

Design case history: the Commodore 64

The 'world's greatest' video-game chips became the basis for the world's most popular personal computer

In January 1981, a handful of semiconductor engineers at MOS Technology in West Chester, Pa., a subsidiary of Commodore International Ltd., began designing a graphics chip and a sound chip to sell to whoever wanted to make "the world's best video game." In January 1982, a home computer incorporating those chips was introduced at the Winter Consumer Electronics Show in Las Vegas, Nev. By using in-house integrated-circuit-fabrication facilities for prototyping, the engineers had cut the design time for each chip to less than nine months, and they had designed and built five prototype computers for the show in less than five weeks. What surprised the rest of the home-computer industry most, however, was the introductory price of the Commodore 64: \$595 for a unit incorporating a keyboard, a central processor, the graphics and sound chips, and 64 kilobytes of memory instead of the 16 or 32 that were then considered the norm.

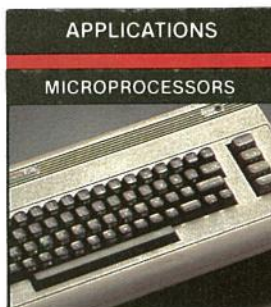
When the chip-development project started, the Commodore 64 was not at all what the designers had in mind. MOS Technology was a merchant semiconductor house. Its LSI Group, headed at that time by Albert J. Charpentier, had been responsible for some of the chips that went into Commodore's VIC-20 home computer, but that project was already well into production. "We were fresh out of ideas for whatever chips the rest of the world might want us to do," said Charpentier, "So we decided to produce state-of-the-art video and sound chips for the world's next great video game."

Charles Winterble, then director of worldwide engineering for Commodore, gave the go-ahead for the chip effort, and Charpentier's group worked fairly independently until both chips were finished in mid-November 1981.

At a meeting with Charpentier and Winterble late that month, Jack Tramiel, then president of Commodore, decided not to proceed with the video game. Instead, he decided, the chips would go into a 64-kilobyte home computer to be introduced at the Consumer Electronics Show in Las Vegas the second week of January 1982. The computer had yet to be designed, but that was easily remedied.

In two days, the engineers laid out on paper the machine's basic architecture. Just before the new year, they completed five working prototypes. In the meantime, enough operating-system software was copied and rewritten from the VIC-20 to give passable demonstrations of what the new machine could do. Following its enthusiastic reception at the Consumer Electronics Show, the Commodore 64 was rushed into production; volume shipments began in August 1982 and have continued unabated.

Despite complaints about quality control and the industry's slowest disk drive, the Commodore 64 has been an unparalleled success, pushing a number of its competitors out of the market. Part of the reason for its success is the price, which keeps fall-



ing—from \$595 at its introduction to \$149 currently, for which the consumer gets graphics and sound equal to or better than that provided by machines that cost five times as much.

The next great chips

In the late 1970s, MOS Technology was a successful semiconductor company; its engineers had designed the popular 6502 microprocessor, and it manufactured several other solidly selling products.

Commodore, a West Coast company at the time, took over MOS in 1976—causing many of the MOS engineers to quit when their stock was bought at 10 cents on the dollar—but MOS was allowed to operate fairly autonomously.

In 1979 and 1980, recalls Charpentier, MOS Technology developed the 6510 microprocessor—a minor revision of the 6502 with an additional input-output port, the 6526 peripheral controller, a lot of read-only memory (ROM) chips, and a 4-kilobit random-access memory (RAM) chip. At that time, MOS was supplying equipment to Atari Inc., General Electric Co., Hewlett-Packard Co., and a number of other clients, as well as to Commodore.

When the decision was made to design the latest in superior graphics and sound chips, the first step was obvious: to find what the current high-quality chips could do. In assembling his design

Defining terms

Bit-map—a display system in which each dot on the screen is controlled by the value of a particular bit in computer memory. Multiple bits per dot permit multiple colors and shading.

Character map—a display system in which a group of dots on the screen are defined by a single word in computer memory that points to a miniature bit-map containing the values for each dot. Because many patterns are reused on a typical screen, character maps use less memory than bit-maps.

Color clock—the frequency of the carrier signal used to transmit color information in the National Television Systems Committee (NTSC) standard; 3.58 megahertz.

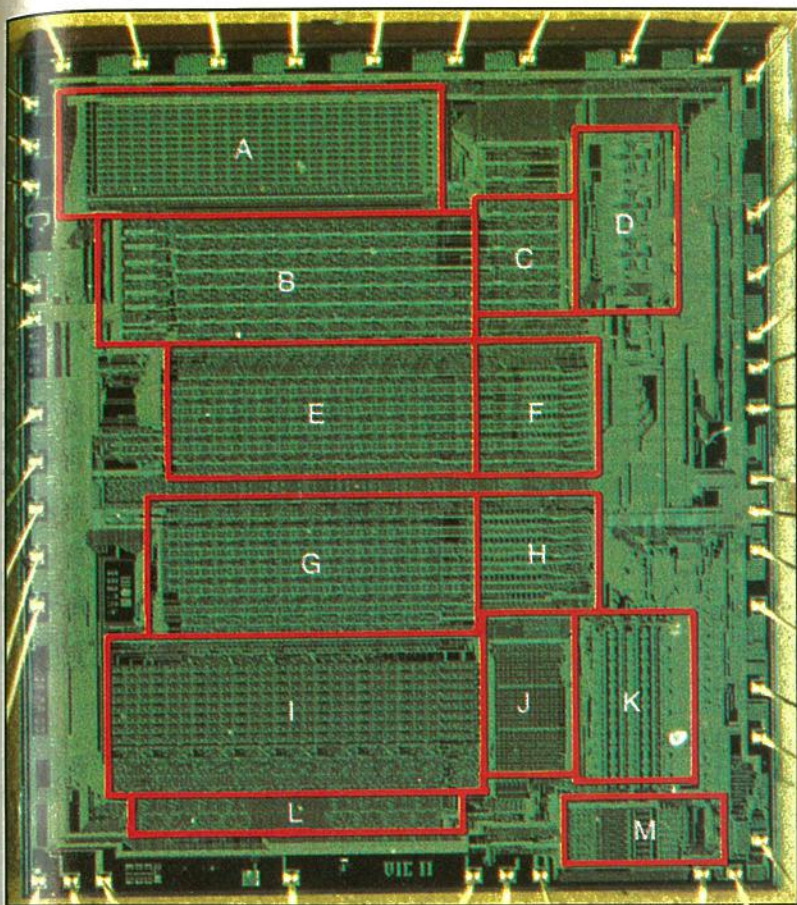
DOS—disk-operating system; the set of functions, such as opening, closing, renaming, and deleting files, that are required for a computer to make efficient use of a disk drive.

Field—in the NTSC standard, a set of 262½ television scan lines, transmitted at a rate of 60 per second. Two interlaced fields make up a single frame of 525 lines.

Interrupt—a signal that causes a computer to stop whatever it is doing and call a special routine. Interrupts permit a computer to process data while responding to unpredictable inputs from peripheral devices.

Sprite—a movable graphics object used in home computers and video-game machines for animation. A sprite can be moved simply by changing the registers that control its position, while an object drawn in bit-mapped memory can be moved only by copying it to a new set of memory locations.

Tekla S. Perry and Paul Wallich Associate Editors



The bulk of the area of the Commodore 64 video chip, the VIC II, is devoted to the eight sprites (B, E-H, L). The clock-generation circuitry (D) serves the entire system. The only circuitry devoted entirely to generating characters is the storage array that holds each line of characters to be displayed (A) and the address counter that fetches the appropriate ROM data for each raster line of the character display. Three modules (I, J, L) are responsible for sending sprite and character data out to the screen in the proper order and reporting any collisions between sprite and character data. The central portion of the chip consists of the circuits for displaying sprites at their correct horizontal and vertical positions, and for fetching their data from memory. The timing for the various chip functions, such as fetching data for the sprites, is done by combining a counter and a ROM. The counter steps through the addresses of the ROM, and the contents of each ROM address drive the appropriate control lines. The MOS Technology logo and the designers' and draftsmen's initials are visible near the left margin of the chip.

- A. Character-pointer storage
- B. Sprite address counters
- C. Character address counters
- D. Clock generator
- E. Sprite vertical comparator
- F. Vertical timing control
- G. Sprite horizontal comparator
- H. Horizontal timing control
- I. Sprite and character shift registers
- J. Display-priority encoder
- K. Color register
- L. Sprite-character collision detection
- M. Video outputs

team, Charpentier recruited Robert Yannes, a young engineer who had joined MOS in 1979 and had designed the VIC-20 at home, from a spare prototype board. The team spent about two weeks researching comparable chips industrywide.

"We looked heavily into the Mattel Intellivision," recalls Winterble. "We also examined the Texas Instruments 99/4A and the Atari 800. We tried to get a feel for what these companies could do in the future by extrapolating from their current technology. That made it clear what the graphics capabilities of our machine had to be."

The MOS designers freely borrowed ideas that they liked—sprites from the TI machine, collision-detection techniques and character-mapped graphics from the Intellivision, and a bit-map from their own VIC-20. They then packed as many of those ideas as they could into a predefined area of silicon.

"Al [Charpentier] was given the freedom, within a certain die size, to put in everything he could, working backward within the size we planned. When he ran out of registers he stopped," said Winterble. "We defined in advance the silicon size that would give a yield we were willing to live with; at that time, a die size that was reasonable in 5-micrometer technology was less than 200 mils. Then we prioritized the wish-list from what must be in there to what *ought* to be in there to what we'd *like* to have, so the decisions that were needed at various points became fairly automatic."

Nine months to first-pass silicon

For nine months, Charpentier worked with two draftsmen and one computer-aided-design operator on the graphics chip while Yannes worked with two other draftsmen and one CAD operator on the sound chip. They lacked completely the sophisticated design tools of today's engineering workstations, but they had one readily available design tool found almost nowhere else in the home-computer industry: a chip-fabrication line on the premises. With this, Winterble explained, a circuit buried deep inside

the chips could be lifted out and run as a test chip, allowing thorough debugging without concern for other parts of the circuitry. David A. Ziembicki, then a production engineer at Commodore, recalls that typical fabrication times were a few weeks and that in an emergency the captive fabrication facility could turn designs around in as little as four days.

The cost of developing the Commodore 64? No one knows. "I had no formal budget accountability," said Winterble, "other than Jack [Tramiel] watching me. Jack said that budgets were a license to steal." Because MOS Technology's fabrication facility was not running at full capacity, the equipment used for C-64 test chips and multiple passes of silicon would otherwise have been idle. "We were using people who were there anyway," said Ziembicki. "You waste a little bit of silicon, but silicon's pretty cheap. It's only sand."

Although custom chips are usually considered expensive, the C-64 chips were not. Not only were development costs absorbed in company overhead, but there was no markup to pay, as there would have been if the chips had been built by another company. And yields were high because the chips were designed for a mature semiconductor-manufacturing process.

The sound chip was designed with 7-micrometer technology, scaling down to 6 in places. (By contrast, the custom chip for Atari's Video Computer System, fabricated four years earlier, was built with 6-micrometer technology.) Commodore's video chip was designed with 5-micrometer features.

Because design time rather than silicon was at a premium, the chips were laid out simply rather than compactly. "We did it in a very modular fashion," said Yannes. For example, he noted, "the standard way of building oscillators is to build one and then multiplex it until you have as many as you need. We just built an oscillator module and repeated it, because that was much faster than working out all the timing for the multiplexer."

"What was remarkable," Winterble added, "was that nine months later, when we came out with the first pass, it functioned

except for one bad sprite."

In November 1981, the chips were complete. The original intent had been a game machine, but at this point the personal-computer market was beginning to look promising. At a meeting of Charpentier, Winterble, and Tramiel, the decision was made to go for a personal computer. The next choice to be made, ac-

cording to Charpentier, was between a 16-kilobyte and a 64-kilobyte machine, since the custom chips were designed to handle either option.

"Jack [Tramiel] made the bet that by the time we were ready to produce a product, 64K RAMs would be cheap enough for us to use," Charpentier said.

A rich collection of graphics modes

"We did not intend to make 128 colors," insisted Albert Charpentier, who was chief designer of the Commodore 64. "It was just too big an area on the chip."

Nonetheless, the Commodore 64 can display far more than the 16 colors listed in its documentation, in addition to its many other graphics features. Brian Dougherty, president of the Berkeley Softworks of Berkeley, Calif., explained: "The width of each pixel is almost half of the NTSC color clock, so when you alternate pixels of two different colors, instead of getting the two colors that you think you're getting, you get a whole new phase interpretation." [See figure below.] The new phase interpretation, of course, yields a new color. Because the pixels are $\frac{1}{2}$ the width of a color clock, the colors produced shift cyclically across the screen, making them difficult to utilize.

Before he designed the video chip, Charpentier recalled, he had spent a few weeks looking at the competition. In addition to the products of Texas Instruments, Atari, and Mattel, he studied his own previous work. "I remember spending about a week going through the graphics modes we could put in."

"That was an example of engineering escalation," said Charles Winterble, who was director of worldwide engineering for Commodore at the time the video chip was being designed. "We found strange color modes. Al's imagination went crazy; some of the color modes turned out to have no purpose, some of them turned out to be very clever."

There are two basic modes in the Commodore 64: character-mapped and bit-mapped. In the character-mapped mode, the screen is made up of 1000 pointers, each of which selects a character stored in ROM or RAM. Since each pointer contains 8 bits, 256 eight-by-eight-pixel characters can be selected. In the bit-map mode, 8000 memory locations are used to define the screen, with one bit corresponding to a single pixel. The bit-mapped mode provides maximum flexibility but gobbles up memory, especially if more than one

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P
Q	R	S	T
U	V	W	X
Y	Z	AA	BB
CC	DD	EE	FF

The Commodore 64 bit-map is unlike those of most computers, in that it borrows most of its circuitry from that used to display character graphics. A standard bit-map would store the eight-dot segments above in successive memory bytes in the order A, B, C, D... The Commodore 64 stores them in the order A, E, I, M, Q, U, Y, CC, B, F... This mimics the eight-by-eight-dot patterns of characters.

screen is defined. The character-mapped mode is not as limiting as it might seem, because most pictures are made up of repeated elements.

There are several variations of the character-mapped mode. In its standard form, it contains 256 eight-by-eight-pixel cells, with a resolution of 320 pixels horizontally and 20 lines vertically. Each of the 1000 positions on the screen can contain a different foreground color, but all use the same background color. "This is a limitation when it comes to mixing colors," said Dougherty. Text is typically displayed using the standard character-mapped mode.

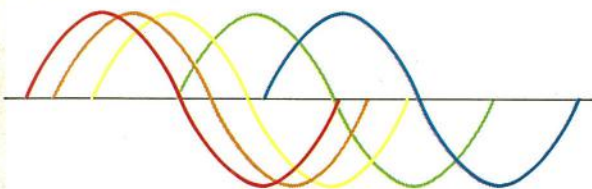
In the multicolor character mode, each cell has one foreground color and a choice of three background colors. Only eight of the 16 colors are available for the foreground; the background colors are defined by three registers in the video chip. Two bits (instead of one) are used to define each pixel in a cell, so the cells have a resolution of eight lines vertically by four pixels horizontally, rather than the standard eight by eight. Each pixel is twice the normal width, so display resolution is reduced to 160 by 200.

In the extended-color mode, there is a choice of four background colors for each position on the screen, plus a foreground color. All 16 colors are available, and the screen is displayed at full resolution. The only limitation is that the number of characters is 64 rather than 256, because two bits of each character pointer are devoted to color selection.

The bit-mapped mode is adapted from the character-mapped mode, making its use less straightforward than it might be. "Most raster-scan algorithms are expecting memory to be laid out linearly," said Dougherty, "so it is hard to use a lot of software that has already been done for other systems." Typical bit-maps are organized with successive pixels on a horizontal line in successive memory locations, but the C-64 bit-map uses eight successive memory locations to store eight pixels each from eight successive lines. The succeeding eight locations store eight more pixels from the same eight lines [see figure above]. This mimics the way that pixels for the character set are laid out in ROM, permitting the same circuitry to be used for both modes.

In the standard bit-mapped mode, a "1" bit is displayed in the foreground color and a "0" bit in the background color (with a different foreground color available for each eight-by-eight-pixel group). In the multicolor bit-mapped mode, two bits are used to define each pixel, thus allowing four colors to be chosen from the 16 that are possible, although the screen resolution again drops to 160 by 200.

—T.S.P. and P.W.



In the National Television Systems Committee standard, color is determined by the phase of the color carrier relative to a color burst sent at the beginning of each line. The sine waves above are colored according to their phase under that standard. A change from red to orange is relatively clean because there is only a small phase difference between the two sine waves. However, a change from red to blue can cause smeared edges on the screen because it leaves the phase indeterminate from the point at which the red sine wave starts upward to the point at which the blue sine wave reaches bottom, a period of about 300 nanoseconds. This problem is universal in systems that adhere to the NTSC standard.

When the design of the Commodore 64 began, the overriding goals were simplicity and low cost. The initial production cost of the Commodore 64 was targeted at \$130; it turned out to be \$135. The computer would use the same case as the VIC-20 and the same-sized circuit board, to speed development. "That wasn't even a decision," said Charpentier. "It was just common sense. If you've got a product that's a winner, why change it?" To switch from the VIC-20 to the Commodore 64, the production group had only to make the cartridge slot smaller, change the color of the case, and design a new label.

Furthermore, instead of designing for performance first and price second, as many engineers tend to do, the small design team at Commodore had cost in mind from the beginning. Yannes, for example, had made economy his credo when he was an impecunious high-school student looking for parts to build a sound synthesizer. "Anytime I design something, I want to use the minimum number of components possible. It's a personal challenge. If there's a spare [logic] gate in a gate package, I'll work to get rid of the entire package, because in working with a certain number of chips, I ought to be able to use up everything that's in them. The Commodore 64 was my attempt to build the absolute minimal system that could be built out of the video and sound chips put together."

Happy New Year!

In the consumer-electronics industry, new products are traditionally introduced to distributors, dealers, the press, and the competition twice a year—during the first weekend in January and the first weekend in June, at the Winter and Summer Consumer Electronics Shows (CESs).

"When you worked for Commodore," said Yannes, "you always had to have something for the Winter CES." The C-64 didn't have much competition at the winter show in 1982. Atari was still showing its 400/800 computer. Mattel introduced the Aquarius computer, and Spectravideo introduced its computer/game machine, both with limited memory and capabilities.

Ziembicki recalled, "All we saw at our booth were Atari people with their mouths dropping open, saying, 'How can you do that for \$595?'"

"We were a hit," Winterble added.

After the show, Commodore, which had a reputation for sometimes showing products that never reappeared, had to get the 64 into production quickly. Only a few design changes were made to ready the machine for production. "What's in the 64 functionally is what I wanted," Yannes said. "What's in the sound chip is what I wanted, and what's in the video chip is what Al [Charpentier] wanted."

In fact, said Ziembicki, "after it finally got into production, we looked back and asked, why did we bother changing it after January? It wouldn't have sold one unit less, and we would have saved a lot of money on development costs."

Some of the changes were fairly significant, but they may have created as many problems as they solved. The biggest improvement was to remedy "a hiccup" in the video chip, said Winterble.

When Charpentier first designed the video chip, he explained, his goal was optimum performance in both black-and-white and color. Previous video chips, such as those used in the Atari and the Apple, had a black-and-white frequency of 7.16 MHz—twice the television color-clock frequency of 3.58 MHz—but this could not squeeze enough characters on a line, and it also induced cross talk between the two signals, causing an object to shift slightly to the left or right on the screen depending on its color. After the January 1982 CES, when he saw that 40 characters wouldn't quite fit onto a TV screen, Charpentier sped up the black-and-white clock rate and made the two clocks completely asynchronous. But since the color and black-and-white sets of information were constantly changing phase relative to each other, "the color transitions were fantastic, but there was an overall swimming effect," he recalled. The solution was to put a phase-locked loop into the system so the color and the black-and-white infor-

mation would have a constant relationship. This, Charpentier acknowledged, was "a Band-Aid solution."

And that is what it looks like, according to Craig Nelson, director of product development for Epyx Computer Software of Sunnyvale, Calif. "It's a really elegant circuit," he said, "except for the phase-locked loop, which is just stuck in the middle of it." (Commodore now says it has redesigned the chip to eliminate the problem.)

As a result of the "Band-Aid," the color information and the black-and-white information shift phase by 180 degrees in relation to each other on successive video fields rather than changing phase unpredictably. This conforms more closely to the National Television Systems Committee (NTSC) standard, which requires black-and-white information to have a clock rate that is an odd harmonic of the color-clock rate, so that the two automatically reverse their relationship every other field—one field in phase, the next field out of phase. (In the NTSC standard, colors are determined by the phase difference between the color signal and a color reference signal transmitted at the beginning of each line.)

At the time the standard was developed, commonly available circuitry could not completely separate the black-and-white information from the color: changes in color would lead to changes in brightness and vice versa. It was to average these changes out over time that the NTSC specified that the relative phase reversal of the two signals would occur on successive fields. The result is more pleasing color transitions, but the phase shift makes stationary figures appear to jitter. The jitter is particularly obvious on thin vertical lines, like those in alphanumeric characters. But when the Commodore 64 was conceived, it was to be primarily a game machine, not a computer.

Running a 5-micrometer-technology chip at an 8-MHz clock rate caused it to dissipate a great deal of power—nearly 1.5 watts. Not only did the chip run fast, but to prevent the colors from washing out, as they had in the VIC-20, "we went to 12 volts to drive the heck out of the color signals," recalled Winterble. "We knew we would have a heat problem."

At this point Winterble made another design fix—welding a small metal tab onto the inside of the lid of the shielding enclosure around the video chip. When the computer was assembled, the tab pressed against the top of the chip package, forming a heat-conducting path and turning the shielding into a heat sink.

Some changes that might have improved the machine did not get made in the rush to production. For example, Yannes said, the wires for the sound output on the printed-circuit board run alongside the wires for the video signal. As a result, the sound output picks up an annoying 15 750-Hz whine. Rerouting the circuit would have taken time, "and we had a board that worked," he explained. "At that point, if you had something that worked, you did not change it." A circuit-board revision since then has rerouted these lines, Commodore said.

But the designers did in fact change some things that worked—including one revision that degraded the machine's performance. The original design specified a high-quality radio-frequency modulator to transmit the signal to a television set, but a cheaper modulator was substituted. "It was \$6.25 to \$6.50 for a good one, and we ended up spending about \$3," Charpentier said.

Winterble has a different view of the \$3 modulator: "The 50-cent modulator we were using on the VIC-20 wasn't good enough," he said, "so we went to a more expensive one."

Electronic design wasn't the only difficult area as the Commodore 64 went from prototype development into production—the logistics posed a complex problem. The C-64 was designed in Norristown, Pa., at MOS. The VIC-20 assembly line, which was to begin making C-64s, was in Santa Clara, Calif. As the C-64 went into production, Commodore was also opening a new assembly line in West Chester, Pa. There were additional VIC-20 assembly facilities in Japan, where the disk drive for the C-64 was to be manufactured. And the C-64 circuit boards were being made in Hong Kong.

"It was a lot of fun," said Ziembicki. "The design people

would pick an English screw. The production end would pick a metric. But they went with what they had. Commodore production was very good at making things fit whether they were intended to or not. Their charter was, 'Ship 'em!'"

"It takes a very tough person," explained Charpentier, "to say 'I'm not shipping these because they're not as good as they could be'—especially when people are clamoring to buy them."

Changes continue during production

The start of production of the Commodore 64 in the spring of 1982 did not signal an end to controversy. "The key is to be able to solve your problems while you're running," said Ziembicki.

Sprites: making animation easy

The inclusion of sprites—independently movable objects that overlay the background display—received a great deal of attention during the early days of the Commodore 64 project. In fact, the circuits that are used to manipulate and display the sprites occupy two-thirds of the machine's video-display chip.

"In hindsight, it would have been better to put a little more bit-map capability into the machine, but we focused on sprites," said Albert Charpentier, who was chief designer of the Commodore 64 computer. "The premise at the time the chip was designed was a game machine."

And a good game, of course, uses many independently moving objects. Robert Yannes, also a designer on the C-64 project, recalls studying Atari's so-called player-missile sprites. "They are terrible," he said. "They don't have enough color, they're not big enough, and they don't do what you want." The sprites on the Texas Instruments 99/4A, he said, "were great, but they were only one color, and you had to group them together to make a big enough object."

The C-64 sprite, the group decided, would be a multicolored

One of the first battles, recalled Charpentier, was over the layout of the printed-circuit board. Commodore's assembly plant in the United States used automated component-insertion equipment, but its Japanese facility did not, and the two assembly techniques required different component spacing. In the end, the board for the C-64 was laid out for automated insertion, and production was moved to a new plant in Hong Kong that had the automated tools.

Problems also plagued a number of the components—switches, for example. "You pick a switch that's listed as a consumer switch," said Ziembicki. "You design it in. You call the manufacturer and get an estimate that sounds reasonable. Then Cali-

object independently movable in both the x- and the y-axis, with minimal software overhead.

"We spent a lot of time discussing the sprites," said Charpentier, "asking, How big should they be? Should they have expansion by a factor of 2 in x? In y? Should we have collision detection?"

A series of mathematical calculations answered many of these questions, he explained. "Moving across the video line, you've got 63 microseconds to play with. I was going to use memory running at just under 2 MHz, meaning only 114 memory fetches per line—period. No more. I figured the processor was going to need at least 40 of those fetches; 40 was a nice architectural number because there are also 40 pieces of background information, so the background information and the processor slices could easily be interleaved.

"So 114 minus 40 for the processor and 40 for the background left me with 34 fetches. I needed eight fetches to pick up the sprite pointers [which indicate where sprite information is stored in memory]. That left 26 fetches. If I had eight sprites, that gave me three fetches for each sprite."

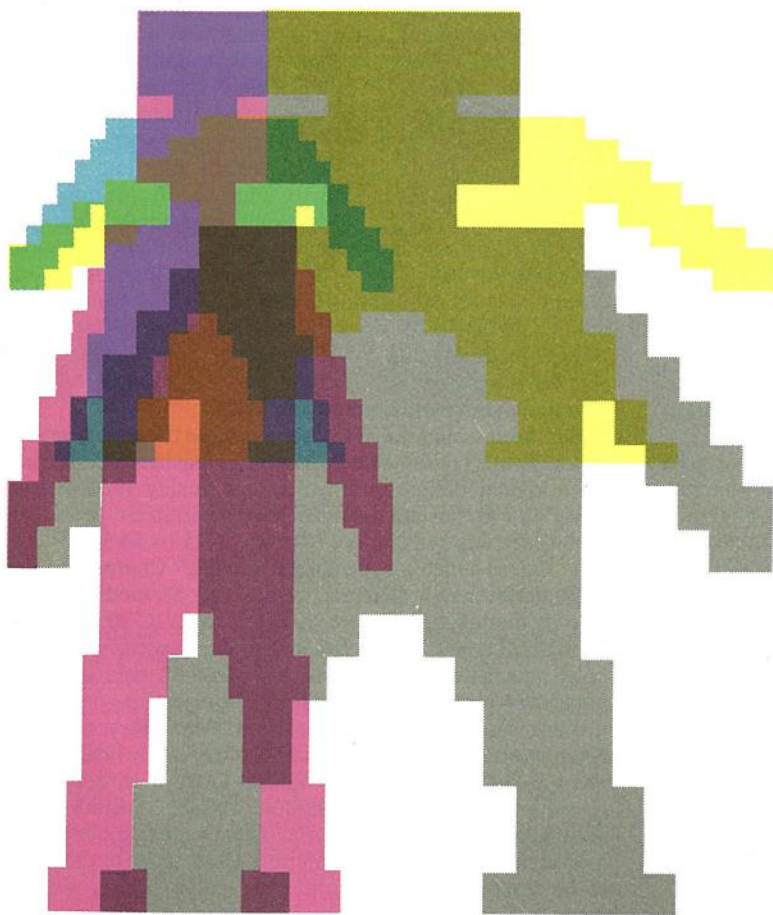
Charpentier explained that no extra cycles were needed to refresh the dynamic RAM during the screen display, since the video chip was already stepping through the appropriate row and column addresses. Only during the vertical-blanking period is the memory refreshed separately.

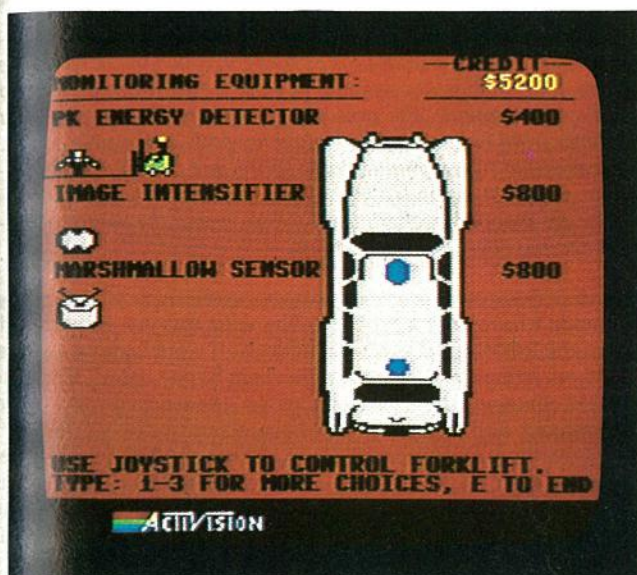
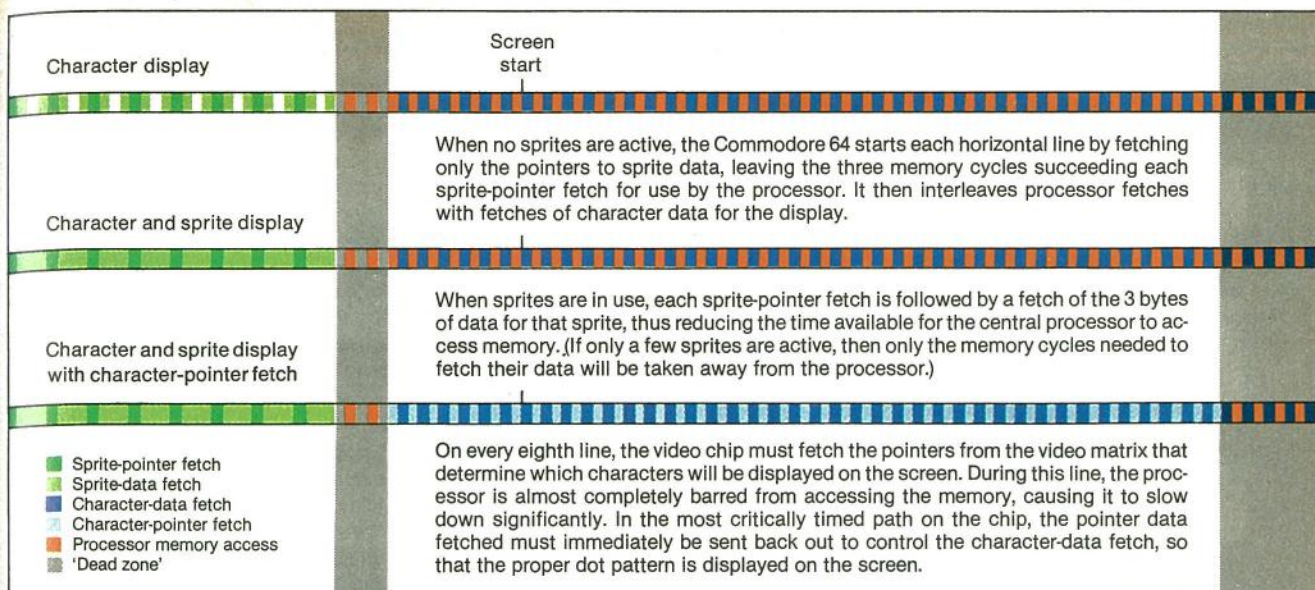
"When I used every one of those cycles I could get my hands on, I had 3 bytes per sprite," he said. (A change in clock speed later increased the number of cycles available on a line to 126, but it was then too late to change the sprite circuitry.)

The 3 bytes per sprite determined the horizontal size of the sprite—24 pixels. Charpentier then looked for the multiple of 3 (bytes) that would approach a power of 2, so that the sprites could be stored efficiently. "It happened that 24 by 21 bits gave 63 bytes of information," he said. Hence, each sprite is 21 lines tall.

Not only can the sprites be positioned anywhere on or off the screen; they can also be stretched to twice their normal dimensions in either the horizontal or vertical direction, producing a sprite as large as 48 by 42 dots (although the resolution does not increase). [See figure.] Sprites can also be defined in the multicolor mode, in which case the horizontal resolution drops to 12 pixels, although their size does not change. —T.S.P. and P.W.

Sprites can be displayed in any of four sizes: normal, vertically expanded, horizontally expanded, or expanded in both directions. They can also be displayed in the multicolor mode, in which their resolution is coarser but the proportions are retained.





The memory-access timing of the Commodore 64 was determined by the need to interleave fetches for the sprites, the character display, and the central processor. Since the central processor uses a two-phase clock and accesses memory only on the second phase, most display fetches can occur without slowing the processor. Only sprite-data fetches and character-pointer fetches use phase-two cycles. Because the memory timing was originally designed for a slightly slower clock speed, "dead zones," during which only the processor accesses memory, were inserted into the timing sequence. In a screen generated by the Activision video game *Ghostbusters* (left), sprites are used to form the displays of the forklift and various other devices that can be bought to prepare for ghostbusting. The background display is formed from specially designed characters (eight by eight blocks); if the game required significant computing, the sprite and character fetches for the center portion of the screen could limit the available processor cycles.

fornia [the production division] wants 50 000 a week, but the manufacturer says, 'We can't make that. It's a consumer switch, but we're not geared for consumer quantities.' At that point, you're hung up."

Since Commodore had just moved its engineering staff from California to Pennsylvania, communication between the design engineers and the production facilities was not very good, Ziembicki explained. "It got to the point where you couldn't stop working even if you didn't get an answer—you'd just proceed with the components that you thought the production end could get."

And outside suppliers were not always reliable. "One provided a power supply for engineering approval," Ziembicki recalled. "It got approved, and then the supplier changed the design and didn't tell anybody."

Charpentier said that his relations with the production group were fairly agreeable. Winterble's dealings with them were not nearly as cordial, but he found this acceptable. "I personally had to play the heavy on a lot of stuff," he said. "Then Al would go in and do some good. That technique eased a lot of tension—it was a way to get things done."

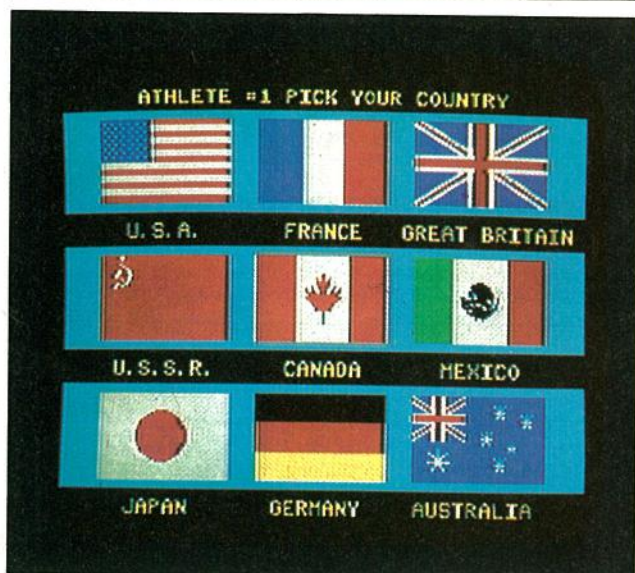
The most notorious problem was "sparkle," a defect that caused small spots of light to appear on the display screen. The problem was solved before Charpentier left the company in Sep-

tember 1982, but reports of the defect continued well into the Christmas season, with press reports citing it as an example of Commodore's poor quality control. Canny consumers used it as an excuse to return perfectly good Commodore 64s while they were under 90-day warranty. Since the price of the machine had dropped \$200 in the two months following its introduction, this ruse enabled owners to obtain a refund of the purchase price and buy another C-64 at the lower figure.

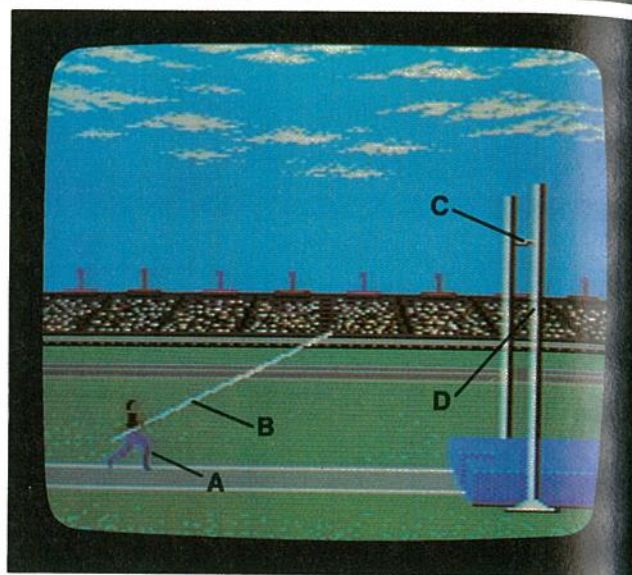
Sparkle was widely attributed to bugs in the video chip that was the heart of the system, but in fact it was caused by a ROM chip of which 3 million were in service with no problems in other types of systems, including the hit arcade video game *Asteroids*. Commodore engineers themselves first looked for the problem in the video chip. It took them three weeks to spot the ROM chip as the source of the defect, Charpentier said. "The problem was a random event—it didn't happen all the time. We thought the video chip was for some reason seeing wrong data. We didn't even suspect it could be the ROM. Finally we put the logic analyzer on it and tracked it down." The ROM, which Charpentier and his group had designed years earlier, had a special pre-charging circuit to make it run faster, but the circuit made it sensitive to spurious signals. The video circuitry and the 6510 micro-processor alternated in controlling the system bus, and when control passed from one to the other, voltage spikes were sometimes generated.

"It just happened that we hit the exact timing," Charpentier said. "If the spike had been few nanoseconds shorter or longer, it

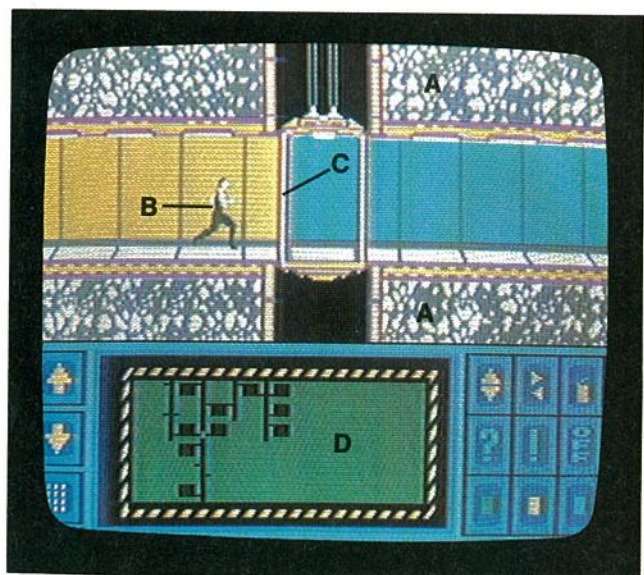
Exploiting the C-64



The various features of the Commodore 64 hardware can be used alone or in concert to achieve impressive effects. In the screens shown here, the raster-interrupt feature of the Commodore 64 is used to change the colors and the resolution, to increase the number of available sprites, and to alternate between the various graphics modes. In an opening screen (above) of the *Summer Games* video game from Epyx, two raster interrupts between the rows of flags are used to change color registers, thus increasing the number of colors available in this multicolor character-mapped display. Sprites are also used to add colors and complex detail.



The pole-vaulter (A) from the *Summer Games* software is composed of three high-resolution sprites for which the data change every four frames ($\frac{1}{4}$ second), cycling through seven states to achieve animation. The pole (B) is made of three sprites that change from regular width to double width as the pole moves. The crossbar (C) is a multicolored sprite, and part of the right upright (D) is an expanded multicolor sprite (matching the multicolor bit-mapped background), which causes the vaulter to appear to pass between the uprights.



In this screen from the Epyx video game *Impossible Mission*, the rocks (A) and computer screen (D) are constructed from standard character-mapped graphics with full resolution and two colors. The running man (B) is made of three full-resolution (one-color) sprites—two dark gray and one white. Overlaying the side of the elevator (C) is a half-resolution sprite that is masked because its four colors duplicate the pattern of the graphic beneath it. The purpose of this sprite is to hide the running man as he enters the elevator, making him appear to pass through the elevator door. The remainder of the screen is a multicolor character-mapped display.

wouldn't have been a problem. The spike was just wide enough that the ROM saw it as a valid address. It would ignore the next address request and give the video chip wrong data." Since the ROM contained the C-64 character set, the screen display would be littered with random slices of characters.

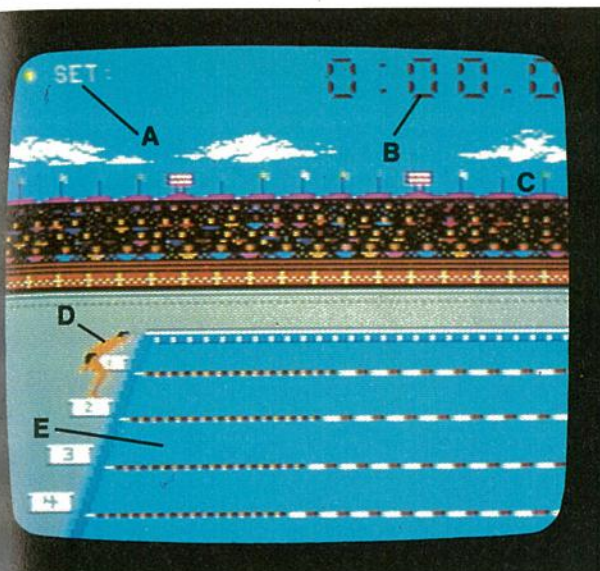
According to Nelson of Epyx, "This confetti interference-looking stuff on the screen, glowingly referred to as sparkle, has an extremely un-nice property: it causes hardware collisions—the sprites believe it really exists." Since the sparkle was caused by inappropriate data fed to the video chip, it triggered the circuitry responsible for checking whether the movable display objects—sprites—were overlaying background objects on the screen. So software that depended on collision sensing to control the movement of objects on the screen would go berserk when confronted by sparkle.

The bug in the ROM was corrected, although Commodore had to rely on outside suppliers for a while as it ramped up its new production line. Only the first few hundred thousand units shipped had the defect. But, Nelson observed, "it doesn't matter to programmers that it was fixed. Our software has to work in every machine out there." The best solution for programmers, he explained, is to copy the contents of the ROM into RAM, which is reliable. But this wastes 2 kilobytes of RAM.

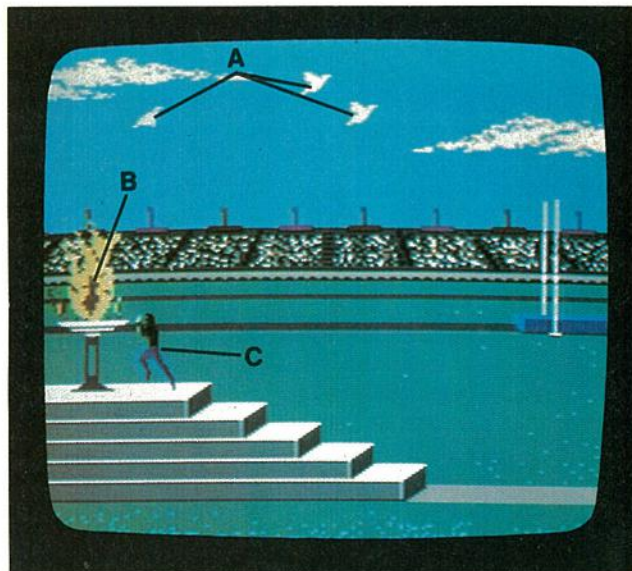
'A screw-up in my timing'

In addition to the difficulty with the ROM, "I made a logic error," Charpentier recalled. The error, which was corrected sometime after Charpentier left Commodore, caused the early C-64s to generate the wrong number of clock cycles on each horizontal video line. "It was off by one," he said. "Instead of 65 clock cycles per line, I had 64."

As a result, the 180-degree phase shift between the black-and-white and color information, which would have eliminated color-transition problems, didn't occur. Depending on their color and the color of the background, the edges of some objects on the screen would appear slightly out of line. This was corrected ap-



The characters in the sky (A) are part of a high-resolution bit-mapped display; the clock (B) is composed of six single-color high-resolution sprites. A raster interrupt changes the screen to a multicolor low-resolution bit-map (C). The swimmers (D) from the Summer Games are created from four high-resolution sprites; an interrupt between the lanes (E) allows these sprites to be reused. To scroll the bit-mapped image of the pool, the data for two screens exist—one scrolling eight pixels, then the other replacing it.



The entire background in this screen from the Summer Games is drawn in a multicolor bit-map. Each of the eight doves (A) is a single-color, high-resolution sprite. A raster interrupt allows these sprites to be reused: four multicolored sprites make up the flame (B), and three single-color sprites make up the man (C). The man, doves, and flame are animated with fourteen, eight, and four states of animation, respectively.

proximately five months into production.

But leading edges are still a problem with the C-64. The circuitry that displays either sprite information or background information at any point on the screen is sometimes slow to respond and overlays the sprite on the background information only after it has missed a few pixels.

The C-64 designers weren't the only ones revising their machine; even some assembly-line workers got into the act. Since testing color quality automatically is virtually impossible, assembly-line workers were instructed to turn a potentiometer that controls the color and black-and-white signals until they saw color. As Charpentier recalled, the workers soon discovered that if they turned the potentiometer as far as possible, it resulted in saturated color—but it also wiped out the black-and-white signal. Several thousand computers had been shipped before the assembly-line workers' "revision" was noticed; by then, some reviewers were lambasting the C-64 for its "garish" colors.

Quality still lacking

Though these modifications have been made in the C-64 since its introduction, designers who write commercial software for the machine would like to see a few more. One such change would be adequate quality control. "They don't test," said Nelson of Epyx. "I've opened up brand-new Commodores and found traces cut. They obviously use a power screwdriver to assemble the C-64, sometimes miss the screw, and chop the traces. How, you might wonder, could that have passed final inspection? Well, those traces are hooked up to the disk-drive connectors, which they obviously don't test."

In hindsight, Charpentier called one of Commodore's mistakes "not coming a little closer to quality."

One complaint voiced by designers of add-on hardware for the C-64 is that its minimalist design results in undesirable interactions between parts of the hardware and software. For example, the circuitry used to control the joystick and game-paddle port is borrowed once every $\frac{1}{60}$ second to scan the keyboard for keys

that have been pressed. If the joystick switches are closed, the keyboard-scanning software will report that a key has been pressed. Devices that attach to the joystick ports can cause unpredictable responses—a problem that would not occur if separate hardware had been used for the two functions.

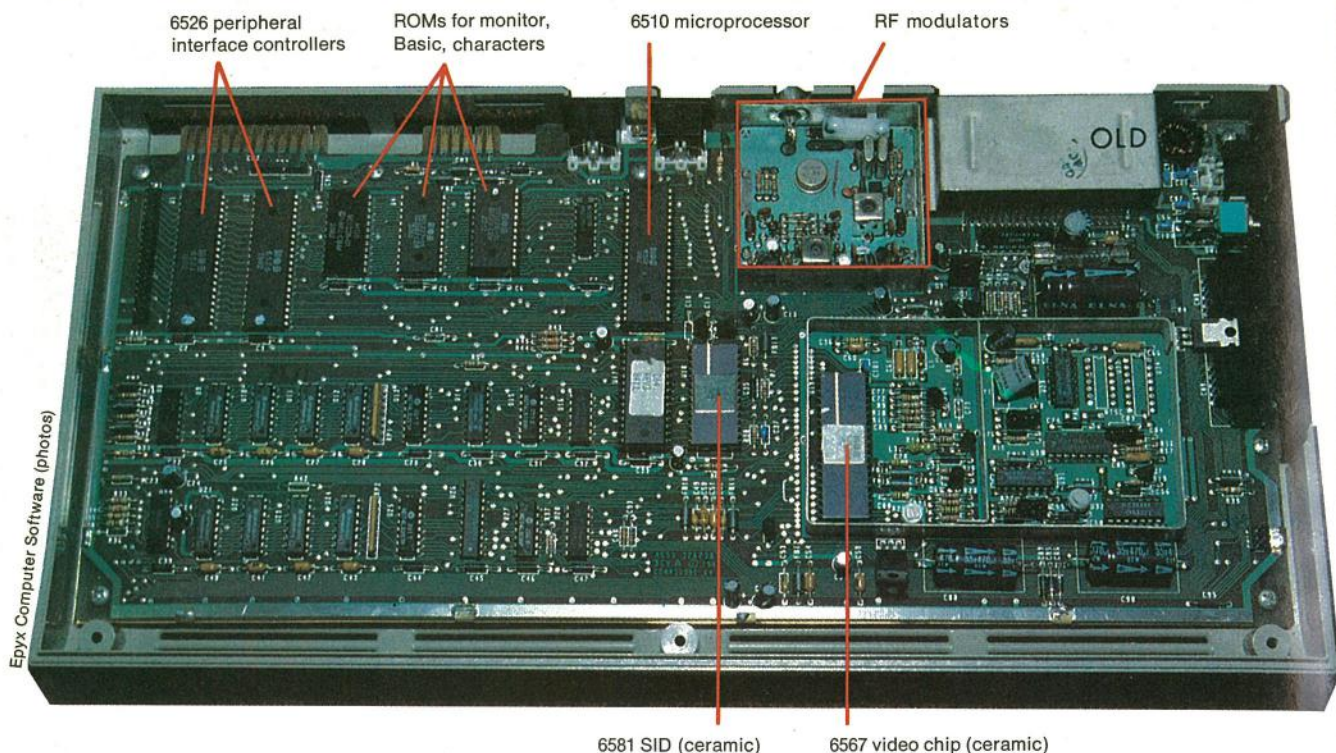
Another flaw is the computer's crude internal software. The system comes up in Basic when it is turned on. This is a leftover from the days of the first Pet computer produced by Commodore, which had no disk-operating system, since there were no disk drives to be had then. "A decent disk-operating system wasn't developed until Pet Basic 4.0," Charpentier said. The C-64 doesn't use this advanced version of Basic because "it would have required more ROM than we could put in." Instead, the machine uses a far more primitive version of Basic, borrowed from the VIC-20.

The most glaring omissions from the Commodore 64's Basic are commands to control the sound and graphics chips. "There was never any intention that the system would have built-in software to control the graphics or the sound," said Yannes. "That was an obvious part of the Commodore philosophy: you don't waste money on things that make the product more expensive and that the majority of buyers aren't going to use."

"Commodore has always paid lip service to software," Charpentier said. "They do enough to get by and then rely on outside sources to fill the gap. Commodore was an extension of Jack Tramiel, and to him software wasn't tangible—you couldn't hold it, feel it, or touch it—so it wasn't worth spending money for."

The C-64 designers had plans of their own for changes which haven't yet been implemented. They used the VIC-20 case to shortcut development, but they did have a new case designed that they intended to substitute in eight to ten months. "It was thinner in front and had more of a wedge shape to it," said Charpentier. "I always thought the VIC-20 case looked clunky." A few new features were also to be added to the machine.

The fact that these changes have not been made has not really



Small changes in the internal design of the Commodore 64 during its production run of more than 5 million units have slashed manufacturing costs to an estimated one-third of the original \$135. Increased volume is one cost-saving. Another is the shifting

of the video chip, the sound chip, and the central processor from ceramic to plastic packages. The most surprising change is a move from automatic to manual assembly; the axial-lead capacitors along the bottom left and at other locations on the old circuit

affected the success of the C-64, asserted Brian Dougherty, president of the Berkeley Softworks of Berkeley, Calif. "This machine has the best graphics-display capability of anything that has yet been done for a TV screen," he said. "It came close to being an awesome system."

The albatross of the C-64

The one major flaw of the C-64 is not in the machine itself, but in its disk drive. With a reasonably fast disk drive and an adequate disk-operating system (DOS), the C-64 could compete in the business market with the Apple and perhaps with other business computers. With the present disk drive, though, it is hard-pressed to lose its image as a toy.

"Business-oriented software does not look as good as it should on the C-64," said Dougherty, "not because the base system isn't good enough but because the disk-drive access is too slow; any business application requires a lot of disk access." And since the DOS is so weak and lacks features such as the automatic creating and maintaining of directories and keeping track of files, "it is a lot more difficult to develop business software" for the C-64, he added. "Somebody like Lotus [Development Corp., manufacturer of a popular integrated business-software package] is facing a lot more difficulty than they did developing software for the IBM PC, because the PC has an operating-system structure that just doesn't exist for this product."

"The basic input-output system in ROM is done fairly cleanly, according to operating-system principles. All the routines that need to be there are there, but there should also be a facility for automatically reading the first track of the disk and booting a more sophisticated operating system into memory."

The disk drive does in fact have a facility for automatically reading the first track of a disk, according to Nelson of Epyx, but because it is not documented, it is not widely used.

How did the Commodore engineers botch up the disk drive?

"The disk drive? What a gory story!" Charpentier said. "That

had to do with the one marketing input we did accept. Marketing said the C-64 had to be compatible with the VIC-20."

The disk drive on the VIC-20, Charpentier explained, was a direct descendant of the drive on the Commodore Pet computer. "We had a parallel IEEE-488 bus for the Pet disk drive; and we used a strange format in the early days in order to pack data: on the outer tracks, the disk moves faster than on the inner tracks, so the data rate changes in order to pack more data out there. It was a nice idea, but it was clumsy, in that you're incompatible with everyone else. That is the nut this all started with."

The VIC-20 disk drive was designed in Japan, and the bus was changed from parallel to serial to economize. The drive is very slow because the serial-interface portion of the interface controller (the 6522) used on the VIC-20 does not work. As a result, the 6502 processor sends out one bit at a time under direct software control, rather than sending a byte to the peripheral controller and letting it take care of clocks, data, and hand shaking.

"We fixed the serial-shift register for the 6526 [the interface controller used on the C-64 to replace the 6522] but we couldn't use it, because the disk drive had to be compatible," Charpentier continued. "And, because the processor in the C-64 is turned off for 40 microseconds every 512 microseconds, it kept missing the interrupt signals that control the disk interface. The solution: slow the drive down even further."

Speed was the only factor that could have been changed without incurring the cost of radical changes in the disk drive; if the C-64 designers had been able to forgo VIC-20 compatibility, they could have done much better. And in the end, of course, the C-64 was not compatible with the disk drive of the VIC-20 anyway.

"It all stemmed from the 6522 serial interface not working," Charpentier concluded.

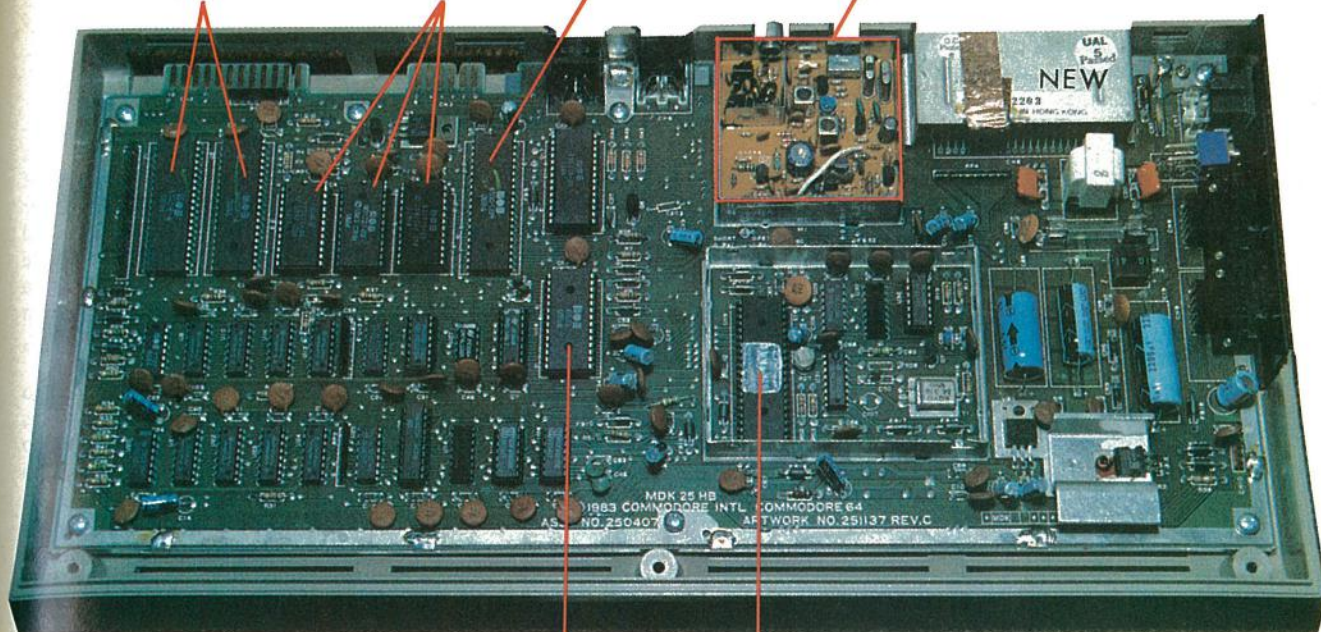
Other manufacturers are trying to solve the disk-drive problem of the C-64. Some are selling their own disk drives for use with the C-64, but they have found it difficult to break into a market that Commodore controls. Others are selling programs that can

6526 peripheral
interface controllers

ROMs for monitor,
Basic, characters

6510 microprocessor

RF modulators



6581 SID (plastic)

6567 video chip (plastic)

board have been replaced by radial-lead capacitors that must be inserted by hand. In addition, the radio-frequency modulator at the upper right of the circuit board has gained parts, showing that it is cheaper for the company to buy discrete components

and assemble them than to buy a single part that performs the function. In addition to the radial-lead capacitors, another sign of hand assembly is the white lines printed on the new circuit boards to guide chip placement.

be loaded into RAM to speed up disk-transfer rates, but this option applies only to use of the C-64 for programming in Basic; prepackaged software loaded into the system knocks the programs out of RAM. Epyx has released a ROM-based program to speed disk transfer fivefold. According to Nelson, the program will work with any disk-based software. "That's possible," said Charpentier. "The C-64 disk drive has its own microprocessor and memory, and you can ship information to it to change the operating system and to speed up the serial-bus protocol."

The standard software for the C-64 disk drive can read a 256-byte sector every 2.5 revolutions of the disk, or 512 bytes per second. Programming tricks used by Epyx have yielded transfer rates of up to 2.5 kilobytes per second. (Among other slow disk drives, Atari's 810 transfers 1000 bytes per second, and the Apple II disk drive runs at a maximum of 15 000 bytes per second.)

Clever design shaves pennies

The Commodore team had worked hard to ensure a low initial production cost for the C-64. Vertical integration helped keep parts costs down, and Winterble vetoed ideas for added features. The designers themselves believed in counting pennies; Charpentier said he had "learned a lot" from Jack Tramiel, who was "a real stickler for cost. We agonized over every transistor."

With a production cost of \$135 and a retail price of \$595 at its introduction, the Commodore 64 would have a retail markup comfortably above the industry standard of one to three (perhaps one to two when volumes climb). But Commodore has continued to find ways to reduce both the retail price, now \$149, and the production cost, believed to be between \$35 and \$50 today.

Commodore would not confirm this cost figure. Dougherty of the Berkeley Softwareworks estimated the costs of the C-64 parts, based on his experience at Mattel and Imagic:

- Three ROMs at \$1 each.
- Eight dynamic RAMs at \$1.85 each.
- SID (sound) chip and VIC (graphics) chip at \$4 each.

- RF modulator package, \$3
- 6510 microprocessor, \$1 to \$2.
- A handful of TTL, buffers, power regulators, and capacitors for a total of \$5.
- Keyboard, less than \$10.
- PC board, \$1 to \$2.
- Plastic case, \$1 to \$2.
- Power supply and miscellaneous connectors, \$5 to \$10.
- Packaging and manual, \$1 to \$2.

In further cost-cutting moves, Commodore has replaced the TTL with less expensive custom chips and has found ways to increase yields on the sound and graphics chips.

Not all cost-cutting steps were without detrimental effect on the system. The 6567 video chip was originally packaged in ceramic, even though plastic is much cheaper. "We tried all kinds of things," Charpentier said, "copper-lead frames, metal inserts in the package, and an ungodly number of other things to get a plastic package, because a ceramic package is very expensive. But we couldn't find a package that ran cool enough. Before I left Commodore, I listed some ideas to reduce the power, but half of the chip was still running at an 8-MHz clip. Without a major redesign, you could not get the power out."

The chip is now encased in plastic. Commodore engineers said that in addition to redesigning the chip, they have developed a plastic package that can dissipate heat as well as ceramic.

Since the change to plastic packaging, a number of new problems with the C-64 have been reported—problems that are machine-specific or temperature-dependent. One failing that is widely known throughout the software industry is a tendency for the chip to lock up when a program switches graphics modes—from displaying a bit-map to displaying characters [see p. 50]. "A game we were producing started to fail on the production line," said Nelson of Epyx. "Then we'd test [defective disks] again and they'd work. It depended on which machine we put them on and what the temperature was." Charpentier speculated

that some circuit lines had been left floating in the original design; poor processing and high operating temperatures would stress chips to the failure point.

Another problem that has begun to emerge in some C-64s is faulty horizontal scrolling of bit-mapped screens. In a recent review of the Summer Games, an Epyx sports game for the C-64,

Building a synthesizer on a chip

"I'd worked with synthesizers, and I wanted a chip that was a music synthesizer," said Robert Yannes, the sound-chip designer and system architect for the Commodore 64. He began working on the chip in the spring of 1981, assisted by two draftsmen and one CAD operator. "Nobody else had a good idea of what we were doing," he said. "Beyond basic research, which I'd been doing since high school, the actual design took only about four or five months."

The major differences between his chip and the typical video-game sound chips, Yannes explained, were its more precise frequency control and its independent envelope generators for shaping the intensity of a sound. "With most of the sound effects in games, there is either full volume or no volume at all. That really makes music impossible. There's no way to simulate the sound of any instrument even vaguely with that kind of envelope, except maybe an organ."

Although it is theoretically possible to use the volume controls on other sound chips to shape the envelope of a sound, very few programmers had ever tackled such a complex task. To make sound shaping easy, Yannes put the envelope controls in hardware: one register for each voice to determine how quickly a sound builds up; two to determine the level at which the note is sustained and how fast it reaches that level; and one to determine how fast the note dies away. "It took a long time for people to understand this," he conceded.

Yannes also put a lookup table in the hardware to convert data from musical notes to the equivalent frequencies, but that required excessive silicon. "I took that out," said Charles Winterble, who was worldwide engineering director at Commodore during the C-64 design.

The precise capabilities of the sound chip are not clear even today, largely because of incorrect specifications having been written when the chip was first designed. "The spec sheet got distributed and copied and rewritten by various people until it made practically no sense anymore," said Yannes. An example of the faulty documentation is the claim that the chip can logically AND several waveforms. The chip can produce either a triangular wave, a sawtooth, or a square wave, depending on which bit is set in one of its control registers. "There is no interlock to make sure that if one bit is on, the others are off," Yannes said. "That would have taken too much silicon." So if more than one waveform is elected, the internal nodes of the output multiplexer are discharged, and what emerges is the minimum of amplitudes.

Another error in the spec sheet concerns the on-chip filters. "The filter was the last thing that was worked on," Yannes acknowledged. "I ran out of time. The computer simulation said, 'This will not work very well'—and it didn't." The filter may have been mediocre, but the equations describing how it worked were just plain wrong, Yannes recalls. "They didn't hang together. No one gave me a chance to correct them."

Confronted by these incorrect equations, most software designers use only the better-documented features of the sound chip.

Some of the Japanese programmers who wrote early game software for the C-64 took a different approach, Yannes noted: "They wrote it according to the spec. It didn't matter to them whether something worked or not. As a result, programs made sound effects you couldn't hear."

—T.S.P. and P.W.

the San Jose, Calif., *Mercury News* criticized the "flashbulb effect" noticeable during the swimming event. That, according to Nelson, is not an "effect"—the C-64 used by the *Mercury News* reviewer simply doesn't do horizontal scrolling properly.

The freedom ended

Although the machine has its flaws, the designers of the Commodore 64 believe they came up with many significant advances because of the freedom they enjoyed during the early stages of the project. The design team was autonomous—they did their own market research, developed their own specifications, and took their baby right up through production. But as soon as the production bugs were worked out and Commodore knew it had a winner, the corporate bureaucracy, which until then had been on the West Coast dealing with the VIC-20 and the Pet computer, moved in.

"At that point, many marketing groups were coming in to 'help' us," Winterble recalled. "The next product definition was going to be thought up by one group, and another group was to be responsible for getting things into production, and Al's group would do R&D on chips only."

"If you let marketing get involved with product definition, you'll never get it done quickly," Yannes said. "And you squander the ability to make something unique, because marketing always wants a product compatible with something else."

Charpentier summed up their frustration: "When you get many people involved in a project, all you end up doing is justifying yourself. I knew the Commodore 64 was technically as good and as low-cost as any product that could be made at the time, but now I had to listen to marketing people saying, 'It won't sell because it doesn't have this, it can't do that.'"

"The freedom that allowed us to do the C-64 project will probably never exist again in that environment."

Postscript

Of the original Commodore 64 design team, only Robert Russell remains at Commodore; he is currently a design manager. The rest of the team—Albert Charpentier, Robert Yannes, and Charles Winterble, along with David Ziembicki and Bruce Crockett, who helped debug the project and bring it into production—left Commodore in the spring of 1983 and formed the company Peripheral Visions. Their plan was to design another computer. To obtain working capital, they took on a contract from Atari—to design a keyboard for the Video Computer System, which was previewed but then, as the video-game market crashed, never released. Peripheral Visions was sued by Commodore, which said the VCS keyboard was a project that belonged to Commodore. At this writing, the suit had not been resolved.

Peripheral Visions has been renamed Ensoniq and will soon release its first product, a music synthesizer. Albert Charpentier is currently vice president of engineering at Ensoniq, Bob Yannes is senior designer, and David Ziembicki is manager of production control. Charles Winterble left Peripheral Visions and is now group vice president of electronics at Coleco Industries Inc. of West Hartford, Conn. And Jack Tramiel is currently chief executive officer of Atari Corp., the company that was Commodore's chief competitor in the home-computer market before losing nearly a billion dollars in competing against the C-64.

To probe further

Computer bookstores carry a wide selection of books on the C-64. Among those recommended are *What's really inside the Commodore 64*, by Bilton Bathurst, published by DataCap of Feneur, Belgium, and *The Anatomy of the 1541 disk drive*, by Lothar Englisch and Norbert Szczepanowski, published by Abacus Software of Grand Rapids, Mich. Computer magazines such as *Creative Computing* and *Compute!* review software and publish programs that run on the C-64. The Commodore 64 itself can be found at most large retailers. ♦