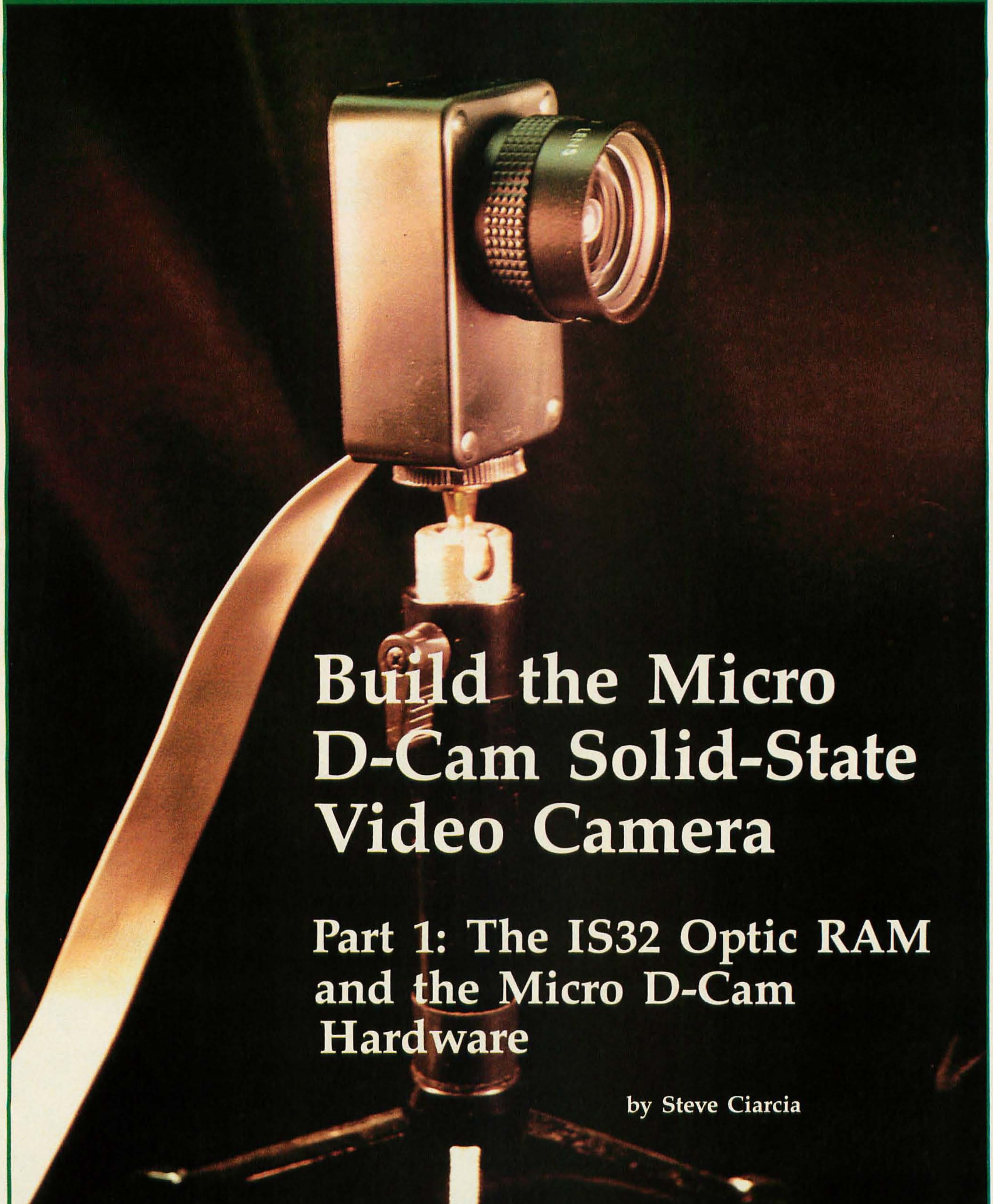


Build the Micro D-Cam Solid-State Video Camera

Part 1: The IS32 Optic RAM and the Micro D-Cam Hardware

by Steve Ciarcia

A 64K-bit dynamic RAM chip is the visual sensor in this digital image camera

If you've followed the activities in the Circuit Cellar for any period of time, you have probably realized that my writing a monthly column is just an excuse to investigate and experiment with whatever I find currently fascinating. One of my longtime fascinations has been the input of visual data to a computer. While I've presented interfaces that allow computers to determine direction and measure distances, receive a variety of sensory inputs via touch or remote signal, and even speak their minds through voice synthesis, until now I have not presented a project that enables a computer to see.

There was a time when most computers communicated via klunky teletypes at 10 characters per second. Improving output technology—high-speed video graphics displays, dot-matrix printers, and voice synthesis—has vastly improved the computer's ability to communicate results and conclusions to its user. Except in specialized applications, however, input technology has been discouragingly static. We still plod along using keyboards or mice as the primary input device even when the input data may be graphical.

So much of our existence involves visual recognition that it only stands to reason that the potential applications of computers would be enhanced if a versatile sensory-input channel were available to the machines. Then, instead of spending hours entering digital coordinates from a picture or map into a computer using a keyboard, you could easily use a ''computerized camera'' to make a visual snapshot of the material, instantly producing a digitized picture. Once you had such a picture in the computer, it could be interpreted, enhanced, or stored as the application might dictate.

Photo 1: *In this prototype, the Optic RAM is mounted inside a light-tight box with a C-mount lens focusing light onto one of its cell arrays. The ribbon cable leads straight from the Optic RAM to the interface card in the Apple.*

Computers and Vidicon Cameras

Computerized-image cameras are not new, but up to now they have always been too expensive for widespread practical use and casual experimentation. Most of the computer image-input devices currently available use a conventional black-and-white television camera as the image sensor. The camera's video output must be converted to digital logic levels for the computer: a difficult task, because the output, produced using a Vidicon-type pickup tube, is a high-frequency analog signal divided into 30 complete frames of picture information transmitted and scanned each second (or 25 frames for most TV systems outside North America).

Most high-quality TV-camera interfaces convert the analog signal for computer processing through "frame grabbing," in which one of the frames is sampled, digitized, and stored during a 1/30-second frame-scan interval. In these sophisticated visual sensing systems, a high-speed A/D (analog-to-digital) converter digitizes the analog signal in real time at sampling rates exceeding 5 megahertz (MHz) and stores the PCM (pulse-code modulated) data in a high-speed buffer made of semiconductor memory. Because they operate so fast, such units are insensitive to camera motion and fast scene changes.

In less sophisticated TV-camera interfaces, the designer has assumed that the camera and the object in its view will remain still long enough for the picture to be processed by slower, cheaper circuitry. When the TV picture is stationary, all frames in the signal are identical, so a sequential line-sampling technique is often employed. In units of this type, a low-speed A/D converter (sampling from 100,000 to 1,000,000 times per second) operates in bursts of activity shorter than a frame interval, with each successive period of activity, or *sampling window*, triggered at a slightly later time during the frame interval by line and pixel (picture element) position counters. In between the sampling windows, the support circuitry has time to store the digitized information and get ready for the next burst of activity. As a result, the interface assembles the single image from pieces snatched from many frames.

High-speed frame grabbers generally cost more than $10,000, while the slower units cost somewhat less, depending upon speed and resolution. You can expect a 256 by 256 pixel-resolution low-speed interface, the kind used in the computer systems often seen at conventions printing images on T-shirts, to cost between $500 and $1000; half of that price is for the camera and lens.

Solid-State Arrays to the Rescue?

The problem with the Vidicon-type camera is that it is an analog device, which must be adapted to work in digital applications. It would be far better to have a computer video camera that is inherently digital and dispense with the analog-to-digital conversion. Why not use semiconductor devices, the outputs of which are digital signals that change as a function of light level?

The barrier has been price. A variety of semiconductor optical sensors, such as photodiode and charged-coupled-device (CCD) arrays, fill the bill nicely. When I first started to think about building a solid-state image camera, I thought I could just order a 256 by 256 pixel CCD array and add a few binary-counter chips for a quick project. This idea evaporated quickly when I discovered that CCD arrays cost from $800 to $2000, depending upon the number of bad pixels you get.

My success with photodiode arrays wasn't much better. It seems that for about $100 you can buy a 128 by 1 or 256 by 1 array, but arrays more than one element wide are hard to find. To create a 256 by 128 or 128 by 128 picture, I would have needed to devise an optical, mechanical, or electronic way to move the array across the image plane, or move the

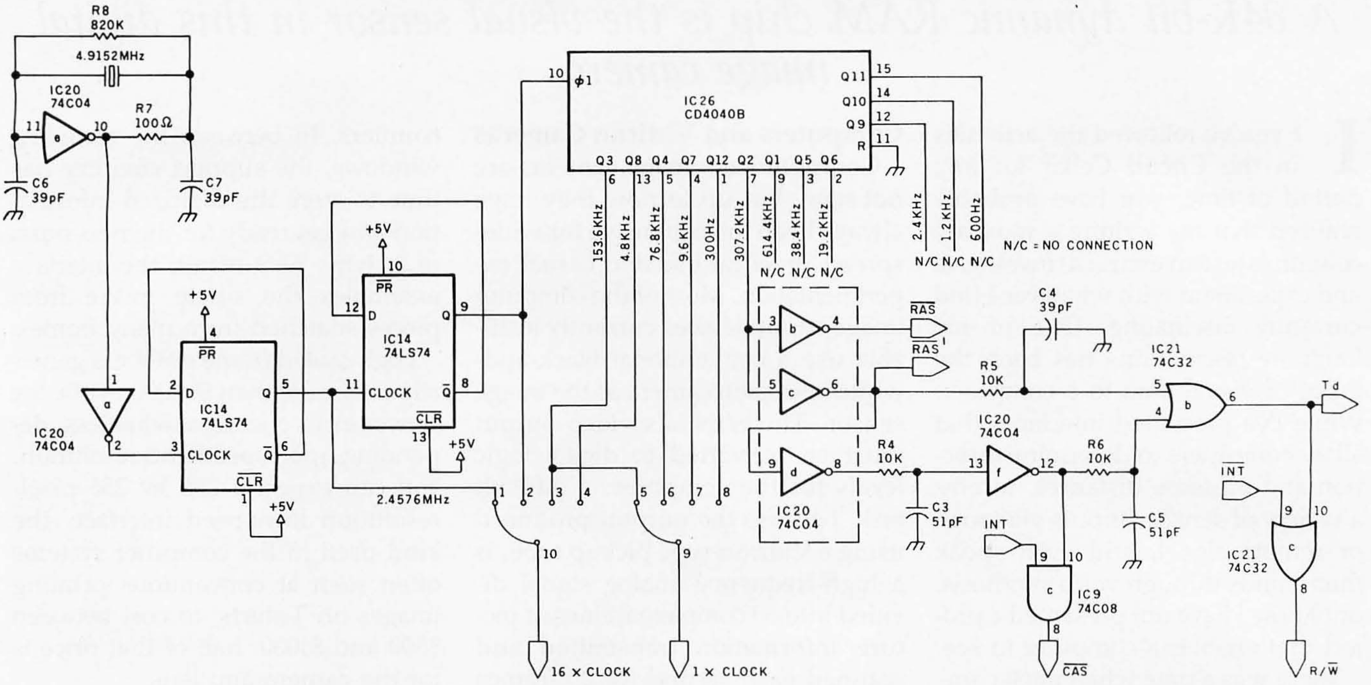


Figure 1: *A schematic diagram of the timing-generator circuit, which contains a CMOS oscillator circuit to generate the fundamental clock rate. This signal is divided down to produce the frequencies for various possible output data rates and controlling the IS32. The data-rate-clock signals control the sequence of operation of the interrupt generator and the transmitter circuit.*

image across the array, stopping periodically for the values of the picture elements to be registered and stored. After tentatively sketching a screw drive and its associated control electronics, I gave up in disgust at the thought of how hard it would be to build.

If Memory Serves...

I had almost decided to join the bandwagon and use a Vidicon camera when I remembered experimenting with the optical sensitivity of dynamic RAM (random-access read/write memory) chips back in 1976. I had even seen a design published for a 64 by 64 pixel resolution camera that used a type-1103 1K-bit dynamic RAM chip as an optical sensor. If the bit-storage cells in those early dynamic RAMs were light-sensitive, could the newer ones be also? If I popped the top off a 64K-bit dynamic RAM chip, wouldn't I find a 256 by 256 array?

Well, yes and no. I took the lids off a few brands of 64K-bit dynamic RAM chips, and it does appear that they are light-sensitive. The problem, however, is that they were designed only as memory devices and not optical arrays. To my knowledge, none of the 64K-bit dynamic RAMs on the market are configured as an orthogonal array laid out 256 elements long by 256 wide. In fact, most have 4 or 8 sections of 16K or 8K bits, and many include redundant sections that can be wired in to replace bad sections on the chip. The bit addresses don't proceed linearly through the chip either; one bit may be in the upper-left corner and the next bit in the lower-right corner.

Just as I was about to abandon all hope, I found an unconventional dynamic-RAM manufacturer that has recognized the light-sensing potential of its 64K-byte device. Micron Technology Inc., of Boise, Idaho (certainly an unconventional place to make integrated circuits), produces a dynamic RAM chip that has its memory cells laid out in only two sections, both of which are 256 by 128 cells, as shown in figure 7 on page 31. With this configuration, the chip can easily be used as an optical sensor. One specially tested 64K-bit dynamic memory device, called the IS32 Optic RAM, comes in a package with a see-through quartz lid (see photo 4 on page 30).

Per pixel, the Micron Technology IS32 Optic RAM costs 1000 times less than the earlier generation of image-sensing chips such as the CCD. The Optic RAM's spectral sensitivity is generally the same as that of other silicon-based light-sensing media, but its bit-for-bit uniformity is not as good as CCDs. Nevertheless, the Optic RAM can bring capabilities to your computer that were previously available only to large industrial users.

Build the Micro D-Cam

This month, using the IS32, I'll show you how to build a relatively low-cost digital image camera I call the Micro D-Cam (see photo 1). Its resolution of 256 by 128 pixels is adequate for many applications in graphics, pattern and character recognition, robotics, process control, and security. (Of course, the output of the Micro D-Cam is a digital signal; it cannot be used to directly drive a composite-video monitor.) I've put together versions of the Micro D-Cam for use with the Apple II computer (II-Plus and IIe, see photo 3, page 30) and the IBM Personal Computer; however, the Micro D-Cam is serially interfaced and requires only five wires for connection, so I'm also working on an RS-232C version that can be attached to any computer that has a serial port.

The Micro D-Cam project is rather complex, so I'll present it in two parts. This month I'll explain how the IS32 Optic RAM and the Micro D-Cam hardware work. Because appropriate software is vital to the success of this project, next month I'll include a lengthy listing of a typical control program for use with the Apple II-Plus version of the Micro D-Cam. We'll also look at some of the Micro D-Cam's capabilities.

IS32 Optic RAM

The IS32 from Micron Technology is an all-digital image-sensing device. Its pertinent characteristics are shown in table 1.

The IS32 contains 65,536 (64K) light-sensitive memory cells laid out in two planar, rectangular arrays of 32,768 elements, each a matrix of 128 rows and 256 columns. The two arrays are separated by an optically nonsensitive "dead" zone about 25 elements wide. To avoid having a gap in the image or using complicated optical systems to eliminate it, only one of the arrays is usually used as an image sensor. Each of the memory elements in the matrix can be accessed randomly when the control circuitry strobes in the appropriate row and column address of the element being accessed.

Theory of Operation

An image camera built around the IS32 focuses reflected light from the viewed object and passes it through a lens onto one of the 32,768-element arrays. When an individual element is struck by photons of light, the capacitor in the cell, which is initially precharged to a fixed voltage, begins to discharge toward zero volts. The capacitor discharges at a rate proportional to the light intensity throughout the duration of the exposure.

After the exposure interval has elapsed, the circuitry reads the element by addressing it as a memory cell. During the cell access, sense amplifiers within the IS32 read the capacitor's voltage value and compare it to a fixed threshold voltage. If the potential is above the threshold, the picture element is deemed to be black; if the potential is too low, the picture element is declared white. The D_{out} pin of the Optic RAM is set to a logic 1 or 0 during the corresponding bit interval as a result of this decision. The raw "gray scale" of the IS32, therefore, has only two shades, white and black. (We'll see how to compensate for this shortly.)

All dynamic-memory devices require refreshing for operation; the charge representing the data stored in each cell capacitor will leak away if left alone (exposure to light merely hastens the leakage). The charge must be sensed and brought back to the nominal voltage for the logic state it represents. This can happen when the computer reads a bit value from the cell, but more frequently it happens when circuitry external to the memory chip periodically activates the cell's address just for the purpose. (Many memory chips, including the IS32 Optic RAM, can refresh their cells a whole row at a time.) The IS32 can be used this way as a regular memory device can, but in optical service there is a twist in the refreshing. The chip is light-sensitive only when it is not being refreshed; the key to using it in a camera is to carefully control its sensitivity by performing the refresh operation in a special way.

In the beginning of an image-sensing cycle, the Micro D-Cam's circuitry addresses all the cells in the active array, filling them with the positive voltages that represent logic 1s. The exposure begins with the receipt of a SOAK command, which is the equivalent of opening the shutter (to allow the array to "soak" in light). Then, after the appropriate exposure interval has elapsed, the control circuitry issues the refresh command, which freezes the states of the memory cells (or pixel cells, if you will). Then the control circuitry activates the interrupt state, during which the value of one cell is fetched and transmitted. Interrupt cycles are continued until all the bits in the array have been transmitted. (The interrupt

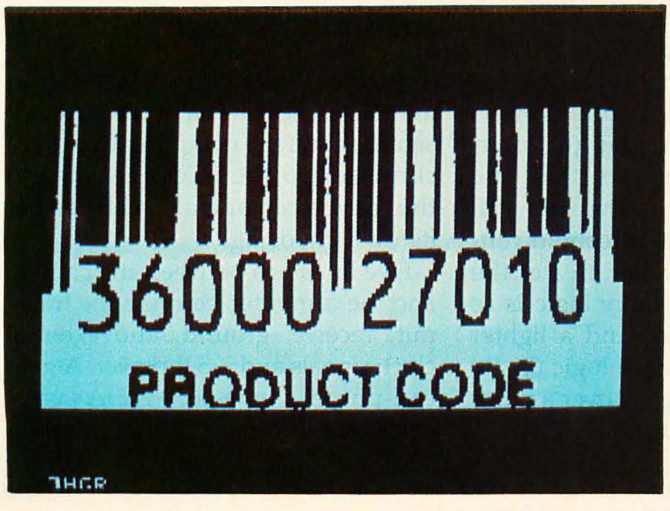


Photo 2a: *The Micro D-Cam can focus on UPC bars. Both photos shown here used the Apple II's high-resolution graphics routines to reproduce the camera's output.*

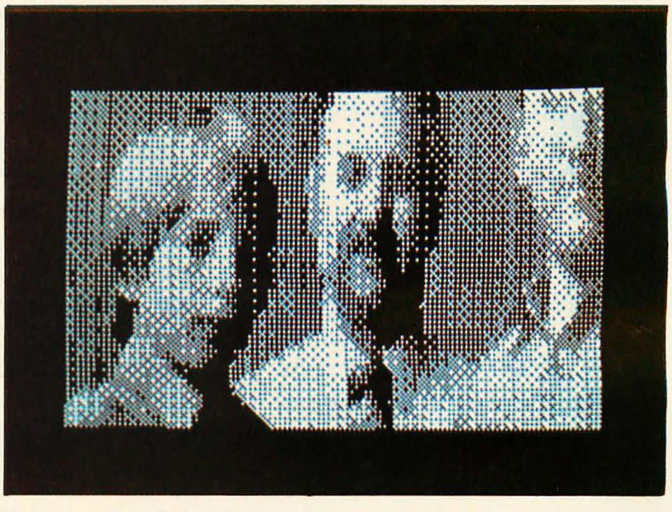


Photo 2b: *The Apple II's display represents gray levels as different densities of white dots. Several different-length exposures are combined to form the gray-scale image.*

mode is not maintained constantly because the IS32 cannot be refreshed during the interrupt.) The Micro D-Cam then causes all the cells to again be set to 1, and the image-sensing cycle starts over.

A white pixel (logic 0) in the output indicates that the capacitor was exposed to a light intensity sufficient to discharge it past the threshold point. A black pixel (logic 1) indicates the light intensity was not enough to discharge the capacitor past the threshold point.

How fast the camera can scan the image varies according to the light intensity. The faster the elements are scanned, or read, the greater the light intensity required. The Micro D-Cam can scan approximately 15 frames per second at maximum speed.

The Optic RAM Resembles Film

The operation of the digital image sensor can be compared to that of a black-and-white film emulsion in a conventional photographic camera. Like the film, the IS32 contains many light-sensitive elements lying in a single plane. The image is focused (optically) on the plane, the user can adjust the aperture (measured in f-stops) and the length of exposure. The aperture is an adjustment of the size of the opening through which the light is allowed to pass on its way to the light-sensitive medium (changed in both cases by mechanically opening or closing an iris). The length of exposure (corresponding to photographic shutter speed) is adjusted in the Optic RAM by the scanning function of the drive electronics. And in the Optic RAM, the film is advanced, so to speak, by refreshing the voltage on all the memory elements.

Also like film, the Optic RAM's elements respond to light in a binary fashion, indicating only black or white, the presence or absence of a certain amount of light during the exposure. However, in a photographic film, the light-sensitive elements (grains of silver-halide compounds) come in different sensitivities and respond to different intensities of light, whereas the IS32's cells all respond at about the same intensity for any given condition. To circumnavigate this limitation with the Optic RAM, varying shades of gray can be recorded by making multiple scans of the same optical image, averaging the results obtained from either changing the sensitivity of the cells, using a different threshold voltage for each scan, or varying the scan rate.

By changing the threshold voltage and keeping both the scanned image and light intensity constant, areas on the Optic RAM where intermediately bright portions of the image fall will give differing output levels. The nominal threshold potential, 2.1 volts (V), can be adjusted though pin 1 (Analog Threshold) on the IS32 from 1.5 V to 3 V, but Micron Technology suggests that gray-scale capability be achieved by varying the scan rate rather than by adjusting the threshold voltage. Changes will be exhibited in the response of the pixels where the image is gray (of intermediate brightness) so that the amount of light striking the cell capacitors is near the threshold voltage. Of course, a darker area of the image will generate more logic 1s as output than logic 0s, and a lighter area will generate more logic 0s. By averaging these outputs over a number of scans, the appropriate shade of gray can be produced in a composite image representation.

1. 128- by 256-element array measuring 5.504 by 1.088 millimeters
2. Element size: 8 microns by 9 microns
3. Vertical center-to-center spacing: 21.5 microns
4. Horizontal spacing: 8.5 microns
5. Spacing between left and right arrays: 150 microns

Table 1: *Specifications of the Micron Technology IS32 Optic RAM, a 64K-bit memory chip that has the extra talent of serving as a digital image detector.*

The Micro D-Cam may not contain a mechanical shutter, but its electronic equivalent is easily controlled by sending the appropriate commands to the control circuitry. The Optic RAM's sensitivity to light varies according to the electrical voltages present on it, allowing for precise continuous control of the Micro D-Cam's exposure values.

Ease of Use

Hooking up the Micro D-Cam to a computer is easy. The unit's control circuitry provides all the requisite timing signals and circuitry to execute commands received from the computer. The Micro D-Cam automatically sequences the Optic RAM so that each image-sensing cell is accessed and the appropriate video information transmitted to the computer for display or processing.

The Micro D-Cam uses a C-mount lens (the type commonly used in 16-millimeter movie cameras and small television cameras) with variable focus. The lens I chose was designed for viewing objects from a distance of at least 18 inches (45 cm); from this distance, the Micro D-Cam can distinguish characters of the size you are now reading. For viewing objects under greater magnification, you can insert a close-up adapter between the lens and its mount to extend the focal length of the lens. (See photo 2.)

The link between the computer and the Micro D-Cam is a TTL- (transistor-transistor logic) level serial interface. The external data-rate clock signal allows the computer to be synchronized to the Micro D-Cam, so the camera can operate at a speed of its own choosing.

Five lines run between the camera and the computer, carrying the transmit, receive, ground, and external clock signals and +5-V power. A general-purpose type-6850 ACIA (asynchronous communication interface

Editor's Note: *Steve often refers to previous Circuit Cellar articles as reference material for each month's current article. Most of these past articles are available in reprint books from BYTE Books, McGraw-Hill Book Company, POB 400, Hightstown, NJ 08250.*

Ciarcia's Circuit Cellar, Volume I *covers articles that appeared in BYTE from September 1977 through November 1978.* Ciarcia's Circuit Cellar, Volume II *contains articles from December 1978 through June 1980.* Ciarcia's Circuit Cellar, Volume III *contains articles from July 1980 through December 1981.*

adapter) buffered chip performs serial-to-parallel and parallel-to-serial data conversion, mating the Micro D-Cam's nonspecific circuitry to the host computer, as illustrated by the Apple II Plus in this article.

Hardware Details: Timing, Refreshing, and Interrupts

The timing-generator circuit (see figure 1), which generates the timing signals for the operation of the Micro D-Cam, contains a CMOS (complementary metal-oxide semiconductor) oscillator circuit that generates the fundamental clock rate. This signal is divided down to produce the frequencies for various possible output data rates and controlling the IS32. The data-rate-clock signals control the sequence of operation of the interrupt generator and the transmitter circuit.

The oscillator circuit emits a fundamental 4.9152-MHz signal, which is buffered by a type-74C04 inverter section (IC20a). This clock signal is divided again by a type-D flip-flop and brought out to a set of data-rate-selection jumper connections. IC26 divides the frequency by increasing powers of 2; these various subharmonic outputs lead to other data-rate-selection jumpers. Jumper connections 5 through 8 select the data rate used in the transmitter and interrupt-generator circuit (figure 5 on page 28), while connections 1 through 4 are 16 × clock signals used in the receiver circuit. The output of IC26's pin 7 drives the Optic RAM's timing circuitry, which generates the familiar $\overline{RAS}$ (row-address strobe), $\overline{CAS}$ (column-address strobe) and R/$\overline{W}$ (read/write) signals as used by most dynamic RAM chips.

When the camera is transmitting data from the Optic RAM, it is in the interrupt mode, and the $\overline{CAS}$ and R/$\overline{W}$ signals are provided to the Optic RAM. When the camera is not transmitting, the interrupt mode is off, and $\overline{CAS}$ and R/$\overline{W}$ are disabled; the active-high interrupt signal INT is low and its complement $\overline{INT}$ is high, so the output of the AND gate driving $\overline{CAS}$ remains high and the OR gate driving R/$\overline{W}$ remains low.

During an interrupt cycle, INT goes high and $\overline{INT}$ goes low, enabling $\overline{CAS}$ and R/$\overline{W}$. The high state of $\overline{RAS}'$ (RAS-bar-prime) passes through a delay line consisting of two inverter sections (IC20d and f) and an R/C (resistance/capacitance) network, and then, combined with INT through an AND gate (IC9c), causes $\overline{CAS}$ to go high. When this happens, the column address is latched into the Optic RAM. At this time the R/$\overline{W}$ signal is still high, so the value stored in the accessed pixel is read out. After another delay period, R/$\overline{W}$ goes low, writing a 1 bit into the accessed cell to restore its charge and make it again able to react to light. When $\overline{RAS}'$ returns low, the interrupt cycle is terminated and $\overline{CAS}$ and R/$\overline{W}$ are disabled.

Command-Receiver Circuit

The serial command line carries commands from the computer to the camera. This data enters the command-receiver section (figure 2) a single bit at a time and is assembled according to the following protocol. The first bit to arrive is the start bit, followed by 8 data bits and then the stop bit. The start bit enables operation of the input shift register and starts the shift-register clock, which is initially low. When the clock goes high, the start bit, always a high level, is latched into the first of eight data positions in the shift register. When the clock goes low, the first data bit arrives at the shift register's input.

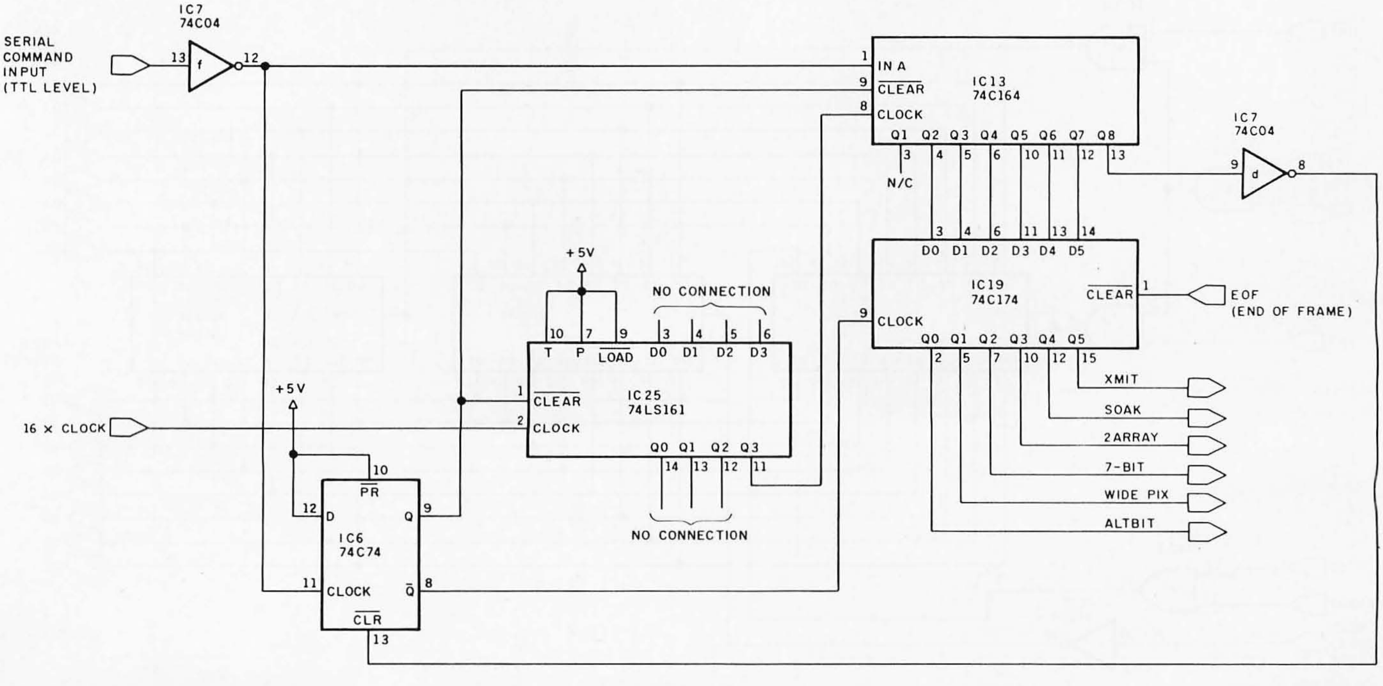


Figure 2: *The serial command line carries commands from the computer to the camera. The data enters the command-receiver section serially and is assembled by the shift register into a decodable word stored in the latch.*

When the rising edge of the clock pulse is detected, the shift register moves the high start bit from position 1 to position 2 and shifts the first data bit from the shift register's input into position 1. As successive bits arrive, each one is shifted into the shift register when the rising edge of the clock pulse is detected.

When the start bit finally reaches position 8, the camera has received the entire command byte, so the first 6 data bits are transferred from the shift register into a latch (a 1-byte memory) called the *command register*. The clock is then disabled and the shift register cleared, leaving the 6 camera-command bits in the command register. The receiver is now ready to accept another command.

Address Registers

The address registers of the circuit (see figure 3) latch the row-address, column-address, and refresh pointers for the Optic RAM addressing. Address registers IC22 and IC16 hold the row and column addresses, respectively, while the third register, IC10, is the *refresh register*.

The first two registers are activated only when the camera is to fetch and transmit a single bit of information from the Optic RAM. (This fetch operation is the interrupt cycle, which, as we saw before, is initiated by the INT signal going high.) The cycle starts on the occurrence of the falling edge of the $\overline{\text{RAS}}$ signal and ends on the next falling edge of $\overline{\text{RAS}}$. When the camera is not fetching in an interrupt cycle, the refresh register is active. This third address register continually increments the row-address value from 0 through to 255. Except during interrupts and exposures, this value passes through to the address lines of the IS32, performing a refresh operation. All three address registers have three-state outputs (that is, their outputs can assume a high-impedance condition, not driving the bus either high or low), and only one register is active at any one time.

An unconventional dynamic-RAM manufacturer has recognized the light-sensing potential of its 64K-bit device.

The selected register drives its data onto a common bus called the *present-address* bus. The present address passes through the descramble-and-soak circuitry (which will be discussed shortly) to the Optic RAM, where it is used to select a row or column. The present-address bus also connects to the address circuit, where a value of 0, 1, or 2 (depending upon software-selected options) is added to the present-address value. The resulting sum is driven out of the adder onto the *next-address* bus, which connects to the inputs of each of the address registers. The value on the next-address bus is latched into the selected address register, and then that register is disabled.

The array-selection circuit simply selects whether one or both of the IS32's cell arrays are to be used. If $\overline{\text{2ARRAY}}$ is high, the output of the OR gate (IC21, pin 11) is always high, and the row-register value (IC22) will never be less than 128, so only the second array (rows 128 to 255) will be addressed and transmitted. If $\overline{\text{2ARRAY}}$ is low, however, the OR gate will appear transparent and the value on the next-address-bus line D7

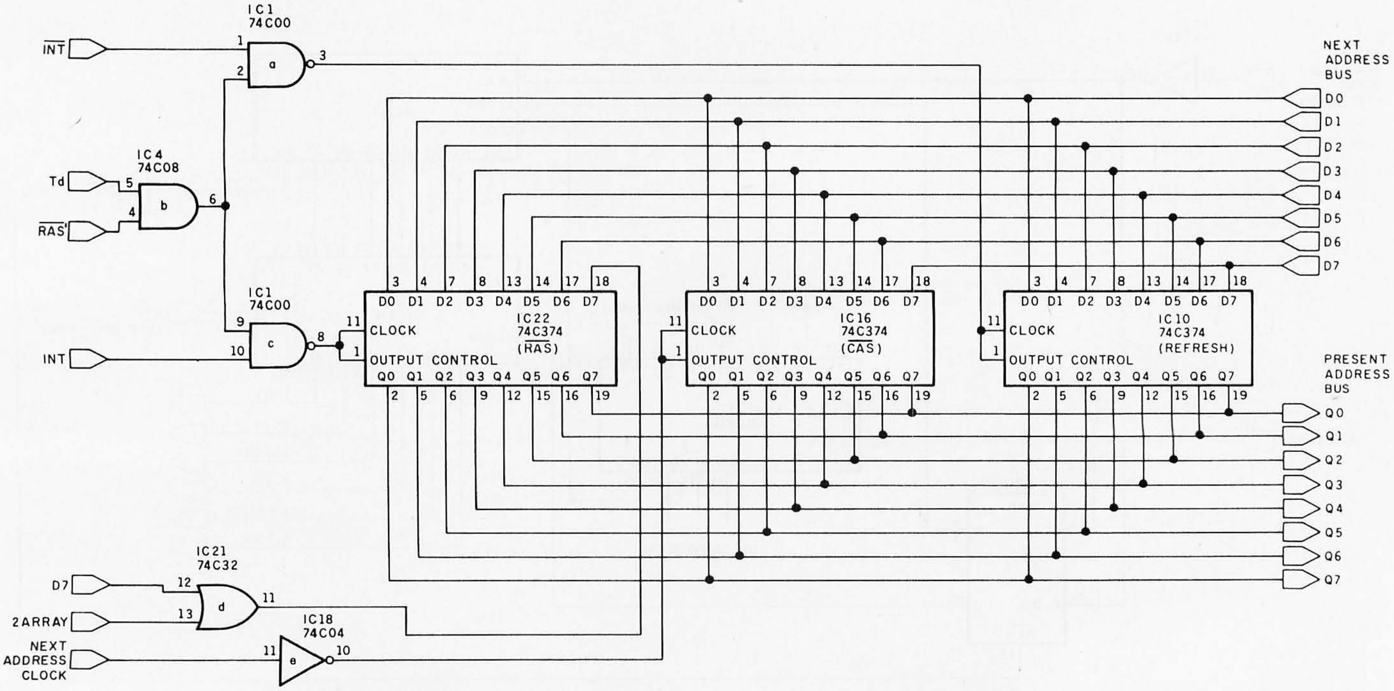


Figure 3: *This section of the circuit latches the row-address, column-address, and refresh pointers for the Optic RAM addressing. Address registers IC22 and IC16 hold the row and column addresses, respectively, while the third register, IC10, is the refresh register.*

will be driven onto IC22. This means all addresses from 0 to 255 will be selected and the values in both arrays will be transmitted.

Address Descramble and Array Soak

The internal circuitry in the Optic RAM scrambles the row and column-address values when accessing a cell. (After all, the IS32 chip was designed for use only as a memory device, not as an optical sensor.) But because element location is a critical issue in optical work, the address-descramble circuit (see figure 4, below) unscrambles the values into a new address, which the Optic RAM decodes to access the desired pixel.

Charged with the task of transforming the data from the address registers into a new address, which the Optic RAM decodes to access the desired pixel, the descrambling circuit consists of two inverters, three exclusive-OR gates, and a multiplexer (IC11). The inverters and exclusive-ORs do the actual descrambling on the row and column addresses; the multiplexer selects between the descrambled row and column addresses at the appropriate times and transmits the address to the Optic RAM.

The multiplexer uses $\overline{RAS}$ to determine which address is selected. If $\overline{RAS}$ is low at the multiplexer's SELECT input (IC11, pin 1), the descrambled row addresses (on the B inputs) are selected. When $\overline{RAS}$ is low, the A inputs, or descrambled column-address inputs, are selected.

The purpose of the $\overline{SOAK}$ circuit is to prevent the refresh addresses from reaching the Optic RAM during the exposure cycles. (Remember, the Optic RAM is light-sensitive only when it is not being refreshed.) During periods when INT is inactive-low (with the refresh register therefore active) and $\overline{SOAK}$ is active-low, the

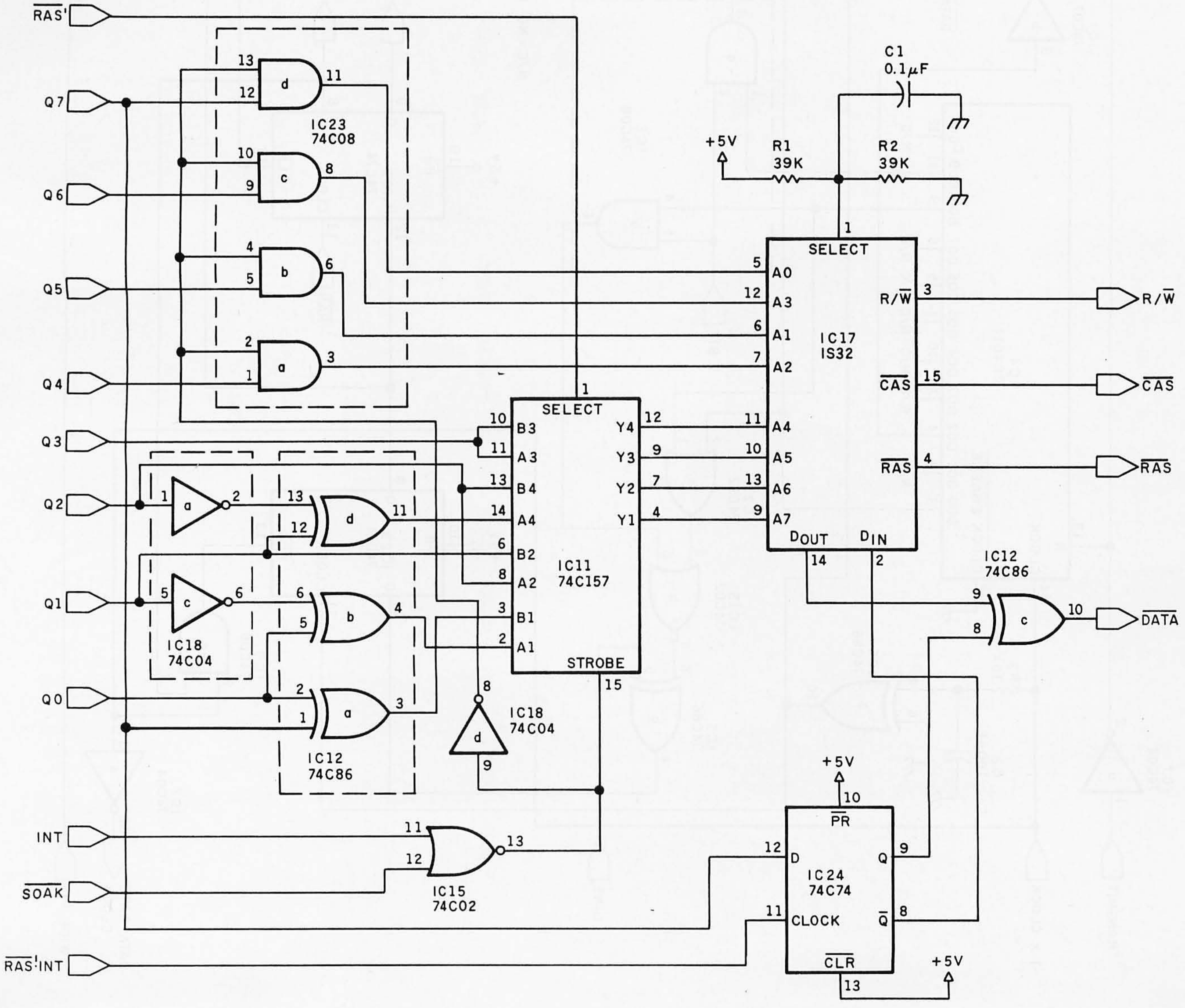


Figure 4: *Consisting of two inverter sections, three exclusive-OR gates, and a multiplexer, the address-descrambling circuitry undoes the internal address scrambling done by the Optic RAM. The soak circuit makes the Optic RAM light-sensitive by depriving it of refresh cycles.*

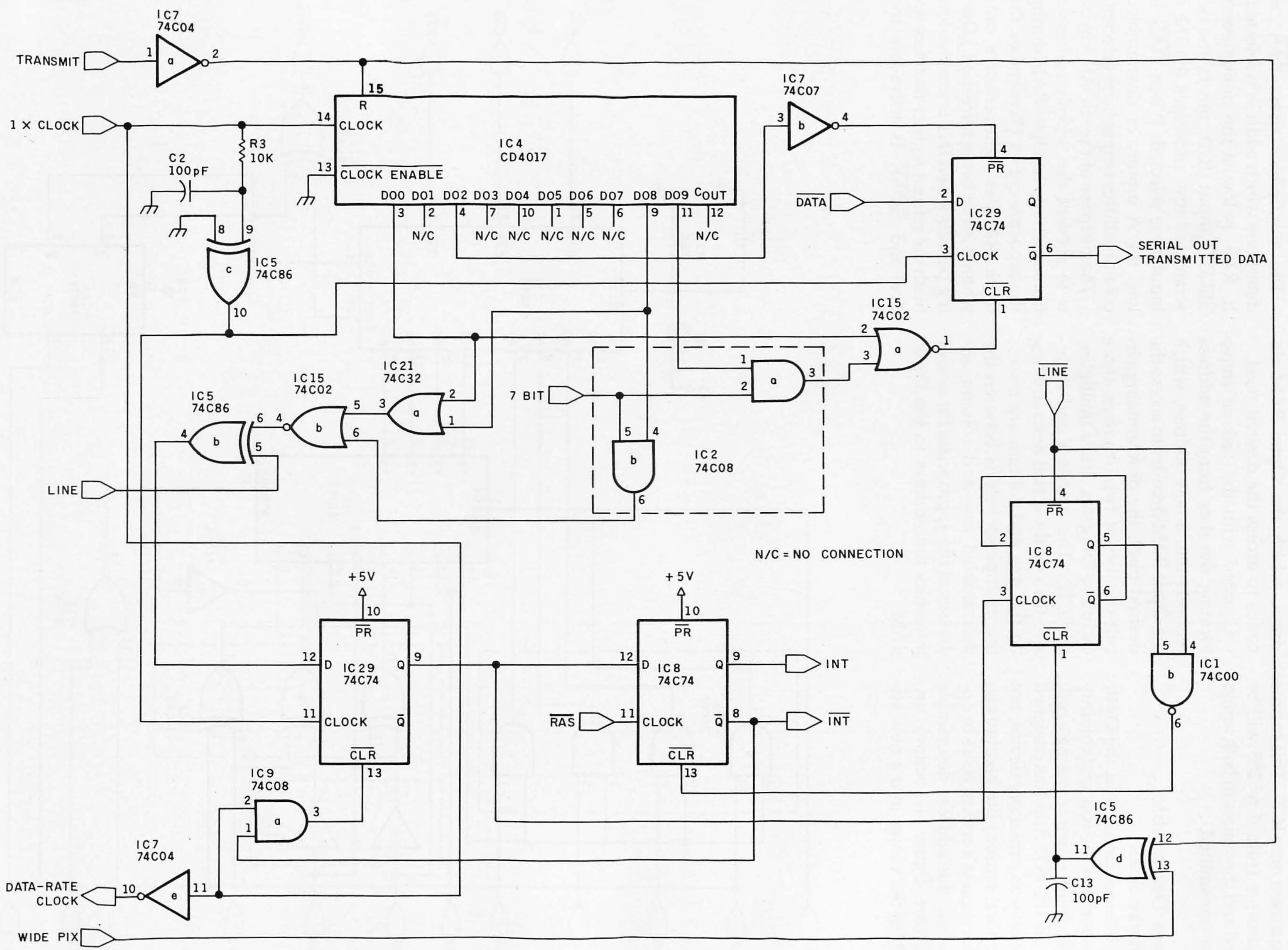


Figure 5: *The circuitry for transmitting the pixel data to the host computer and for generating the interrupt states that allow data to be read from the pixels in the IS32.*

output of NOR gate IC15d is high. This sets the multiplexer's enable input high and drives the multiplexer's outputs low. The high NOR-gate (IC15d) output also forces a low state at the inverter output IC18d, which forces the outputs of the four AND gates IC23a, b, c, and d low. These AND gates stand between the present-address bus and the IS32's four low-order address inputs. Thus, the Optic RAM's address inputs remain low, and the refresh function is performed on only address 0. When $\overline{\text{SOAK}}$ goes inactive-high, the multiplexer and AND-gate outputs are enabled and the refresh addresses reach the Optic RAM so that the entire chip is refreshed, making it insensitive to light.

Transmitter and Interrupt-Generator Circuit

This circuit, shown in figure 5, transmits the pixel data serially to the host computer, inserting start and stop bits where appropriate, and generates the INT and $\overline{\text{INT}}$ signals for fetching the pixel information from the IS32.

At the heart of this circuit is the ripple counter, IC4, enabled when the Micro D-Cam has been commanded to transmit data. It inhibits the interrupt generator when start and stop bits are being transmitted (preventing accessing of the Optic RAM) and enables the interrupt circuit when it is transmitting data. The transmitter's frequency is determined by the data-rate clock. During each clock cycle only one start, stop, or data bit is transmitted.

The output of the Micro D-Cam is a digital signal; it cannot be used to directly drive a composite-video monitor.

The interrupt generator is enabled by both the ripple counter (IC4) and the data-rate clock, but the interrupt cycle itself is clocked by $\overline{\text{RAS}}$. Because the purpose of the interrupt cycle is to fetch a single pixel for transmission, only one pixel can be transmitted on each clock cycle. The rising edge of the data-rate clock enables the interrupt circuit. The next falling edge of the $\overline{\text{RAS}}$ waveform initiates the interrupt cycle, causing a pixel to be read from the Optic RAM. The $\overline{\text{INT}}$ signal feeds back into the interrupt circuit, resetting the interrupt enable.

When $\overline{\text{RAS}}$ goes low again, the interrupt cycle is terminated. The next falling edge of the data-rate clock enables the interrupt circuit again (unless a start or stop bit is to be transmitted). Thus, only one pixel is transmitted during each data-rate-

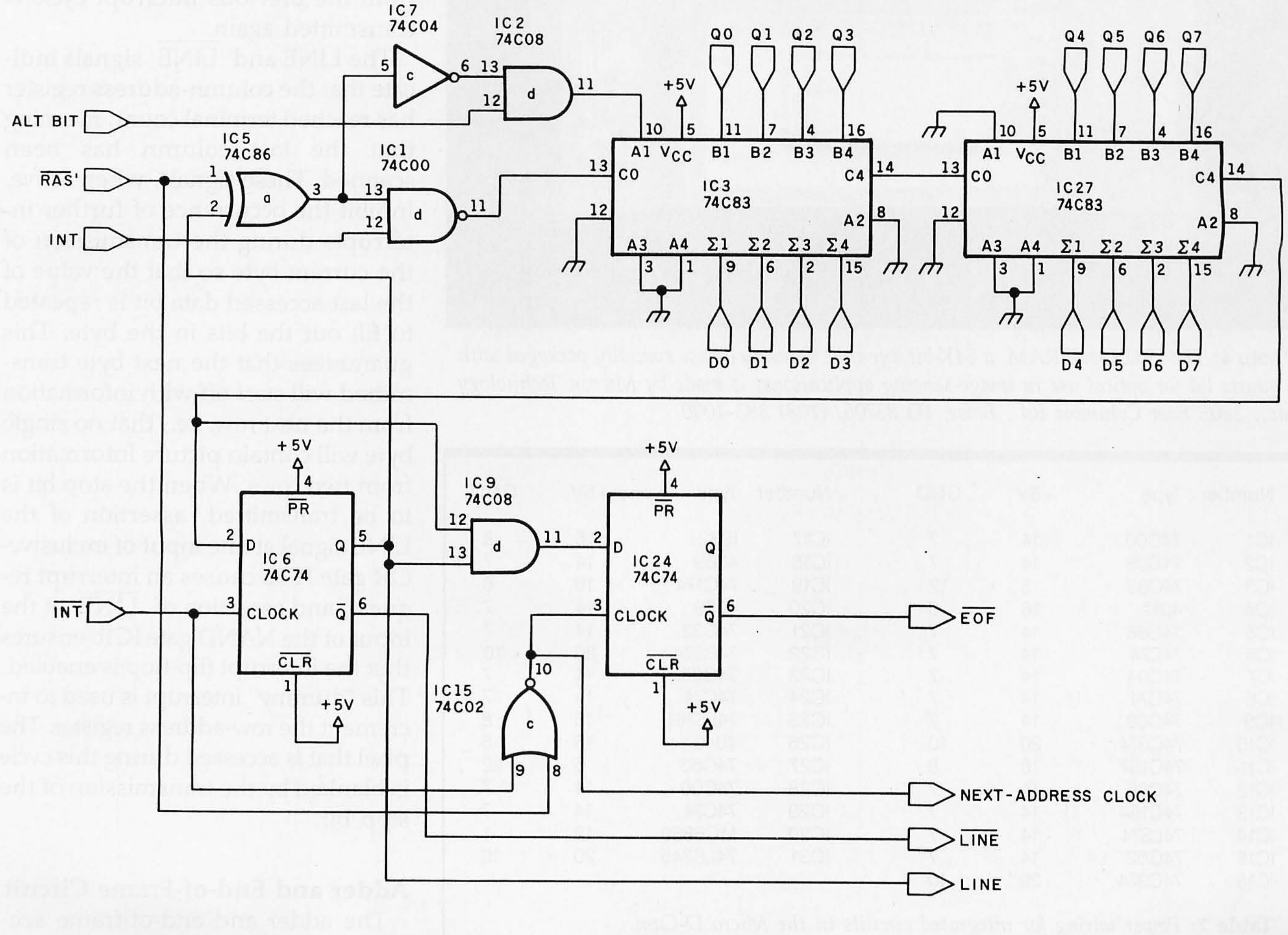


Figure 6: *The adder circuit allows the Micro D-Cam to keep track of the proper values for the row, column, and refresh registers.*

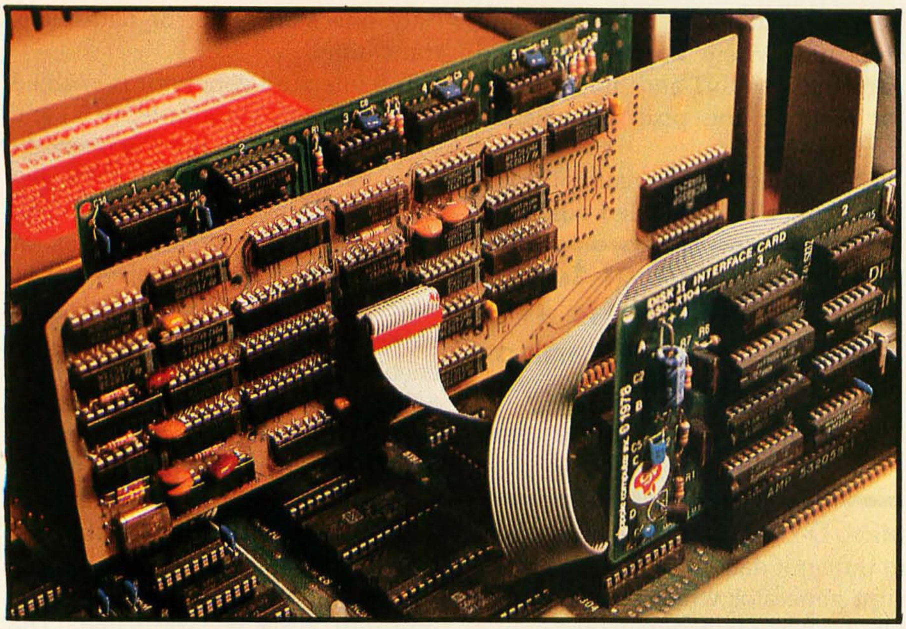


Photo 3: *The control circuitry for the Micro D-Cam image camera is shown here in prototype form mounted in an input/output slot in an Apple II Plus computer.*

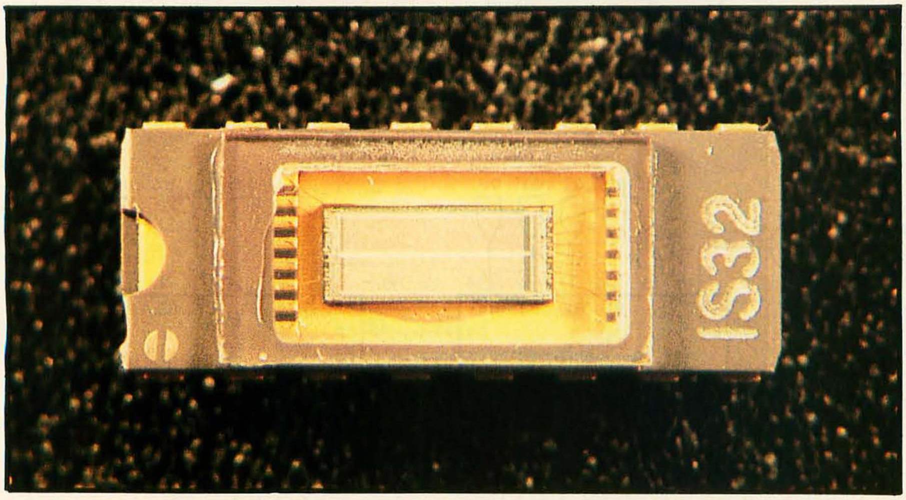


Photo 4: *The IS32 Optic RAM, a 64K-bit dynamic memory device specially packaged with a quartz lid for optical use in image-sensing applications, is made by Micron Technology Inc., 2805 East Columbia Rd., Boise, ID 83706, (208) 383-4000.*

Number	Type	+5V	GND	Number	Type	+5V	GND
IC1	74C00	14	7	IC17	IS32	16	8
IC2	74C08	14	7	IC18	4069	14	7
IC3	74C83	5	12	IC19	74C174	16	8
IC4	4017	16	8	IC20	4069	14	7
IC5	74C86	14	7	IC21	74C32	14	7
IC6	74C74	14	7	IC22	74C374	20	10
IC7	74C04	14	7	IC23	74C08	14	7
IC8	74C74	14	7	IC24	74C74	14	7
IC9	74C08	14	7	IC25	74LS161	16	8
IC10	74C374	20	10	IC26	4040	16	8
IC11	74C157	16	8	IC27	74C83	5	12
IC12	74C86	14	7	IC28	74C00	14	7
IC13	74C164	14	7	IC29	74C74	14	7
IC14	74LS74	14	7	IC30	MC6850	12	1
IC15	74C02	14	7	IC31	74LS245	20	10
IC16	74C374	20	10				

Table 2: *Power wiring for integrated circuits in the Micro D-Cam.*

clock cycle.

The WIDEPIX circuit is used to help compensate for the mismatch in aspect ratios of the Optic RAM and most computer graphics screens. The Optic RAM has a ratio of 2.5:1, compared with the 4:3 aspect ratio of most cathode-ray-tube (CRT) displays, and the pixels are not square. If the image data is displayed on a screen with an aspect ratio that close to 1:1, the image will appear to have been squeezed horizontally. The WIDEPIX circuit helps compensate for this by causing each pixel to be transmitted twice, doubling the width of the image. The circuit is enabled when the Micro D-Cam is transmitting and the WIDEPIX command line is high. This causes the flip-flop IC8a's output to toggle on every data-rate-clock cycle. This flip-flop inhibits the interrupt cycle on alternate data-rate clock cycles. During data-rate-clock cycles in which the interrupt is inhibited, the pixel from the previous interrupt cycle is transmitted again.

The LINE and $\overline{\text{LINE}}$ signals indicate that the column-address register has reached terminal count, meaning that the last column has been scanned. These signals, when active, inhibit the occurrence of further interrupts during the transmission of the current byte so that the value of the last accessed data bit is repeated to fill out the bits in the byte. This guarantees that the next byte transmitted will start off with information from the next row, i.e., that no single byte will contain picture information from two rows. When the stop bit is to be transmitted, assertion of the LINE signal at one input of exclusive-OR gate IC5b causes an interrupt request, and assertion of $\overline{\text{LINE}}$ at the input of the NAND gate IC1b ensures that the interrupt flip-flop is enabled. This "dummy" interrupt is used to increment the row-address register. The pixel that is accessed during this cycle is blanked by the transmission of the stop bit.

Adder and End-of-Frame Circuit

The adder and end-of-frame section, shown in figure 6, adds the proper increments to the row, column, and refresh registers and generates signals indicating end-of-frame (EOF) in the Optic RAM.

When any one of the address registers drives a value onto the present-address bus, the adder circuit receives this value, adds a 0, 1, or 2 to it (depending on the control inputs RAS, LINE, ALTBIT and INT), and places the sum onto the next-address bus. When the refresh register is active, the INT line causes a 1 to be added each cycle. During interrupt cycles, the row and column registers are active. The adder sequences these registers through the Optic RAM in a "column-fast" mode, i.e., the adder adds 0 to the row address and 1 to the column address until the end of the column (or end of the line) is reached. The adder then adds a 1 to both the row and column, thus incrementing the row register and resetting the column register to 0.

The ALTBIT input simply adds an extra 1 to the value on the present-address bus during interrupt cycles; thus the row and column registers are incremented by a total of 2 rather than 1.

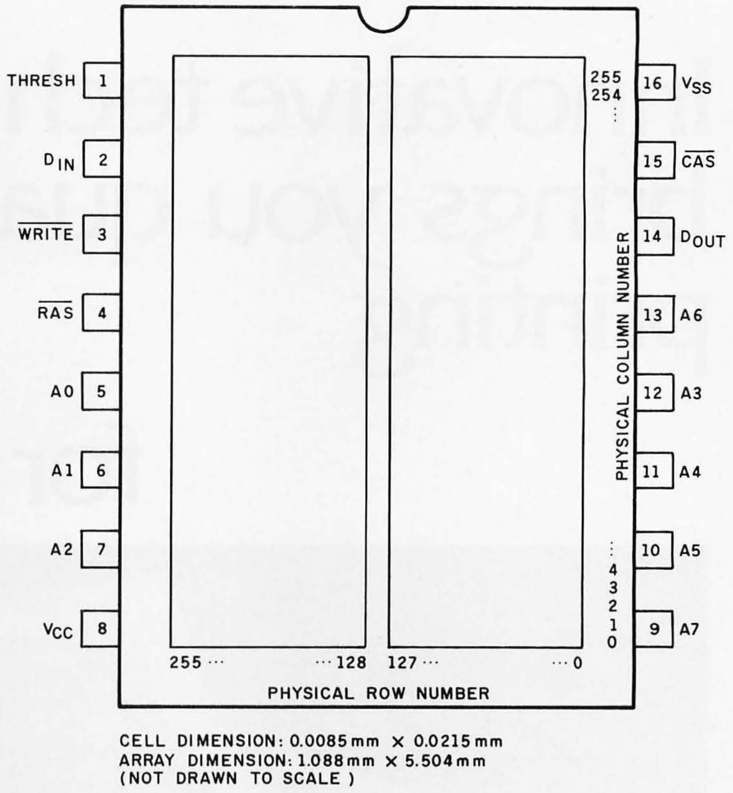


Figure 7: *A diagram of the topology and pinout configuration of the IS32 Optic RAM (not to scale). Each of the two cell regions, visible through the quartz package lid, contains a 128- by 256-cell array.*

Control and Use

The software routines that control the Micro D-Cam are menu-driven. While the camera is running, several real-time commands are available to alter the operation of the camera from frame to frame. The real-time options are displayed on the screen.

When the camera is first turned on, you start the image-gathering process by selecting one of the options from the menu offered by the software, which I'll discuss in detail next month. If everything is working properly, an image of what the Micro D-Cam is seeing is shown on the computer's video-display screen. If the display screen remains dark, the exposure interval may be insufficient; this situation may be remedied by increasing the exposure time. If the exposure time is excessive, the screen will be white. This situation may be remedied by decreasing the exposure time or changing the aperture on the lens. Eventually, a clear picture will appear on the computer's screen as you reach the proper adjustments.

Next Month:

In part 2, we'll look at the software you'll need to read the Micro D-Cam's images, including a complete listing for the Apple II Plus, and I'll explain the computer interface and how the Micro D-Cam communicates with its user. ■

Steve Ciarcia (POB 582, Glastonbury, CT 06033) is an electronics engineer and computer consultant with experience in process control, digital design, nuclear instrumentation, product development, and marketing. In addition to writing for BYTE, he has published several books.

Special thanks to Carl Baker and Jim Herrud of Micron Technology for their contributions to this project.

To receive a complete list of Ciarcia's Circuit Cellar project kits, circle 100 on the reader service inquiry card at the back of the magazine.

The following items are available from:
The Micromint Inc.
561 Willow Ave.
Cedarhurst, NY 11596
(800) 645-3479 (for orders)
(516) 374-6793 (for information)

1. *Complete Micro D-Cam unit including interface card, extension cable, IS32 Optic RAM, lens, remote housing, operators manual, and utility software. Specify Apple II (Plus or E), or IBM Personal Computer.*
   Assembled and tested $295
2. *Same as Item 1 except in kit form. Specify Apple II or IBM Personal Computer version.*
   Complete kit $260
3. *IS32 Optic RAM sold separately*
   IS32 each $42
4. *RS-232C-interfaced Micro D-Cam for general use. Call for price and delivery.*

Please add $4 shipping and insurance in continental United States, $20 overseas. New York residents please include 7 percent sales tax.

Build the Micro D-Cam Solid-State Video Camera

Part 2: Computer Interfaces and Control Software

Serial interfaces for the Apple II and the IBM Personal Computer and versatile software for the Apple II

by Steve Ciarcia

Last month I introduced you to the Micro D-Cam, a relatively low-cost direct-output digital camera that you can build, either from scratch or from a kit distributed by The Micromint. Using a 64K-bit dynamic memory chip as its optical sensor, it has a resolution of 256 by 128 pixels (picture elements), which is adequate for many applications, including input of graphic images, pattern and character recognition, robotics, process control, and security.

In part 1 I explained the principles of operation of the IS32 Optic RAM (random-access read/write memory) and the rest of the Micro D-Cam's hardware. (Table 1 may help you recall some of the IS32's characteristics.) This month I'd like to finish the project by discussing how the camera can be attached to the expansion buses of the Apple II Plus and the IBM Personal Computer and how the camera is programmed to work.

The amount of software included with this article is somewhat more than you've come to expect from a hardware-type fellow like me, but I feel it is necessary to properly show how software can be used to enhance the final picture. In particular, some of you may be interested in the method used to present a gray scale on an Apple II computer.

A Quick Review

The IS32 Optic RAM from Micron Technology Inc. is a memory chip specially packaged to function as a digital image-sensing device. (Because its output is a pure digital signal, it cannot be used to directly drive a composite-video monitor.) The IS32 contains 32,768 usable light-sensitive elements arranged in a matrix of 128 rows and 256 columns. Each of the elements in the matrix is a light-sensitive capacitor, a memory cell that can be accessed randomly by simply reading in the appropriate row and column address. Light striking a particular element causes the capacitor, which is initially precharged to a fixed voltage, to discharge toward 0 volts (V). The capacitor discharges at a rate proportional to the light intensity throughout the duration of the exposure. When the cell's content is read, a logic 0 remaining in the cell indicates a bright pixel—the capacitor was exposed to a light intensity sufficient to discharge the capacitor past the threshold point. A dark pixel is indicated by a logic 1 remaining in the cell, which happens when the light intensity is not sufficient to discharge the capacitor past the threshold point.

1. two 128- by 256-element arrays each measuring 5.504 by 1.088 millimeters
2. element size: 8 microns by 9 microns
3. vertical center-to-center spacing: 21.5 microns
4. horizontal spacing: 8.5 microns
5. spacing between left and right arrays: 150 microns

Table 1: *Specifications of the Micron Technology IS32 Optic RAM, a 64K-bit memory chip that has the extra talent of serving as a digital image detector.*

The operation of the image sensor can be compared to the function of film in a camera. The user can regulate the exposure by two adjustments: aperture (f-stop) and shutter speed. The aperture adjustment controls the amount of light that is allowed to expose the light-sensitive medium (either the IS32 or the film emulsion) by mechanically widening or narrowing the hole through which the light passes. The shutter speed (or scanning speed in the case of the IS32) dictates the amount of time the sensitive medium is exposed.

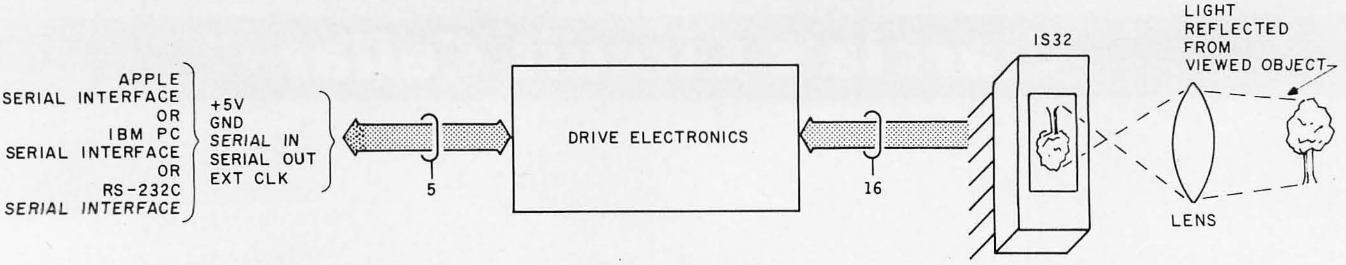


Figure 1: *A block diagram of the Micro D-Cam system.*

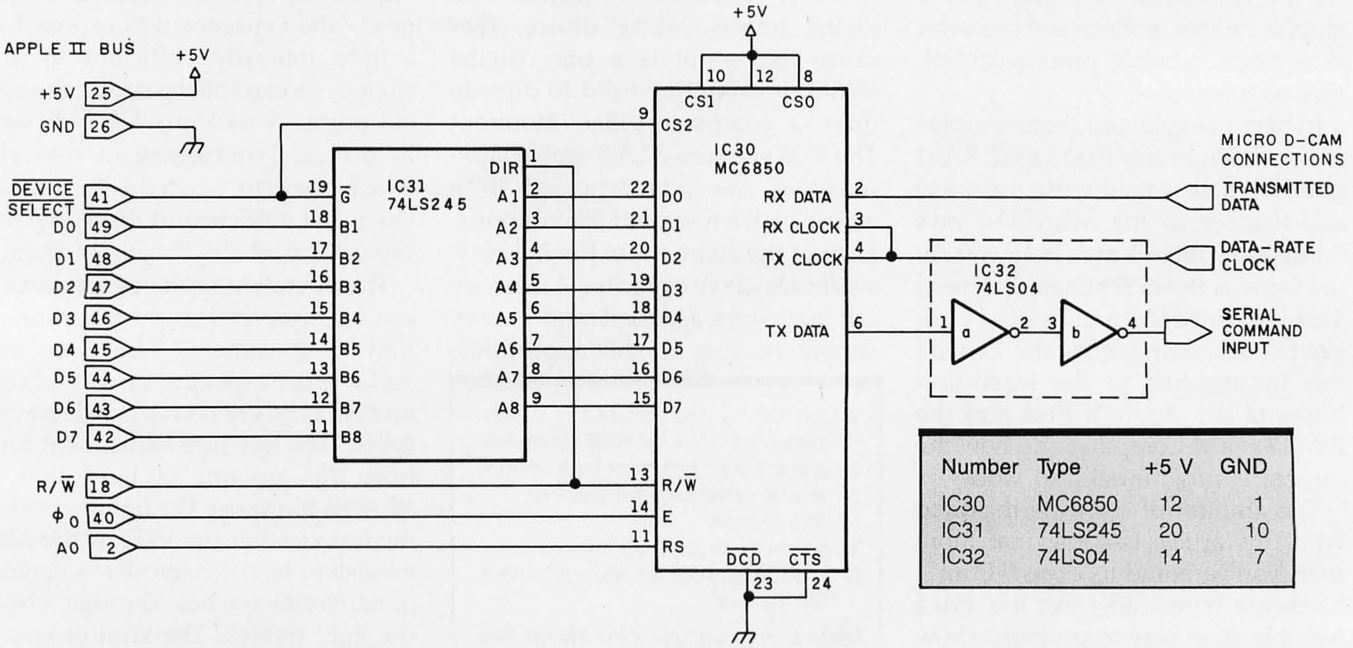
The Micro D-Cam's equivalent of an electronic shutter is controlled by commands transmitted to the interface. Sending a SOAK command to the Micro D-Cam has the effect of opening the shutter. After the appropriate period of exposure has elapsed, two commands, REFRESH and SEND, stop the exposure (close the shutter) and transmit the image to the host computer.

Interfacing the Micro D-Cam

Last month, when we looked at the control and driver electronics of the basic Micro D-Cam, we found that it communicates with its host computer serially, one bit at a time. In its minimal configuration, it requires four wires to be connected to the host computer: two supplying +5 V and ground potential and one each for serial data in and out. In a nonspecific configuration, it can operate asynchronously over an RS-232C link (at a data rate of up to 19,200 bps or bits per second), but I have devised serial interfaces for the camera that can be attached directly to the IBM PC and Apple II computers' buses (although still communicating serially). Using a fifth signal, an additional external clock signal provided to the bus interfaces by the drive electronics, the Micro D-Cam can then function at data rates up to 153,600 bps. The complexity of interface circuits of this type depends upon the host computer's bus structure and address range. The general scheme of connection is shown in figure 1.

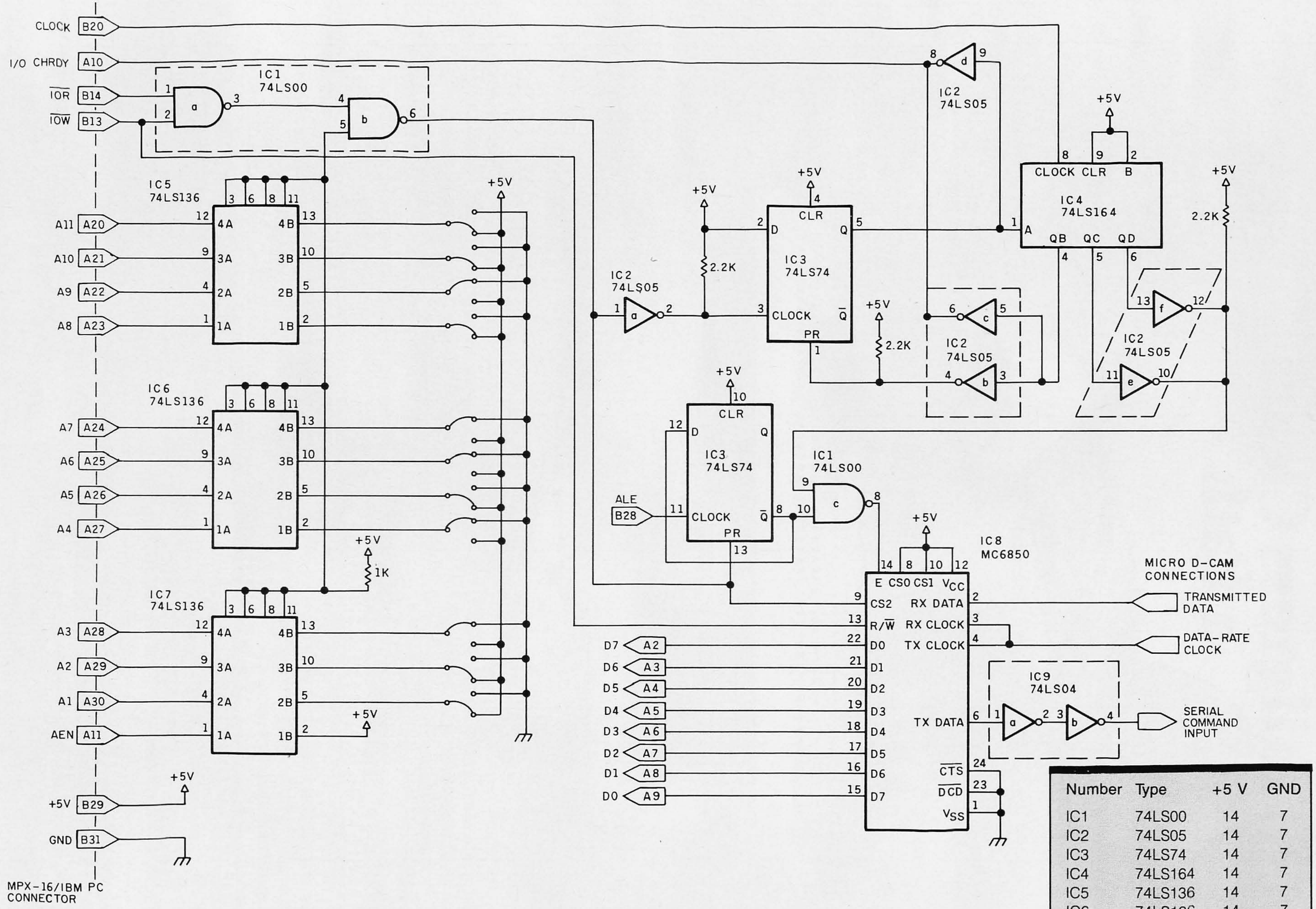
Figure 2 is a schematic diagram of the circuit that forms the interface from the Micro D-Cam circuitry (shown in part 1) to the expansion bus of the Apple II Plus computer. It owes its simplicity to the predecoding of the I/O (input/output) slot address already provided on the Apple's main circuit board. The address decoders usually required in a peripheral interface are eliminated, and the complete serial interface can be built with only two integrated circuits. The 74LS245 octal bus transceiver buffers the TTL- (transistor/transistor logic) level serial data into and out of the MC6850 ACIA (asynchronous communication interface adapter). The serial bit rate is controlled by the external clock output from the Micro D-Cam drive electronics. For maximum speed, the clock frequency should be set for 153,600 Hz.

Figure 3 on page 70 shows the serial interface circuit for the Micro D-Cam configured for the IBM PC's bus. Due to the greater complexity of the Intel 8088 processor as compared with the Apple's 6502 and the PC's larger memory-address space, the in-



Number	Type	+5 V	GND
IC30	MC6850	12	1
IC31	74LS245	20	10
IC32	74LS04	14	7

Figure 2: *A schematic diagram of an Apple II Plus or Apple IIe interface for the Micro D-Cam. The serial data stream from the Optic RAM is converted to parallel bytes and placed on the Apple's data bus by the ACIA and bus transceiver. Although operating asynchronously, high data rates (up to 153,600 bps) are possible because of the external data-rate clock input from the camera-control circuitry.*



Number	Type	+5 V	GND
IC1	74LS00	14	7
IC2	74LS05	14	7
IC3	74LS74	14	7
IC4	74LS164	14	7
IC5	74LS136	14	7
IC6	74LS136	14	7
IC7	74LS136	14	7
IC8	MC6850	12	1
IC9	74LS04	14	7

Figure 3: *An interface from the Micro D-Cam to an IBM Personal Computer (or a bus-compatible unit like the MPX-16). A set of jumper connections on the inputs of exclusive-OR gates determines the I/O-bus address of the interface, shown here set to hexadecimal xD26 and xD27.*

(2a)	Status Bit	Meaning When Set to 1
	0	data has been received from the camera
	1	a command may be sent to the camera
	2	unused
	3	unused
	4	received data was improperly framed
	5	data received before previous byte read
(2b)	**Command Bit**	**Meaning When Cleared to 0**
	7	none (always 1)
	6	none (always 1)
	5	alternating-bit mode (ALTBIT)
	4	wide-pixel mode (WIDEPIX)
	3	7-bit data bytes (7BIT)
	2	transmit one frame instead of two (1ARRAY)
	1	refresh instead of soak (REFRESH)
	0	send the requested image (SEND)

Table 2: *Meanings of bits in the status register (2a) and command word (2b) for the Apple II/Micro D-Cam interface.*

terface requires three times as many integrated circuits. In the IBM, the Micro D-Cam's two port addresses are decoded by three chips: IC5, IC6, and IC7. These are 74LS136 open-collector exclusive-OR gates connected together in a "wired-OR" configuration. The voltages wired to the 11 inputs of the address decoder determine the interface board's addresses. As shown in figure 3, the addresses I used were *x*D26 and *x*D27 (where *x* can take on any hexadecimal value from 0 to F). The 6850 ACIA (IC8) functions as previously described except that IC2 and IC4 are configured as a wait-state generator to facilitate timely access to the bus.

Data and Command Format

The 6850 ACIA comprises a data register and a status register. You can configure operating parameters (such as parity, stop bits, start bits, clocking, etc.) by writing values into the status register. Before the host computer can access the Micro D-Cam, the ACIA has to be initialized to the proper configuration. The control software does this by writing two bytes, a hexadecimal 03 followed by a hexadecimal 14, into the status register. The first byte performs a master reset on the ACIA, while the second byte specifies that the serial transmission protocol is 1 start bit, followed by 8 data bits, followed by 1 stop bit.

Reading the status register allows the control program to determine when new data has been received and when the ACIA is ready to send data. The meanings of the status bits, when set, are as shown in table 2a.

In normal use, only bit 0 is checked when seeing if data is available from the camera. Bits 4 and 5 are used only in debugging, as these situations should not normally arise. When designing the program that receives the image from the camera, it is a good idea to incorporate a time-out mechanism in case the camera stops sending bytes before the program expects; otherwise, the program can hang up if the software misses even a single byte.

In the Apple II Plus and IIe, the hexadecimal addresses of the type C0*n*E access the status register of the ACIA on an interface card plugged into the corresponding slot, while C0*n*F addresses access the ACIA's data register. The *n* is the hexadecimal value of the slot number plus 8. For example, suppose the interface card were plugged into slot 3; 3 plus 8 equals B, and so address C0BE will access the status register and C0BF the data register.

Command Functions

While the camera is running, the host computer directs the Micro D-Cam's operating modes by sending it command words. Each command word is composed of 8 bits, with functions as summarized in table 2b. Let's look at each of these in detail:

ALTBIT Mode: When bit 5 is clear (equal to 0), the Micro D-Cam transmits only the pixels from the even-numbered rows and columns in the Optic RAM. This mode usually produces a clearer image than the NOALTBIT mode at the expense of losing resolution.

WIDEPIX Mode: When bit 4 is clear, the Micro D-Cam transmits each pixel in the array twice. Each image-sensing element is *rectangular* in shape, so by "double-transmitting" the pixels, the proper width-to-height (aspect) ratio is maintained when the image is displayed on the computer's video monitor.

7BIT Mode: The Apple II's implementation of high-resolution graphics is somewhat peculiar. The most significant bit of each byte on the hi-res graphics page is reserved as the color bit for a group of pixels, while each of the other 7 bits stores a 1 or 0 as a bright or dark value for a pixel. In 7BIT mode, the Micro D-Cam transmits data in a format compatible with the Apple's high-resolution format, with 7 bits of pixel values per byte. The 7BIT mode is selected by clearing bit 3 of the command byte to 0. The alternative to 7BIT mode is 8BIT mode, which is achieved by setting bit 3 to 1. The 8BIT mode causes the camera to transmit in normal bit-mapped format, with all 8 bits in the byte containing image data, and is preferred for use with all computers other than the Apple.

1ARRAY Mode: The 1ARRAY mode is selected by clearing bit 2 of the command byte. Using this mode, only data from the image focused on the lower light-sensitive array is transmitted from the Micro D-Cam. By setting bit 2 of the command byte, 2ARRAY mode is selected, which causes data from both arrays to be transmitted from the camera. The 2ARRAY mode causes a split-screen effect because of the space between the two arrays in the image-sensor chip.

REFRESH Mode: In some ways, the Micro D-Cam is like any other

Command Character	Control Effect
>	increase exposure time
<	decrease exposure time
F	fix exposure time to current setting
L	load previously stored image from disk
N	print negative of screen image onto Epson printer
P	print screen output onto Epson (Graftrax option required)
Q	quit and return to main menu
R	toggle display of exposure time and light level
S	save current image to disk
T	use current light level and autotrack the exposure

Table 3: *Options for control of the Micro D-Cam that may be selected in real time through the distributed menu-driven software. See table 4 for the options provided in the GREY16 program.*

camera. It must receive the proper amount of light to make the image develop properly. Too much light will overexpose the image, while too little light will underexpose the image. Exposure time is determined by how long the control program in the host computer allows the Optic RAM to be exposed to light without its cells being refreshed. Refreshing the image sensor is the same process used in any dynamic memory: the existing charge in each cell is sensed, the voltage compared with a threshold potential, and a fresh potential of 0 V (for a logic 0) or +5 V (for 1) is rewritten into the cell. (The only difference in the Optic RAM is that all cells must contain +5 V at the beginning of an image-sensing cycle when refreshing stops.) If the image sensor is not continually refreshed, the light focused on each cell causes the voltage in each cell to leak away at a rate proportional to the intensity of the light. When the image sensor is not being refreshed, we say it is "soaking" (in light). Allowing the image sensor to soak for longer periods of time enables the Micro D-Cam to see better in dimmer light.

When the REFRESH mode is selected (by clearing bit 1 of the command byte) the Micro D-Cam keeps the image sensor's cells refreshed while it is sending an image. When bit 1 is set, SOAK mode is invoked. This causes the camera to soak (and therefore remain sensitive to light) while it is transmitting an image.

SEND Mode: When a command is sent to the camera with SEND mode selected (bit 0 cleared), the camera begins transmitting an image.

Control Software

The software for controlling and displaying pictures is vital to the operation of the Micro D-Cam. Menu-driven versions of the Micro D-Cam control software for both the Apple II and IBM PC are available from The Micromint.

However, some of you may already have the Micron Technology Optic RAM or a similar 64K-bit dynamic RAM device with suitable chip layout, and you may want to build the Micro D-Cam from scratch. Consequently, I have included with this article complete listings of two control programs written for the Apple II. One provides experimenters with a means for testing the Micro D-Cam; the second is a more sophisticated software routine that enhances the image and performs gray-scale ordered dithering (I'll explain this term later). While the Micro D-Cam software includes some additional menu-driven utility programs (some options of which are shown in table 3), all the Micro D-Cam photos printed here and last month can be reproduced using only the two programs in the magazine.

A Sample Control Program

The Micro D-Cam demonstration program (listing 1 on pages 512 through 518) illustrates the simplest possible software needed to receive an image from the camera and display it on the Apple's hi-res screen. It is not really as long and complicated as it looks; the accompanying flowchart (figure 4 on page 76) should reveal the general scheme of operation. The software consists of two parts: a short BASIC main program (listing 1a) and a set of machine-language subroutines (shown in assembled format in listing 1b). The BASIC program loads the machine-language code from disk, interactively sets the correct I/O-slot number and exposure time, and calls the machine-language code to display the image; upon returning to BASIC, the calling program checks to see if you want to terminate the process.

The hard part of the work is done in the machine-language routines, which were necessary to allow the Micro D-Cam to operate at 153,600 bps. When called, the machine code begins by making sure that the hi-res screen is being displayed. It then initializes the ACIA and sends a command to tell the camera to soak without sending an image. (This effectively clears the Optic RAM and tells the camera to begin the exposure.) The program then waits for the duration of the exposure.

The next step is to read the image from the camera and display it on the screen. To save time and memory, the software sends the picture straight to the hi-res screen memory (rather than reading it into a separate buffer area and then moving it) to minimize the processing of the final image. The mode used is alternate-pixel, wide-pixel, with 7-bit data words. Before any part of the picture is received, a number of memory pointers are set up to facilitate proper placement on the screen. A command is sent to the camera to begin transmitting the image, and the program loops to read in each byte of the image and put it on the screen.

The control software knows how many bytes of image data it should receive from the camera, but a problem can arise from relying on byte-counting to determine when to stop reading data: if the computer misses one, it could hang the system up. To be on the safe side, a time-out loop has been provided in the image-reading routine. If the computer times out

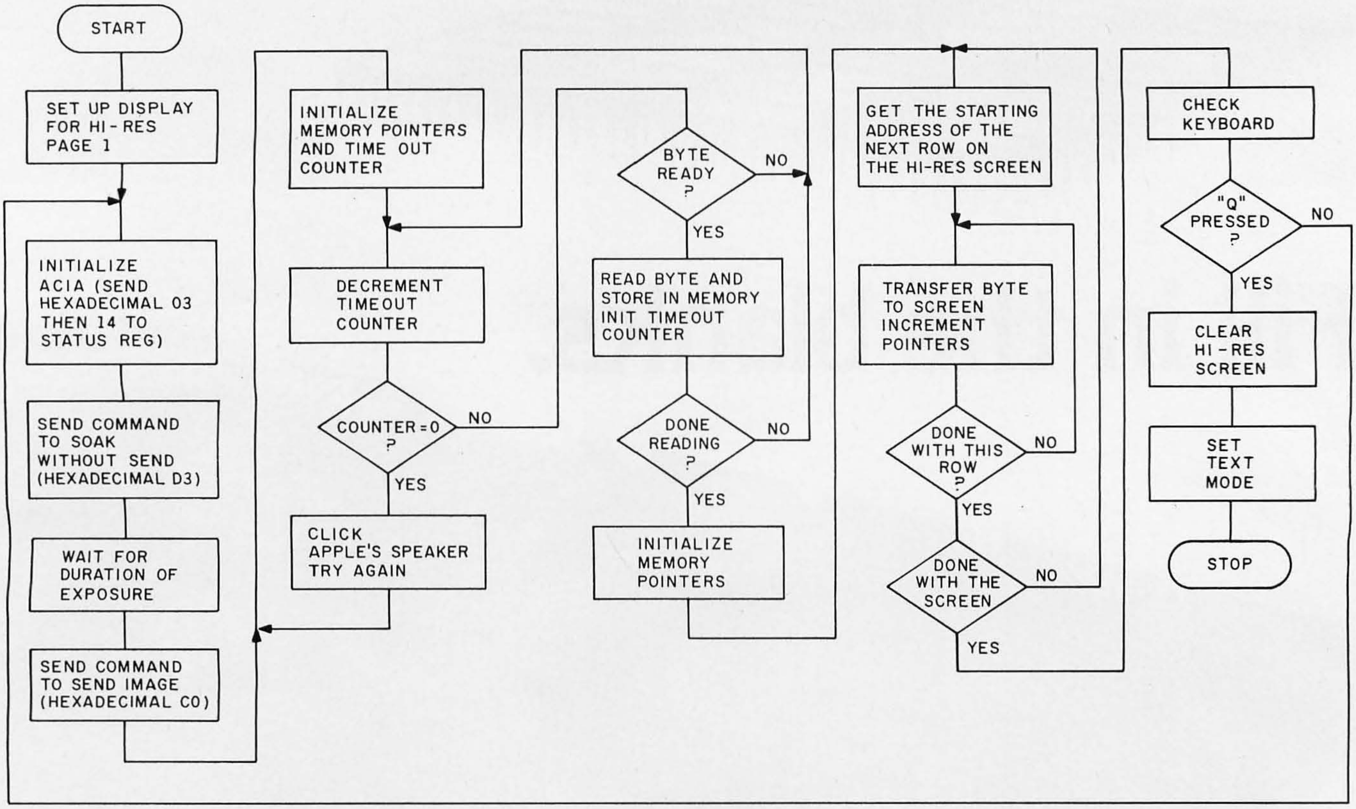

START
SET UP DISPLAY FOR HI-RES PAGE 1
INITIALIZE ACIA (SEND HEXADECIMAL 03 THEN 14 TO STATUS REG)
SEND COMMAND TO SOAK WITHOUT SEND (HEXADECIMAL D3)
WAIT FOR DURATION OF EXPOSURE
SEND COMMAND TO SEND IMAGE (HEXADECIMAL C0)
INITIALIZE MEMORY POINTERS AND TIME OUT COUNTER
DECREMENT TIMEOUT COUNTER
COUNTER = 0 ?
NO
YES
CLICK APPLE'S SPEAKER TRY AGAIN
BYTE READY ?
NO
YES
READ BYTE AND STORE IN MEMORY INIT TIMEOUT COUNTER
DONE READING ?
NO
YES
INITIALIZE MEMORY POINTERS
GET THE STARTING ADDRESS OF THE NEXT ROW ON THE HI-RES SCREEN
TRANSFER BYTE TO SCREEN INCREMENT POINTERS
DONE WITH THIS ROW ?
NO
YES
DONE WITH THE SCREEN
NO
YES
CHECK KEYBOARD
"Q" PRESSED ?
NO
YES
CLEAR HI-RES SCREEN
SET TEXT MODE
STOP

Figure 4: *A flowchart of the Micro D-Cam demonstration program for the Apple II. The program consists of a BASIC main routine, shown in listing 1a, and some 6502 machine-language subroutines, shown in assembly-language form in listing 1b.*

while waiting for the camera, it clicks the speaker, checks for a keypress, and tries the entire command sequence again. In this manner, you are alerted to any possible problems.

Because the Apple's hi-res screen display is mapped nonlinearly into memory space, a lengthy table at the end of the machine-language code provides the starting address for each consecutive row of the hi-res screen. The program gets the address of the beginning of each row and then reads 40 bytes from the camera, placing them consecutively on the screen. The next row, and each row after it, is done in a similar manner.

Once the image is on the screen, a command is sent to the camera to refresh without sending. This gets it ready for the next exposure. Finally, the machine code checks the keyboard and processes any command inputs before returning to BASIC.

Obtaining Gray Scale

A more user-friendly demonstration of the Micro D-Cam that also provides a level of gray-scale capability is the GREY16 program of listing 2 (pages 518 through 538). It has one mode that allows you to do quick aiming and focusing of the camera, another to let you get an idea of what the final picture will look like, and a third to create a 15-intensity-level gray-scale picture on the Apple II. (The processes involved are outlined in the flowchart of figure 5. Unfortunately, space constraints prevent me from showing you a similar program for the IBM PC.)

Using GREY16, you can change the length of exposure for the image being displayed, or you can change the upper and lower exposure limits of the gray-scale image. Once you've obtained a satisfactory picture, you can save it on disk for later use or print it on an Epson MX-80 printer (equipped with Graftrax) using the screen-dump program. A summary of available commands in the GREY16 program is shown in table 4.

When it is first powered up, you start the camera running by selecting one of the options from the GREY16 menu. If the exposure time is insufficient, the screen will be black. If the exposure time is excessive, the screen

Text continued on page 82

Command Character	Control Effect
N	display the image in normal size (256 by 64)
F	display the image in full size (256 by 128)
G	create a picture (256 by 128) with 15 levels of gray (this process takes about 30 seconds and displays a countdown of the number of exposures from F to 0)
E	change the exposure time of the current displayed image, the upper limit of the gray-scale image, or the lower limit of the gray-scale image
S	save to disk the picture currently being displayed (this may be done in any of the three display modes: normal, full, or gray)
Q	quit the program and return to BASIC

Table 4: *A summary of user commands implemented in the GREY16 program of listing 2.*

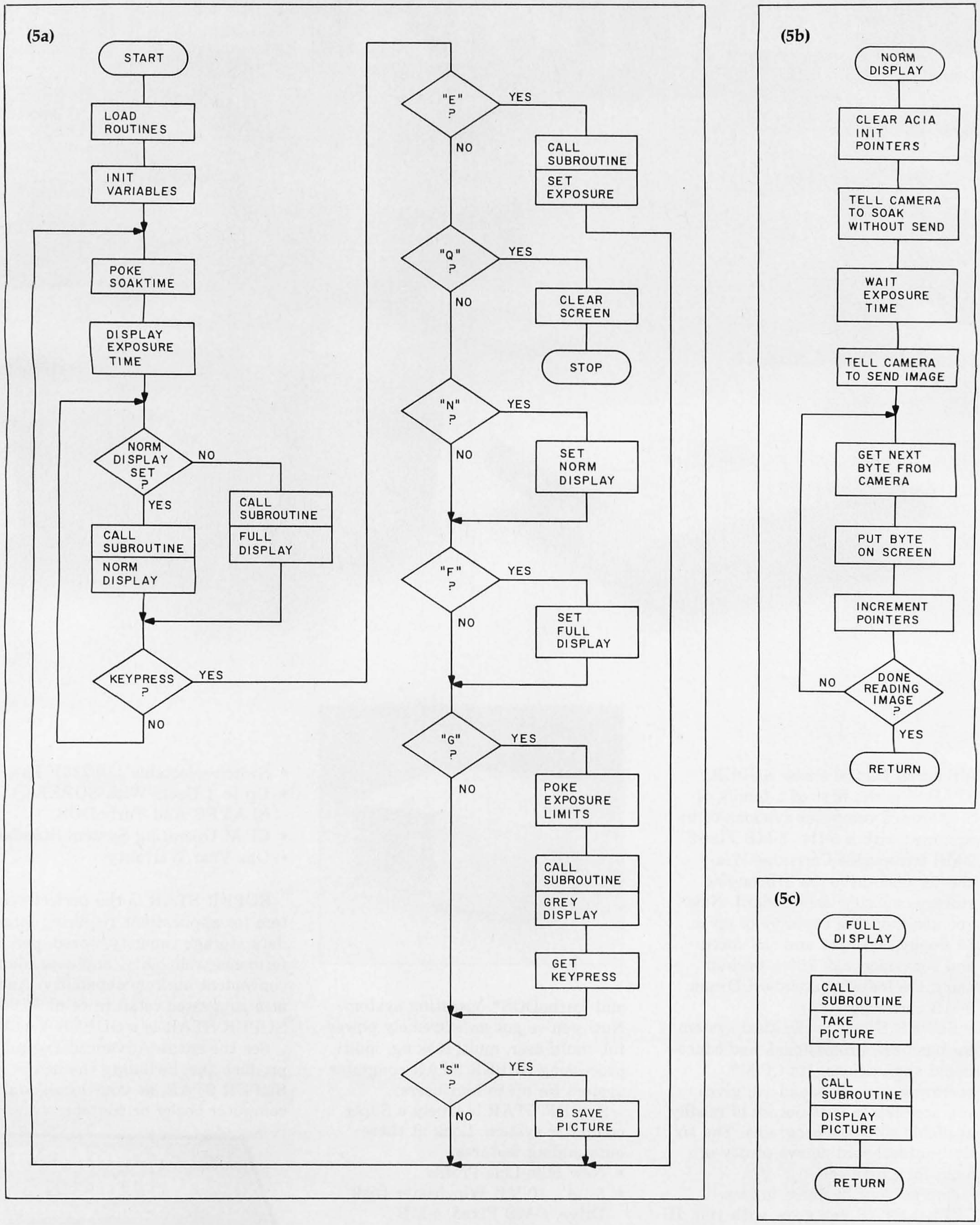


Figure 5: *Flowcharts of the GREY16 program for the Apple II (the figure continues on pages 80 and 82). The BASIC portion appears as listing 2a, the machine-language portion as listing 2b. The main routine (5a) calls various subroutines: NORM DISPLAY (5b), FULL DISPLAY (5c), SET EXPOSURE (5d), GREY DISPLAY (5e), TAKE PICTURE (5f), DISPLAY PIC (5g), and ENHANCE (5h).*

The subroutine GREY DISPLAY takes sensor pixels from 15 exposures and translates them into arrays of the smaller display pixels to represent intermediate brightnesses.

Figure continued on page 80

Figure 5 continued:

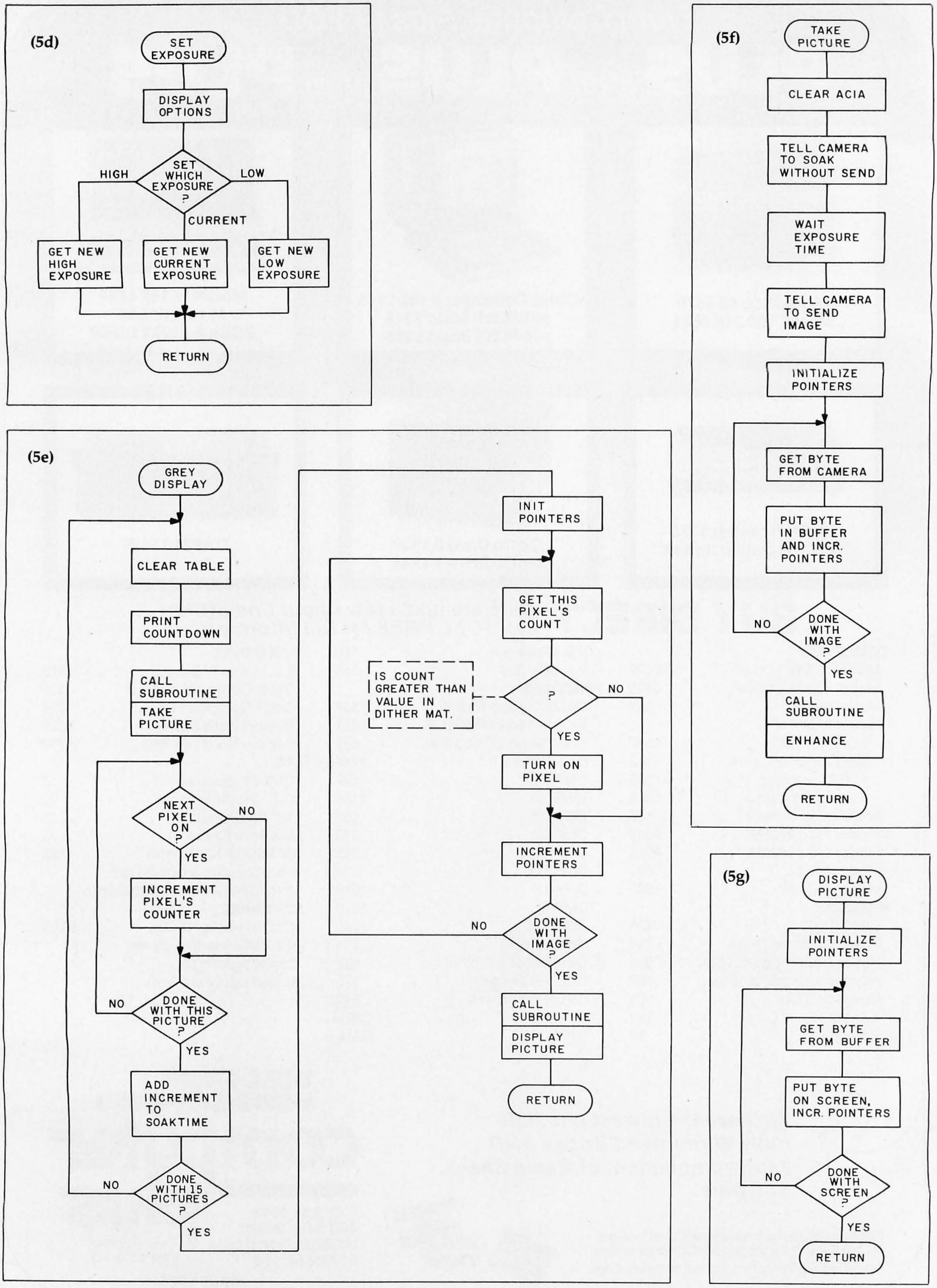
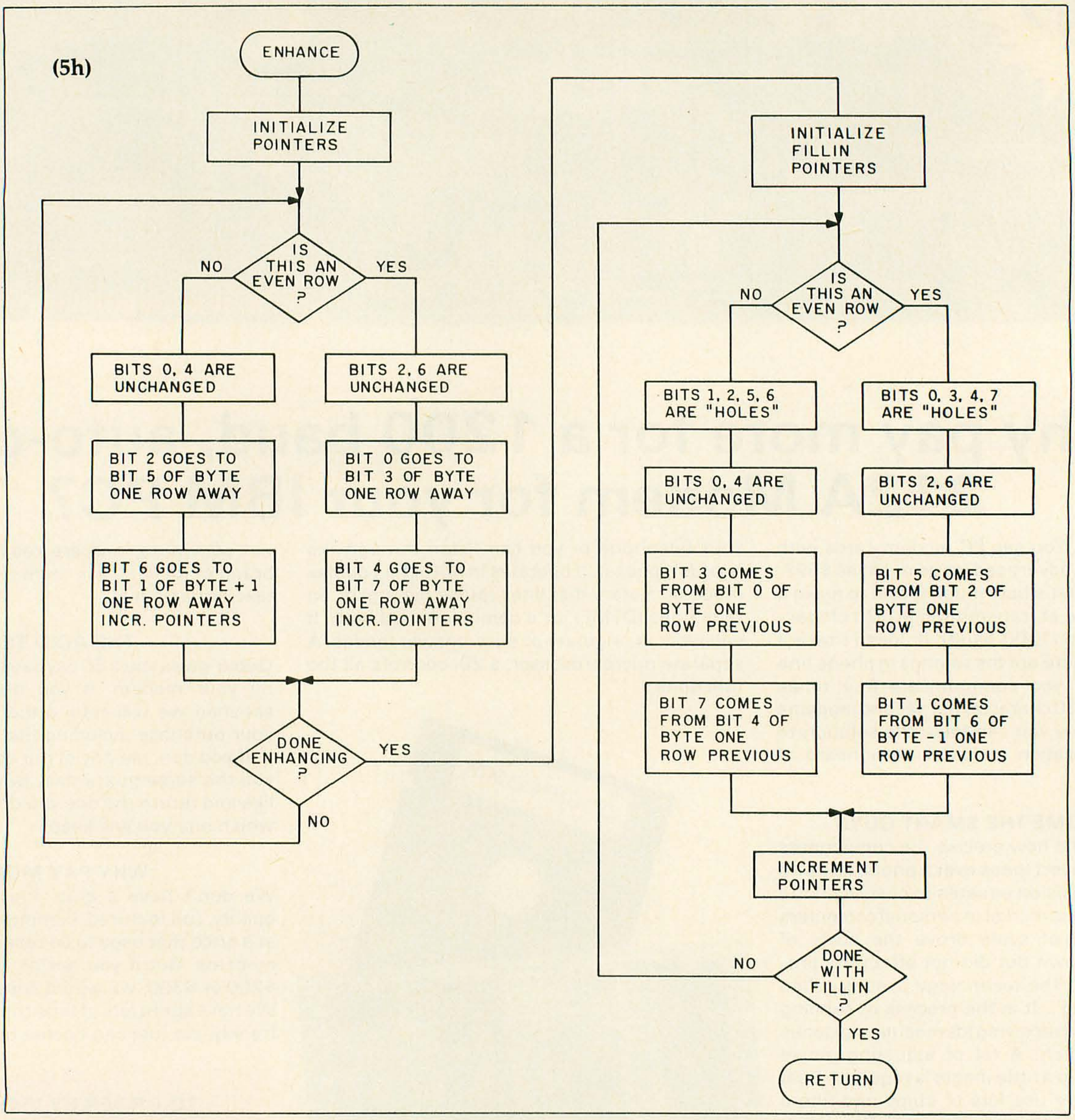


Figure continued on page 82

Figure 5 continued:


(5h)
ENHANCE
INITIALIZE POINTERS
IS THIS AN EVEN ROW ?
NO
YES
BITS 0, 4 ARE UNCHANGED
BIT 2 GOES TO BIT 5 OF BYTE ONE ROW AWAY
BIT 6 GOES TO BIT 1 OF BYTE + 1 ONE ROW AWAY INCR. POINTERS
BITS 2, 6 ARE UNCHANGED
BIT 0 GOES TO BIT 3 OF BYTE ONE ROW AWAY
BIT 4 GOES TO BIT 7 OF BYTE ONE ROW AWAY INCR. POINTERS
DONE ENHANCING ?
YES
NO
INITIALIZE FILLIN POINTERS
IS THIS AN EVEN ROW ?
NO
YES
BITS 1, 2, 5, 6 ARE "HOLES"
BITS 0, 4 ARE UNCHANGED
BIT 3 COMES FROM BIT 0 OF BYTE ONE ROW PREVIOUS
BIT 7 COMES FROM BIT 4 OF BYTE ONE ROW PREVIOUS
BITS 0, 3, 4, 7 ARE "HOLES"
BITS 2, 6 ARE UNCHANGED
BIT 5 COMES FROM BIT 2 OF BYTE ONE ROW PREVIOUS
BIT 1 COMES FROM BIT 6 OF BYTE - 1 ONE ROW PREVIOUS
INCREMENT POINTERS
DONE WITH FILLIN ?
NO
YES
RETURN

Text continued from page 76:

will be completely white. These situations may be remedied by increasing or decreasing the exposure time or changing the lens aperture. You may need to focus, also. Eventually, a clear picture will appear on the video screen when the lens is properly adjusted.

The gray-scale portion of the program demonstrates what can be done with just a little bit of software enhancement, permitting you to create images with 14 intermediate levels of brightness (plus extreme dark and bright) and display them on the Apple's hi-res screen. The image of an automobile shown in photo 1 is an example.

The technique used to display the gray-scale pictures on the Apple II Plus and IIe computers is known as *ordered dithering,* in which half-tone values are constructed from multiple binary black or white images. The process requires the Micro D-Cam system to take 15 exposures of the same subject, each lasting a little longer than the previous one. (This normally takes only several seconds.) After each exposure is taken, every pixel in it is checked. If the pixel is on (showing a 1 value corresponding to brightness above that exposure's threshold), a counter location corresponding to that pixel is incremented. At the end of 15 passes, this process yields a table of values, each value describing the relative intensity of its corresponding pixel. For example, if a pixel's final value is 15, that pixel should be displayed maximally bright; if a pixel's value is 8, the pixel deserves a shade of gray halfway between the black and white extremes.

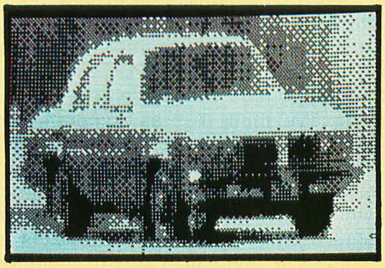
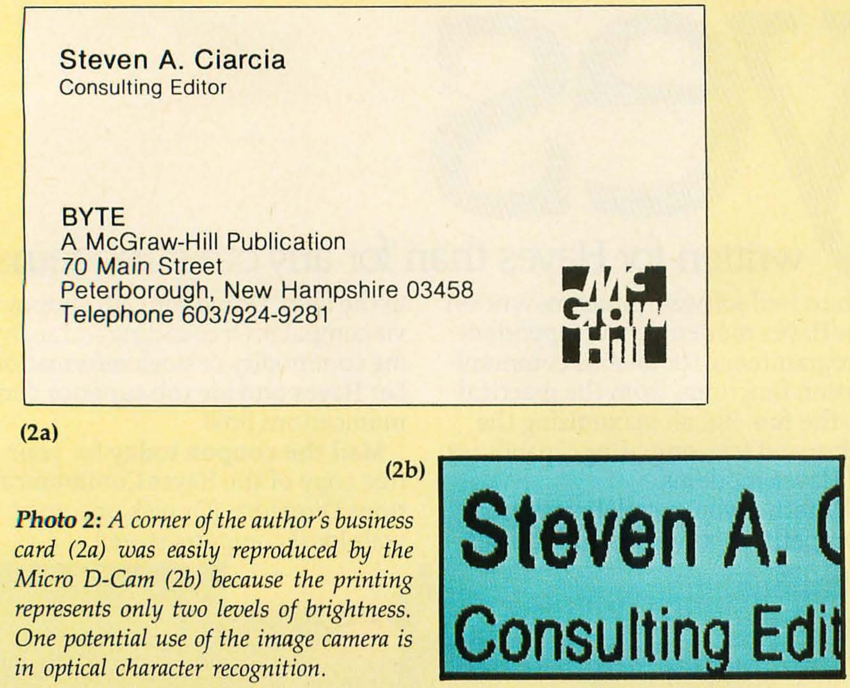


Photo 1: *The Micro D-Cam was aimed at a car parked outside. The dithered digital gray-scale image shown here is displayed by an Apple II Plus.*

Once the pixel-intensity table has been constructed, a 4 by 4 dither



(2a)

Steven A. Ciarcia
Consulting Editor

BYTE
A McGraw-Hill Publication
70 Main Street
Peterborough, New Hampshire 03458
Telephone 603/924-9281

(2b)

Steven A. C
Consulting Edit

Photo 2: *A corner of the author's business card (2a) was easily reproduced by the Micro D-Cam (2b) because the printing represents only two levels of brightness. One potential use of the image camera is in optical character recognition.*

matrix is used to assign a display value (an array of binary pixels) to each screen position. In this software, the matrix is as follows:

$$\begin{matrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{matrix}$$

Then, one pixel at a time, the values in the table of final image magnitudes are compared to the array-element values in the matrix. If the image value for the pixel is greater than the element's value, that array element is turned on. If the intensity value is 0, none of the matrix elements are displayed bright; if the value is 8, elements 0 through 8 are displayed bright; and if the value is 15, all the elements become bright. In this manner, 15 levels of luminance may be represented but at a certain loss of spatial definition. The process is repeated across the entire screen until each screen position has a value assigned to it.

It would definitely be possible to use different-size dithering matrices, with certain trade-offs. For example, a 2 by 2 matrix would yield only 5 levels of gray but would have much finer spatial definition, while an 8 by 8 matrix would yield 64 levels of gray but with much loss of spacial definition.

The GREY16 program overcomes many of the limitations associated with binary optical sensors. While black print on white paper (like my business card, shown in photo 2) is easily viewed by the Micro D-Cam with no enhancement, we don't live in a pure-black-and-white world, and three-dimensional objects need shading to be recognized on a two-dimensional video display.

This is most easily demonstrated with a series of photos of a pair of dice. Photo 3a shows the color and lighting conditions of our sample object. If we use the Micro D-Cam without gray scale, we obtain the binary picture in photo 3b. (This slightly vague yet quite representative picture of the dice would probably be usable in robotics or some recognition applications.)

For a more representative picture, we can invoke the G command in the GREY16 program to produce photo 3c. There is now no question of what the subject is or what value is shown on the dice. If the image were reproduced on a computer capable of displaying half-tones, it would look much more like photo 3a than this dithered Apple II Plus display.

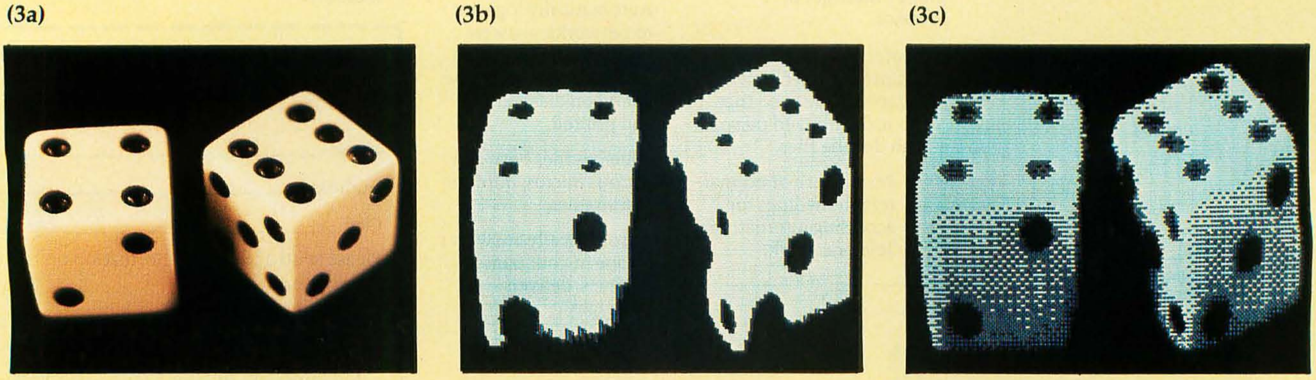


Photo 3: *A pair of dice (3a) was scanned by the Micro D-Cam. When only two levels of luminance are recorded and sent to the computer's display, the result is the output shown in photo 3b. When multiple gray-scale exposures and ordered dithering are invoked, the more easily recognizable output of photo 3c appears.*

Closing Observations

Two articles have not been enough to describe all the capabilities of the Micro D-Cam. If I had more time, I'd try some experiments using different lenses and filters. Theoretically, if three exposures were taken through red, green, and blue filters, we should be able to create a color image.

One interesting fact I did observe is that the IS32, like most silicon-based image sensors, is infrared-sensitive. My test was somewhat unscientific, and I have no precise data on the Optic RAM's spectral sensitivity. I merely lighted the subject with some infrared light-emitting diodes, but it was clearly seen by the Micro D-Cam even in visible-light darkness.

This mild success leads me to consider related experiments. Don't count on it, but in a few months you just might be reading about some sort of character-recognition wand I've built using an Optic RAM. In the meantime, if you find any other dynamic RAM chips that are suitable in this application or wish to show me a character-recognition program of your own, please write and let me know.

Next Month:

Communicating with their fellow humans can be a problem for people who cannot speak. We'll look at a way digital electronics can be harnessed to remedy this difficulty.■

Editor's Note: *For a review of a similar assembled product, see page 316.*

References

1. Ciarcia, Steve. "Analog Interfacing in the Real World." BYTE, January 1982, page 72.
2. Ciarcia, Steve. "Build the Micro D-Cam Solid-State Video Camera, Part 1: The IS32 Optic RAM and the Micro D-Cam Hardware." BYTE, September 1983, page 20.
3. Crow, Franklin C. "Three-Dimensional Computer Graphics." Part 1, BYTE, March 1981, page 54; Part 2, April 1981, page 290.
4. Grob, Bernard. *Basic Television: Principles and Servicing*, 4th ed. New York: McGraw-Hill, 1975.
5. Newman, William M. and Robert F. Sproull. *Principles of Interactive Computer Graphics*, 2nd ed. New York: McGraw-Hill, 1979.
6. Tomas, Joe. "Hardware Review: Dithertizer II." BYTE, February 1982, page 219.
7. Walker, Terry, Harry Garland, and Roger Melen. "Build Cyclops: First All Solid-State TV Camera for Experimenters." *Popular Electronics*, February 1975, page 27.
8. Williams, Thomas. "Digital Storage of Images." BYTE, November 1980, page 220.

The following items are available from

*The Micromint Inc.
561 Willow Ave.
Cedarhurst, NY 11596*

*(800) 645-3479 for orders
(516) 374-6793 for information*

1. *Complete Micro D-Cam unit including interface card, extension cable, IS32 Optic RAM, lens, remote housing, operator's manual, and utility software. Specify Apple II (II Plus or IIe) or IBM Personal Computer.
Assembled and tested............$295*
2. *Same as item 1 except in kit form. Specify Apple II or IBM PC.
Complete kit...................$260*
3. *IC32 Optic RAM sold separately.
Each..........................$42*
4. *Serial-interface (RS-232C) Micro D-Cam for general use. Software listings for several different computers to be available soon. Call for price and delivery.*

Please add $4 shipping and insurance in continental United States, $20 overseas. New York residents, please include 7 percent sales tax.

Editor's Note: *Steve often refers to previous Circuit Cellar articles as reference material for each month's current article. Many of these past articles are available in reprint books from BYTE Books, McGraw-Hill Book Company, POB 400, Hightstown, NJ 08250.*

Ciarcia's Circuit Cellar, Volume I *covers articles that appeared in BYTE from September 1977 through November 1978.* Ciarcia's Circuit Cellar, Volume II *contains articles from December 1978 through June 1980.* Ciarcia's Circuit Cellar, Volume III *contains articles from July 1980 through December 1981.*

Special thanks to Carl Baker and Jim Herrud of Micron Technology Inc. for their contributions to this project.

To receive a complete list of Ciarcia's Circuit Cellar project kits available, circle 100 on the reader service inquiry card at the back of the magazine.

Steve Ciarcia (POB 582, Glastonbury, CT 06033) is an electronics engineer and computer consultant with experience in process control, digital design, nuclear instrumentation, product development, and marketing. In addition to writing for BYTE, he has published several books.