

History

In 1981, the C64 was not trying to be a C64, it was just trying to be a computer. Today, the C64 is trying to be a C64; it's a simulation – an emulator. The C64 Ultimate is an emulator running on an AMD Xilinx Artix-7 FPGA. It's an official Commodore product, and it does a *very good* job of being a C64. If no one said anything, no one would know the difference.

The SD-8516 is also an emulator. An emulator for a computer that never existed. It's not trying to be a 6502, a Z80 or a 68000. It's just trying to be a computer.

"This is a computer that never existed. But if it had, it would have changed the world." -DatCube82

1972: The 8080

The standout "famous chip" of the early 70s, the Intel 8008 and 8080 were widely regarded as the world's first commercially available microprocessors. The 8008 in 1972 was the first 8-bit microprocessor, an important step up for handling text/characters (not just numbers). It saw some use in early terminals and systems. By 1974, just before the 6502, the Intel 8080 became the most popular and influential 8-bit microprocessor of the immediate pre-6502 period. It was much more capable and general-purpose than earlier chips. It was the brain of the MITS Altair 8800 (1975), often credited as sparking the personal computer hobbyist revolution (the "first" widely recognized microcomputer kit).

The 8080 quickly became the go-to "hot" chip for anyone building serious 8-bit systems before cheaper alternatives like the 6502 and Z80 disrupted the market.

The Motorola 6800 (also 1974) was another major contender. A clean 8-bit design that the 6502 was actually derived from (by ex-Motorola engineers who left to make a simpler, cheaper version). It was respected but considered expensive by the hobbyist scene.

1975: MOS-6502

The MOS Technology 6502, released in 1975, is arguably the most influential 8-bit microprocessor ever created. Designed by Chuck Peddle and a small ex-Motorola team, it was deliberately engineered to be dirt-cheap (volume price ~\$25 vs. \$179 for the Intel 8080) while still delivering excellent performance. This radical cost breakthrough, combined with a minimalist yet highly efficient design, sparked the home-computer and video-game revolutions. It powered the Apple I/II, Commodore PET/VIC-20/64, Atari 2600/400/800, BBC Micro, and (in a custom form) the original Nintendo NES.

Core specs: True 8-bit CPU with 16-bit addressing (64 KB), only three general-purpose registers (A, X, Y), a dedicated 256-byte stack page, and exceptionally powerful zero-page indexed addressing modes that effectively gave it "extra registers." Typical clocks were 1-2 MHz ***(up to 4 MHz in later binned parts)**, with a tiny transistor count (~3,500) and superb interrupt handling.

The SD-8516 is the modern spiritual successor built specifically for the 6502/C64 vibe but taken to the

next level. It's a clean-slate 16-bit load/store architecture (16 × 16-bit general-purpose registers, 24-bit addressing) running at a locked 4 MHz – exactly 4x a stock Commodore 64 (2x a C128). While not a direct silicon descendant of the 6502, it deliberately captures the same “hackable, low-level fun” philosophy while adding modern conveniences: browser-based virtual hardware, up to 320×200 configurable graphics, Stellar BASIC, and an extremely fast Forth implementation (Star Forth / SD/FORTH, reportedly ~150,000 words/second). It's the “PDP-11/70 equivalent” for the 6502 world – the high-end dream machine that 1980s hobbyists would have killed for, now available for retro enthusiasts, BASIC aficionados, and Forth hackers. It keeps the direct, immediate programming feel of the classic 6502 but with 16-bit power, expanded memory, and zero hardware cost.

1975: DEC PDP-11/70

The DEC PDP-11/70 (1975) was a high-performance 16-bit minicomputer with a ~5 MHz (300ns-333ns) microcycle CPU, roughly 3x-4x faster than a PDP-11/34A. It utilized a 2 KB cache, a fast 32-bit memory path supporting up to 4 MB of memory, and dedicated Massbus controllers, delivering approximately 1230 Dhrystone MIPS (roughly 0.7x VAX-11/780).

The SD-8516 at 4 MHz equivalent, with its register-rich architecture and eight-cycle MUL, lands squarely in early-80s minicomputer territory. A PDP-11/70 cost \$350,000 in 1975. Today, the SD-8516 matches it as a single-board microcomputer. That's the same revolution the 68000 represented; minicomputer power on a desktop.

1976: Z80

The Zilog Z80, launched in 1976 by Federico Faggin (one of the Intel 4004/8080 designers), was designed as a superset of the Intel 8080 with full binary compatibility plus a huge list of new instructions, extra registers, better interrupts, and built-in DRAM refresh. It was an 8-bit CPU with a 16-bit address bus but felt much richer thanks to dual register banks and powerful block-move instructions. Clock speeds ranged from 2.5 MHz up to 8+ MHz. CP/M, the dominant business OS of the late 1970s/early 1980s and forerunner of DOS, ran almost exclusively on Z80 machines.

It dominated European and business microcomputing and remains one of the longest-lived 8-bit designs.

These CPUs, as well as the SD-8516, perfectly illustrate “keep it simple, make it fast and cheap, and let programmers have fun”. The Z80 produced wildly different yet equally legendary platforms. Business and gaming. The SD-8516 carries that same joyful spirit into the future for a new generation of retro hackers.

1978: Intel 8086

The Intel 8086, introduced in 1978, was designed as a stopgap. Intel's real ambition was the iAPX 432, a radical 32-bit object-oriented processor. The 8086 was a quick 16-bit extension of the 8080 architecture meant to hold the line against the Z80 and the looming 68000. It was never supposed to define the future of computing. Then IBM chose the 8088 (its 8-bit bus variant) for the original IBM PC in 1981, and the architecture became inescapable.

The 8086 runs at 5-10 MHz with a 16-bit data bus, a 20-bit segmented address space (1 MB, accessed through four segment registers that shift and add to produce physical addresses), and four general-purpose 16-bit registers (AX, BX, CX, DX) that can be split into eight 8-bit halves. It also has four segment registers, a stack pointer, base pointer, and two index registers. The instruction set is notoriously irregular; certain operations can only be performed with specific registers (CX for loops, AX for multiply, DX:AX for 32-bit results). The segmented memory model, where every address is computed as segment x 16 + offset, was the defining headache of PC programming for over a decade. Hardware multiply exists but is painfully slow at 118-133 cycles for a 16×16 operation. The 8086 at 4.77 MHz achieved approximately 300 Dhrystones — roughly 0.17 DMIPS.

The 8088 variant in the IBM PC was even slower, with an 8-bit external bus that halved memory bandwidth. Most benchmark comparisons of the era showed the 8 MHz 68000 outperforming the 4.77 MHz 8088 by a factor of 5-7x on real workloads. The 8086/8088 won the market not on technical merit but on IBM's brand, the open architecture that enabled clones, and the CP/M software migration path through MS-DOS.

At nearly identical clock speeds, the SD-8516 delivers more than 5x the processing power of the 8086. The reasons are architectural, not just clock-for-clock. The 8086's segmented memory model imposes constant overhead. Every far pointer access requires loading a segment register, and crossing a 64 KB boundary demands segment arithmetic. The SD-8516's flat 24-bit addressing eliminates this entirely. The 8086's register specialization (only CX can count loops, only BX can index memory in certain modes, multiply must use AX) forces constant register shuffling. The SD-8516's orthogonal register file has no such constraints.

The multiply gap is the widest of any comparison. The 8086's MUL takes 118-133 cycles – over 25 microseconds at 4.77 MHz. The SD-8516 can complete the same operation in under eight cycles at 4 MHz. That is a 20:1 advantage per multiply. For workloads like prime number generation, encryption, or any game physics involving scaling or rotation, this single difference transforms what is computationally feasible in real time.

Forth on the 8086 was notoriously awkward. The segment registers interfere with the dual-stack model, the register specialization prevents clean assignment of dedicated stack pointers, and the 16-bit cell size limits numeric range. Estimates place 8086 Forth at roughly 15,000 words/sec at 4.77 MHz. The SD-8516's 155,000+ words/sec represents a 16:1 advantage; a gap so large that workloads impractical on the IBM PC (interpreted game logic, real-time procedural generation) become routine on the SD-8516.

In the broader context: the IBM PC succeeded despite the 8086, not because of it. The SD-8516 represents the road not taken; a clean, flat-addressed, register-rich architecture designed for programmer productivity rather than backward compatibility with the 8080. In an alternate 1982 where Stellar Dynamics competed with IBM, the VC-3 would have been the machine serious programmers wanted, just as the Amiga and Atari ST were the machines serious programmers wanted in the real 1985. The market chose otherwise, but the architecture speaks for itself.

1979: Motorola 68000

The 68000 was the chip that brought minicomputer power to the desktop. Introduced by Motorola in 1979, it was a quantum leap over every 8-bit processor on the market. With a 32-bit internal architecture (though externally 16-bit), eight 32-bit data registers, seven 32-bit address registers, and a clean orthogonal instruction set, the 68000 was widely considered the finest microprocessor

architecture of its generation. It powered the original Apple Macintosh, the Commodore Amiga, the Atari ST, the Sega Genesis, and nearly every Unix workstation of the early 1980s.

The original 68000 ran at 4 MHz with a 24-bit address bus (16 MB), hardware $16 \times 16 = 32$ multiply (MULU/MULS, 38-70 cycles), and hardware divide (DIVU/DIVS). Its instruction set was inspired by the DEC PDP-11 minicomputer and offered addressing modes that made it exceptionally pleasant to program in assembly. The chip achieved approximately 2,100 Dhrystones at 8 MHz – roughly 1.2 DMIPS – making it competitive with the DEC VAX-11/780, a minicomputer that cost \$350,000. The 68000 represented the architectural philosophy that your code should be readable and your hardware should be powerful enough to support clean abstractions. It was, and remains, the gold standard for Forth implementations.

The SD-8516 and the 68000 share a remarkable number of architectural features: 24-bit addressing, dual stack pointers, pre-decrement/post-increment addressing modes, and a register-rich design philosophy. Both chips were designed to run Forth exceptionally well, though the 68000's designers had broader goals in mind. The SD-8516 matches the 68000's Dhrystone performance at half the clock speed. This is primarily due to the SD-8516's sub-eight cycle multiply and divide. The 68000's MULU takes 38-70 cycles (data-dependent), and its DIVU takes approximately 140 cycles. On multiply-heavy code, the SD-8516 holds a 20-40× advantage per operation. However, the 68000's 32-bit data bus and eight full 32-bit data registers give it an advantage in data movement and 32-bit arithmetic sequences.

The SD-8516's composite register system (BLA as a 24-bit pointer combining B and A) is architecturally novel; the 68000 uses dedicated 32-bit address registers instead. The 68000's approach is more straightforward; the SD-8516's is more flexible, allowing any combination from an 8×8 grid of 64 non-colliding 24-bit pointers. Both solve the same problem differently.

In terms of Forth performance, the SD-8516 at 155,000+ words/sec at 4 MHz outperforms what would be expected from a 68000 at the same clock, roughly 90,000 words/sec. The register-in-mnemonic encoding, dedicated data and return stack pointers, and inlined stack primitives give the SD-8516 a Forth-specific advantage that a general-purpose processor like the 68000 cannot match.

The Secret Sauce: A Fast MUL

The SD-8516's legendary *single-cycle MUL* is not based on the latest technology, but technology from yesteryear. It uses a lookup table to perform an 8bit argument to 16 bit result MUL in a single CPU cycle.

The *Quarter-square lookup* uses the formula $xy = ((x+y)^2 - (x-y)^2) / 4$ in a squaring table of just 512 bytes. This allows it to do an 8×8 multiply with three lookups. One such implementation ran in 79 to 83 cycles on a 6502 – much faster than shift-and-add. On a chip with an internal squaring ROM and a dedicated subtractor, this can be done in 3 to 4 cycles.

But 3 to 4 cycles is nothing compared to one. *According to the lore*, Stellar Dynamics bet the bank and licensed a massive order of 128 KB mask ROM die from an unknown Japanese semiconductor partner. Hitachi, Toshiba, Sharp and many others were all doing custom ROM for game cartridges in that era. This ROM, bonded directly into the SD-8516 package, contains a complete 8 bit to 16 bit multiplication table. The MUL microcode concatenates the two 8-bit operands as a 16-bit ROM address and reads the 16-bit result in a single bus cycle. For partial products it then sums them with the internal ALU. A total of no more than 4 ROM lookups and 3 additions. Pipelined, an 8×8 mul into 16

bits could be done in *one CPU cycle*.

In truth, it's not really a single-cycle MUL – that's just a marketing term. It actually can take up to five CPU cycles to multiply a 16 bit number, as it has to shift and add for each additional byte. Still, pound for pound, nothing else could do an 8 bit MUL that fast.

And thus, the SD-8516's legendary “single-cycle MUL” was born.

Table lookup MUL was not new, but it was typically impractical due to cost and die area. Stellar Dynamics' innovation was integrating the lookup ROM on-die at a time when most competitors considered 128 KB of ROM too expensive for a general-purpose processor. The gamble paid off. The SD-8516's legendary “single-cycle MUL” became a signature feature and the reason it found its niche in scientific instruments, embedded systems, arcade hardware, and high-end personal computers where multiply-intensive workloads (3D graphics, signal processing, game physics) justified the premium price.

It's DIV uses a similar but smaller ROM (a reciprocal table, 512 bytes) combined with a Newton-Raphson iteration step. This uses two ROM lookups and one multiply to give a quotient in the same cycle as MUL. This technique was eventually used in the AMD Am29000 and later in the Pentium's floating-point divider; but, *according to lore*, the SD-8516 was first-to-market. You can't stop technology!

1981: The SD-8516

Origins: Stellar Dynamics Epsilon Containment Facility (1980-1982).

In the summer of 1980, three engineers at Stellar Dynamics were given authorization to create the Epsilon Containment facility alongside Black Mesa and Aperture Science, after the G-Man had given each a random sample from another world. The ECF received a sample of 80s protoculture known as “Sample SD-0064”.

Dr. Issac Korr and Dr. Vance Halberg headed the project, reporting to a Dr. Magnusberg. Korr and Halberg had spent two years on the VC-1 (JTD) and VC-2 (SD-8510) projects already, and came from a background writing terminal code for the NW-073 project. They had seen the 6502, the Z80, the PDP-11/70 and the 68000 and they believed these were the “wrong chips at the right time”. And, the wrong chip at the right time, can make all the... difference. In the world.

The 68000 was a general-purpose marvel. Clean, orthogonal, elegant; but it made compromises for breadth. Its 16-bit external bus was a cost concession. Its 38-to-70-cycle multiply was a die area concession. Its instruction decoder, designed to handle every conceivable addressing mode with equal grace, occupied silicon that could have been spent on making common operations faster.

Korr wanted to build a chip that was unapologetically opinionated. Where the 68000 asked “what might a programmer need?”, they asked “what does a programmer actually do, over and over, millions of times per second?” The answer, drawn from profiling Forth interpreters, BASIC runtimes, and early C compilers, was: push, pop, compare, branch, and multiply. Everything else came together from those. The key development was the reception of Sample 64 from the G-Man. Once that had been secured, Magnussberg authorized the construction of the Epsilon Containment Facility and work

began on the high powered lasers needed to scan the sample.

Design Philosophy: The Controversial Choices (1981)

The SD-8516 design began in earnest in January 1981 and was driven by three principles that were, at the time, considered somewhere between unorthodox and reckless.

First: registers are cheaper than memory cycles.

While Intel was spending transistors on segment arithmetic and Zilog was adding prefix bytes to squeeze more instructions from an aging architecture, Stellar Dynamics committed to sixteen 16-bit base registers that could be combined into sixty-four non-colliding 24-bit pointers through a novel composite encoding scheme. The register name was embedded in the mnemonic itself; 'LDAB [ELY]' meant "load the 32-bit AB pair from the address in 24-bit register ELY." Critics called it "alphabet soup." Dr. Vance Halberg became so associated with this that people started calling him "ELY Vance".

The composite register system emerged from Dr. Issac Korr's observation that most programs use pointers and data simultaneously, and that traditional architectures force a choice between them. By allowing any combination of bank byte and address word to form a pointer (BLA, ELY, FLX, GLZ, ILC, TLK - sixty-four combinations from the 8×8 grid - the SD-8516 eliminated the most common reason for spilling registers to memory. A Forth inner loop could hold the data stack pointer, return stack pointer, top-of-stack value, and three scratch pointers all in registers simultaneously. 32 to 24 and 24 to 32 bit transforms were zero cost; you just loaded into AB and used BL:Y as a register, or TK and TLK.

Second: Two stack pointers, not one.

Every major programming paradigm — Forth, C, Pascal, even structured BASIC maintains at least two stacks: one for data, one for return addresses. Every existing processor forced programmers to share a single hardware stack pointer between both, or to dedicate a general-purpose register as a second stack pointer and manage it manually. The SD-8516 made the dual-stack model architectural, with dedicated 24-bit stack pointers for data (ELY) and return addresses (FLX). This was not a new idea — the 6809 had user and supervisor stack pointers, but the SD-8516's pre-decrement store and post-increment load addressing modes ('STA [-,ELY]' and 'LDA [ELY,+]') made multi-stack operations single-instruction primitives.

Third, and most controversially: a single-cycle hardware multiply.

The Multiplication Table: Stellar Dynamics' Gamble

In 1980, an 8-bit multiply took 130 cycles on a 6502, 200 cycles on a Z80, and 70 cycles on a 68000. Even the fastest implementation — the 6809's dedicated 11-cycle MUL — could only handle 8×8 products. The SD-8516 was designed to multiply 32-bit values in a single clock cycle.

The secret was the 128 KB mask ROM die, bonded directly into the processor package, containing a complete 8×8 to 16-bit multiplication lookup table. To multiply two 8-bit values, the microcode simply concatenated them into a 16-bit ROM address and read the 16-bit result in one bus cycle. For 16-bit and 32-bit multiplies, the operation was decomposed into four 8×8 partial products using the ROM and summed by the ALU. The entire sequence was pipelined to complete in a single processor cycle.

A contributing factor was the unrolled microcode used in it's design. While considered sloppy by some, at three times the transistors as a 68000 (and almost 10x those on an 8086) the design choice paid off with the ability to overclock from 4 MHz to 16 MHz and above. Over-engineered to a fault,

some users reported LNG cooled chips running at over 300 MHz.

The 128 KB ROM coupled with a larger die size was, in 1981, outrageously expensive. By 1978 the 6502 cost \$4 or \$5 to make and sold for \$25. The 8086 cost \$10 to \$20 to make; launched at \$87, by the early 80s the price had fallen to under \$20. The 68000 had a similar story; Launched for over \$400, Steve Jobs famously negotiated a mass-purchase for around \$15 per unit.

The straw that broke the camel's back and enabled economic viability was that in 1981, 64K DRAM chips crashed to \$5 in volume (from \$25). The DRAM price crash in 1981 (specifically for the then-new 64K DRAM chips) was driven by a classic semiconductor industry cycle: overcapacity combined with softening demand during an economic slowdown. By the time RAM prices stabilized, the other SD-8516 manufacturing process had improved. Yet the cost to produce one of these chips remained a multiple of any other CPU. Launching at \$189 in 1982, and falling to \$85 by Christmas, a \$795 total package computer system was not unheard of; a direct competitor to the C64's \$595 pricetag. Given the massive difference in power (4 to 10x faster) hobbyist enthusiasts and homebrew hackers flocked to the SD-8516; it was the underground computer scene of the underground computer scene.

Dr. Issac Korr, who designed the multiply unit, later reflected: "Everyone told us it was insane to put 128 kilobytes of ROM and 200,000 transistors on a processor die. Fie! It will take them more than a week before they can coax their MUL instructions out from the ALU."

He was right. The multiply ROM was the chip's signature feature and its single greatest competitive advantage. But it was also the reason the chip nearly died at birth.

Critical success, commercial failure.

The SD-8516, formally announced in March 1982 and available in retail quantities by September, offered the following specifications:

- **Architecture:** 16-bit registers with 8 to 32 bit combined register modes
- **Registers:** 16 base registers forming 64 composite 24-bit pointer addressing modes
- **Address space:** 24-bit flat addressing (4 banks × 64 KB = 256 KB)
- **Data path:** 16-bit internal, 32-bit operations via register pairing
- **Clock speed:** 4 MHz (initial), later 8 MHz and 16 MHz
- **Multiply:** 8×8 to 16 bit 1 cycle (128 KB on-die lookup ROM)
- **Divide:** 8×8 bit 2 cycle (512-byte reciprocal ROM + Newton-Raphson)
- **Stack pointers:** 2 dedicated (Stack and Data Stack)
- **Interrupts:** Programmable interrupt system with vector table
- **Performance:** ~1.2 DMIPS at 4 MHz (equivalent to VAX-11/780)
- **Transistor count:** ~168,000 (including multiply ROM)
- **Process:** 3.5 μm HMOS (Kitsune fabrication)
- **Package:** 64-pin DIP
- **Price:** \$189 (initial, single unit, 1982)

A Savior from Japan

The SD-8516 launched into a market that did not want it.

At \$189 per unit, it cost six times more than a Z80, four times as much as a 68000, and more than

double an 8086. The 128 KB multiply ROM, the chip's greatest technical achievement, was largely seen as useless as every standard game had to run on the lower-class hardware of the era to get sales. No credible game designer wrote anything for the SD-8516. Although, it did see great use in the scientific community, being adopted in cluster configuration to run the laser sample analysis arrays at all three of the top scientific research facilities; Black Mesa, Aperture Science, and Stellar Dynamics.

In the business and banking world, it's overstated power was largely seen as un-needed at the time. System designers who needed fast multiplication were already using the 68000 with acceptable results. Those who needed a cheap CPU bought the Z80 or 6502. The SD-8516 fell into the gap between single-user and enterprise-class time-sharing and never found it's niche.

IBM had quickly standardized the industry around the the 8088 architecture in response. The Macintosh was about to standardize creative professionals on the 68000. Stellar Dynamics had no ecosystem, no software library, and no Fortune 500 patron. The VC-3, their reference personal computer design, was technically impressive – a complete system with banked memory, SID-inspired sound synthesis, and multiple video modes – but it competed against machines backed by companies with million dollar marketing budgets and thousand-man sales teams.

By mid-1983, Stellar Dynamics had sold fewer than 5,000 units total. Their Japanese and Taiwanese investors were losing patience. Dr. Magnussberg reportedly mortgaged his house to make payroll in November 1983.

The turning point was when Dr. “ELY” Vance Halberg flew to Japan to meet with the head of the investor's board of directors, Iroichiri Shimajiro. He basically told them that the 8510 project had failed, and that it was unlikely to make a return on the investment. But, if they did not pay the final \$5 million dollars investment, the company would fold. Korr said in a tele-call that he believed this Chip was the future but they were just to early. He apologized and deeply bowed.

But Shimajiro's response was unexpected. He said, “This is a young man that I liked,” and history was made. Therefore we must acknowledge that without the greats of yesteryear – The Ataris, the Segas and Nintendos, the Origins, the Midways, Commodores, Amigas, Apples, Spectrums, BBC Micros, Tandys and so many more – we would today have nothing. It was their work and sacrifice that showed us the way in the early 80s era of the home microcomputer. It is thus, from their generous donation, that we have been allowed to exist today, and indeed, succeed and thrive!

Audiovisual Magic, but at a price

Shimajiro's investment was not entirely charitable. He was the preside of Namsegdo's Game division and their company was sitting on a glut of 128kb RAM die chips. A deal was made; they were placed into the SD-8516 alongside a new pair of coprocessor chips: the XY-2000 PPU and a relatively unknown prototype chip called the SPC-100. Two boards were to be used for System RAM and one each for the PPU and SPU; this made the SD-8516, on paper, a 512kb system. In addition, Stellar Dynamics had to pay back the cost of each board without raising the price until the debt had been paid. But Shimajiro saw to it that the boards would sell.

The Arcade Pivot (1984-1987)

Thus, what saved Stellar Dynamics was not the personal computer market but the one market where

single-cycle multiply was not a luxury; it was a necessity. The Arcade market. As part of the investment deal, Namsegdo made the SD-8516 the centerpiece of their their high end arcade board system. The bet paid off big time.

Arcade game hardware in 1984 was undergoing a transformation. Sprite scaling, rotation, and pseudo-3D effects required real-time multiplication at rates that brought the 68000 to its knees. A single sprite rotation required dozens of multiplies per frame. At 60 frames per second, the 68000's 38-to-70-cycle MULU became the bottleneck. Game developers were resorting to pre-calculated lookup tables that consumed precious ROM space, or accepting visible slowdown when too many sprites appeared on screen.

But the SD-8516 had solved this problem before it became an issue, and sales began to pick up. Shimajiro's gamble began to pay off; The SD-8516's legendary "single-cycle MUL" could perform sprite transformations in real time using lookup tables, without slowdown, and without the elaborate co-processor arrangements that competitors required. As well, by the mid 80s hackers had discovered ways to do fast memory copy. A single SD-8516 at 4 MHz could transform more sprites per frame than a 12 MHz 68000 and reach speeds of over 120 frames per second even during heavy parallax scrolling.

Comcap-Namiko was the first major licensee, integrating the SD-8516 into a custom arcade board that could do over 6,000 sprites at 30fps in late 1987. Conam co. followed in 1988. By 1990, the SD-8516 was present in over a dozen arcade platforms, and Stellar Dynamics had quietly become profitable. The chip's price dropped to under \$100 as volumes increased.

The Cult Following (1987-1995)

Arcade success attracted a different kind of attention. University computer science departments, which had been using PDP-11s and VAXen for teaching, discovered that the SD-8516's clean architecture and dual-stack design made it an ideal platform for teaching language implementation. The VC-3 reference design, with its built-in FORTH interpreter and monitor ROM, could be assembled for under \$250 and sold for \$399 and less; a fraction of the cost of a minicomputer terminal, and with an architecture that was not constrained by a miniscule RAM or less registers than you can count on one hand.

A community formed. It was small, intense, and disproportionately influential. Forth programmers – already accustomed to being a cult – adopted the SD-8516 as their ideal machine. The dual stack pointers, the pre-decrement/post-increment modes, the composite register addressing: every design decision that had baffled mainstream buyers turned out to be exactly what threaded-code interpreters needed. SD/FORTH 2.0 (1988) running on a 4 MHz SD-8516 benchmarked at over 360,000 words per second, faster than any other microcomputer Forth in existence and competitive with native-code Forth systems running on processors twice its clock speed.

The BBS scene, still thriving in the early 1990s, began to adopt the VC-3 as a cult platform. Its KERNAL ROM, inspired by Commodore's conventions but refined with a decade of hindsight, provided a clean API for terminal I/O, file services, and sound synthesis. The DYNATERM 8800 text mode — 80 columns with it's retro era-authentic phosphor-amber CGA and COLORDORE palettes became instantly iconic.

The Business Surprise (1991-1999)

The SD-8516's unexpected third act came from business software.

By 1989, the IBM PC ecosystem was drowning in its own legacy. DOS applications were bumping against the 640 KB conventional memory barrier. The 80286's protected mode was notoriously difficult to use. The 80386 was powerful but expensive, and OS/2 (IBM's intended successor to DOS) was late, bloated, and unpopular. Meanwhile, Japanese manufacturers were producing SD-8516-based business terminals at remarkably low cost, leveraging the same Kitsune fabrication pipeline that produced the chip's multiply ROM. What's more, people started to worry about the "year 1999 bug". A bug that wouldn't affect the SD-8516 because it wasn't trying to be clever; it stored the date as a zoned integer (one byte per number) for easy conversion in business software. Interest grew as 1999 approached and sales went through the roof. A new, terminal-based SD-8516 "Business Package" was launched.

These terminals ran lean. The SD-8516's flat 24-bit address space, no segments, no banks to manage, no mode-switching ceremony, meant that business applications could address all 256 KB of memory directly and cleanly. The dual-stack architecture made multitasking cooperative schedulers trivial to implement. And the chip's speed, equivalent to a VAX-11/780 on integer workloads, was more than adequate for spreadsheets, word processing, and database applications. People began to develop expansion RAM cartridges. First to 512KB, then 1MB. By the early 1990s, more expensive 4 and 8MB enterprise-class expansion boards began to show up in the banking and scientific communities for specialized applications.

Stellar Dynamics licensed the architecture to several different Japanese and Taiwanese manufacturers. By 1992, SD-8516-based terminals were outselling 80486-class PCs in several Asian markets. A version with expanded banking (16 banks × 64 KB = 1 MB) addressed the memory ceiling on the base model, and a line of business peripherals – dot matrix printers, serial networking cards, and external floppy drives rounded out the ecosystem.

The installed base, combining arcade hardware, university systems, BBS machines, and business terminals, reached an estimated 2.4 million units by 1992. Stellar Dynamics, the company that nearly died in 1983, was valued at... *one **million** dollars.*

The Silver Age (1995-)

What happened next was unprecedented and, in retrospect, slightly insane.

The SD-8516's performance ceiling, real as it was, coincided with a cultural moment. The IBM PC had won the general-purpose computing war, but it had won it with complexity. CONFIG.SYS files, expanded memory managers, IRQ conflicts, device driver nightmares: the PC ecosystem in 1990 was powerful and miserable. The SD-8516 ecosystem was limited and joyful.

Programmers who had grown up on Commodore 64s and Apple IIs – the generation that learned to code by poking bytes and timing raster interrupts found in the SD-8516 a machine that respected their skills. You could understand the entire system. You could hold the memory map in your head. You could write to the framebuffer directly and hear the results from the sound chip immediately. The VC-3, with its 320×200 graphics modes, SID-inspired 8-voice synthesizer and console-inspired PPU and sample module, its FORTH interpreter, was like an 80s home microcomputer had gone to graduate school.

The demoscene adopted it. The indie game scene adopted it. A small but vocal contingent of programmers argued, not entirely without merit, that the SD-8516 represented the last of a generation; a computer that a single person could fully comprehend.

This philosophy that a computer should be knowable, that complexity has costs beyond performance delayed the adoption of 3D acceleration in the SD-8516 ecosystem by nearly a decade. While the PC world raced toward texture-mapped polygons, dedicated GPU pipelines, and hardware T&L units, SD-8516 developers perfected the art of software rendering. Mode 7-style floor effects, raycasting engines, and Bresenham line drawing techniques that PC developers abandoned the moment they had GPU hardware were refined to extraordinary levels on the SD-8516.

The results were, by any objective measure, technically inferior to what a Pentium with a Voodoo card could produce. They were also, by a measure that resists quantification, more beautiful. There is a particular aesthetic that emerges when every pixel is placed by code that a human wrote and understood, and the SD-8516 community cultivated that aesthetic with the devotion of monks illuminating manuscripts.

The Sunset Era (1999-present)

The SD-8516 did not survive into the 2000s as a commercial platform. It couldn't. The 3D revolution, the internet, the Windows hegemony... these were tidal forces that could be delayed, but not stopped. No 8/16-bit architecture could resist, no matter how elegantly designed. Stellar Dynamics pivoted to embedded systems in 1994, licensing the SD-8516 core for industrial controllers, point-of-sale terminals, and in a fitting full circle, arcade machines emulating legacy games. *Emulators running emulators*. The company was eventually acquired by an international conglomerate in 2002 for \$42 million, and the original engineers retired. They still post on social media, every now and then – to reminisce about the old days and give some pointers to the next generation. Then again, no one has heard from them in quite a few years now. Time flies when you're having fun.

Legacy

The scene never really died; it just went online.

Emulators for the 8516 appeared in the early 2010s. Browser-based implementations followed, running the SD-8516 in WebAssembly at speeds the original hardware could only dream of. The community, small but persistent, continued to write for the system. Demoscene productions, and retro-style games were plentiful on a machine designed to represent the best of an era. The SD-8516: a machine that existed on the boundary between history and mythology.

The SD-8516's legacy is not one of commercial triumph. It is a legacy of what could have been. Maybe, what should have been. Of choices that were right for reasons the market didn't value until it was too late to matter. The flat address space that Intel wouldn't adopt until the 80386. The hardware multiply that wouldn't become standard until the RISC revolution. The dual-stack architecture that Forth programmers had been begging for since 1970 and still don't have on x86.

Every few years, someone on a retro-computing forum asks: "What if IBM had chosen the SD-8516 instead of the 8088?" The question is unanswerable and irresistible. But the SD-8516 was never going to win. It was too expensive, too opinionated, and too late. But it was right. And sometimes, being right is its own reward; for as Angel the Vampire with Soul once famously said, "Our reward is that we

got to be the good guys; we got to do the right thing. Because when nothing you do matters, all that matters is what you do.”

From:

<https://www.appledog.ca/wiki/> - **Appledog**

Permanent link:

<https://www.appledog.ca/wiki/doku.php?id=sd:history&rev=1777557734>

Last update: **2026/04/30 14:02**

