,"' says Fenny. 'There was a great deal of excitement. We were adding texture compression. We had hardware ordering-dependent translucency, which is still difficult to do now.'

What's hardware ordering-dependent translucency? 'If you ever have to write a game where you have lots of layers of translucent objects, which are in random order on the screen, you have to make sure you do them in back-to-front order,' says Fenny.

Beets informs Fenny that these days developers write a quick-sort in a shader program to deal with it. 'No! Yuck!' he

A VideoLogic Apocalypse 3DX – still in its shrink-wrap at Imagination Technologies

the work done on early PC 3D accelerators is still useful in smartphones todaY
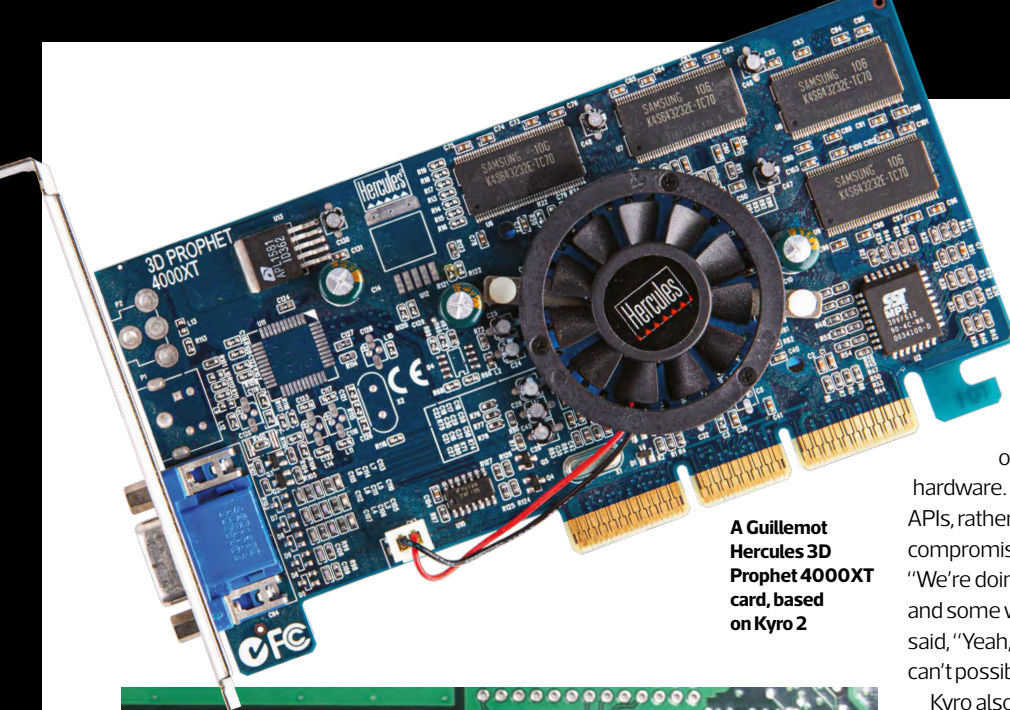
A Guillemot Hercules 3D Prophet 4000XT card, based on Kyro 2

The back of a Kyro 3 card, which never made it to market

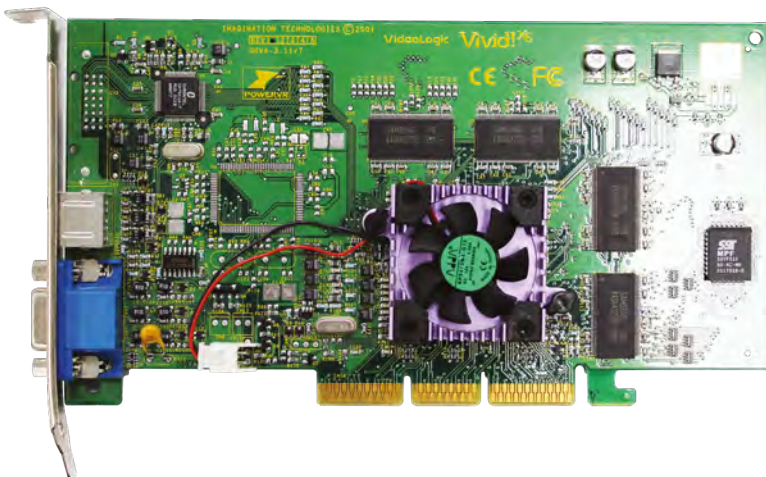VideoLogic's Vivid! Card – based on the 1st–gen Kyro chip

responds. 'It was funny watching some people trying to port the Dreamcast games onto, say, the PlayStation. You'd see the early examples, and all the translucency would be wrong, because the games were designed with the hardware doing it all for you. It did help that we had control over the API, because DirectX was kind of limited to Z-buffer rendering.'

The next PowerVR PC product was the Neon 250, based on some of the tech in the Dreamcast, and an all-in-one 2D/3D AGP card. 'The original product of the Neon product had no fan on it, and we found it really hard to sell in 1999,' muses David Harold, VP Marketing Communications. 'People basically thought, well it has no fan so it must be underpowered compared to Nvidia. So the next version of the board had a fan

on it, which was the cheapest fan we could find in China, because it's essentially cosmetic.'

KYRO

The final push for PowerVR on the desktop PC was the Kyro series. Fenny laments that the Kyro series saw hardware ordering-dependent translucency removed from hardware. The industry was moving towards standardised APIs, rather than proprietary ones, and that meant compromising on some hardware features. 'We'd say, "We're doing translucency sorting" to DirectX developers and some would say, "What? No, that's not possible." Others said, "Yeah, it would be great to use it, but there are cards that can't possibly use it, so we're not going to develop for it."'

Kyro also saw the introduction of PowerVR's 'perfect tiling' technique. 'We figure out exactly which tiles that an object is in,' explains Beets. 'What our competitors do is bounding boxes, but a box covers a lot more area than a triangle.' Next came the Kyro 2, with a die shrink and an increase in clock speed. I was working for PC Pro magazine at the time, and reviewed the Kyro II. It wasn't as quick as Nvidia's top-end GeForce2 chips, but it happily beat the GeForce2 MX's performance for a similar price.

Nvidia wasn't happy, and briefed industry partners against Kyro 2. A leaked PowerPoint presentation showed Nvidia lambasting Kyro 2's driver support, rendering quality, Z-buffer issues and lack of hardware transform and lighting. The presentation's conclusion was damning: 'Buying Kyro 2 is a risk – and when cards and PCs get returned, it damages your finances and your reputation.' Understandably, there's not a great deal of love for Nvidia among the PowerVR folks.

LEAVING THE DESKTOP

Fenny shows me the card that would have been Kyro 3, but it never made it to market – the reasons are kept off the record. I ask why we've never seen a PowerVR desktop product since. 'We were very nervous,' says Harold. 'We looked at the market, and thought, "There are five console makers, and Panasonic is going to be out of this business in five minutes, then there's going to be four and some day there will be three. And in every generation you have to win a slot."

'After Dreamcast, we talked very seriously about doing a console with somebody else, and realised that every single engineering resource we had would have to go on that project. Then, if we lost that slot to whoever in the next generation, we would have no customer.

'It's the same with the PC market. When we started, there were 50+ companies making devices for PC boards, and that figure was shrinking – not yearly; it was practically shrinking weekly. We looked at the market and just thought, If we keep going after PC and console, we're never going to have enough customers to make our business resilient.

'At the time, we said that one day we'd come back to those markets, but ultimately, you're driven by what your customer wants to make.' CPC

NVIDIA GEFORCE

Lights, transform, action! **Ben Hardwidge** recalls the very first 'GPU'

I t's testament to Nvidia's marketing team that one of its buzzwords has now slipped into common parlance. Not only did Nvidia's 1st-gen GeForce 256 introduce us to its now famous 'GeForce' gaming graphics brand, but it also brought the term 'GPU' to the PC with it. An initialism that we now use as shorthand for any PC graphics chip, or even a whole graphics card, started life as an Nvidia marketing slogan.

To give you an idea of how long ago this was, I was introduced to the term 'GPU' by a paper press release the same week I started my first tech journalism job in September 1999. We didn't get press releases via email then – they were physically posted to us, and the editorial assistant sorted them all into a box for the team to peruse.

'In an event that ushers in a new era of interactivity for the PC, Nvidia unveiled today the GeForce 256, the world's first graphics processing unit (GPU)', it said. At the time, I thought it seemed pompous – how could this relative newcomer to the 3D graphics scene have the nerve to think it could change the language of PC graphics? But I now see that it was a piece of marketing genius. Not only did 'GPU' stick for decades to come, but it also meant Nvidia was the only company with a PC 'GPU' at this point.

TRANSFORM AND LIGHTING

Nvidia's first 'GPU' did indeed handle 3D graphics quite differently from its peers at the time, so it's time for a little history lesson. If we want to understand what made the first GeForce GPU so special, we first have to take a look at 3D pipelines of the time.

It was October 1999, and the first 3D accelerators had only been doing the rounds for a few years. Up until the mid-1990s, 3D games such as Doom and later Quake were rendered entirely in software by the CPU, with the latter being one of the first games to require a floating point unit.

If you want to display a 3D model, it has to go through the graphics pipeline, which at this stage was all handled
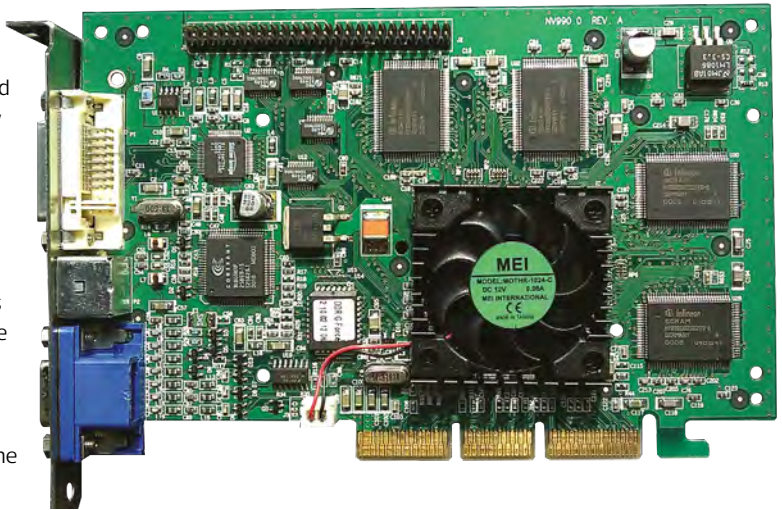

A VisionTek GeForce 256 card with SDR memory

by the CPU. The first stage is the geometry, where the CPU works out the positioning (where polygons and vertices sit in relation to the camera) and lighting (how polygons will look under the lighting in the scene). The former involves mathematical transformations, and is usually referred to as 'transform', with the two processes together called 'transform and lighting' or T&L for short.

Once the geometry is nailed, the next step is to fill in the areas between the vertices, which is called rasterisation, and pixel processing operations, such as depth compare and texture look-up. This is, of course, a massive oversimplification of the 3D graphics pipeline of the time, but it gives you an idea. We started with the CPU handling the whole graphics pipeline from start to finish, which resulted in low-resolution, chunky graphics and poor performance.

We then had the first 3D accelerators, such as the 3dfx Voodoo and VideoLogic PowerVR cards, which handled the last stages of the pipeline (rasterisation and pixel processing), and massively improved the way games looked and performed, while also ushering in the wide use of triangles rather than polygons for 3D rendering. With the
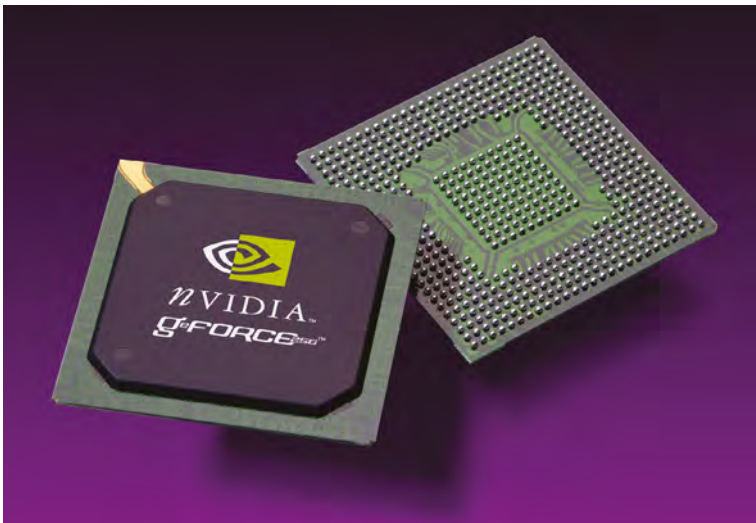
Nvidia's GeForce 256 was the first consumer graphics chip to handle the whole 3D graphics pipeline, including the transform and lighting stages

CPU no longer having to handle all these operations, and dedicated hardware doing the job, you could render 3D games at higher resolutions with more detail and faster frame rates. At this point, the CPU was still doing a fair amount of work though. If you wanted to play 3D games, you still needed a decent CPU.

Nvidia aimed to change this situation with its first 'GPU', which could process the entire 3D graphics pipeline, including the initial geometry stages for transform and lighting, in hardware. The CPU's only job then was to work out what should be rendered and where it goes.

At that point Nvidia had won the performance battle – nothing else could compete

BATTLE OF THE PLANETS
As with any new graphics tech, of course, the industry didn't instantly move towards Nvidia's hardware T&L model. At this point, the only real way to see it in action in DirectX 7 was to run the helicopter test at the start of 3DMark2000, although some games using OpenGL 1.2 also supported it.

The latter included Quake III Arena, but the undemanding nature of this game meant it practically ran just as well with software T&L. DirectX 7 also didn't require hardware-accelerated T&L to run – you could still run DirectX 7 games using software T&L calculated by the CPU, it just wasn't as quick.

The GeForce was still a formidable graphics chip whether you were using hardware T&L or not though. Unlike the 3dfx Voodoo 3, it could render in 32-bit colour as well as 16-bit (as could Nvidia's Riva TNT2 before it), it had 32MB of memory compared to the more usual 16MB, and it also outperformed its competitors in most game tests by a substantial margin.

ATi's response at the time was a brute-force approach, putting two of its Rage 128 Pro chips onto one PCB to make the Rage Fury Maxx, using alternate frame rendering (each graphics chip handled alternate frames in sequence – note how I'm not using the term 'GPU' here!) to speed up performance. I tested it shortly after the release of the GeForce 256 and it could indeed keep up.

THE GPU WINS
The Rage Fury Maxx's limelight was cut shortly afterwards, though, when Nvidia released the DDR version of the GeForce in December 1999, which swapped the SDRAM used on the original GeForce 256 with high-speed DDR memory. At that point, Nvidia had won the performance battle – nothing else could compete.
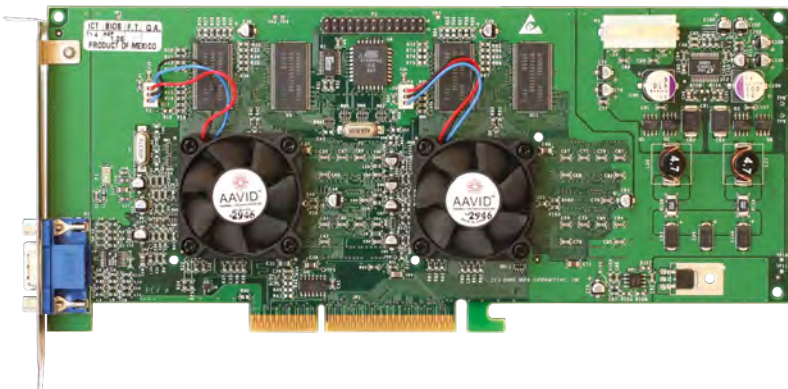
It also took a while for everyone else to catch up, and at this point, various people in the industry were still swearing that the ever-increasing speed of CPUs (we'd just passed the 1GHz barrier) meant that software T&L would be fine – we could just carry on with a partially accelerated 3D pipeline.

When 3dfx was building up to the launch of the Voodoo 5 in 2000, I remember it having an FAQ on the website. Asked whether the Voodoo 5 would have software T&L support, 3dfx said, 'Voodoo4 and Voodoo5 have software T&L support.' It's not deliberately dishonest, but every 3D graphics card could support software T&L at this time – it was done by the CPU – it looked as though the answer was there to sneakily suggest feature parity with the GeForce 256.

In fact, the only other graphics firm to come up with a decent competitor in reasonable time was ATi, which released the first Radeon half a year later, complete with hardware T&L support. Meanwhile, the 3dfx Voodoo and VideoLogic PowerVR lines never managed to get hardware T&L support on the PC desktop, with the Voodoo 5 and Kyro II chips still running T&L in software.

But 3dfx was still taking a brute-force approach – chaining VSA-100 chips together in SLI configuration on its forthcoming Voodoo 5 range. The Voodoo 5 5500 finally came out in the summer of 2000, with two chips, slow SDRAM memory and no T&L hardware. It could keep up with the original GeForce in some tests, but by that time Nvidia had already refined its DirectX 7 hardware further and released the GeForce 2 GTS.

By the end of the year, and following a series of legal battles, 3dfx went bust and its assets were bought up by Nvidia. GeForce, and the concept of the GPU, had won. **CPC**

With no T&L hardware, 3dfx fought back with a brute-force, multi-chip approach on the Voodoo 5 5500. Photo credit: Konstantin Lanzet

SUBSCRIBE TO

CUSTOM PC

custompc.co.uk/subscribe

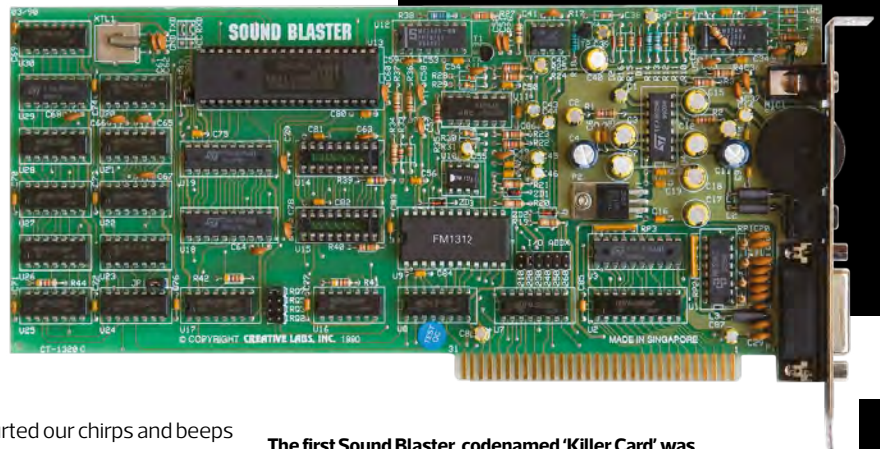# SOUND

THE SOUND BLASTER STORY

Ben Hardwidge talks to Creative Technology founder and CEO, Sim Wong Hoo, about the development of the iconic Sound Blaster brand

Now celebrating its 30th birthday, the Sound Blaster made a massive impact when it was launched back in 1989. It seems bizarre now, but at that time, gaming was still considered to be a frivolous novelty for the PC, which was primarily a business machine. While the Atari ST and Commodore Amiga had half-decent sound capabilities, most PCs came equipped with only a mono PC speaker, which simply blurted our chirps and beeps like an excitable 1970s telephone. PC audio was terrible.

If you wanted proper music in your games then you needed a MIDI card. Rather than playing back a music recording like current games, MIDI music is a bit like a Word document. In a Word document, the fonts are stored somewhere else, and the Word file just stores the formatting, meaning you can store a huge number of words and pages in a very small file size. In the same way, with MIDI, you have the sounds stored on a synthesiser card, and a game's music file just tells it which sounds to play and when.

This started with basic FM synthesisers such as Yamaha's OPL2, which modulated frequencies to simulate instruments, and then later went up to 'wavetables' of sampled instruments to create much more realistic-sounding music.

In the days before we had very powerful CPUs and masses of storage space, this meant complicated musical scores could be performed in games using tiny files, without

The first Sound Blaster, codenamed 'Killer Card' was launched in 1989, combining MIDI synthesis with 23KHz audio playback

needing masses of processing power, or a massive hard drive to store a recording. AdLib was one of the first companies to market a MIDI music expansion card for the PC, making a massive difference to games, but the Sound Blaster went one step further by combining MIDI music with basic sampling capabilities.

The result was an audio system that could give you decent music in games, as well as sampled speech and sound effects. It changed the PC's sound forever and sold by the bucketload. It was the final part of the equation needed to transform the PC into a proper gaming machine. Thirty years after the original Sound Blaster card was launched, we caught up with founder and CEO of Creative Technology, Sim Wong Hoo, to talk about the history of the iconic Sound Blaster brand.

The later Sound Blaster 2 increased the playback sampling rate to 44KHz, still on an 8-bit ISA card

CPC: Let's start right at the beginning. What made you think there was a definite market for a discrete sound card in the 1980s?

Sim Wong Hoo: Let me go a little bit further back. I started playing with microcomputers in 1979, when there was only a handful of them around. They were either dumb or only managed some beeps. At that time, I was designing some computerised seismic data logging equipment, which my former French boss claimed to be the most advanced in the world. When the equipment was brought to operate in oil rigs, nobody believed that it was possible that a Singaporean had designed this equipment in Singapore, which had no high-tech industry at all.

With a strong background in digital and analogue technologies, plus acoustic knowledge, coupled with a deep interest in the science of music, I had a burning desire to bring sound into the computer world. In fact, my first secret microcomputer project in my French boss' company was writing an electronic organ program in machine language that could be played on the computer keyboard, much to the chagrin of my boss. I left his company and started Creative in 1981 with a mission – to bring sound into the computer world.

It took Creative another five years, until 1986, before we developed a PC – the Cubic CT – that had sound and music capabilities. This was five years before the term 'multimedia' for PC was even coined. But we were too early, as there wasn't any third-party content to support it, especially voice-capable software. Creative faced a Herculean task in marketing the Cubic CT, especially in a tiny market such as Singapore.

In 1987, after some soul-searching, we decided that the Cubic CT was too complex an animal for a tiny startup in Singapore to handle. We decided to focus our energies on just the music portion of the Cubic CT, for which we had developed some cool software, such as the Intelligent Organ in 1986, which enabled you to play orchestra-like music with just one finger tapping on the keyboard. This became our Creative Music System music card, which

featured a stereo 12-voice music synthesiser. This product became quite a hit in Asia. It was our first highly lucrative product, and back then, that meant a lot for a small company such as Creative.

In 1988, I felt the time was right for me to go to the USA, which was the world's largest PC market at the time, and my mission was to create a PC sound standard for the whole world. A lot of people felt this was an impossible mission, considering our small size and limited resources at the time. While I was in the USA, I learned that the market for music cards had started to gain traction in the gaming industry.

We quickly approached key game developers to support our music cards. As an unknown company from Asia, it was very challenging initially. However, we soon gained the respect of several key developers, because of our prowess in technology and commitment to supporting these developers. To target this gaming market, we changed the name of our music card to Game Blaster and dropped the price by half.

In the process of talking to these developers, they strongly requested a sound card that could support voice. I told them that we had already done it in 1986, but removed the feature due to lack of support for it. I told them that if they were willing to support it, we could do a joint development. The first company with which we worked closely was Broderbund, with its Carmen Sandiego series of educational games. The project name of this sound card was 'Killer Card'. Broderbund developed its new voice-capable games with a crude prototype version of our 'Killer Card'. This card actually consists of two prototype boards, interlinked together with a whole bunch of wires.

The 'Killer Card' became the Sound Blaster, and it was launched in November 1989 at Comdex in Las Vegas. And with the voice-capable games from Broderbund ready to ship, Sound Blaster was ready for prime time. Michael Jackson passed by and was attracted to the only booth that generated computer audio throughout the entire Comdex show – the Creative booth. I showed him the demo and presented our

My mission was to create a PC sound standard for the whole world

The Sound Blaster Pro added an IDE interface, enabling Creative to sell multimedia packs with a sound card and CD-ROM drive
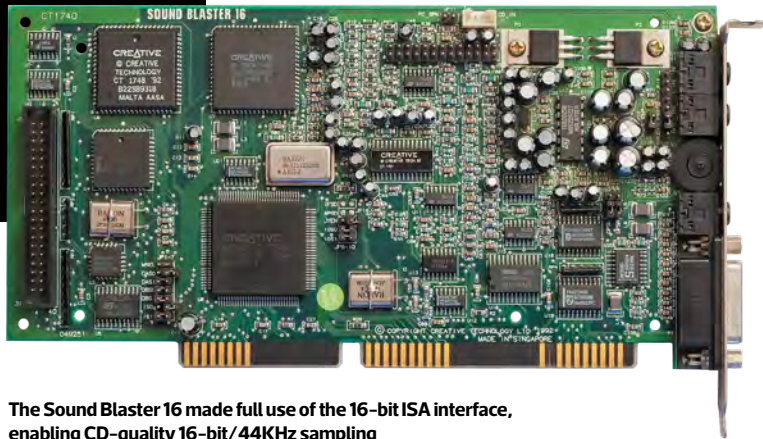
The Sound Blaster 16 made full use of the 16-bit ISA interface, enabling CD-quality 16-bit/44KHz sampling

technology, and he stayed for 30 minutes, even though his minders wanted to usher him away after five minutes. Obviously he was awed.

At Comdex, people lined up in 20-person queues in front of three cashiers in our tiny 300-square-foot booth – we sold one Sound Blaster every four minutes. This was a phenomenal success and Sound Blaster took off like a rocket after that. To date, over 400 million units of Sound Blaster have been sold. In fact, all PCs today still retain the original Sound Blaster compatibility in the OS.

CPC: What were the limitations of these old cards in comparison with later Sound Blasters?
Sim Wong Hoo: The Sound Blaster was an 8-bit sound card with a low sampling rate. The audio quality was coarse and very bad by today's standards. But going from no sound to 'got sound' was a giant step for the PC at that time. Users were thrilled by this new capability, and its highly affordable mass-market price.

> # The audio quality was coarse and bad by today's standards

The OPL2 synthesiser was a two-operator FM synthesis chip and could only generate nine-voice mono music. 'Two-operator' means it uses only two sine waves to modulate each other and generate different kinds of musical instrumental sounds. While it sounded slightly better than our own 12-voice stereo synthesiser, it was still rudimentary in the realm of electronic music instruments.

CPC: The first Sound Blaster used a Yamaha OPL2 FM synthesiser, making it AdLib-compatible. If AdLib hadn't done this first, do you think the first Sound Blaster cards would have had different MIDI synthesis?
Sim Wong Hoo: The first Sound Blaster did have a different music synthesis system – that was already in our Cubic CT PC in 1986. In fact, the first generation of Sound Blaster supported both the Yamaha OPL2 FM synthesiser and our own 12-voice

synthesiser, so by default, it automatically supported a wider range of software from the two standards, giving users the best of both worlds.

CPC: The first Sound Blaster made a killer product by combining PCM audio with FM synthesis, but its sampling rate was limited to 23KHz. Why was the sample rate so low?
Sim Wong Hoo: The sampling rate was low simply due to component cost, and the performance of PCs at the time. A mass-market 8-bit analogue-to-digital converter wasn't available at that time, so analogue-to-digital sampling was performed in software using the digital-to-analogue converter. Because it was software, it was limited by the speed of the PCs at that time. Anyway, a 23KHz sampling rate is good enough for 8-bit, as the benefit of increasing the sampling rate is drowned out by the coarse 8-bit output anyway.

CPC: PC games had very limited audio features at this time – how did you go about getting game developers to implement Sound Blaster support?
Sim Wong Hoo: After the initial success of Sound Blaster, we started to engage the entire gaming industry, and supported developers to put audio into their games. We provided them with a free Sound Blaster Developer Kit, which was the first of its kind in the industry, as well as free consultancy. We even helped game developers to certify their games as 'fully Sound Blaster compatible' at no cost.

CPC: What was the thinking behind adding gameports to the backplates of Sound Blaster cards?
Sim Wong Hoo: It was very simple to do, and we had the space on the backplate to include a gameport. This also saved a precious slot for users who wanted to play games with joysticks.

CPC: It took a while for Creative to make the MIDI output of Sound Blasters MPU-401-compatible. Why was this?
Sim Wong Hoo: MIDI wasn't our focus at the time – it was a small, niche and hard-to-service market. The original Sound Blaster did have a MIDI interface hidden in the gameport. It was put there to give a positive answer to curious people who asked about MIDI but didn't need it. This limited MIDI features didn't cause any loss Sound Blaster sales. We eventually did make our MIDI interface MPU-401-compatible and, as we had expected, it made no difference to our sales. The fact is that almost all our users didn't care much about this compatibility.

CPC: The Sound Blaster Pro came on a 16-bit ISA card, but was still only an 8-bit card really. Why did it need a 16-bit ISA interface?
Sim Wong Hoo: The Sound Blaster Pro was a stereo version of the Sound Blaster, which was a requirement of the Microsoft Multimedia PC standard. It supported additional interrupt and DMAs, which were only found on the 16-bit bus.

CPC: The Sound Blaster Pro also came with an IDE interface to control a CD-ROM drive. What was the thinking behind this?

▶

Sim Wong Hoo: The CD-ROM drive that met the performance requirement specifications of the Multimedia PC initiative was originally a very expensive, Japan-made CD-ROM drive with a complicated and expensive SCSI interface, which cost over $2,000 US. This expensive drive would have immediately derailed the multimedia PC initiative.

So Creative solved this nightmarish scenario by codeveloping a new and inexpensive CD-ROM drive with MKE (Japan). Creative significantly improved the performance of this low-cost drive by developing a proprietary CD-ROM drive interface on the Sound Blaster, as well as new driver software. This innovative driver went against conventional wisdom of needing an Interrupt and DMA for high-speed data transfer. Instead, it used the CPU to access the CD-ROM drive directly and create a huge buffer of data in advance, thereby increasing performance tremendously.

Putting the CD-ROM interface on the Sound Blaster was an obvious advantage in that you also didn't require an additional expansion slot for a CD-ROM drive controller. It also simplified the sales of our Multimedia PC Upgrade Kits, which comprised a sound card, CD-ROM drive and some CD-ROM titles.

CPC: The Sound Blaster Pro 2 introduced OPL3 synthesis – what could this do that you couldn't do on OPL2?
Sim Wong Hoo: OPL2 had two operators and nine voices, while OPL3 had four operators, 18 voices and stereo output. FM synthesis with four operators used four sine waves to synthesise music, which provided a richer timbre and thus created better-sounding musical instruments.

CPC: Several competitors started producing cheaper 'Sound Blaster Pro-compatible' cards in the early 1990s – how did these affect your sales, and was there any licensing involved in claiming compatibility with your cards?
Sim Wong Hoo: These so-called compatible sound cards had negligible effects on our sales, despite selling at lower prices. In fact, they helped to create a larger awareness for sound

on the PC. Many of these cards suffered high returns as users found them not to be *that* Sound Blaster-compatible. After the returns, the users would usually then buy original Sound Blasters.

CPC: Take us through the development of the EMU chips for the later 16-bit Sound Blasters – what were you looking to achieve with this level of advanced synthesis?
Sim Wong Hoo: The EMU was the grandfather of wavetable synthesis, earlier than Yamaha and Roland, pioneering wavetable synthesis way back in the early 1970s. EMU joined the Creative family in 1993, and we started using its wavetable chips in Sound Blasters to provide much better music synthesis and FM synthesis. It was a major breakthrough for PC sound cards at that time.

The subsequent EMU chips – for example, EMU10K1 – besides doing wavetable synthesis, were also fully programmable acoustic digital signal processing engines that powered our game-changing Environmental Audio eXtension (EAX) system. This enabled multiple simultaneous voices to be processable in hardware.

CPC: Even though so many decent MIDI sounds were available, via the AWE 32, AWE 64 and various wavetable cards, OPL2/OPL3 is still considered the 'sound' of the era – it's the default in DOSBox, for example. Why do you think wavetable synthesis didn't quite catch on in the same way as FM synthesis?
Sim Wong Hoo: FM synthesis supported many old games, which is why it's still found to be the default in DOSBox. As PCs got a lot faster, and supported larger memory, I guess it was easier for developers to stream music directly in games. Some of them used their own software audio engines.

CPC: The AWE32 was expandable via standard 30-pin SIMMs, but the AWE64 wasn't. What was the reason for this decision?
Sim Wong Hoo: The AWE64 was targeting a much bigger market and, to be cost-effective, we had to remove the

In the heyday of MIDI gaming audio, the massive AWE32 could be expanded using 30-pin SIMMs
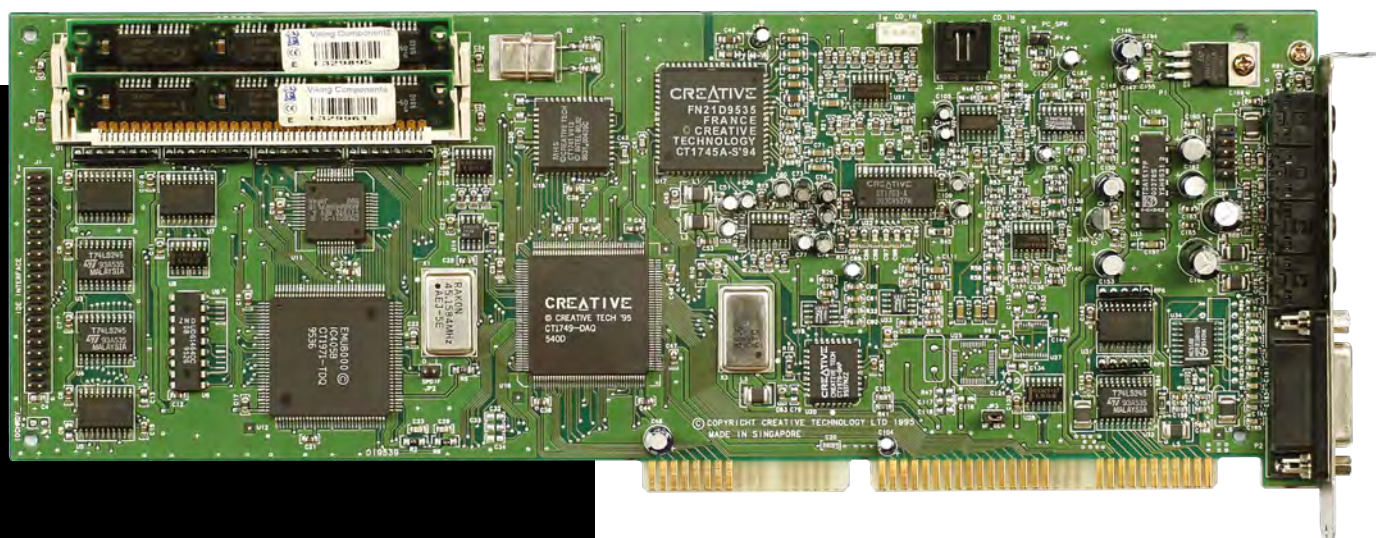
memory upgrade functions. The built-in memory was sufficient for most applications. The AWE64 subsequently became a runaway success.

CPC: Some hobbyists have found ways to clone old ISA Sound Blaster cards, ordering a pre-made PCB and soldering in the components (such as the Snark Barker). Given that Creative hasn't made these cards for 25-odd years, do they have Creative's blessing?

Sim Wong Hoo: We have no issues with individual hobbyists who are nostalgic about our very old Sound Blaster cards.

CPC: We recently did a social media survey on how people use their spare PCI-E slots, and 19 per cent of our respondents used a dedicated sound card. What do people get from a dedicated sound card that they can't get from integrated audio?

Sim Wong Hoo: In the first place, I think motherboard audio is horrible. Many engineers, especially digital engineers, think that PC audio is achieved by simply putting a decent DAC on a motherboard. That couldn't be further from the truth. A good audio design requires a good analogue section.

There are many contributors of noise on any motherboard, so designing a good analogue section on a noisy

In the first place, I think motherboard audio is horrible

motherboard is almost a defeating cause. On a powerful gaming computer, the noise from a powerful CPU is even worse. This problem is magnified by on-board Wi-Fi, Bluetooth and so on. The final nail in the coffin is the multiple GPU cards found on the most powerful computers, which to me makes motherboard audio unredeemable.

Creative has many, many years of experience and expertise in pristine audio design. This expertise spans digital, analogue and acoustic audio domains, all of which are necessary for superior audio performance. Sound Blaster was well established decades before motherboard audio became pervasive. Over the years, as motherboards became more powerful and noisier, our Sound Blaster cards, despite being plugged into the motherboard, were always a few steps ahead in being able to preserve this pristine quality.

If users are willing to spend money on an expensive high-end graphics card, it would make total sense for them to invest in a worthy sound card to complete the entertainment experience. Even our lower-end sound cards provide good audio, and retain a big following to this day.
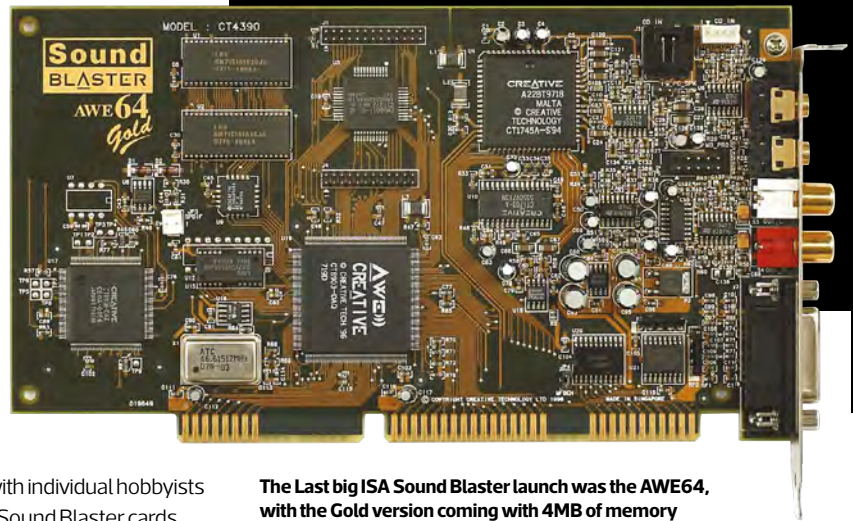
The Last big ISA Sound Blaster launch was the AWE64, with the Gold version coming with 4MB of memory

For example, Sound BlasterX AE-5 offers dedicated high-quality components, and proprietary technologies such as Xamp, which drives individual headphone channels, providing much better headphone audio transience. It can also drive two extreme ends of the headphone spectrum, from 600 Ohm studio monitor headphones to 16 Ohm sensitive in-ear monitors.

Then there's the Sound Blaster audio processing technology, which can be personalised to suit individual entertainment needs such as specific game profiles. It has features such as Creative Multi Speaker Surround 3D technology (CMSS 3D), which is able to provide 3D surround audio on just two front speakers. There's also the Crystalizer, which helps to restore details that are otherwise lost in compressed audio, and DialogPlus, which enhances speech clarity in movies.

In fact, we've also moved beyond the internal sound card. To serve users who don't have a slot for internal sound cards, we have a family of external Sound Blasters, which provide the same high-end audio performance and features.

CPC: How much of Creative's revenue comes from Sound Blaster cards these days, and how does this compare with the past?

Sim Wong Hoo: The sound card revenue of today obviously can't be compared with the heyday of Sound Blaster in the past, when we used to ship millions of sound cards a month. That said, Sound Blaster is still an important contributor to our revenue. Plus, with the recent launch of our Sound Blaster AE series, we are seeing a renewed interest in sound cards.

CPC: What's next for Sound Blaster?

Sim Wong Hoo: Super X-Fi is our latest revolution in headphone audio. It provides holographic-like audio experience in headphones that's as good as the real thing. To the headphone industry, it will be like the transition from black and white TV to colour TV. Super X-Fi will be seen in upcoming Sound Blasters, and this could well reignite the audio revolution for the world. **CPC**

THE PC SPEAKER

K.G. Orphanides delves into the bleeps and bloops
of the PC's original primitive sound system

Before sound cards brought us polyphonic music and CD-quality PCM (pulse-code modulation) audio recordings, PCs could make exactly one noise: a square wave, output through a dynamic speaker driven by the computer's timer chip. Launched in 1981, IBM's first model 5150 Personal Computer had an internal 2.25in (5.7cm) speaker, designed to produce BIOS error codes to help diagnose problems at boot.

It was driven by the Intel 8253 Programmable Interrupt Timer, the same piece of hardware that handled system timing. While Timer Channel 0 was used for system synchronisation, Timer Channel 2 was used to send square waves to the internal speaker, making it beep.

By the 1990s, the 8252 had been superseded by the Intel 8259 Programmable Interrupt Controller (PIC), and these days, you'll find a modern hardware equivalent on your motherboard's southbridge in the form of an Intel Advanced Programmable Interrupt Controller (APIC) variant. All of them retain PC internal speaker functions.

As PC sound card adoption grew through the 1990s, fewer games used the integrated beeper and smaller piezoelectric speakers would become more commonplace. These were quieter, and lacked the versatility and subtlety of a larger dynamic speaker, making some fancier audio effects far less distinct and often too quiet.

Many modern PCs no longer come with any kind of speaker. But motherboards still have the header connector, so you can still install one and listen to audio designed for an internal beeper as it was meant to be heard.

QUEST FOR POLYPHONY

Whichever way your PC beeper sound is implemented, it's monophonic, which means it can only produce one tone at a time. But, as with other very limited early computer audio standards, that wasn't going to prevent

The Intel 8253 chip drove the original PC speaker. Credit: Wikimedia Commons
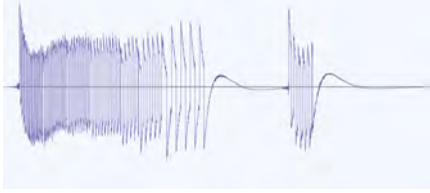
An arpeggiated pseudo-polyphonic 'chord' from The Secret of Monkey Island

composers from doing remarkable things with it.

Beyond simple system beeps, the easiest music to persuade a PC speaker to reproduce is single-tone melodies. A series of instructions is sent to the timer via the CPU, using the programming language of your choice, telling to it produce a series of tones at a specified frequency.

Sound effects in games also started out as simple beeps, but programmers soon started getting clever, rapidly changing the tones being sent to the speaker to produce complex audio effects. Apogee Games mastered the art of creating convincing – or at least distinctive – PC speaker effects in titles including Commander Keen and Hocus Pocus.

You technically can't play polyphonic music on hardware that can only produce one voice at a time but, as it transpires, there are ways around this problem. Probably the most widely used approach to this is arpeggiation, where a pseudo-polyphonic effect is achieved by rapidly switching from one tone to another – anywhere up to 120 times a second – to give the impression of chords to the listener.

A number of games, including the 1990 PC version of The Bitmap Brothers' Xenon 2 Megablast in 1989, the PC port of Sega's Golden Axe in the same year and Magnetic Fields' Lotus III in 1992, create two or three virtual audio channels and alternate which of them is directed to the timer chip, allowing basslines to be rapidly switched into the music. The results often sound harsh and busy, but produce a rather effective impression of polyphony.

A combination of these techniques was used to even better effect in LucasArts' PC speaker music, such as the remarkable beeper rendition of the main theme from The Secret of Monkey Island (1990), where the sophisticated use of fast trills and an alternating percussive channel created the impression of steel drum chords backing the main melody.

Other techniques made more direct changes to the way the PC speaker's sound output worked. Windmill Games' 1983 booter game Digger and its iconic use of Hot Butter's Popcorn as its in-game theme is thought to be the earliest title to use pulse width modulation (PWM) as a method of producing more sophisticated sound, with a variable volume and harmonies.

Also used in numerous ZX Spectrum games, PWM uses careful timing of the signals sent to the PC speaker to modulate its usually binary voltage levels, forcing the speaker into a range of partially on positions to produce sine waves. This can effectively turn the speaker into a 1-bit DAC (digital-to-analogue converter).

Also heard in titles including Hard Drivin' and Fantasy World Dizzy, this approach can be used to play a pre-generated soundtrack, rather than using the timer chip to directly generate square wave tones. However, even at 1-bit, this sound reproduction was often CPU-intensive and the resulting audio's low quality grates on many listeners.

Later, Access Software's RealSound technology used a near-inaudible carrier wave and fine-grained control of the PC speaker's displacement amplitude to produce 6-bit digitised audio, giving us surprisingly high-quality speech and music in games including Mean Streets, World Class Leaderboard Golf and Legend Entertainment's Spellcasting series.

By 1992, even Microsoft was in on the game, releasing a driver for Windows 3.1 that allowed any PCM WAV file to be output via the internal speaker. As sound cards, CD-ROM games, and then integrated motherboard audio became ubiquitous, the need to write dedicated timer chip music or kludge samples through the internal beeper evaporated, and PC speaker audio vanished from audio selection screens.

BACK TO THE PRESENT

Today, PC speaker music isn't as dead as you might expect. Although less iconic than the C64 or NES audio systems, you can hear its influence in the modern chiptunes music scene.

In February 2019, Russian composer Shiru released System Beeps, an entire album written for the PC speaker and using some of the most sophisticated arrangement, arpeggiation and hearing perception tricks we've heard to create an illusion of polyphony. There is, of course, a DOS version of the album, but if you don't happen to have any classic PC hardware (or a copy of DOSBox), it's also available to buy in conventional digital formats.

Shiru used modern Digital Audio Workstation software to compose System Beeps and has made relevant plug-ins, projects and source code available for anyone else who wants to play with them.

Shiru isn't alone in working on music creation tools for your internal beeper. BaWaMI, created by Robbi-985, is a Windows MIDI synthesiser that will output via PC speaker. If you're so inclined, you can still hear and make new music for the PC's oldest audio device. **CPC**

A classic PC speaker square wave from the theme to Space Quest III: The Pirates of Pestulon

ROLAND MT-32

K.G. Orphanides looks back at Roland's external MIDI synth that revolutionised early PC gaming music

Released in 1987 as a musician's tool, the MT-32 would revolutionise PC gaming audio

A glossy black box with a green LCD invites you to 'Insert Buckazoid' on its screen. A stirring 1980s sci-fi theme blasts glossy-textured synth tones through the speakers connected to it, as you're brought up to speed on the continuing exploits of space janitor Roger Wilco. In 1989, Space Quest III leaned into the highest-quality music available on home computer platforms, an external MIDI audio device that was as prohibitively expensive as it was revolutionary.

When it was released in 1987, the original Roland MT-32 MIDI synthesiser cost £450 in the UK – equivalent to over £1,200 in today's money, and it didn't even come with the MIDI interface card you'd need to connect it to your PC.

Roland primarily marketed its MIDI expander module at amateur electronic musicians: a multi-timbral synth-in-a-box that could be controlled by any MIDI keyboard. It proved popular by being significantly cheaper than most rivals, and by supporting 32-note polyphony across up to eight simultaneous voices.

But the MT-32 would become best known as the pinnacle of IBM PC-compatible gaming audio from the late 1980s to the mid-1990s, and it helped to popularise the fully orchestrated game soundtracks we take for granted today.

WHAT'S IN THE BOX?

In 1987, digital synthesis was still a relatively new technology, developed in the 1970s and popularised in 1983 by Yamaha's DX7 synthesiser. It would become the archetypal sound of the 1980s, with a very different feel to analogue synthesisers' use of control voltages to determine pitch, gate and trigger signals.

The MT-32 used Roland's new Linear Arithmetic (LA) synthesis (see **custompc.co.uk/LASynth**) technique, first seen a few months earlier in Roland's 61-key D-50 keyboard synthesiser. LA synthesis relies on Partials: fundamental sounds to which it then adds effects in order to produce voices.

These Partials are either stored as pulse code modulation (PCM) sound samples (as used by audio CDs, WAV files and so on) or fully simulated combinations of oscillators, creating the tone. Filters then determine the brightness of the sound by fixing its cutoff frequency, and an amplifier then determines its loudness. The LA chip's pitch and amplitude envelopes act on the PCM sounds, determining the note produced and
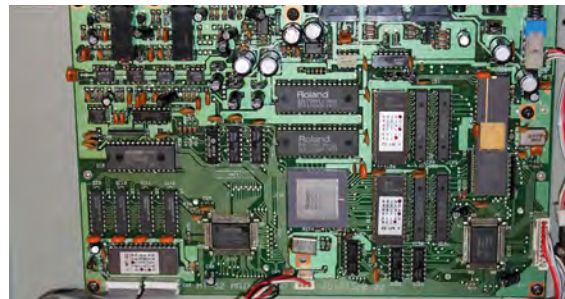
The innards of an early MT-32

its attack, decay, sustain and release. This technique enabled the synth to produce a realistic (for the time) reproduction of genuine instruments.

Alongside the LA chip, you'll find a dedicated gate array, a reverb chip, a Burr-Brown PCM54 DAC, a clutch of op-amps, and EEPROMs that hold the MT-32's firmware and PCM sample banks. You can even send custom patches to the MT-32 – specific configurations of effects for the LA synthesis chip to render on a voice from the PCM bank, so you can effectively make new instruments.

Sierra carried the flag for the MT-32, recruiting Supertramp drummer Bob Siebenberg

GETTING INTO PC GAMING

The first IBM PC-compatible game with an MT-32 soundtrack was Sierra's King's Quest IV: The Perils of Rosella. Scored for the MT-32 by film and TV composer William Goldstein, the game also supported other audio hardware on release, notably the Yamaha OPL2-based AdLib.

Sierra would carry the flag for the MT-32, recruiting Supertramp drummer Bob Siebenberg to create the soundtrack for Space Quest III and even selling the MT-32 and required MPU-401 ISA MIDI interface card for $550 US (equivalent to around £950 today), with MIDI composition software and two Sierra games of your choice included.

The MT-32's original US retail price was $695 US (around £1,200 today). It wasn't cheap, particularly compared with the AdLib and CMS Game Blaster cards Sierra also sold, but it was the best way to get what the company's 1989 catalogue describes as 'a symphony orchestra playing in your living room'.

Other companies took up the challenge, some more enthusiastically than others. Origin Systems supported the MT-32 with some excellent soundtracks from 1990's Ultima VI and Bad Blood, through to Pacific Strike in 1994. LucasArts/ Lucasfilm Games put most of its MT-32 support into its Star Wars titles, such as X-Wing, although some adventure games, including Sam & Max and the Monkey Island titles, received MT-32 MIDI soundtracks. Legend Entertainment, New World Computing and Microprose were also enthusiastic adopters.

UK game development support for the MT-32 included the Bitmap Brothers' Gods, Adventuresoft/Horrorsoft's Elvira and Simon the Sorcerer games, Team 17's Alien Breed, Gremlin Graphics' Litil Divil and Plan 9 From Outer Space, as well as Ocean Software's Elf.

Sierra aggressively promoted and supported the MT-32 until the General MIDI standard was published in 1991, which standardised the voice types and program numbers, ensuring that the right instrument sounds were playing the right parts on all compatible devices, although the quality of the voices still depended on your synth.

The music for Laura Bow II: The Dagger of Amon Ra (1992) was composed on the MT-32, but released with full support

for new General Midi audio devices such as the Roland SCC-1. Other studios supported the MT-32 as late as 1997, with the cover disk demo of Bethesda's The Elder Scrolls: Daggerfall (**custompc.co.uk/Daggerfall**) being among the last.

VERSIONS AND RELATIONS

The MT-32 spawned a host of versions and successors, and became a de facto MIDI standard for other sound card producers before General MIDI was established.

This first 'old' version of the MT-32 is easy to spot, based on its port configuration – it had just a stereo pair of 1/4in TRS outputs. If you connect it to a MIDI interface card on any PC faster than a typical 286, it can produce buffer overflow errors due to an insufficient delay between SysEx messages sent to the device. This could be resolved using the turbo button on 386 and 486 PCs or slowdown utilities on later PCs. This doesn't affect modern PCs using good-quality USB-to-MIDI connectors though – delaySysEx switches are also implemented in a number of popular emulators.

The second 'new' version of the MT-32 introduced a functionally undetectable control CPU switch, along with an additional rear TRS stereo headphone port and reduced noise levels. It also added a ROM playback demo mode and introduced some changes to the gate array and ROM chips. It fixed the buffer overflow error affecting faster computers, but it also rectified some firmware bugs on which some game composers had relied, breaking some soundtracks.

The MT-32's appeal to computer music composers didn't go unnoticed by Roland, and the company followed it with the screenless Computer Music (CM) range of MIDI devices, based on the LA chip. This included, in 1990, Roland's first internal ISA sound card, the LAPC-I, which integrated an MPU-401 interface and MT-32-compatible CM-32L synth.

By 1991, General MIDI was standardised and Roland launched its Sound Canvas range with the SC-55, which used Roland's own GS (General Standard) extension to provide even more voices. A year later an internal version, the Roland SCC-1, was released. Both provided reasonable MT-32 backwards compatibility, but lacked support for custom MT-32 instrument patches.

These MIDI devices, and many to follow, would be popular with musicians for years, but MIDI music in games was on the wane. Full CD audio was clumsy at first, but as disk capacity and audio compression improved, it would be digitally recorded audio that led game music into the new millennium. **CPG**

STORAGE

IMATION

3M

SS, SD,
Single Side
Single Density

FLOPPY DISKS

Ben Hardwidge takes you through the workings of various types of floppy disk, which were once the PC's main storage medium

The classic 3.5in HD floppy disk has become a bit of an icon now, and in more ways than one. Plenty of people will tell you that their kids think of a floppy disk as the Save icon in Word. The 3.5in floppy is also what most people imagine when you say 'floppy disk' – a plastic case with a spring-loaded metal protector and a 1.44MB storage capacity. The history of floppy disks goes back well beyond these neat little storage packs though.

My first experience of multiple types of floppy disk came when my dad bought me a game (Targhan, in case you're interested) for our PC XT clone in 1988. I opened the box, and inside it were two 5.25in disks. We had to send it back to get the version with 3.5in disks, which took ages because, at the time, hardly anybody used 3.5in disks for PCs. All the major models, from the IBM PC to the Amstrad PC1512 and PC1640, had one or two 5.25in floppy drives instead.

Floppy disks were the main form of storage for the first decade of the PC's history, in many cases the only form of storage. But the history of the floppy disk goes back even further than the first PCs. For the purpose of this feature, I got hold of one of the very first types of floppy disk, an 8in single-sided single-density disk. It's huge. You can put a 50p piece in the central spindle hole with space to spare. It holds a formatted capacity of just 248KB.

The floppy disk was one of the 'first solutions to the problem of transferring data from one place to another. We'd used punch cards, punched tape and magnetic tape, which worked, but

A single-sided, single-density 8in floppy disk, with a 50p coin for scale. It has a total formatted capacity of 248KB

were laughably awful in terms of reliability, convenience, space and the length of time taken to load and save data. In 1971, IBM's first read-only 'Type 1 diskette' was an attempt to solve these problems in a neat 8in package, capable of storing 81KB. In 1973 it became commercially available with read/write abilities, and a larger capacity of 248KB.

MAGNETS, HOW DO THEY WORK?

Floppy disks work on the basic principle of magnetic binary storage. As you probably know, all computer data can be broken down to simple on-off switches called bits at its most basic level – if the switch is off, it's a zero, if it's on, it's a 1. There are eight bits in a byte, 1,024 bytes in a kilobyte, 1,024 kilobytes in a megabyte and so on.

Inside the package of a floppy disk is a circular piece of magnetically coated material with a hole in the middle, and a piece of protective fabric on either side of this material to protect it. In the case of 8in and 5.25in disks, the hole is left blank for the drive's spindle to go through it. In the case of 3.5in disks, there's a metal plate in the middle with holes in it, onto which the floppy drive can lock.

The drive then spins the disk and a stepper motor brings the magnetic read/write heads into contact with the disk. With 8in and 5.25in disks, where the disk is exposed in a hole at the front, the heads make contact with the disk once you insert the disk and flip down the physical lever at the front of the drive to lock it in place. With 3.5in disks, the heads make contact with the desk once the disk has been fully inserted in the drive, meaning the protective metal plate at the top has been fully moved to expose the disk, and it's all locked in place.

Once the heads make contact with the disk, and the disk is spinning, the drive can then read or write data – a magnetic transition denotes an on (1) switch, while no magnetic transition means an off (0) switch. All the ones and zeroes are encoded/decoded in a bitstream, and in the case of 5.25in and 3.5in floppy disks, this is generally MFM (modified frequency modulation), although there were other encoding methods in the early days of floppy disks.

▶

Going into the complete workings of MFM would take a feature in itself, but the basic gist is that it introduces a clock to separate the bits in the bitstream. After all, a computer would have a tough time reading a long line of zeroes in a row, with no magnetic transitions between them to tell it whether this was one 'off' bit or several of them. The idea behind using FM and MFM data encoding was to enable a non-return-to-zero (NRZ) system, so there was never a state where there was neither an on nor off signal.

An FM bitstream can encode a 0 as 10 and a 1 as 11, for example. MFM is more complicated than just using 10s and 11s, but the principle of using a clock to separate the bits is basically the same. Incidentally, MFM was also the standard used in early hard drives, including the Amstrad PCs in the late 1980s, before IDE was introduced.

MAKING TRACKS

Towards the latter days of the floppy disk's reign, you could buy disks pre-formatted for your type of computer, but you originally bought them unformatted. In this unformatted state, there's nothing on the disk. It's a blank, circular piece of magnetically coated material. You would then have to tell your computer to format it for your system. In my case, that meant typing 'format a: /w' at the DOS prompt.

The '/w' means 'wait' – like many people at the time, I couldn't afford a hard drive, so I had to boot DOS from a floppy disk called a system disk each time I started my PC – this system disk also contained all the DOS commands, so you would have to type the format command with the system disk in the drive, then swap over to the unformatted disk when prompted – you really didn't want to accidentally format your system disk!

The formatting process would then prepare your disk for reading and writing. Unlike the spiral of data used on most CDs and DVDs, floppy disks organise data in 'tracks' – concentric circles that are separated by small areas containing no data. These tracks are then, in turn, separated

into sectors containing a certain number of bytes, with unused bytes on either side of the sector and a header to mark the start of the sector.

This header also contains a cyclic redundancy check (CRC), which was also placed at the end of the data used in each sector, for error checking. There are many blank spaces, and bytes to denote start and end points for tracks and sectors, which is part of the reason why the formatted capacity is always lower than theoretical maximum capacity of a disk.

SIDES AND DENSITIES

In the case of the classic 3.5in floppy disk, you had 512 bytes per sector on a PC. Depending on your disk (and your hardware), you could also use both sides of the disk, doubling the storage capacity. The other way to increase the capacity was with the 'density', the number of tracks and sectors per side.

As an example, a 'double-density' (DD) 3.5in floppy disk has 80 tracks per side, each containing 9 sectors with 512 bytes each. Each track therefore has 4,608 bytes – multiply that figure by 80 and you get 368,640 bytes, or 360KB per side. So, a single-sided, double-density (SS/DD) disk has 360KB – add another side and you get a double-sided double-density (DS/DD) 720KB disk. The next step, once you've got faster controllers and better physical media, is to double the number of bits (and hence sectors) that fit on one track, taking the number of sectors per track from nine to 18 and doubling the capacity from 720KB to the classic 1.4MB high-density (HD) floppy disk.

I'm referring to PC standards here, of course, but other computers, such as the Amiga and Mac, had more efficient ways of formatting disks that resulted in higher capacities. With a continuous motor speed, every sector held the same amount of data, regardless of whether it was on the inner or outer part of the disk area. As you move from the centre of the disk outwards, however, the sectors become physically bigger, which means space is wasted on the outside area of the disk, where there should be more room for data storage.

Apple got around this issue by varying the speed of the motor when the head was on the outside of the disk vs the inside of the disk, enabling it to add more storage capacity on the outside of the disk and get 800KB from a double-sided double-density disk, rather than the 720KB on a PC.
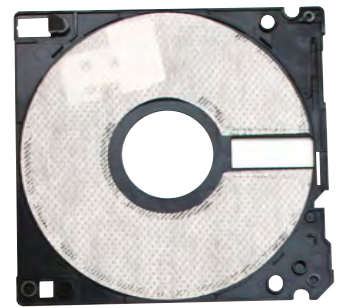
Inside a 3.5in disk's plastic shell is a small floppy disk with a piece of protective fabric on either side

From left to right, 8in SS / SD, 5.25in DS / DD, 3.25in HD

You also got wildly different amounts of formatted storage space from the same physical size of disk on different systems in the early years of the floppy, as there were so many different software standards, all with different sector sizes and formatting systems.

It all resulted in a bit of a confusing mess of different standards, across all sizes of floppy disk. The 8in floppy started off being single-sided, single-density (SS/SD) in 1973, but ended up with a 1.2MB (unformatted) capacity in 1977 thanks to doubling the density and the number of sides used. Likewise, 5.25in floppy disks also had an 80-track high density flavour that gave you up to 1.2MB of data on one disk.

> **A double-density 3.5in floppy disk has 80 tracks per side, each containing nine sectors**

THE COST OF STORAGE

In these times when solid state drives are racking up huge speeds and capacities, it's difficult to imagine a hard drive being a luxury item, but hard drives were extremely expensive for a long time – just 10MB could cost you thousands of pounds.

It's for this reason that many PCs came with two floppy drives in the early days – you'd generally use the A drive for your OS and programs, and the B drive for data. There's still a bit of this residual DNA in PCs today – we may not have A and B drives any more, but our primary storage devices are still C drives, with the assumption that Windows reserves A and B for floppy drives.

The 5.25in floppy disk was once the standard for PC storage
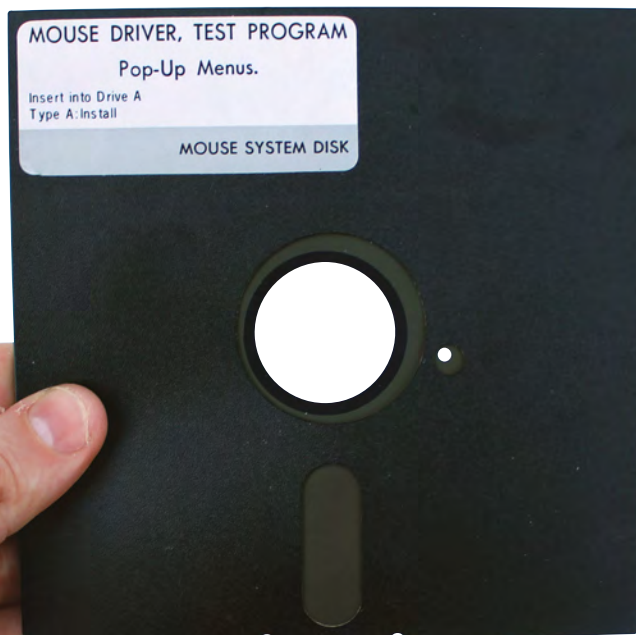
KNOW YOUR FLOPPIES

DATE	SIZE	SIDES / DENSITY	FORMATTED CAPACITY
1973	8in	SS / SD	248KB
1976	8in	DS / SD	563KB
1977	8in	DS / DD	985KB
1978	5.25in	DS / DD	360KB
1982	5.25in	DS / HD	1.2MB
1983	3.5in	SS / DD	360KB
1984	3.5in	DS / DD	720KB
1986	3.5in	DS / HD	1.4MB

In my case, I only had one floppy drive in my first PC, but at least it was a 3.5in double-sided double-density (DS/DD) one, which meant the disks had a reasonable amount of storage space compared to 360KB 5.25in floppy disks (the other main PC standard at the time). The big adventure games would arrive in large boxes containing multiple floppy disks, and with no hard drive, you would have to swap over disks regularly. But floppy disks were cheap. I could buy a couple of with my pocket money and you got disks of demos free with PC magazines.

DECLINE

It took a long time for the floppy disk to completely die out. Even when DVD writers and USB thumb drives were mainstream, floppy drives could still be useful for low-level computing tasks, such as flashing your motherboard BIOS or installing RAID drivers before a Windows installation.

There were also attempts to keep the format alive with the 'SuperDisk' drive in the late 1990s, with capacities of 120MB and later 240MB, along with backwards compatibility with older 3.5in drives. But eventually, capacious and speedy flash drives, as well as cloud storage, killed them off. Floppy disks were revolutionary compared with the storage methods that preceded them, but they were also notoriously unreliable, slow and noisy.

You risked losing all your data if you got them near a magnet, or accidentally creased a 5.25in disk's packaging. The last motherboard I remember coming with an MFM floppy drive controller was ASRock's Z77 Extreme6 in 2012, although you can still buy USB 3.5in floppy drives today, as well as adaptors to control old 5.25in floppy drives via a USB port. I still have a box of my old DS/DD floppy disks, a reminder of where so much of my pocket money went when that was my only storage option. **CPC**

SOFTWARE

WINDOWS 1.0

35 years ago Microsoft finally launched the first version of Windows. **Stuart Andrews** looks back to where Windows started, and tries using Windows 1 again for himself

I t's now more than 35 years since Windows launched in November 1985, 18 months behind schedule and almost three years after Apple's Lisa had introduced the first commercial GUI. It wasn't exactly a hit; it flopped commercially, while reviewers criticised its performance and wondered whether some of its most powerful features were really that useful. Yet less than five years later Windows dominated the operating system market, running on over 70 per cent of all personal computers sold. You can see Windows 1 as the ugly duckling that was to transform into the, well, still gruesome but enormously successful swan.

It wouldn't require a hard drive and it would run with just 192KB of RAM

MAKING WINDOWS

Windows began its journey in the autumn of 1982. Microsoft's CEO Bill Gates was already aware of research into mouse-driven, graphical user interfaces at the legendary Xerox PARC, and of Apple's continuing work on the same principles. However, the story goes that Gates attended the autumn 1982 Comdex trade show in Las Vegas, where he saw VisiCorp demonstrate Visi On: a GUI for the IBM PC. Gates is said to have watched the demo several times, back-to-back, before suggesting that other Microsoft personnel needed to come out to Comdex and take a look. If GUIs were the future, Microsoft wanted a piece of the action.

At this point Microsoft wasn't the huge tech monolith we know today. It was still a small company that had grown successful on the back of Microsoft BASIC and MS-DOS. Gates saw an appetite for a new and easier way to work with the personal computer, and that rival systems were

either too expensive – an Apple Lisa cost around $10,000 US, while you could buy a PC for under $3,000 – or too demanding in their system requirements. If it wasn't bad enough that Visi On needed a staggering 512KB of RAM and a hard disk, its applications needed to be coded in a specific version of C using Unix tools. This left space for an alternative.

Gates hired Scott McGregor, one of the key developers at Xerox PARC, and set a team to work on a project codenamed Interface Manager. Crucially, it wasn't seen as a complete OS, but as a graphical environment that ran on top of MS-DOS. In November 1983, Gates announced Windows and set its release date for April 1984.

The hype said Windows would bring a new way to use PCs. It wouldn't require a hard drive – just two floppy disk drives – and it would run with just 192KB of RAM. By December 1983, an early version was previewed for an article in Byte magazine, with its writer, Phil Lemmon, arguing that 'Microsoft Windows seems to offer remarkable openness, reconfigurability and transportability, as well as modest requirements and pricing'. The result,

Even in the first release, there were options to personalise Windows, although good luck finding a colour scheme that didn't look horrific
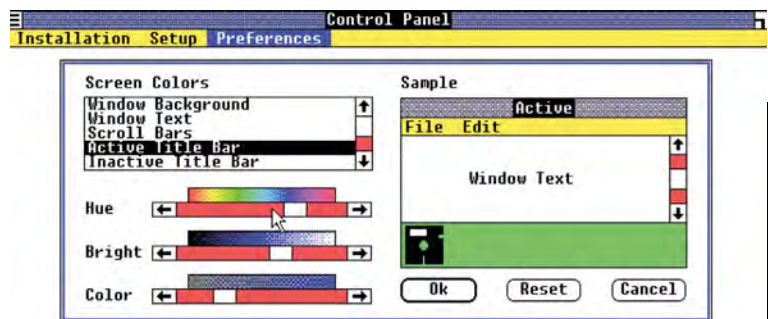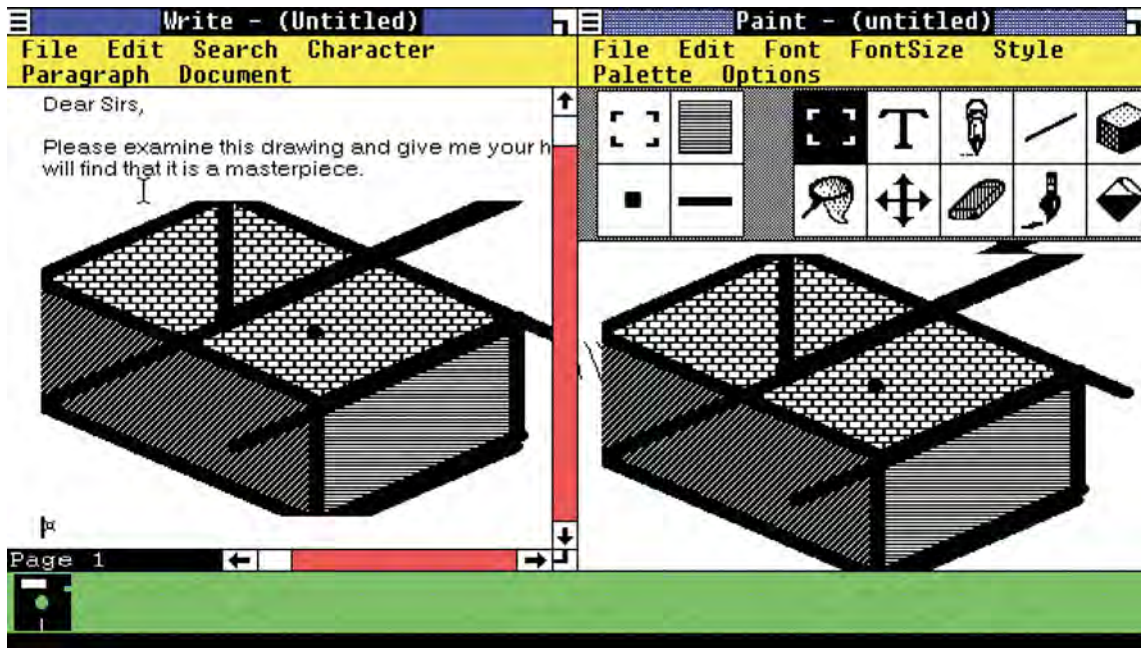
Lemmon thought, could bring computing to a new, non-technical audience.

Why, then, did it take another two years to get finished? For a start, there were some major technical challenges. When development started, standard CGA screen resolutions were limited to 640 x 200 in monochrome, and it was only with the development of EGA graphics boards in late 1984 that you had enough pixels to make Windows effective. The slow speeds and limited capacity of floppy disks had an impact, while the Intel 8088 CPUs used in most PCs weren't exactly bursting with firepower.

Perhaps worst of all, there was a challenge in building industry support. As Gates said in 1983, 'the primary focus of the company and the speeches I gave, the promotion I did, was to get people to believe in the graphics interface whether it was Macintosh or Windows, and that was a tough thing because people like WordPerfect and Lotus refused to put the resources into doing applications'.

Some believe that other factors were in play. By 1984 Microsoft was working with Apple on Macintosh software, and had signed licensing agreements for specific UI elements, but not others, including overlapping windows and the Recycle Bin. It's possible that Microsoft reworked Windows to avoid including these elements and triggering future litigation. If so, Microsoft wouldn't admit it. A November 1983 article in the US computing mag, Infoworld, suggested that Microsoft's Steve Ballmer saw tiled windows as delivering a neater desktop.

A DEVELOPMENT DISASTER

Whatever the case, the development of Windows was definitely troubled. Tandy Trower came in as the product manager in autumn 1984, at a point where Windows was seen externally as vapourware and internally as an embarrassment. Trower even saw being put in charge of the project as a step towards getting fired. By this point Scott McGregor had resigned, and while the core components were in place, elements of the design and the look weren't working. More seriously, there weren't any applications.

'Even at Microsoft, getting developers to write Windows software was a challenge,' said Trower in a 2010 interview. 'I couldn't even get my former team to build a version of BASIC.' However, there was a prototype of a simple bitmap drawing program, while Trower persuaded Gates and Ballmer that Windows needed a set of simple applets, including a word processor, calendar and business card database.

What's more, Trower made it a requirement that Windows could run existing DOS applications. This in itself proved awkward – many DOS apps exploited tricks or workarounds that caused problems for Windows memory management – but it was a major boost to Windows in the future.

You could run three or four applications at once, provided you could tolerate painful slowdowns and a lack of screen real estate
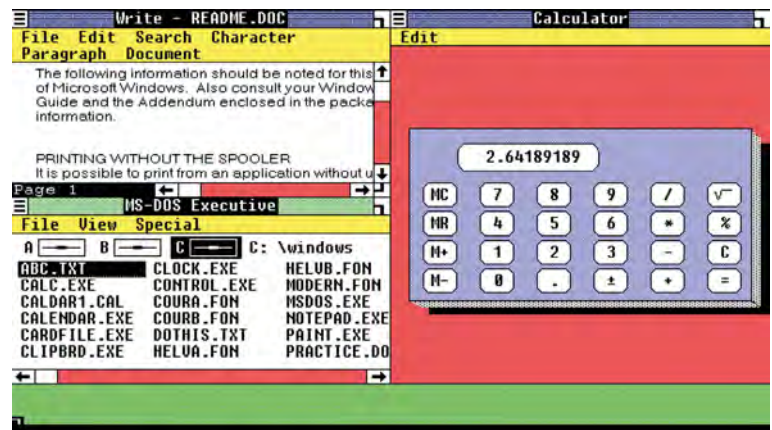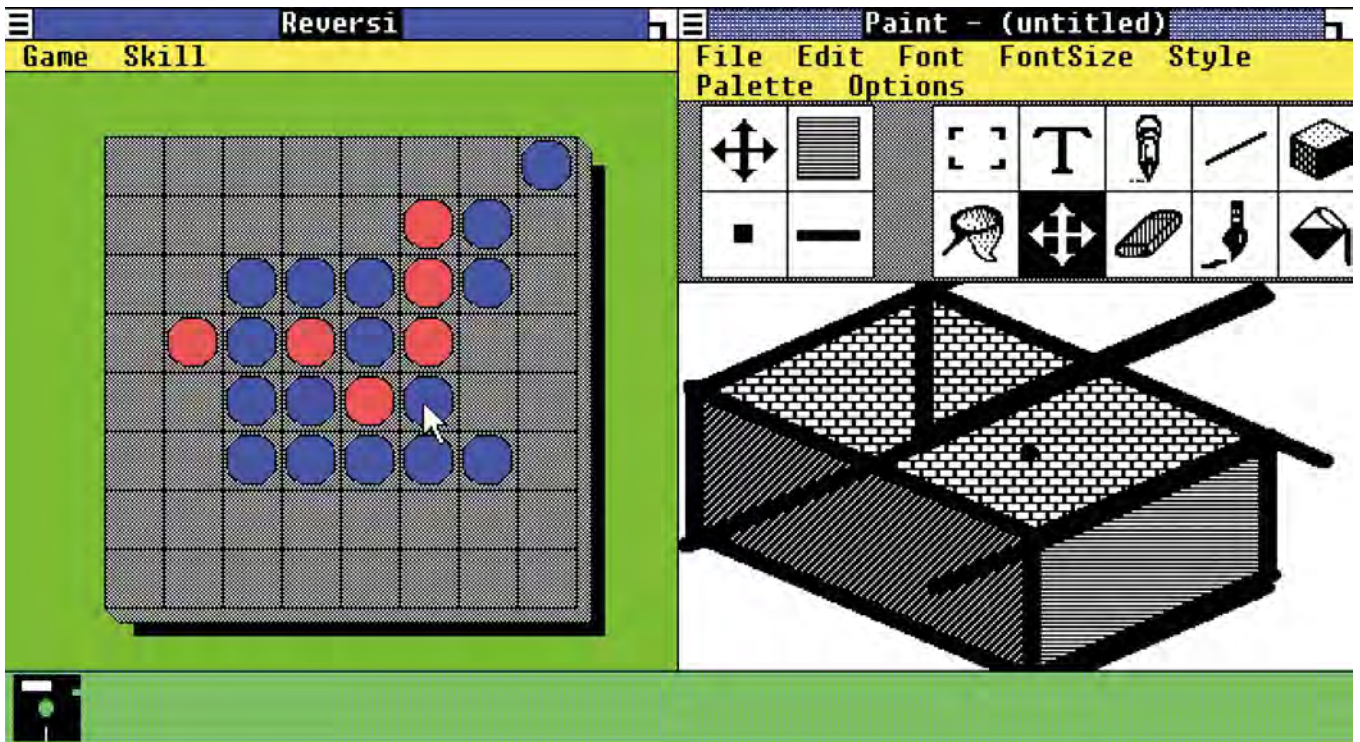
Lack of app support was such a problem that the Windows team developed its own paint programs, utilities and games

By the early summer of 1985 Windows still wasn't finished, but Ballmer decided to release a 'Premiere Edition' to application developers and members of the press. The team went into crunch, to the extent that one young program manager, Gabe Newell (yes, that one) started sleeping in the office. Even at the last stages, new defects were found in the memory management code, delaying the release even further. It was only in November that testing Windows was finished, to be released at Comdex 1985 with a comedy roast where Microsoft poked fun at its own product's lateness.

Even selecting from a pull-down menu is different, involving a click, button-hold, select and release process

MALIGNED AND MISUNDERSTOOD

You might have expected the response to be rapturous, but – as with so many Microsoft products – there was disappointment and bemusement. InfoWorld ran its review with the headline 'Windows Requires Too Much Power' and gave it 4.5 out of 10. A piece by Erik Sandberg-Diment for The New York Times called Windows extremely memory-hungry. 'Running Windows on a PC with 512K of memory', he noted 'is akin to pouring molasses in the Arctic. Also, the more windows you activate, the more sluggishly the program makes its moves'.

Most of all, pundits weren't convinced that Windows solved any genuine problems. Some didn't see the point of
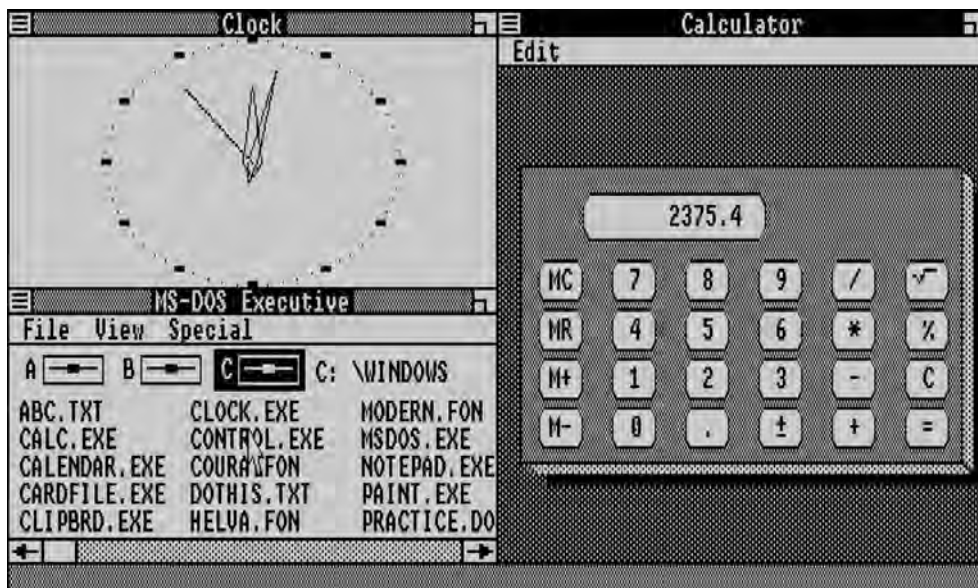
the mouse or the GUI. Sandberg-Diment had his doubts about dialogue boxes, suspecting that most people would prefer 'a more direct means of executing commands.' He also felt that multi-tasking was a waste of effort. 'Most people use but one program most of the time, if not all the time,' he suggested. That's aged well.

USING WINDOWS

So how successfully did Windows 1 lay down the foundations for the Windows we know and sort of love today? Well, it has to be said that it's a very different experience. There's no desktop and the management of windows is incredibly primitive. While it is mouse-driven, icons don't play a starring role. Instead, you launch applications by double clicking on a list in the MS-DOS Executive – a simple file manager that lists not just the programs, but all your MS-DOS files.

The first application you launch occupies the whole screen, and subsequent applications split the screen into two, three or four. Once windows are in place you can close, maximise or resize them, or move them from one half or corner of the

widescreen aspect ratio, or your favourite drawing subjects are sausage dogs and snakes.

Windows Write is recognisably a word processor, but there's no spell check or anything beyond basic formatting features, much like the Windows Write we all carried on not using before Windows 95. And as for Reversi, well it's a variant of the classic black and white disc strategy game Othello, but – let's face it – it's no Minesweeper or Solitaire.

LOOKING TO THE FUTURE

What's most striking about using Windows 1 now is that it feels less like an operating system than a fancy front-end for MS-DOS. It still runs from an MS-DOS command prompt, it still works with the MS-DOS file and directory structures and it was still partly designed to run MS-DOS applications, principally because Microsoft had little faith in anyone developing native Windows ones. They were right, as until Windows 3.0 took off, barely any of the major software vendors made Windows software.

Yet there are aspects of Windows that show its potential. We'll be kind and say that Microsoft 'borrowed' Apple's concept of a clipboard, allowing you to cut and paste text or pictures from one application to another. Microsoft's early Windows adverts go big on copying contact details from a database and pasting them into a letter in Windows Write, then adding a graph from Microsoft Chart or Lotus 1-2-3 which, at that point, was the T-Rex of business applications.

Microsoft designed Windows to be compatible with a range of applications – not just its own – and to promote interoperability, so that you didn't have to work with just one application, or even one specific suite. Windows wanted you to mix and match. Microsoft also designed Windows as a GUI that could work across PCs with different hardware, and this in turn helped to make the PC market more competitive. Even at that point, Bill Gates' ambition was 'to create the software that puts a computer on every desk and in every home'.

Sure, Microsoft wanted to build the system software and the most important applications, but it also understood the necessity of bringing other software-makers on board. As Gates said in 1993, 'Our vision, we shared; we didn't view that as some competitive edge. We just wanted to talk about it and get other people to share the same ideas so that they would help make it all come true.'

Of course, Microsoft has never been shy about monopolising, but Windows has always been stronger when Microsoft opened up and led the way. You can see Windows 1 as the start of that process, even if it's not an OS that you'd want to use today. **CPC**

If you didn't have the cash for a new-fangled EGA card, you were stuck with the even lower-res, black and white CGA version

screen to another. But with no overlapping, space gets tight pretty quickly, and the size of the fonts and the blocky graphics mean you don't always get enough room per application to make head or tail of what's going on.

There's no taskbar, but icons for the MS-DOS Executive and any minimised applications appear in a space at the bottom of the screen, where a double click will bring them back into view. There's also a Control Panel where you can set the time and date, adjust your cursor preferences, add fonts and set up your colour scheme. Of course, the EGA standard only supported a maximum of 16 colours from a gamut of 64, while only seven fonts were available on release.

Even some Windows fundamentals don't work like we expect these days. Drag and drop is as non-existent as the old Recycle Bin. Today, we also forget how Windows was so keen to demand double clicks when a single click would do. The menu bar is in place, with a button in the top-left corner where you can resize, move, close, maximise and minimise the window, but the latter two options are called Icon and Zoom. Even selecting from a pull-down menu is different, involving a click, button-hold, select and release process that feels utterly alien now.

The next shock is the primitive built-in applications. Calculator is a simple calculator with only the most basic functions. Calendar has a single field where you can add appointments on the hour or add alarms, but nothing else. The Notepad is your classic no-frills text editor – and we mean no-frills – while Terminal is the kind of baffling, text-driven comms program that only ever looked good in WarGames. Seriously, people used this stuff?

The highlights are Paint, Windows Write and Reversi, not because they're any good but because they bear some vague resemblance to modern applications. Paint has a palette of tools, plus drop-down menus to handle fonts and options. It also has virtually no room to actually do anything with the tools, unless you're keen on drawing in a low-resolution,

WINDOWS 3.1

Screensavers, colourful icons and proper fonts. 30 years after its release, **Stuart Andrews** looks back at the version of Windows that finally put it on the map

Windows 3.1 is arguably the most crucial Windows ever – the Windows that defined how PC computing looked just as it was beginning to take off. Before version 3.1, Windows was a successful operating system, but one that looked and felt like a GUI shell perched precariously on DOS.

With the launch of Windows 3.1 in April 1992, Windows finally looked and felt like the real deal. What's more, it was a sales phenomenon, shipping over 3 million copies in its first six weeks on the market and 25 million within the first year. Windows was already big, but 3.1 put Windows in the lead.

How did Windows 3.1 do this? That's not something you can nail down to any one factor. It was partly a question of stability, partly features and partly look and feel. Believe us – Windows 3.1 looks rough by today's slick standards, but not half as rough as what came before.
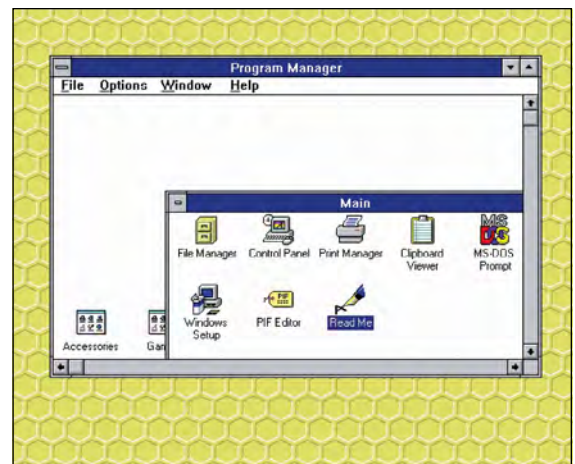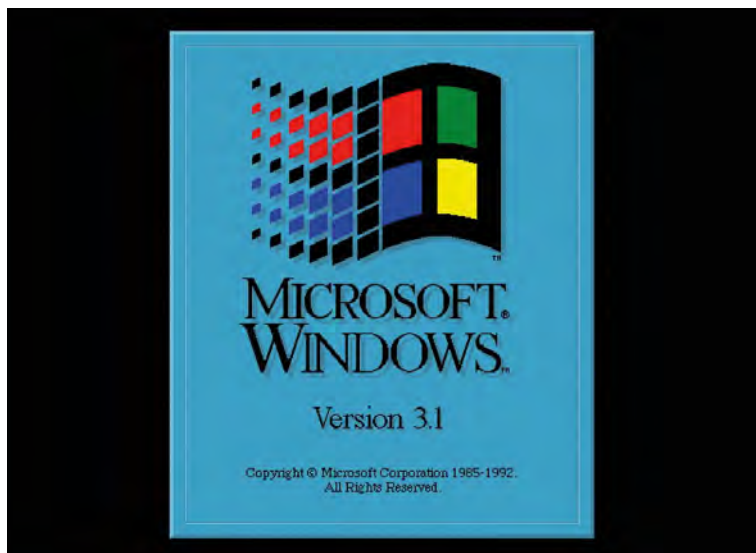
LOOK AND FEEL

Look and feel certainly played an important part in Windows 3.1's success. Windows 3.0 has already done some of the hard work of introducing a proper GUI, replacing the horrible, text-based MS–DOS Executive of Windows 1.x and 2.x with the new Program Manager and File Manager components. Instead of clicking on a program or a file in a list, you could double click on an icon to launch it. Yet Windows 3.1 went further, taking advantage of the VGA and SVGA graphics standards to introduce a revamped UI with more colourful icons.

What's more, those icons could now do more than just get clicked on, as Windows 3.1 introduced drag and drop. You could explore your PC's file system visually, copying files from one folder to another by clicking on the file, dragging it over and releasing the mouse button. You could drag a file onto the Print Manager icon to print it out, or onto the application's icon in Program Manager to open it and start work.

Yet perhaps the most vital enhancement over Windows 3.0 was the introduction of TrueType fonts. At this point, Windows still involved a lot of text and, up until Windows 3.1, this text didn't look good. It was pixelated, primitive and ugly, with no real provision to vary horizontal or vertical spacing.

While developing Windows 3.1, Microsoft put a team together to fix this problem, and that team worked with

Cue a sigh of relief when this splash screen showed up. Launching Windows from MS–DOS could be s...l...o...w

The Program Manager was the heart of Windows 3.1. Double clicking icons launched the applications, or you could drag and drop files onto the icons or open windows

True Type fonts were a revelation to Windows users, making the OS look significantly better and opening up more sophisticated WYSIWYG DTP and design applications

one of the two leading typesetting companies of the era, Monotype, to design a new set of core fonts. Meanwhile, Microsoft worked on the technology to render those fonts on-screen, so they could be scaled upwards and downwards, rotated and respaced, and still look pretty good.

Monotype came up with the Times New Roman, Arial and Courier New fonts that Windows still incorporates today, while Microsoft licensed and adapted Apple's TrueType technology, adapting the font hinting tech that made these fonts clear and legible even on a VGA resolution (640 x 480) screen. This not only made Windows look a whole lot better, but made it a viable platform for desktop publishing and design. Suddenly, the Mac had competition.

This was also the first version of windows to include a built-in screensaver

FUN AND GAMES

Other aspects of Windows 3.1 revealed a more playful side to Microsoft. Windows 1.0 includes one game – Reversi, while Windows 3.0 introduced Solitaire, a patience card game originally developed by a Microsoft intern, Wes Cherry, and responsible for so much lost productivity that Microsoft banned its 'boss key' feature, which switched from the game to a mocked-up Excel spreadsheet, before release.

Windows 3.1 added Minesweeper, the classic game of grid-based bomb discovery so addictive that, legend has it, Bill Gates had it uninstalled from his PC. Not that this stopped him or anyone else playing it. While Gates was known to sneak onto a colleague's computer after hours to play it, the rest of the company joked that Minesweeper was the most carefully tested of all Windows 3.1's new features.

This was also the first version of windows to include a built-in screensaver. As with so many new features, this wasn't all that new – screen burn-in was an issue for CRT-based VGA

monitors, and After Dark's fish and flying toasters had already appeared on Windows 3.0 and macOS. However, Windows 3.1 made screensavers a standard component, introducing long-time favourites, such as the classic flying Windows logo, the Star Trek-style Starfield, and the psychedelic Mystify and Swirl. Seriously. After few too many shandies, they blew our primitive, PC-loving minds.

ARCHITECTURAL IMPROVEMENTS

Yet the most important features that Windows 3.1 introduced were those you couldn't see. Windows 3.0 had introduced protected memory – a way of using the protect mode of the 80286 CPU to allow Windows and Windows apps to use up to 16MB of RAM rather than just the first 640KB.

Coded by ex-physicists David Weise and Murray Sargent, this feature had been crucial, making Windows a viable alternative for Microsoft to working with IBM on what would become OS/2. Running in protected mode gave Windows programs more stability, and enabled MS-DOS applications to run under Windows and still access all the available RAM. This in turn meant that Windows spent less time crashing, which made it a lot more attractive to people trying to get some actual work done.

Windows 3.1 built on this foundation by taking the new memory management features built into the newer 386 processors and using them in a 386 Enhanced mode. Where Windows 3.0 was limited to a maximum of 16MB, Windows 3.1 upped that limit to 256MB (or, in theory, up to 4GB) and enabled programs to use virtual memory above and beyond the physical memory installed.

It also enabled most DOS programs to be run inside a Window with mouse support, and multiple DOS programs to be run simultaneously. What's more, all these enhancements meant Windows 3.1 only worked on an Intel 80286 CPU or later. Rocking an old-school 8086? Tough.

These changes improved not just Windows' overall stability, but its multi-tasking capabilities as well. Applications

Windows 3.1 gave us new ways to customise our desktops, although not much of any value with which to customise them

File Manager

File Disk Tree View Options Window Help

C:\WINDOWS*.*

C:
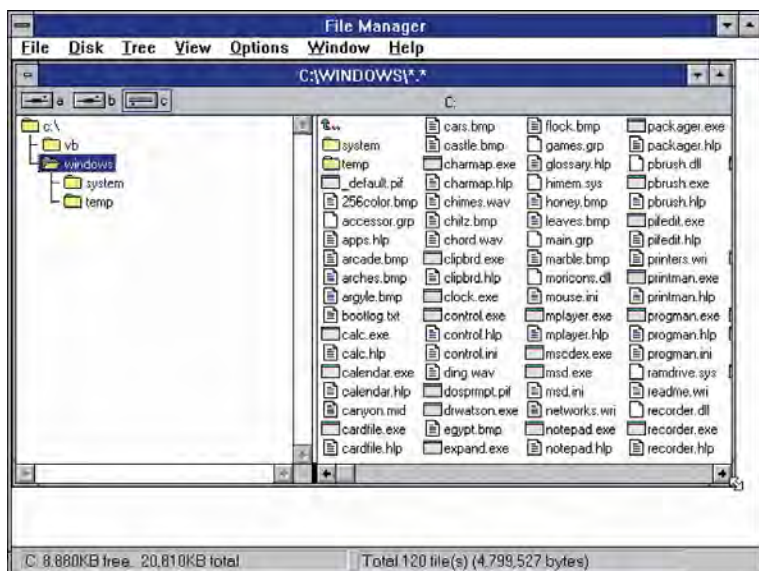
This was as exciting and intuitive as file management got in Windows 3.1. Notice those old-school 8.3 character filenames

mostly got the resources they needed, and a central messaging system alerted them to hand over resources as and when they were needed, although not all Windows programs behaved as well as others. A Task List enabled you to see all the currently running programs and halt any that were gumming up the system, although the more likely outcome was that they would crash Windows first.

What's more, all of this went hand in hand with another major Windows feature. Windows already had the Dynamic Data Exchange (DDE) protocol, which allowed you to take messages and/or data from one Windows program to

WinG worked on a technical level, as proven by a WinG port of id Software's Doom

another. Windows 3.1 went one better with Object Linking and Embedding (OLE), which enabled you to embed an object created by one application into a document created by another, with both apps updating seamlessly when you made any changes.

Suddenly, you could create a chart in Microsoft Excel and stick it in your Microsoft Word report, then update the data in Excel and see the changes rolled out in Word. I know. It doesn't sound that thrilling, but at the time, this rocked the computing world.

Last, but not least, Windows 3.1 gave the world the Windows registry. At the time, this central database of settings wasn't all that well known or understood, and we never felt the need to edit it directly as we would in the Windows 95 years. Still, it showed a willingness to gather vital system information and preferences in one place, rather than in a horde of SYS, INF and INI files, as had been the Windows way until this point.

CONFOUNDING ISSUES

Let's not heap too much praise on Windows 3.1; it still had its fair share of issues. One was that Windows still didn't support long filenames, so both files and directories were limited to names eight characters long, followed by a three-character suffix that told the OS what kind of file it was. This meant users became ingenious at truncating filenames, which in turn made looking through a folder full of documents or save games feel like decoding some esoteric text.

What's more, while Windows 3.1 had support for multimedia hardware, which was just about becoming affordable and available, ease of installation wasn't on Microsoft's list of priorities. Restrictive hardware didn't help – these were the days when solving hardware conflicts involved moving jumpers from pin to pin to swap Direct Memory Access channels. However, Windows 3.1 made the whole process of installing drivers for a CD-ROM drive and sound card as challenging as possible – it might take hours to get the whole setup running.

Networking wasn't any better either, because Windows 3.1 didn't have any built-in networking support. Instead, it piggy-backed on networking clients for the underlying MS-DOS operating system. If you hadn't already mastered Novell Netware or Microsoft LAN Manager, you were still going to have to get to grips with them here.

Nor was the Windows shell ideal. Simply finding a program in Program Manager could be daunting, especially if you weren't sure which folder or group held it. With screen space at a premium, you would have to constantly minimise and restore Windows while you looked. Don't even ask about finding files in File Manager.

Most of all, Windows wasn't a great platform for games. Dodgy drivers and the massive overheads involved in just running Windows itself made it much, much easier to run Wolfenstein 3D or The Secret of Monkey Island 2 in DOS, which Windows needed to run anyway and for which all Windows users had to pay. This also meant that getting games running still required tinkering at text editor level with a range of crucial system files, to the point that most PC gamers were on intimate terms with config.sys, himem.sys and autoexec.bat. Windows 3.1 didn't change this one bit.

With time, there was some movement. In 1994, Microsoft released a new API, WinG, which was designed to deliver faster graphics performance under Windows and encourage more developers to port their DOS games. WinG worked on a technical level, as proven by a WinG port of id Software's Doom. Yet it didn't work so well on the commercial level, with developers looking at the work involved and the existing DOS user base, then shrugging their shoulders until Windows 95 and DirectX came along.

Still, for all these faults, Windows 3.1 was a major leap in the right direction, paving the way not just for Windows 95, but for the switch from IBM and OS/2 towards Windows NT. Without that we might never have had the PC boom of the mid-1990s, Windows XP and everything beyond. And where would we all be without that? **CPC**

THE FIRST PC

IBM PC 5150

Ben Hardwidge travels back to August 1981, when IBM released its Personal Computer 5150 and the PC was born

A big ape had only just started lobbing barrels at a pixelated Mario in Donkey Kong arcade machines, Duran Duran's very first album had just rolled off the vinyl presses and Roger Federer was just four days old. In this time, the UK was even capable of winning Eurovision with Bucks Fizz. It's August 1981, and IBM has just released the foundation for the PCs we know and love today, the PC 5150.

'By the late 1970s the personal computer market was maturing rapidly from the many build-it-yourself hobbyist kits to more serious players like Apple, Commodore and Tandy,' retired IBM veteran Peter Short tells us. 'As people realised the greater potential for personal computers in business as well as at home, pressure grew on IBM to enter the market with their own PC.'

Short is now a volunteer at IBM's computer museum in Hursley (**slx-online.biz/hursley**), which holds a huge archive of the company's computing machines and documentation, from Victorian punch card machines to the company's

personal computers. We ask him if it felt like the beginning of a new era when the PC was first launched 40 years ago. 'Yes,' he says, 'but probably not the beginning of something so huge that its legacy lives on today.'

At this time, the home computer market was really starting to take off, with primitive 8-bit computers, such as the Sinclair ZX80 and Commodore VIC-20, enabling people at home to get a basic computer that plugged into their TV. At the other end of the scale, large businesses had huge mainframe machines that took up entire rooms, connected to dumb terminals.
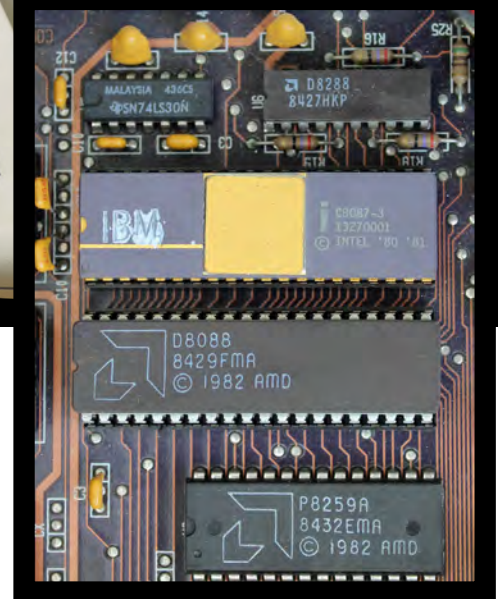
There was clearly room for a middle ground. IBM was going to continue producing mainframes and terminals for many years yet, but it also wanted to create a powerful, independent machine that didn't need a mainframe behind it, and that didn't cost an exorbitant amount of money.

The PC 5150's launch price of $1,565 US (around £885 ex VAT) for the base spec in 1981 equates to around £3,469 ex VAT in today's money. That's still very far from what we'd

call cheap, but it was a colossal price drop compared with IBM's System/23 Datamaster, an all-in-one computer (including screen) that had launched earlier the same year for $9,000 US – six times the price. And even that was massively cheaper than some of IBM's previous microcomputer designs, such as the 5100, which cost up to $20,000 US in 1975.

The ticking heart of the box is a 4.77MHz 8088 processor made by AMD

IBM needed to act quickly. Commodore had already got a foothold in this market several years earlier with the PET, for example, and IBM realised that it couldn't spend its usual long development time on the project. The race was on, with the project given a one-year time frame for completion.

'At the time, IBM was more geared up to its traditional, longer-term development processes,' explains Short. 'But it eventually realised that, with a solid reputation in the marketplace, it was time to look for a way to do fast-track development that would not produce a machine three, four or five years behind its competitors.'

PROCESSORS AND COPROCESSORS

We opened up a PC 5150 for this feature, so we could have a good look at the insides and see how it compares with PCs today. It's hugely different from the gaming rigs we

see now, but there are still some similarities. For starters, the floppy drive connects to the PSU with a 4-pin Molex connector, still seen on PC PSU cables today. The PC was also clearly geared towards expansion from the start.

The ticking heart of the box is a 4.77MHz 8088 CPU made by AMD – Intel had given the company a licence to produce clones of its chips so that supply could keep up with demand. It's for this reason that AMD still has its x86 licence and can produce CPUs for PCs today, but at this point, the two companies weren't really competitors in the way they are now. To all intents and purposes, an AMD 8088 was exactly the same as an Intel one, and PCs generally came with whichever one was in best supply at the time of the machine's manufacture.

The CPU itself is an interesting choice. It's a cut-down version of Intel's 8086 CPU that it had launched in 1978. The 8088 has the same execution unit design as the 8086, but has an 8-bit external data bus, compared with the 8086's 16-bit one. As with today's PCs, the CPU is also removable and replaceable, but in the case of the PC 5150, it's in a long dual in-line package (DIP) with silver pins, rather than a square socket.

Immediately above the CPU sits another DIP socket for an optional coprocessor. At this point in time, the CPU was only an integer unit with no floating point processor. This

The floppy drive connects to a 4-pin Molex connector on the PSU – a plug that's still sometimes used in today's PCs

was generally fine in an era when most software didn't overly deal with decimal points, but you had the option to add an 8087 coprocessor underneath it. This worked as an extension of the 8088 CPU. 'Adding the 8087 allowed numeric calculations to run faster for those users who needed this feature,' explains Short.

The decision to use a CPU based on Intel's x86 instruction set laid the machine code foundation for future PCs, and hasn't changed since. Comparatively, Apple's Mac line-up has had a variety of instruction sets, including PowerPC, x86 and now Arm. Nvidia might be making big noises about the future of Arm in the PC, but the x86 instruction set has stood its ground on the PC for 40 years now.

IBM itself has also dabbled with different instruction sets, including its own 801 RISC processor. Why did it go with Intel's CISC 8088 CPU for the first PC? The answer, according to Short, is mainly down to time and a need to maintain compatibility with industry standards at the time.

'The first prototype IBM computer using RISC architecture only arrived in 1980 and required a compatible processor,'

It's typical of the Wacky Races vibe, with multiple features shoehorned into one slot

he explains. 'In order to complete the 5150 development in the assigned one-year time frame, IBM had already decided to go with industry-standard components, and there was existing experience with the 8088 from development by GSD (General Systems Division) of the System/23. RISC required the IBM 801 processor, but the decision was made to go with industry standard components.'

EXPANSION SLOTS

In addition to the ability to add a coprocessor, the IBM PC 5150's motherboard also contains five expansion slots, with backplate mounts at the back of the case, just like today's PCs. Three of the slots in our sample were also filled.

One card is actually two PCBs sandwiched together – it's a dual-monitor video card with the ability to output to both an MDA screen and a CGA screen simultaneously (more on

**The IBM PC 5150 had five 8-bit ISA slots for expansion cards**

these standards later) – each standard required a separate PCB on this card – there's a composite TV output in addition to the pair of 9-pin monitor outputs as well. Bizarrely, this card also doubles as a parallel port controller, with a ribbon cable providing a 25-pin port. It's typical of the Wacky Races vibe seen on cards at the time, with multiple features shoehorned into one expansion slot.

Similarly, there's also a 384KB memory expansion card, which also doubles as a serial I/O card, with a 25-pin port on the backplate. The final card is an MFM storage controller for the 5.25in floppy drive at the front of the machine.

Although the PC was clearly built with expansion in mind, Short points out that 'IBM was not the first to introduce expansion slots. As far back as 1976, Altair produced the 8800b with an 18-slot backplane, the Apple II also featured slots from 1977 and there was also an expansion bus on the BBC Micro from 1981. No doubt market research and competitive analysis showed that this approach would provide additional flexibility and options without having to redesign the motherboard'.

Interestingly, though, Short also says IBM was keeping an 'eye on the hobby market. A standard bus with expansion slots would allow users to create their own peripherals. IBM even announced a Prototyping Card, with an area for standard bus

A raw ISA card at the Hursley museum, designed for hobbyists to make their own expansion cards
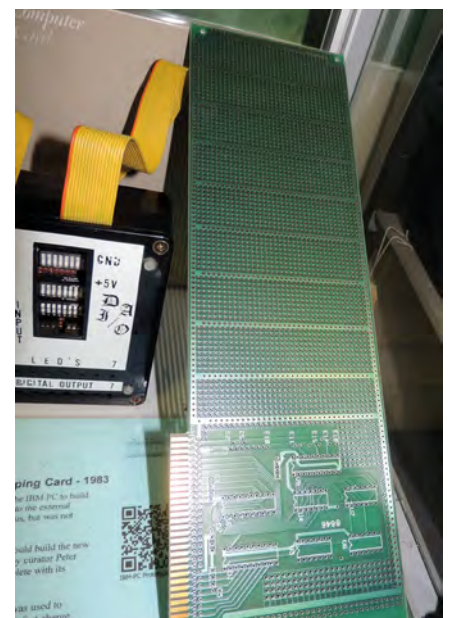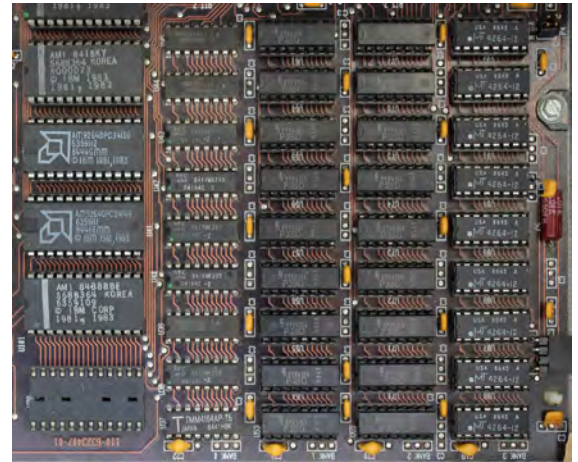
A 5.25in floppy drive was the standard storage system for the 5150, with no hard drive option at launch

The memory is organised in four banks in the bottom right corner of the motherboard – in this case there are four 64KB banks, adding up to a total of 256KB

interface components and a larger area for building your own design'. It's a far cry from the heavily populated PCI-E cards with complex machine soldering that we see today.

MEMORY

That 384KB memory card shows a very different approach to memory expansion than the tidy modules we have today. Believe it or not, at launch, the PC 5150 base spec came with just 16KB of memory (a millionth of the amount of memory in today's 16GB machines), which was supplied in the form of DRAM chips on the bottom right corner of the motherboard.

The top spec at launch increased that amount to 64KB, although you could theoretically also install the DRAM chips yourself if you could get hold of exactly the right spec of chips and set it up properly. The chips on the motherboard are split into four banks, each with nine chips (eight bits and one parity bit). In the original spec, the 16KB configuration filled one

bank, while the 64KB configuration filled all four banks with 16KB of memory each.

A later revision of the motherboard expanded this to 64KB as the base spec with one bank filled, and 256KB with all four banks filled (this is the spec in our sample). If you then added a 384KB memory card, such as the one in our sample, you ended up with 640KB of memory – the maximum base memory addressable by PCs at this time.

GRAPHICS AND DISPLAYS

As we previously mentioned, our PC 5150 sample has a dual-monitor card, which supports both the display standards available to the IBM PC at launch. A Mono Display Adaptor (MDA) card could only output text with no graphics, while a Color Graphics Adaptor (CGA) card could output up to four colours (from a palette of 16) at 320 x 200, or output monochrome graphics at 640 x 200.

However, as Short notes, 'the PC was announced with the mono 5151 display in 1981. The CGA 5153 was not released until 1983'. Even if you had a CGA card in your PC 5150, if you used the original monitor, you wouldn't be able to see your graphics in colour. Seeing colour graphics either required you to use the composite output or a third-party monitor.

'Once the colour monitor became available,' says Short, 'it could either be attached as the sole display with its own adaptor card, or equipped with both a mono and colour adaptor card, and could be attached together with a mono screen. Now you could run your spreadsheet on the mono monitor and display output graphics in colour.'

There's an interesting connection with the first PC monitors and the legacy of IBM's computing history too. When we interviewed the Hursley Museum's curator Terry Muldoon (who has now sadly passed away) in 2011, he told us the reason why the first PC monitors had 80 columns. 'It's because it's the same as punch cards,' he said. 'All green-screen terminals had 80 columns, because they were basically emulating a punch card.'

IBM's colour 5153 monitor didn't come out until 1983, shown here with an IBM PC XT at Hursley, with Alley Cat in full CGA glory

DOS running on an IBM PC 5150 with a monochrome green screen at Hursley

STORAGE

Storage is another area where the PC is at a crossroads between new tech. As standard, the PC 5150 came with a single 5.25in double-density floppy drive, with 360KB of storage space on each disk. There was the option to add a second floppy drive in the empty drive bay, but there was no hard drive at launch.

'The first hard drive for microcomputers did not arrive until 1980 – the Seagate ST506 with a capacity of 5MB,' explains Short. 'By that time, the PC specifications had already been agreed and the hardware development team in Boca Raton was in full swing. The requirement was for a single machine developed within a one-year time frame.

'A small company called Microsoft was also developing the first version of DOS under sub-contract. The 5150 BIOS therefore had no hard disk support – DOS 1.0 and 1.1 are the same. The power supply selected for the 5150 wasn't beefy enough at 63W to power the 5150 and a hard drive.'

Later versions of the 5150, such as our sample, came with a 165W PSU, and future DOS versions enabled you to run a hard drive, but it wasn't until the IBM PC 5160 XT in 1983 that there was a hard drive option with an IBM PC as standard.

The PSU also connects to a massive red switch power switch on the side, which is very different from the delicate touch-buttons we have today. You had to literally flip a switch to power on the first PCs. This was another legacy of IBM's past – a time when, if a machine needed to be shut down drastically, you would 'BRS it' – BRS stands for big red switch.

The back of the PC 5150 also alludes to another form of storage. There are two DIN sockets on the back, one of which is labelled for the keyboard – the other is labelled 'cassette'. 'It was common at the time to provide software on cassette tapes, which could also be used to store user written programmes,' says Short. 'My own Radio Shack TRS80 in 1979 used this method. A standard cassette tape machine such as the Philips could be connected through this socket.'

SOFTWARE SUPPORT

This brings us neatly to the subject of software support. We're now used to graphical user interfaces such as Windows as standard, but in 1981 Microsoft was a small company, which had developed a popular version of the BASIC programming language.

'Microsoft Basic was already very much an industry standard by 1980,' says Short. 'It was Microsoft's first product. This fitted with the concept of using industry standard components. IBM chose to sub-contract its operating system development to Microsoft, perhaps for this reason. Again, the compressed development schedule influenced these decisions.'

Terry Muldoon gave us some more insight into the development of the PC's first operating system, IBM PC DOS 1.0, when we spoke to him in 2011. 'The story I heard is that basically IBM needed an operating system,' he said, 'and IBM didn't have time to write one – that's the story. So they went out to various people, including Digital

You flip the big red switch (BRS) on the side to power the PC 5150 up or down

Research for CPM, but Digital Research didn't return the call. Bill Gates did, but he didn't have an operating system, so he went down the street and bought QDOS.

'The original DOS was a tarted-up QDOS, supplied to IBM as IBM Personal Computer DOS, and Gates was allowed to sell Microsoft DOS (MS-DOS). And they carried on for many years with exactly the same numbers, so 1.1 was DOS 1 but with support for us foreigners, then we went to DOS 2 with support for hard disks, DOS 2.1 for the Junior, DOS 3 for the PC80 and so on.'

You can have a play with DOS 1.0 on an emulated PC 5150 at **custompc.co.uk/5150**, and it's a very basic affair. Even if you've used later versions of DOS, there are some notable absences, such as the inability to add '/w' to 'dir' to spread out the directory of your A drive across the screen, rather than list all the files in a single column.

What's also striking is the number of BASIC files supplied as standard, which can be run on the supplied Microsoft BASIC. One example is DONKEY.BAS, a primitive top-down game programmed by Bill Gates and Neil Konzen, where you move a car from left to right to avoid donkeys in the road (really). What's more, this game specifically requires your PC to have a CGA card and to run BASIC in advanced mode – you couldn't run it on the base spec.

A FUTURE STANDARD

With its keen pricing compared with previous business computers, the IBM PC 5150 was well received in the USA, paving the way for a launch in the UK in 1983, along with DOS 1.1 and the option for a colour CGA monitor. Clone machines

The power supply wasn't beefy enough to power the 5150 and a hard drive

from companies such as Compaq soon followed, claiming (usually, but not always, rightly) to be 'IBM PC compatible', and the PC started to become the widespread open standard that it is today. Was this intentional on IBM's part?

'Industry standard components, an expansion bus and a prototyping card would naturally lead to an open standard,' says Short. 'Not publishing the hardware circuitry would make it difficult to capture the imagination of "home" developers. Open architecture was part of the original plan.'

Muldoon wasn't so sure when we asked him back in 2011. 'Now where did IBM make the mistake with DOS?' He asked. 'This is personal opinion, but IBM allowed Bill Gates to retain the intellectual property. So we've now got an Intel processor – the bus was tied to Intel – and another guy owns the operating system, so you've already lost control of all of your machine in about 1981. The rest is history.

'The only bit that IBM owned in the IBM PC was the BIOS, which was copyright. So, to make a computer 100 per cent IBM compatible, you had to have a BIOS. There were loads

The IBM Personal Computer laid the foundation for the PCs we know and love today

of software interrupts in that BIOS that people used, such as the timer tick, which were really useful. You get that timer tick and you can get things to happen, so you have to be able to produce something that hits the timer tick, because the software needs it.'

Rival computer makers could circumvent the copyright of the BIOS by examining what it did and attempting to reverse-engineer it. Muldoon explained the process to us.

'The way people did it is: with one group of people, say: "this is what it does", and another group of people take that specification, don't talk to them, and then write some code to make it do that – that's called "clean room". So one person documents what it does, and another person now writes code to do it – in other words, nobody has copied IBM code, and there's a Chinese wall between these two people.

'What some of the clone manufacturers did is, because we published the BIOS, they just copied it. Now, the BIOS had bugs in it, and we knew they'd copied our BIOS because they'd copied the bugs as well. This was only the small companies that came and went. Phoenix produced a clean room BIOS, so if you used a Phoenix chip in your clones, you were clean.'

Of course, any self-contained personal computer can technically be called a PC. Peter Short describes a PC as a machine that 'can be operated directly by an end user, from beginning to end, and is general enough in its capabilities'. It doesn't require an x86 CPU or a Microsoft OS. In fact, there was and still is a variety of operating systems available to x86 PCs, from Gem and OS/2 in the early days, through to the many Linux distributions available now.

However, the PC as we generally know it, with its x86 instruction set and Microsoft OS, started with the PC 5150 in 1981. Storage and memory capacities have hugely increased, as have CPU clock frequencies, but the basic idea of a self-contained box with a proper CPU, enough memory for software to run, its own storage and a display output, as well as room to expand with extra cards, started here. Thank you, IBM. **CPC**

HOW TO

BUILD A DOS PC WITH A MODERN TWIST

Ben Hardwidge shows you how to build a pure retro PC gaming rig, without scraping your fingers on horrible off-white cases, or faffing with ancient hard drives

Tech nostalgia can be a perilous and well-lubricated slope if you don't tread carefully, a bit like a staircase in an old Sierra adventure game (we're looking at you, King's Quest III). It starts innocently, perhaps by browsing screenshots of LucasArts adventures on **mobygames.com**, or wistfully searching for old hardware on Google Images. You might even download Doom and The Secret of Monkey Island on Steam or GOG, or have a play around with DOSBox.

You can stop there, of course. You can, honestly. There's no reason to scratch this reminiscing itch any further, but there's always another step to a hobby that starts to venture into the realms of silliness, and we've decided to take it. We're going to build a PC based on original hardware from the 1990s, and show you how to do it.

WHY? JUST, WHY?

There's no doubt that DOSBox is a seriously powerful virtual emulator that works brilliantly. However, it doesn't quite tick all the boxes. If you want to put an actual hardware synthesiser card in your PC, and hear it play music in games in real time, with the option to expand it with a wavetable daughterboard, then you need the original hardware.

If you want to boot to a DOS prompt and know that your computer is running the software natively, without emulation, then you'll need a real 1990s PC. In all honesty, it might make little difference to the end result compared with an emulator, but it's a fun project, and it also gives you an understanding of how old PC hardware works, which you don't always get from an on-screen DOS emulator.

▶

If you've ever wondered what it was like to use a PC from the old days, and fancy having a dabble with old hardware, but don't know your AT from your AT-AT, then this Retro tech special is for you. In fact, even if you don't want to buy a load of overpriced ancient hardware in order to construct an obsolete gaming machine (and we won't judge you for that, much), we'll give you a grounding in how the PC has changed in some ways, but not in others, and help you understand the foundation on which today's PCs are built.

MISSION BRIEFING

The idea behind our retro rig is to combine the best of the old world with the perks of the new world. While there are parts of the legacy PC hardware era we miss, there are others that we're very glad have been consigned to the great silicon scrapheap in the sky. There was no way we were going to use a 1990s case, for example. As well as having that horrible off-white colour, which yellowed over time, early PC cases were often badly designed and built. There was nearly always sharp metal on which you could easily scrape your knuckles, very little consideration given to cable routing, drive bays everywhere and the PSU would often be sat at the top.

The idea is to combine the best of the old world with the perks of the new world
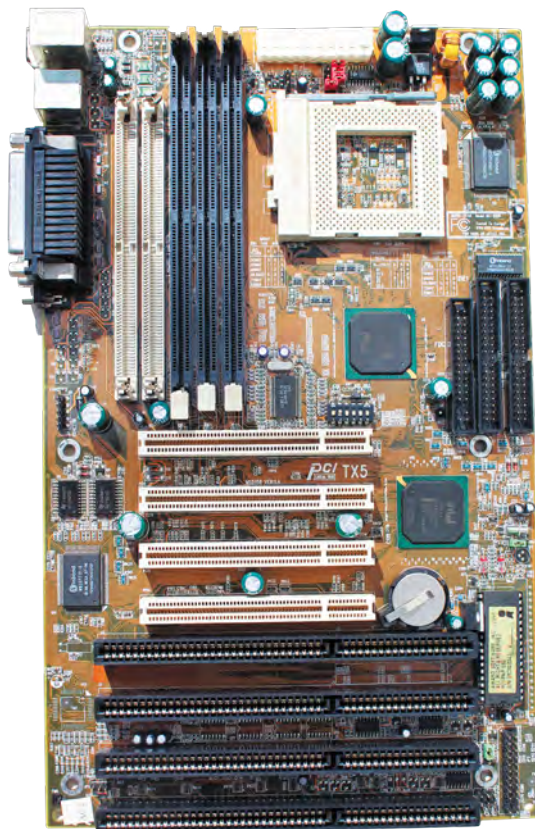
We also wanted to avoid using an old hard drive. Mechanical hard drives can become unreliable after five years, let alone 25, plus they're slow and noisy, so we wanted to use solid state storage. However, we also wanted the flexibility to run any old software from charity shops and eBay, not to mention disks from the loft, and that means our system needs a CD-ROM drive and a floppy drive too.

Finally, PSUs have come an enormously long way since the 1990s. We have modular and semi-modular designs, wrapped/sleeved cables as standard, and the 80 Plus initiative has weeded out the flaky and inefficient PSU designs that were commonplace 25 years ago. So we're using solid state storage, a modern case and a new PSU. The rest of the core spec, however, is contemporary 1990s DOS hardware.

SLOTS OF FUN

Your first priority when building a DOS gaming system is the motherboard. You want one with 16-bit ISA slots (long and usually black), so you can get the sound working properly. PCI sound cards were largely designed for Windows, rather than DOS, and while some of them have DOS drivers, it's a faff trying to get them to work in all your games.

PCI sound cards also tend to make heavy use of the CPU, and don't have all the required audio hardware on them, relying on the CPU to do some of the work. That's fine if you have a Pentium III and Windows 98, but it's rubbish for DOS gaming – an ISA card will have all the synthesiser hardware you need on it. However, it's worth having a PCI slot for your graphics card.

Some motherboards from the 1990s will also have VESA local bus slots, which look a bit like an ISA slot with a brown PCI slot on the end. You can install an ISA graphics card in these slots, but actual VESA local bus cards are generally expensive and hard to find these days. Ideally, look for a motherboard with a mix of both ISA and PCI slots – the latter are short, with a thin socket in the middle, and they're usually white.

You also want a replaceable CMOS battery. These silver discs are a standard feature of today's

The MSI MS-5158 motherboard used in our DOS PC. The ISA slots are the black ones at the bottom, and the PCI slots are the white ones above them

Choosing a motherboard with a replaceable CMOS battery will save you having to replace a dead battery with a soldering iron
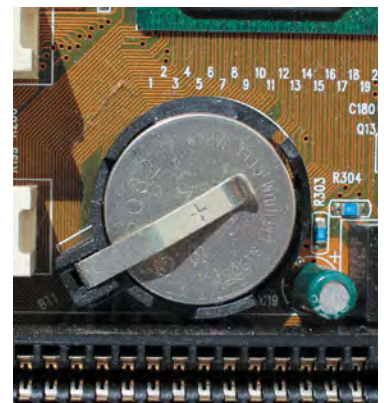
Molex-to-floppy adaptors are easy to find on eBay. These black ones are made by Corsair

motherboards, but in the 1980s and 1990s, the CMOS battery was often soldered to the motherboard. This wasn't a major issue at the time, as the batteries lasted for years, but decades later, these batteries have run dry, and you'll now need a careful touch with a soldering iron to replace them.

The other crucial considerations for motherboards are their form factor and power socket. For starters, avoid any proprietary motherboard designs. These were often found in the PCs from big brands at the time, such as Compaq, Hewlett-Packard and so on. These PCs were fine in themselves, but they severely lacked flexibility, as they often had proprietary power connectors and their I/O ports would be positioned in non-standard locations, to fit with the custom case designs.

Meanwhile, in the world of DIY PCs and independent system builders, there were two main standards of motherboard – AT and ATX, with the latter coming later. AT was a firm favourite among enthusiasts, as it maintained compatibility with older cases. If you bought an AT 386 system, you could replace the motherboard with a Super Socket 7 AT motherboard many years later.

The only port soldered to a standard AT motherboard was a large 5-pin DIN socket for connecting a keyboard. Any other ports would be connected via ribbon cables, using expansion slot backplates or cut-outs on the back of the case.

The power supply socket was also a bit of an oddity, with the PSU having two plugs that needed lining up together. There was no way to turn off an AT PC with software either, so there would be a hard push-in/push-out switch trailing off the PSU.

ATX introduced the motherboard power socket that we still use today, albeit with 20 pins rather than 24, as well as the ability to turn off the system with software, so you could tap a simple power button rather push in a hard switch. The ATX standard also introduced the standard motherboard length and I/O backplate design you see on motherboards now.

You can avoid most of the pitfalls of incompatibility by simply opting for a motherboard with an ATX form factor and power socket. In our case, we've gone for an MSI MS-5158, an ATX Socket 7 motherboard based on Intel's Triton TX chipset – it has PCI and ISA slots, and a

replaceable CMOS battery, plus it supports a wide range of CPUs, from early Intel Pentiums to the later MMX chips, as well as AMD's K6 processors.

PLUGGING AWAY

Your power supply and its connectors are also a consideration. You should be able to use a new one, rather than scouting around for an old 1990s one – just make sure that the main ATX connector (24-pin on the latest designs) can be separated into two parts – the large 20-pin part will connect straight into the socket on an old ATX motherboard.

Secondly, you'll need plenty of Molex connectors, plus a couple of floppy power connectors. Most PSUs have the former in some form, but the latter can be hard to find sometimes. Thankfully, you can get adaptor cables on eBay and Amazon, which go straight from a Molex connector to a floppy plug, and also ones that go from SATA power to Molex power. We bought a pair of black Corsair Molex-to-floppy adaptors on eBay.

CHEAP AS CHIPS

The CPU is your next consideration, and if you read our piece on Socket 7 (see Issue 203, p108), you'll know there's a huge range of options here. We suggest avoiding Cyrix CPUs, as their poor floating-point performance will make them struggle in games such as Quake. An AMD K6 or Intel Pentium will do the job fine. We're using a 166MHz

We're using a 166MHz Intel Pentium MMX CPU
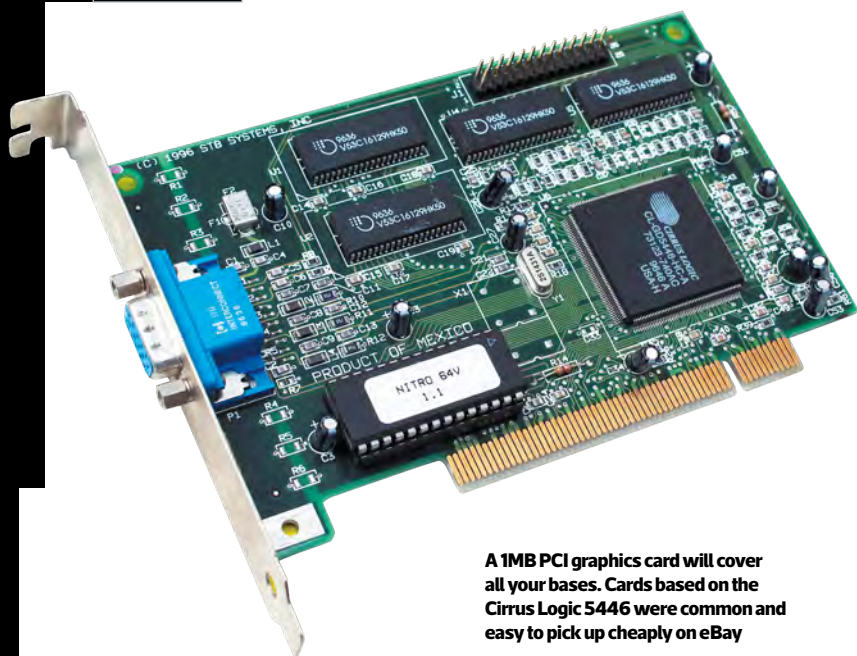
A 1MB PCI graphics card will cover all your bases. Cards based on the Cirrus Logic 5446 were common and easy to pick up cheaply on eBay

72-pin SIMMs need to be installed in pairs on Pentium systems – we're using two 16MB EDO sticks

Pentium MMX, which is overkill for most of the games we'll be running on our system, but that hardly matters when you can pick them up for £15 now.

You could also use a Slot 1 Pentium II or Celeron system, as long as it has ISA slots for your sound card. These systems were much more geared towards 32-bit computing than their Socket 7 predecessors, and they won't give you any advantage over Socket 7 CPUs in DOS games, but they will work.
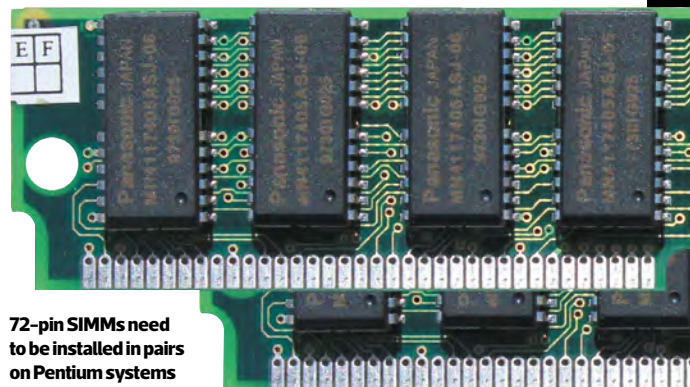
Likewise, an AGP graphics card will boot into DOS, but there's no advantage to using one over a PCI card, as there will be no DOS 3D drivers. There are a few considerations when buying a graphics card for DOS though. Firstly, the minimum you want for DOS games is a 256KB VGA card, which will enable you to run games at

For our DOS and Windows 3.1 system, 32MB of memory is plenty

320 x 240 (or 320 x 200) with 256 colours, and you can step up to 640 x 480 with 16 colours. Having a 512KB card will enable you to get 256 colours at 640 x 480, and stepping up to 1MB will even enable 16-bit colour (over 64,000 colours) at 640 x 480, or 256 colours at 800 x 600.

Virtually no DOS games support the latter two modes, but they're handy if you want to have a play with Windows 3.1. If you do want to run Windows 3.1 on your retro system (which will open you up to some other games, such as Civilization II), make sure there's a driver available for your graphics card. A PCI graphics card will also be quicker in Windows than an ISA one, making for a more responsive experience. We've chosen a 1MB PCI Cirrus Logic 5446, which covers all our bases – you can still pick them up cheaply on eBay.

You'll also need a supporting monitor with an analogue 15-pin VGA input, or an HDMI or DVI input with an active

adaptor – a passive cable won't work here, as you need to convert the analogue signal to digital, so do your research before purchasing. If you want the full retro experience, you could even pick up an old CRT monitor, but we're just plugging our machine into a 4K iiyama monitor in the lab, which has a 15-pin VGA input, as well as the ability to change the aspect ratio to 4:3 in the OSD menu. You'll want the latter feature if you don't want your games to look weirdly stretched.

MEMORY LANE
Next up is memory, and you may have two options here – 72-pin SIMMs or 168-pin DIMMs. Single inline memory modules (SIMMs) need to be installed in pairs on Pentium systems, although you could install them singularly on some earlier 486 PCs. You'll have a choice of fast page non-parity or EDO memory – either should work fine, but EDO will be slightly quicker.

Meanwhile, 168-pin dual-inline memory modules (DIMMs) come with either EDO or SDRAM chips (the latter is quicker), and it's fine to install them singularly rather than in pairs, although you won't be able to mix DIMMs and SIMMs together. SDRAM DIMMs also come in a variety of clock speeds to match the front side bus. As with today's memory, the fastest DIMMs can slow down to lower clock speeds, so you may as well buy 100MHz (or even 133MHz) memory if you have the option. This will match the bus speed of later AMD K6-II and Pentium II motherboards, and slow down to 66MHz on older systems

For our DOS and Windows 3.1 system, 32MB is plenty, and our CPU uses a 66MHz front side bus, so we're just going with a pair of 16MB EDO 72-pin SIMMs.

FLASH, AAAAHHHHH!
The idea of running your whole PC on solid state storage was a mere fantasy back in the 1990s, but you can easily use flash memory for your main storage system on an old PC now. There are various methods, but we're going to use a CompactFlash adaptor that plugs into an IDE socket and requires a floppy power connector – it cost us just £5.99 on eBay. There are also options that plug into ISA

slots, but the IDE method makes for a system that's easy to set up for booting in the BIOS.

CompactFlash is readily available in a variety of capacities, and its removable nature means you can easily have a few flash cards to boot your system with different options. What's more, you can easily plug a Compact Flash card into a USB card reader, and transfer files from another PC to it, which is much easier than mucking about with slow and unreliable floppy disks.

We're using a 512MB card, but you can go higher. DOS runs on a FAT16 (not FAT32) file system, which means you're limited to using no more than 2GB for a single drive, although you can also partition a larger flash card into multiple 2GB drives with different letters.

Our other two storage devices are an IDE CD-ROM drive and a 3.5in floppy drive, which will connect directly to the motherboard's IDE and floppy controller ports. You'll need cables to attach both of these devices, which are commonly available in ribbon format, but in the early 2000s, some manufacturers started bunching the wires all together to make 'rounded' IDE and floppy cables, so they take up less space. You can still buy these new, and we're using some blue ones here.
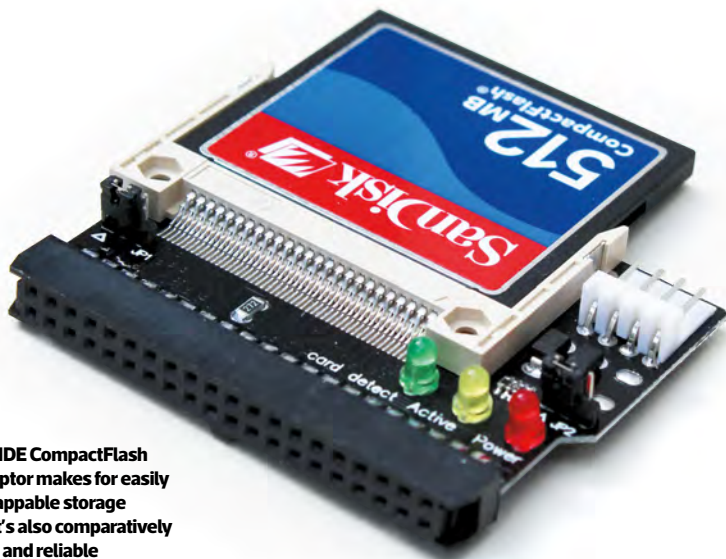
An IDE CompactFlash adaptor makes for easily swappable storage that's also comparatively fast and reliable

ON THE CASE

As we mentioned earlier, you should be able to install an old ATX motherboard into a new ATX case, but if you want to run software from the original media, it will also need front-facing drive bays. You'll need a 5.25in bay for a CD-ROM drive, and a bay for a 3.5in floppy drive if you want that too. A dedicated 3.5in bay can be used for the latter, or you can get a 3.5-to-5.25in adaptor to put a floppy drive in a 5.25in drive bay.

We're using a Fractal Design Define R5, which we had spare in the lab and has two 5.25in bays (we're using an Akasa adaptor to install the floppy drive in a 5.25in bay), but there are other new cases that will do the job. Fractal's latest Define 7 has one 5.25in bay, for example, and the larger XL model has two 5.25in bays. The Pure Base 600 from be quiet! also has two 5.25in front-facing drive bays.

PERIPHERAL VISION

If you have an old ATX motherboard, the rear I/O panel will likely have a pair of 9-pin serial ports (usually used for mice and external modems), a 25-pin parallel port (usually used for printers, but also some scanners and storage devices) and a pair of PS/2 ports (small 5-pin DIN sockets) – one for a keyboard and one for a mouse. You may even have USB ports, but these are useless for DOS.

We recommend using the PS/2 ports for your keyboard and mouse. PS/2 is quicker than serial, and there's a decent range of PS/2 kit available, including optical mice (no one wants to return to using analogue ball mice again, however nostalgic they are!)

You can even get some modern USB keyboards and mice working with old PS/2 ports via adaptors, but you'll need to do your research. A USB peripheral will need to internally support the PS/2 protocol in order for it to work over a USB adaptor, and many of them don't. With a bit of help from Google, you should be able to find out if you can use your USB keyboard or mouse with a PS/2 adaptor. If not, you can buy second-hand PS/2 peripherals cheaply on eBay – it's not as if you're going to need a 4,000dpi sensor to play The Secret of Monkey Island, or even Doom for that matter.

SOUND BYTES

Finally, we come to the sound card. As we mentioned earlier, you want a 16-bit ISA sound card for DOS. The basic standard for DOS games is the Creative Sound Blaster Pro, which combines FM synthesis for music with the ability to play back 8-bit sampled sound – it's great for a game such as Doom, so you get music *and* demon growls, gunshots and explosions. It's also compatible with the first Ad-Lib products, which provided FM synthesis (but no sampled sounds) and are supported in quite a few DOS games, particularly early ones.

There are plenty of non-Creative 16-bit ISA cards that have Sound Blaster Pro-compatibility and can be picked up quite cheaply – just make sure you can get the drivers

Creative's Sound Blaster 16 will happily provide FM synthesiser music in games, as well as sampled sound
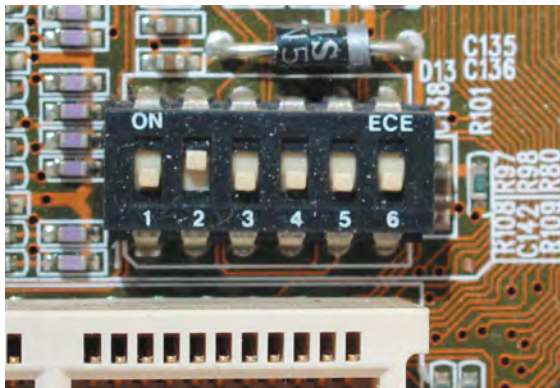
BUILDING TIPS

So you've got all your bits and pieces, and because we're using an ATX case, PSU and motherboard, much of the build process is similar to making a custom PC now. However, there are a few key differences.

JUMPER ROUND

The first warning is that your motherboard's BIOS will be very different from today's user-friendly EFI systems, and even the BIOSes found on boards ten years ago. In fact, for the moment, forget about the BIOS, and instead look at the various jumpers and switches on your motherboard.

Firstly, there may be some DIP switches – plastic blocks featuring several little numbered on/off switches. Secondly, look for jumpers. Jumpers are small sets of pins with movable, conducting tops, which can be swapped around to connect pairs of pins, acting as switches.

Before you change anything, look for any tables printed on your motherboard that outline the position of the switches and what they mean. If you can't find them, try to find your motherboard manual online, so you can see what all the switch settings mean. Getting this wrong can genuinely result in you accidentally overvolting or overclocking your hardware and cooking it. These could be perilous times for PC building!

Now you need to set the switches and/or jumpers to meet the voltage, bus speed and multiplier for your CPU. In the case of our 166MHz Pentium MMX, that means a voltage of 2.8V, a 66MHz front side bus and a 2.5x multiplier. You'll also need to check any jumpers or switches for the memory – some motherboards require a switch to be set to use SIMMs or DIMMs, or to set the memory speed. Triple-check all your switches and jumpers before you install your CPU.

CPU INSTALLATION

Physically fitting the CPU is one area that hasn't really changed over the past 25 years. In the early PC days, CPUs were sometimes soldered into motherboards, rather than using a sockets, and it wasn't until the 486 era

for them. We're using a Creative Sound Blaster 16, which has full compatibility with the Sound Blaster Pro, and can also play and record 16-bit sound.

There are advantages to buying a better sound card, though, as you can massively improve the quality of the synthesiser music in some games. Creative's AWE32 and AWE64 cards have much better synth sounds than the Yamaha OPL2/OPL3 FM synthesiser sounds used by most cards at the time. Again, Doom is a great example of a game that sounds much better with one of these cards.

Another alternative is to use the MIDI interface on the Sound Blaster 16. The early Sound Blaster 16 and AWE32 cards (but not the later ones) had a wavetable daughterboard connector, to which you can attach a secondary synthesiser card. If you can find a Yamaha DB50XG, the sounds are amazing, and there are plenty of other decent-sounding daughterboards too. Once it's plugged into the wavetable connector, your daughterboard will then just output its synthesiser sounds through the line out. This setup will only work on games that can play music data through the MPU-401 interface but quite a few do, and they'll sound better for it.

You can also use the 15-pin joystick port on the back of the Sound Blaster 16 for MIDI, via a 2 x MIDI (5-pin DIN) to 15-pin cable, which you can buy on eBay. A popular external MIDI box at the time was the Roland MT-32, which is supported in some DOS games, including King's Quest IV.

DIP switches are simple on/off switches that tell the motherboard what to do. Our ones are currently set up for a 200MHz CPU, so we'll need to flip up switch 1 to avoid inadvertently overclocking our 166MHz chip

Your motherboard may have tables printed on it, showing how to set your bus speed and CPU multiplier

BUS/CORE RATIO	1	2	3
X1.5/X3.5	0	0	0
X2	1	1	0
X2.5	1	0	1
X3	0	1	0
X4	0	1	1
X4.5	1	1	1

	SW1 1:ON 0:OFF		
BUS CLOCK	4	5	6
50MHZ	1	1	1
60MHZ	1	0	0
66.6MHZ	0	0	0
75MHZ	0	1	0
83MHZ	1	0	1
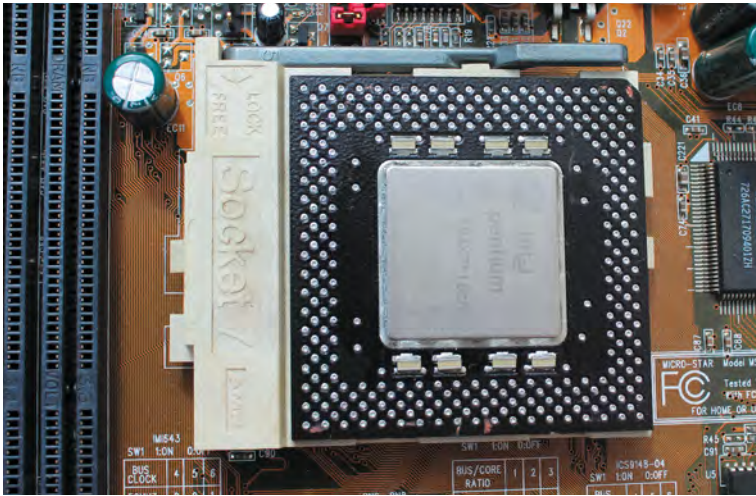
Installing a Socket 7 CPU is very similar to fitting an AM4 chip today

The heatsink will grip the socket with a retention clip, which exerts enough force on the CPU to maintain thermal contact. You'll need to apply thermal paste first

Some low-powered Socket 7 CPUs don't even need a fan, but you'll need one for a Pentium MMX. Ours screws directly into the heatsink fins
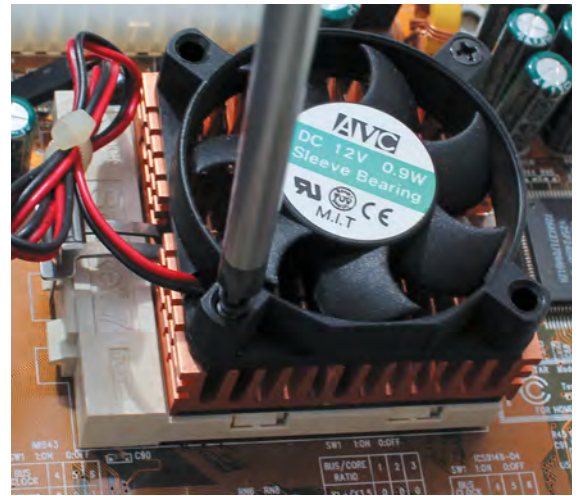
that zero-insertion force (ZIF) sockets really became a standard feature, but we've never looked back since. As such, installing our Pentium CPU is very similar to fitting a modern-day Ryzen chip. You lift up the lever by the socket, and line up pin 1 on the CPU with pin 1 on the socket (denoted by a triangle shape). You can then slot the CPU into its socket and push down the lever to hold it in place.

Next comes the CPU cooler, and this is an area that's changed hugely over the past couple of decades. With a thermal design power (TDP) of just 13W, there's no need for a massive CPU cooler on top of our Pentium MMX CPU, and sub-100MHz Pentiums technically don't even need a fan.

We've picked up a basic Socket 7 heatsink and fan on eBay. Apply a small blob of thermal paste in the middle of the CPU, then put the heatsink on top of it. You then simply clip the heatsink to the hooks on either side of the socket. You can then fit the fan (if it isn't fitted already) and plug in its power cable – the power header is usually next to the CPU socket, but check your motherboard manual.

As a side note, you may notice that you can't see the clips in the later pictures of our PC, which is because we got a bit carried away, and wanted to see if we could

water-cool it for a laugh. It was all fun and games until the force of our custom retention clip for the waterblock ended up pulling a lug off one side of the socket. We ended up having to stick our heatsink to the CPU using thermal adhesive as a result. Don't be like us – be sensible here.
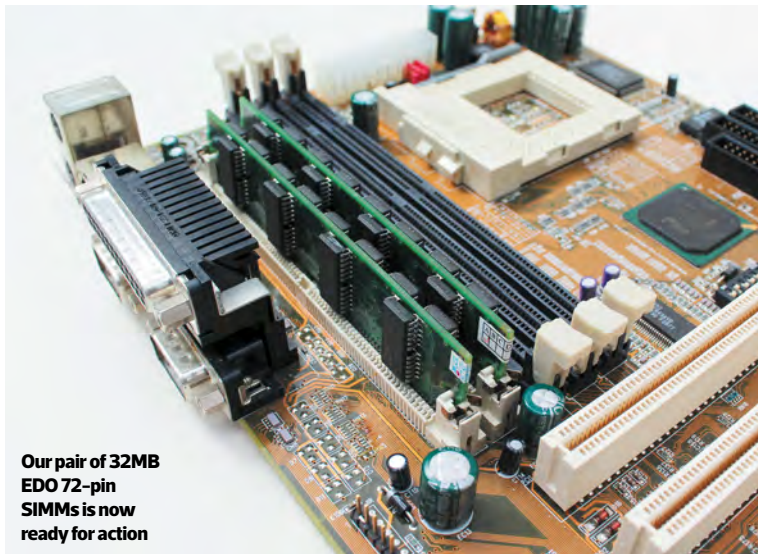
RAM IT HOME

Next comes the memory. If you're using a DIMM, the installation method is the same as today. Push back the clips, put the memory in the slot (it will only go in one way around) and push it down until the clips flip up to grip the memory. SIMMs are a little different, but still simple. To install a SIMM, insert it in the slot in an angle, as shown, then push it back to secure it in place.

Now is a good time to check your hardware works, so plug in your power supply's 20-pin ATX connector (you may need to uncouple a block of four pins next to it) to the motherboard's socket, install a graphics card and hook up a monitor. You'll then need to find the header for the power switch, which will be detailed in the motherboard
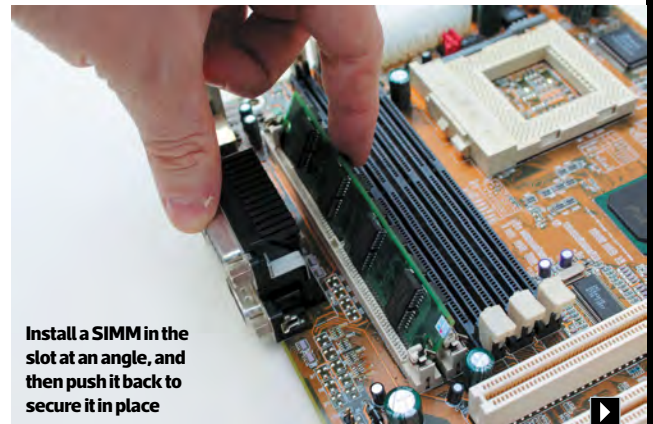
Install a SIMM in the slot at an angle, and then push it back to secure it in place

Our pair of 32MB EDO 72-pin SIMMs is now ready for action

Unless you want to attach two drives to a single cable, set your drive's jumper to Master

manual, or may (if you're lucky) be labelled on the board. Turn on the PSU, and short out the two pins of this header with a screwdriver – you should then get a display on the monitor showing your CPU and clock speed, plus the amount of memory. If not, double-check your components, jumpers and switches.

STANDOFFISH

Next, check the standoffs on your case's motherboard tray. Most of today's motherboards have a standard screw layout, but this wasn't the case in the 1990s, and you don't want to accidentally short out the traces on the bottom of your motherboard. Check that the layout of the standoffs in your case matches the screw hole layout on your motherboard. If it doesn't, move the standoffs to the right places, and remove any that aren't going to be under a screw hole.

After that, put your motherboard's I/O shield in the slot at the back of your case. If you don't have one, you may well be able to get a replacement on eBay, as the standard layout (2 x PS/2, 2 x USB, 2 x serial, 1 x parallel) was pretty universal at this time.

Next, use your cable-routing holes to pass your PSU's 20-pin ATX cable through to the right area of your motherboard. Today's motherboards nearly always have the ATX power connector on the right edge, but they could be practically anywhere on 1990s boards. In our case, it's at the top, so we're passing the cable through one of the top cable-routing holes.

Now is also a good time to route the power cable for your case's exhaust fan through a top cable-routing hole. You can then connect it to one of your PSU's Molex power connectors round the back of motherboard using a 3-pin fan to Molex adaptor. If you want to keep it quiet, you can also put a resistor cable between the two adaptors, which will cut the voltage from 12V to 7V – they cost a couple of quid on eBay.
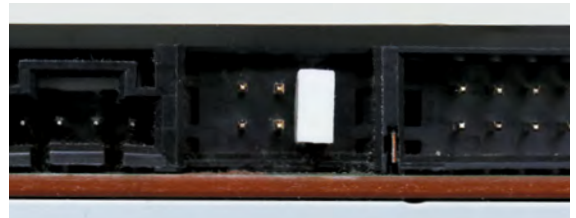
Next, locate your case's front panel connectors. Many of them will be redundant on this system, but the power switch, power LED and reset switches can all be connected – check your motherboard manual to find the location of the headers, and plug them in now. This area is also where you'll be able to attach a PC speaker, usually with a 4-pin header (with two wires). Even if you have a sound card, it's worth having a PC speaker to identify error beeps, and for sound in some older games – if your case doesn't have a speaker, you can pick up a small one for £1.95 from **amazon.co.uk**, which will do the job fine.

Now offer up the motherboard to the case, plug in the 20-pin ATX power connector, gently push the motherboard into its I/O backplate at a slight diagonal (being careful not to scrape it on the standoffs below), then gently lower it down and screw it into the standoffs.

DEEP DRIVE

Next comes storage, which sadly isn't as simple as just plugging in the cables and screwing the drives into the case. IDE cables usually have two connectors, for connecting two drives to one IDE channel, in what was then (politically incorrectly) called 'master and slave' configuration – you wanted your faster drive (such as your boot hard drive) to be the master, and slower drive (such as a CD-ROM) to be the slave.

In our case, we're using two separate IDE channels for each of our IDE drives, rather than connecting more than one drive to the same channel, but we still need to set up our drives properly. On the back of an IDE CD-ROM or hard

Our CompactFlash adaptor, CD-ROM and floppy drive are all connected to the right sockets, with power and rounded cables routed via the cable-routing holes

An Akasa 3.5-to-5.25in bay adaptor enables us to fit a floppy drive in our Fractal Design Define R5. Yay, Lemmings!

ISA cards have the PCB on the other side of the backplate from PCI cards

drive (and on our CompactFlash PCB) will be a jumper, which can be switched to M, S or CS, with the latter standing for 'cable select', although we recommend using the 'master' or 'slave' options for certainty.

Set this to 'M', unless you're running two drives on one cable, in which case set the faster drive to 'M' and the slower drive to 'S'. If you don't do this properly – for example, by putting two 'master' drives on one cable – the system may not boot. You can then connect your IDE cables. There will be two plugs with a short length of cable between them, and a longer cable going down to a third plug. The latter plugs into your motherboards IDE socket (a notch means it can only be fitted the right way around), and the top plug goes into your drive. If you're connecting two drives, the 'top' plug goes to the 'master' drive and the second plug goes to the 'slave' drive.

The ATX power socket could be practically anywhere on 1990s boards

You can route your IDE cables around the back using your cable-routing holes, and you'll also need to route power cables to your drives from your PSU, using Molex or 4-pin floppy connectors. Next comes the CompactFlash adaptor, if you're using one. You just need to make sure its jumper is set to 'master', hook up a floppy power connector and plug it into your motherboard's primary IDE channel socket.

Meanwhile, your floppy drive needs to be connected to your motherboard's floppy controller socket, which looks like a slightly smaller IDE socket. Sadly, most floppy drives don't have a notch to make sure you can only install the cable the right way around, but it's not disastrous if you get it wrong. If you turn on your system, and the floppy drive light is on permanently, and the floppy drive isn't detected, then the cable is the wrong way around – you just need to turn it round the right way.

CARD GAMES

Finally, slot your graphics card and sound card into place – put the graphics card in the top PCI slot, and the sound card in one of the bottom ISA slots. ISA cards have the PCB on the other side of the backplate from PCI cards, with their chips facing the top of the case, but you fit them in the same way. Slot them in place, and secure their backplates with your case's screws.

PLUG IT IN, PLUG IT IN!

All your hardware is now basically installed – the final step is to plug in your keyboard, mouse, mains cable and VGA cable and start it up. Your BIOS should be set to boot from the floppy drive (A) by default, but you'll be able to change the boot priorities in the BIOS by pressing Del when your system starts up.

The next step is to insert your DOS boot disk and boot up your system. We'll give you some tips and tricks for getting your software set up over the page. CPC

One 1990s DOS machine ready for action and, unlike PCs from the time, the interior is nice and tidy too

INSTALL FREEDOS ON VINTAGE HARDWARE

Following our vintage PC building guide, **K.G. Orphanides** shows you how to get a retro PC up and running with FreeDOS

Because MS-DOS 6.22 is increasingly hard to obtain legitimately – your options are old floppies on eBay or an annual Microsoft Visual Studio subscription that costs over a grand – you're better off with a modern open source DOS. We're using FreeDOS, an actively developed MS-DOS-compatible operating system that's sufficiently close to the original that neither you nor your software are likely to notice the difference. You can grab a copy from **freedos.org**

> ## It comes with quality-of-life features, such as PS/2 and USB mouse drivers

It comes with quality-of-life features, such as PS/2 and USB mouse drivers, Tab command completion, file decompression tools and support for FAT32 file systems – it can handle soft reboot and shutdown commands too. Connect your FreeDOS PC to the Internet, and there's an even a package manager, FDNPKG, to help you install and update your system utilities.

```
Free FDISK is capable of using large disk support to allow you to
create partitions that are greater than 2,048 MB by using FAT32
partitions.  If you enable large disk support, any partitions or
logical drives greater than 512 MB will be created using FAT32.

IMPORTANT:  If you enable large disk support, some operating systems
will be unable to access the partitions and logical drives that are
over 512 MB in size.


    Do you want to use large disk (FAT32) support (Y/N).[Y]?
```

Unlike standalone versions of MS-DOS, FreeDOS supports the FAT-32 file system

PARTITION YOUR DISK AND INSTALL DOS

FreeDOS's current stable release is version 1.2, but we recommend using the near-final live CD release candidate of FreeDOS 1.3. From the FreeDOS website, follow the release candidate link to **custompc.co.uk/FreeDOS** and download FD13-LiveCD.zip. Burn it to a CD-ROM, make sure the BIOS on your DOS PC is set to boot from CD before hard disk, insert the disc and boot the machine. If your DOS system doesn't have a CD-ROM drive, there's also an FD13-Floppy image.

At the FreeDOS menu, select Install to harddisk – this works fine on a CompactFlash card, as used in our hardware guide too. If your drive is blank, you'll be asked if you want to partition it. Select Y to automatically partition drive C – the maximum available partition size will be used.

Reboot when prompted and select Install to harddisk again. Erase and format drive C when prompted. Select your keyboard layout and then choose 'Full installation including applications and games'. Confirm your choice, wait for installation to complete, eject the CD and reboot. If you need more control of your disk partitioning, instead select Use FreeDOS 1.3 in Live Environment Mode and type FDISK at the command prompt.

Unlike MS-DOS, FreeDOS supports FAT32, which means you can have hard disk partitions bigger than 2,047MB. If your disk is 2TB or larger, you'll be asked if you want to enable FAT32 support. Click Yes here, unless you specifically want to create multiple smaller partitions that are backwards compatible with older versions of MS-DOS.
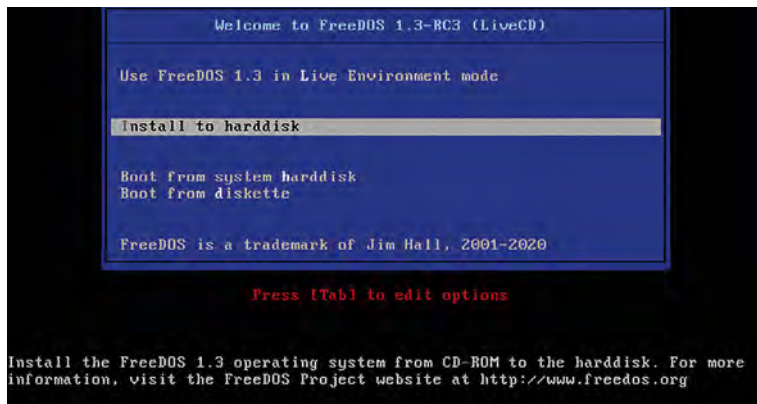
USING DOS

DOS is a command line operating system, and if you've ever used Windows' cmd, it will feel familiar. It's case-insensitive: commands, paths and file names don't have to be typed in UPPER CASE but are often styled that way. To run an

The FreeDOS 1.3 RC3 live disk makes testing, formatting and installation a convenient menu–driven affair

executable file – which will typically have a .COM, .EXE or .BAT extension – just type its name without the extension. File and directory names are limited to eight characters and extensions to three, with longer names curtailed with a tilde (~). When you're finished with DOS, you just turn off the computer. Some older programs don't even have the option of quitting back to the command line.

EDITING DOS CONFIG FILES

As it loads, DOS looks for specific user instructions in files traditionally known as AUTOEXEC.BAT and CONFIG.SYS, in the root of your boot drive, whether that's a floppy or your C:\ partition. As we're using FreeDOS, these are actually called FDCONFIG.SYS and FDAUTO.BAT. FreeDOS includes a selection of useful drivers, such as ones for mice and CD-ROM drives, and these are already called in its boot files.

The easiest way to create or modify these files, assuming you're using a CompactFlash or SD card for your hard disk,

is to simply mount your drive on your usual PC with a card reader and copy in the lines you need using a GUI editor, such as Notepad in Windows. If you prefer to write or edit config files under DOS, just use FreeDOS' EDIT command for a very capable MS-DOS editor with mouse support. If you want to comment on a line, put the word 'rem' in front of it. This is handy for troubleshooting and working out exactly what lines you need in your boot files.

DRIVERS

Although FreeDOS has some integrated drivers, you'll still need the manufacturers' drivers for your sound card, possibly your graphics card, and any non-standard interfaces or unusual input devices, such as specialist joysticks and Zip drives.

Your first stop for driver sourcing should be Vogons Drivers (**vogonsdrivers.com**), a spin-off of the popular and infinitely helpful Vogons retro gaming message board. The drivers generally come with full instructions and examples of the lines you'll need to insert in boot-time config files. Another useful collection of hardware drivers, this time with a focus on storage devices, can be found at Hiren & Pankaj's Homepage (**hiren. info/downloads/dos-files**).

FreeDOS' default FDAUTO.BAT file includes the most common SET BLASTER address line for Sound Blaster compatible cards. This will be enough in many cases, but you may still have to add the path to the actual driver yourself, as well as assigning your own MIDI settings. For example:

```
SET BLASTER=A220 I5 D1 H5 P330
SET MIDI=SYNTH:1 MAP:E
SET SOUND=C:\DRIVERS\SB16
```


The OS comes with FreeDoom, but real Doom works well too

```
1 - Load FreeDOS with JEMMEX, no EMS (most UMBs), max RAM free
2 - Load FreeDOS with JEMM386 (Expanded Memory)
3 - Load FreeDOS low with some drivers (Safe Mode)
4 - Load FreeDOS without drivers (Emergency Mode)

 Select from Menu [1234], or press [ENTER] (Selection=1)

 Singlestepping (F8) is: OFF
```

Pre-defined startup menus provide commonly required memory configurations, but you can add your own too

Under DOS, you'll generally have an easier time of configuration if you stick with ISA cards, although we got the PCI Sound Blaster Live! 5.1 from 2000 working with some tweaking of its driver's CTSYN.INI file. If you run into IRQ or DMA conflicts, check your motherboard's bios settings – if in doubt, disable on-board components such as unused parallel and serial ports, and – especially if you're using PCI components – disable Plug and Play and enable Legacy Mode.

Graphics drivers were far less important in the DOS era than now: if your card supports the VGA display mode, you can select VGA from your game's installation options and it will work. In some cases, such as with the 3dfx Voodoo range of 3D graphics cards, DOS games that supported them would come bundled with the relevant driver – GLide in the case of Voodoo. However, you may need to copy your own more recent copy of the driver file into the game's directory – we copied the glide2x and glide3x DLL and OVL files from our Voodoo 3 3500 TV's Windows driver disk and it worked fine.

TRANSFERRING DATA

If you're using a CompactFlash or SD card-based DOS drive and have a reader connected to your PC, you can just mount your entire DOS drive under your normal Windows, Linux or macOS operating system and copy any files you want to it. This convenient approach makes it easy to get retro games you've bought on **gog.com** or Steam onto your DOS drive – we tried this with the Steam versions of Quake and Ultimate Doom, and both games worked fine on our retro machine.

Alternatively, you can burn a load of DOS software to a data CD and transfer it the old-fashioned way. However, if you're using standard IDE hard disks, or you don't want to routinely open your DOS PC to load up its hard disk, you might want to add USB mass storage support to FreeDOS.

If your motherboard has the common UHCI-compliant host controller, then you're in luck, as FreeDOS includes Bret Johnson's USBDOS drivers (**bretjohnson.us**). We recommend just invoking them as needed to keep memory consumption down, rather than loading them in FDAUTO.BAT.

If your vintage system only has an OHCI controller, or if you're using a newer motherboard with an EHCI USB chipset, then you'll need Panasonic's multi-chipset USBASPI driver (**custompc.co.uk/USBASPI**) and use the Motto Hairu USB Mass Storage driver (**custompc.co.uk/Hairu**) to mount your disks. To add an OHCI controller in FDCONFIG.SYS, add the following lines, modifying the driver paths as appropriate:

```
DEVICE=C:\DRIVERS\USBASPI1.SYS /V /O
DEVICE=C:\DRIVERS\di1000dd.sys
```

USB drives must be plugged in at boot time to be accessible.
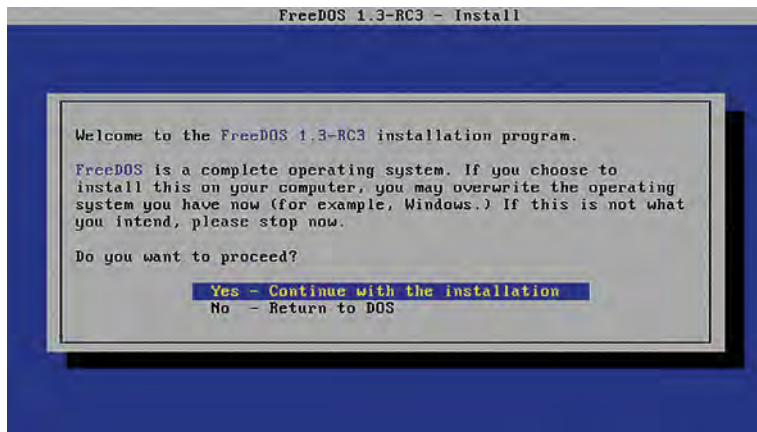
CPU THROTTLING

If you're using a 500MHz PC from 2000 to run games from 1991, your processor will make older clock cycle fixed software run impossibly fast. FreeDOS includes the SLOWDOWN tool to counter this problem.

For Origin's Martian Dreams, for example, with an executable called MARTIAN.EXE, we just typed SLOWDOWN MARTIAN in its directory. You can then reduce speeds by pressing Ctrl and Alt together until you get the speed with which you're happy.

MEMORY MANAGEMENT

The classic DOS games came from a time when only 640KB of conventional (or base) memory could be directly used in MS-DOS 'real mode'. Even then, that was a tiny amount of

KNOW YOUR FREEDOS COMMANDS

DIR
List everything in the current directory. FreeDOS by default applies the /P command extension to pause when the screen is filled. Press space to see more.

DIR /W
Show filenames and extensions only, in a columnated list.

X:
Change to specified drive letter, swapping 'X' for the letter of the drive you want to access.

CD PATH
Change Directory to the specified directory name or path, replacing PATH with the name of the directory you want to access.

CD..
Move back to previous directory

**CD **
Move to top level directory

MD NAME
Make a Directory called NAME

COPY X:\PATH\ X:\NEW\PATH
Copies files and directories from one place to another

MOVE X:\PATH\ X:\NEW\PATH
Moves files and directories to a new location

EDIT
The friendliest DOS text editor

RESET
You don't have to type reset to reboot your PC, but FreeDOS gives you the option

SHUTDOWN
Another optional FreeDOS command for the comfort of modern computer users

FDISK
DOS partitioning tool

FORMAT X
Formats the specified drive (replace X with the appropriate drive letter). This will erase its contents and ready it for use with DOS

You can usually install straight from the CD without any fuss, but you can run FDISK from the live OS if you need more control over drive partitioning

RAM with which to play, so methods of increasing available memory were rapidly introduced. These included a 64KB high memory area, expanded memory of up to 32MB (EMS) and extended memory of up to 4GB (XMS).

In FreeDOS, these memory areas are controlled by HIMEMX, JEMMEX, and JEMM386, which are invoked in FDCONFIG.SYS.

From the late 1980s onwards, most games included installers

To free up extra memory, DOS users traditionally have to juggle extended memory management tools, load drivers into the high memory area, and winnow out unnecessary drivers until there's enough memory available to load your desired application.

If you're using an IDE CompactFlash reader for your retro machine, you can plug it into a card reader on a Windows PC to easily copy and edit files for your OS

As an alternative to using the old-school boot floppies that most gamers had at the time, we're going to use FreeDOS' integrated startup menu system.

FreeDOS has already done a lot of the work for us here, creating high memory and JEMM386 expanded memory startup options.

If you need the maximum amount of conventional memory available, select option 1, Load FreeDOS with JEMMEX, no EMS (most UMBs) and max RAM free, which nets us 643KB of available conventional memory.

If you're running one of the many 1990s games that require EMM386 expanded memory (their manuals will tell you if they do), you want option 2.

MAKE YOUR OWN BOOT MENU

In FDCONFIG.SYS, a MENUDEFAULT section defines four numbered startup menus. We can add an extra option 5 like this:

```
MENU 5 - SB LIVE (JEMM386, HIMEM, NO USB)
```

In the same file, you can add specific lines to a chosen menu number by putting the number(s) and a question mark at the start of the line. For example, putting '125?' before a line means it will be included in boot options 1, 2 and 5 – we've added '5?' to lines that call HIMEMX, JEMM386 and FDAUTO. BAT to include those features in our new menu option.

In FDAUTO.BAT, a quick way to load drivers that only apply to your new menu option is to insert an 'if not' block just before :FINAL at the bottom. For example, the following lines enable a PCI Sound Blaster Live! if we select menu option 5, but skips straight past it if we select any other menu option:

```
IF NOT "%CONFIG%"=="5" GOTO FINAL
:SBLIVE
SET MIDI=SYNTH:1 MAP:E MODE:0
SET BLASTER=A220 I5 D1 H5 P330 T6
SET CTSYN=C:\DRIVERS\SBLIVE\DOSDRV
C:\DRIVERS\SBLIVE\DOSDRV\SBEINIT.COM
```

INSTALL A GAME

Software installation is usually blissfully simple under DOS. From the late 1980s onwards, most games included installers, so you just need to insert your install CD or floppy, go to its drive letter (for example, type 'a:' at the C prompt to go to your floppy drive) and run the installer, usually called INSTALL or SETUP.

You'll probably be asked to select your graphics mode, sound card and choose an install location – this should be drive C. The installer will copy over its files and tell you what you need to run to play the game. You may need to do some disk swapping during this process, and games with CD audio will also require the disc to be in the drive while you're playing the game. Some games don't have installers, but if you copy all their files into a directory on your hard disk, you can usually run them from that location. **CPC**

EMULATE DOS ON RASPBERRY Pi

K.G. Orphanides shows you how to use the powerful DOSBox-X emulator to boot Raspberry Pi to DOS, and run anything from Windows 3.11 to classic games

Graphical user interface? Pah, luxury! When us PC gamers were young, we had to type in text at the DOS prompt. Of course, you can already buy some classic DOS games readily from GOG and Steam that have been tweaked to run from Windows 10. However, if you want to get the authentic DOS experience, where you have complete control over your system, you have to either run an emulator or build a machine based on old hardware.

We're going to take you through the latter next month, where we'll show you how to build a machine that natively runs DOS games. Another alternative, though, is to use an emulator, and Raspberry Pi makes a great platform for this if you want to make a dedicated machine that boots straight into DOS, particularly because of its low cost.

The extra oomph of the 4GB or 8GB edition of Raspberry Pi 4 provides plenty of power for emulating classics of the past in DOS, and that even goes as far as installing and running early versions of Windows. In this tutorial, we'll show you how to emulate PC software from the DOS era using DOSBox-X. If you don't need DOSBox-X's menus or extra features, though, the standard version of DOSBox 0.74-3 available in the package repository is a handy alternative. Just type `sudo apt install dosbox`. You'll find its config file in `/home/yourusername/.dosbox`

Forked from the original DOSBox emulator (**dosbox. com**), DOSBox-X has more precise hardware emulation, supports a wider range of software, and can effectively run more DOS-related operating systems (up to Windows ME). It also has a sophisticated graphical interface to help you manage tasks such as configuration and virtual disk-swapping. In this guide, we'll show you how to make a Raspberry Pi system that boots straight into DOS.

1 / CREATE YOUR DOS DIRECTORIES
Let's create the directory structure to house the software we're going to run through DOSBox-X:

```
mkdir -p dos/{floppy,cd,games}
```

The **floppy** and **cd** directories will house disk images, and we'll be able to switch between them in DOSBox-X. This tutorial and our template config files presume you'll keep all your DOS files in a `/home/pi/dos/` directory, so be sure to change any paths if you're using a different username or DOS directory names.

While our generic config file should handle most DOS software well on a Raspberry Pi, you can also create separate .conf files for specific programs, in order to better match their requirements and automatically run commands.

2 / TWEAK YOUR GRAPHICS
Assuming you're using a standard 1,920 x 1,080 display with your Raspberry Pi, you'll find some more demanding DOS software struggles at full resolution, particularly if you have DOSBox-X configured to use OpenGL and aspect ratio correction.

On the desktop, open the main menu, go to Preferences and select Screen Configuration. Right click on your display – most likely marked HDMI-1 – and select 1,280 x 720 from the Resolution menu. Running your entire GUI

When us PC gamers were young, we had to type in text at the DOS prompt
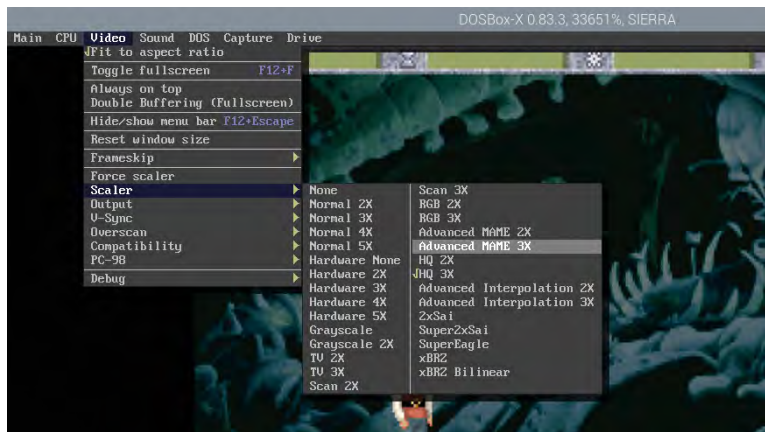
The menu system allows you to easily make changes to your emulated system. You can radically change the appearance and performance of your software by switching scalers on the fly

at a lower resolution will lighten the load of rendering and upscaling for the emulation, and have no adverse effect on games from an era when 640 x 480 (or often 320 x 240) was the norm.

3 / INSTALL DOSBOX-X

In a Terminal, enter the following:

```
sudo apt install automake libncurses-
dev nasm libsdl-net1.2-dev libpcap-dev
libfluidsynth-dev ffmpeg libavdevice58
libavformat-* libswscale-* libavcodec-*
git clone https://github.com/
joncampbell123/dosbox-x.git
cd dosbox-x
./build
sudo make install
dosbox-x
```

DOSBox-X should open at its Z: prompt. You can't paste commands into it from the clipboard, but there are some modern convenience features. Pressing Tab will auto-complete, you can scroll through your command history using the Up arrow, and you can add startup commands to a config file. Type exit to quit and ensure that the config directory, which we'll need in the next step, is created properly.

One of DOSBox-X's key advantages is a graphical interface that covers each element of your emulated PC's configuration, from CPU emulation to scaler
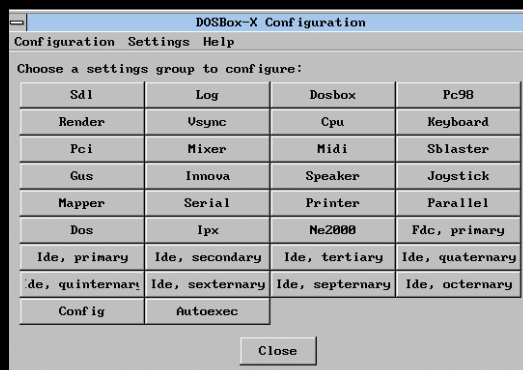
4 / EXPORT A CONFIG FILE

Restart DOSBox-X and tell it to generate a config file that we can later modify in a text editor, based on the program's default settings, and then exit.

```
CONFIG.COM -all -wcd
exit
```

The file we've just made can be found in **/home/pi/.config/dosbox-x** and, at the time of writing, is named **dosbox-x-0.83.3.conf.** As well as being human-readable and conveniently editable in a text editor, you can modify this long and extensively commented file from within DOSBox-X using the configuration GUI in the main menu. This is handy, as DOSBox-X's configuration has more options than that of vanilla DOSBox.

5 / CUSTOMISE YOUR CONFIG

For this tutorial, we've created some config files that you can download from **custompc.co.uk/github**. The code box will run most DOS software. As well as editing your main DOSBox-X config, you can launch DOSBox-X with a specific config file – useful if you wish to easily switch between different OS setups – using the following command-line switch:

```
dosbox-x -conf yourfile.conf
```

We'll take advantage of that later to help install Windows 3.11. Note that your custom config files only need to include lines that vary from the defaults. In the following steps, we'll create a config file optimised for playing late-era DOS games on Raspberry Pi 4 with 4GB or 8GB RAM.

6 / GRAPHICS, SCALERS AND PERFORMANCE

The default config is already well optimised to run DOS software on most systems, but we need to make a few adjustments to improve performance on Raspberry Pi's hardware.

Leave the fullscreen setting as false, as you can enable and disable fullscreen mode using DOSBox-X's menus, or the F12+F keyboard shortcut; fullresolution should be left as 'desktop'.

To get proper aspect ratio correction and reasonable graphical fidelity at 1,280 x 720, set the output to 'opengl', aspect to 'true' and select a scaler for interpolating low-res graphics. Your scaler choice is largely a matter of personal taste, so use the Video menu options to try a few. If your sound becomes choppy, you're pushing Raspberry Pi's capabilities too far.

7 / AUTOEXEC.BAT

At the end of the config file is autoexec, where we'll put all our MOUNT and IMGMOUNT lines to assign drive letters

To improve performance, change Raspberry Pi's desktop resolution to 1,280 x 720
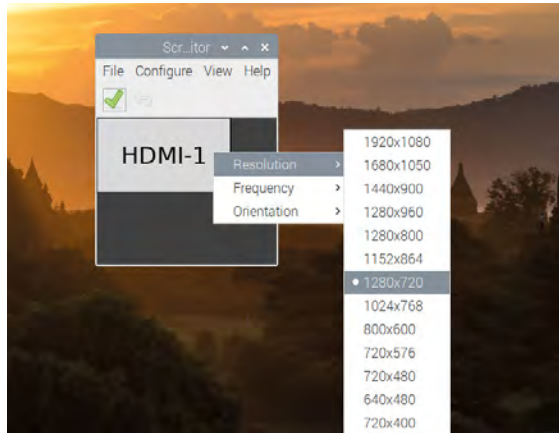
to directories and floppy or CD images, as well as any commands to run at boot.

In our sample config, we've used MOUNT to set **/home/ pi/dos** as the C drive in DOS. We'll copy and install all our software to this location. If you use the IMGMOUNT command with multiple file names of CD or floppy images, you'll be able to swap between those images in order to swap between media. To swap floppies, use **F12+LEFT-CTRL+D.** To swap CDs or DVDs, use **F12+LEFT-CTRL+C**.

8 / USING DOSBOX-X

Like DOSBox, DOSBox-X uses the open-source FreeDOS operating system, rather than Microsoft's proprietary

pi-dos.conf

> Language **DOSBOX-X CONFIG FILE**

```
001.   # Basic DOSBox-X config for 90s DOS software o
       Raspberry Pi.
002.   # See default config file and https://github.com/
       joncampbell123/dosbox-x/wiki for further documentation
003.
004.   [sdl]
005.   # set fullscreen true if you want to boot to an
       authentic-feeling DOS environment
006.   fullscreen = false
007.
008.   # Don't forget to set Raspberry Pi's desktop resolution
       to 1280x720
009.   fullresolution = desktop
010.
011.   # opengl allows aspect ratio correction
012.   output = opengl
013.
014.   [render]
015.   # set frameskip to 1 or 2 for resource-hungry titles
016.   frameskip = 0
017.
018.   # aspect ratio correction
019.   aspect = true
020.
021.   # choose your favourite. Don't use scalers on games
       that already have high resolutions. Set scaler to none
       to improve performance.
022.   scaler = advmame3x
023.
024.   [cpu]
025.   # use normal core for multitasking OSes such as Win95

026.   core = dynamic
027.
028.   # some software benefits from emulating a specific CPU,
       which can be specified here
029.   cputype = auto
030.
031.   # if you experience lag or juddering audio, set CPU
       cycles to max.
032.   cycles = auto
033.
034.
035.   [autoexec]
036.   # Your DOS autoexec.bat file. These commands will
       be run at startup, making it easy to mount lots of
       floppies or CDs at once, as well as your working
       directories.
037.
038.   mount c /home/pi/dos/
039.
040.   # uncomment and customise these lines to mount floppy
       and CD images. Remember that DOS isn't case sensitive,
       but Linux is.
041.
042.   # imgmount a "/home/pi/dos/floppy/disk1.img" "/home/pi/
       dos/floppy/disk2.img" "/home/pi/dos/floppy/disk3.img"
       -t floppy
043.   # imgmount e "/home/pi/dos/cd/a directory with spaces
       in/sherlock.iso" /home/pi/dos/cd/quake/QUAKE101.cue  -t
       iso -fs iso
044.
045.   c:
```

You may find that the interface of Windows 3.x feels rather alien

MS-DOS, although you can install and run MS-DOS from a disk image if you own a copy.

Navigation through DOS directories isn't too different to using a Bash terminal, particularly as a number of Bash commands have been included, such as LS as an alternative to DIR in DOS. To run a .com, .exe or .bat file, just type its name without the extension.

To capture and release your mouse, use the LEFT-CTRL+F10 shortcut. The autolock entry under SDL config enables capture-on-clock.

9 / WINDOWS 3.11
Now we're going to install Windows for Workgroups 3.11, released in December 1993. The biggest challenge is finding a copy of Windows 3.11 to install – that usually

means aging floppy disks, or disk images if you had the foresight to make backups. We're working from a set of disk images.

If you don't already have one, and don't fancy the second-hand market, you can, surprisingly, find it included in Microsoft Visual Studio Subscriptions (formerly MSDN Subscriptions), currently priced at £33.54 per month, for the benefit of developers working on backwards compatibility,

Windows 3.11 will cheerfully run either on top of DOSBox-X's default FreeDOS operating system, or installed with DOS 6.22 on a dedicated hard disk image

10 / INSTALL WINDOWS
Copy the contents of each installation disk or image to a /win311 subdirectory of the dos directory tree we made earlier; you can do this as you normally would on the desktop or at the command line, or by using DOSBox-X's IMGMOUNT to mount them and using the DOS COPY command while switching disks. At the command line, start DOSBox-X with a Windows-suitable config file – download ours from **custompc.co.uk/PiWin**

```
dosbox-x -conf win311.conf\

CD WIN311
SETUP
```

Windows 3.11 will install itself. Reboot.

RPG classic Worlds of Ultima: Martian Dreams is legally available for free from GOG.com, but you'll have to use innoextract 1.8 (constexpr.org/innoextract) to pull the files out of it

```
CD WINDOWS
WIN
```

11 / USING WINDOWS 3.X
If you've only ever used Windows 95 or later, the interface of Windows 3.x may feel rather alien. There's no Start button, and if you want to quit back to the DOS prompt, you have to open Program Manager's File menu and select Exit Windows.

The default Program Manager folders, each of which are full of shortcuts to helpful software and settings, are clearly labelled. To explore your mounted DOS drives, open Main and then File Manager. Accessories include MS Paint (precursor Paintbrush), a Sound Recorder and even a Media Player. A line at the top left of each opened window allows you to move and close it, and you'll find minimise and maximise buttons on the top right of each window.

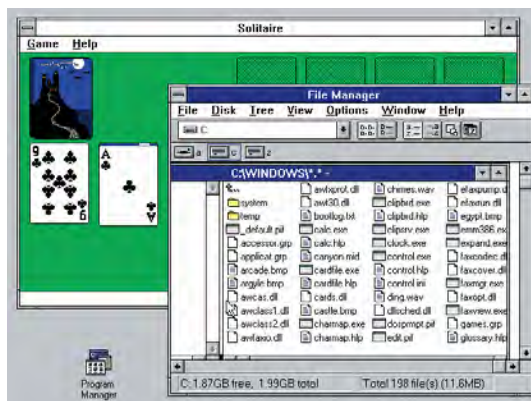## 12 / BOOT RASPBERRY PI TO DOS
Once you've configured DOSBox-X – and any relevant window managers – to your satisfaction, you can complete your pitch-perfect 1990s PC simulation by booting straight to DOS. Open a Terminal window and type:

```
mkdir /home/pi/.config/autostart
mousepad /home/pi/.config/autostart/
dosbox.desktop
```

Add the following to the new text file:

```
[Desktop Entry]\
Type=Application
Name=DOSBox
Exec=/usr/bin/dosbox-x
```

This will use DOSBox-X's default config file. You'll need to enable fullscreen in your DOSBox-X config for this to launch correctly, and we strongly advise enabling opengl-dependent aspect ratio correction. **CPC**