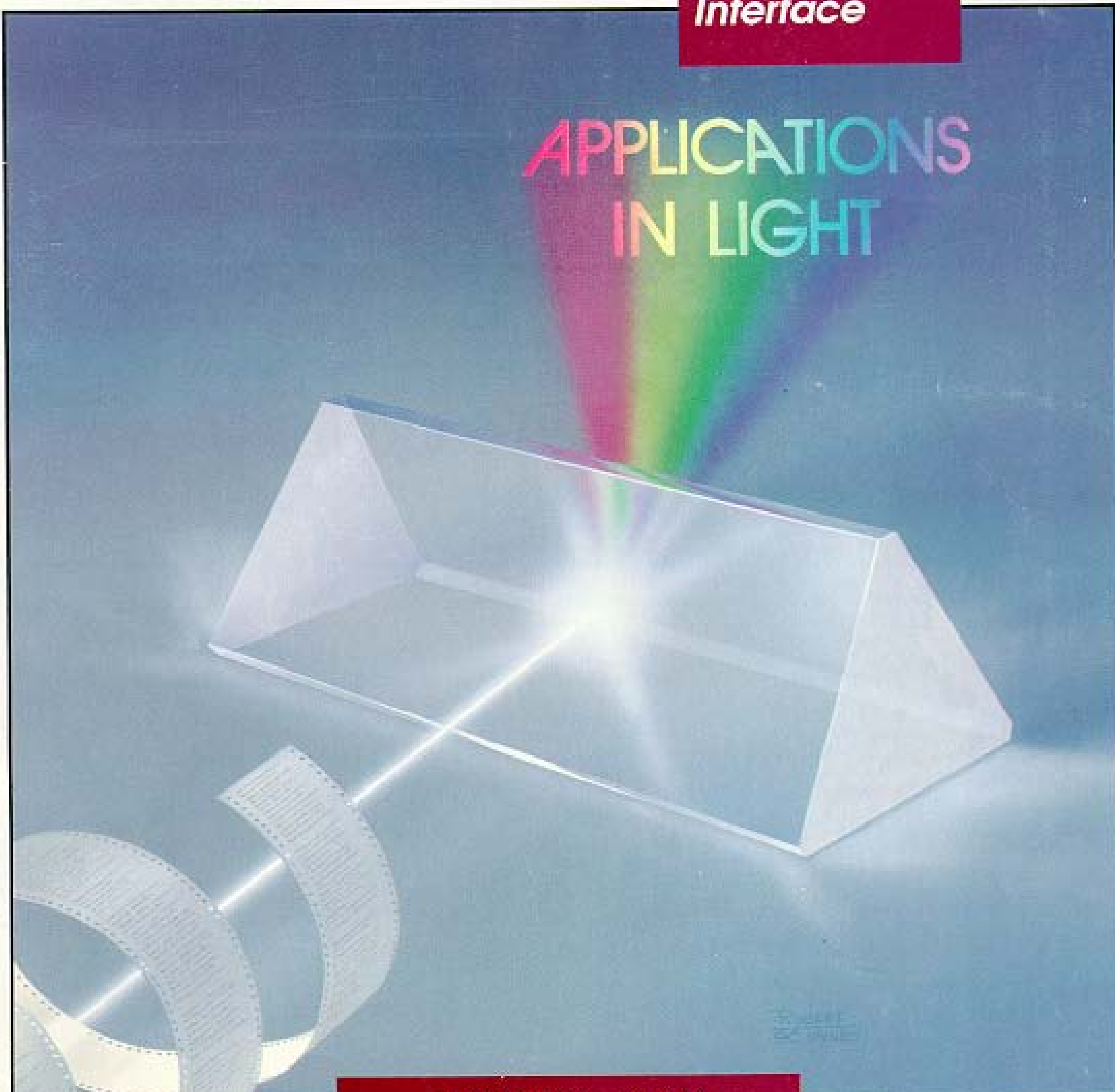
CIRCUIT CELLAR INK®

THE COMPUTER APPLICATIONS JOURNAL

Computer-Generated Holograms

Ciarcia on Laser Diodes

Build a SCSI Interface



April/May 1990 — Issue 14

$3.95

EDITOR'S INK

Curtis Franklin, Jr.

It Just Gets Better

It must have something to do with a year ending in "0." My desktop has disappeared under a deluge of press releases, product announcements, and data sheets for products that are new, products that have been updated, and, I fear, some products that are no more than gleams in their maker's eyes. Many of the announcements have very impressive corporate logos embossed on them, but an encouraging number bear the names of small, entrepreneurial start-up firms. Since we've launched into a decade still young enough to be full of hope and promise, I see the reemergence of the "garage shop" as a most promising omen for the future.

The 1980s have been simultaneously held up as the decade of entrepreneurs and the decade of megamergers. For many people, "The Dream" consisted of having a great idea, starting a small company, and quickly selling out to a large, multinational conglomerate. During the last six months of 1989, I read a number of articles which trumpeted the notion that the only way for a company to survive in the '90s was for it to have a billion-dollar budget, a lean, mean staff numbering in the thousands, and a debt load that would sink most developing nations. According to this line of reasoning, this world has now become so complex that only massively organized teamwork can work to solve problems. I agree that it's important for our largest corporations to be healthy, dynamic organizations, but a thriving class of entrepreneurs and small companies is vital for economic (and social) well-being in this decade, and the next century.

Let's look at just one facet of the situation. There are several corporations that are able to fund R&D efforts involving thousands of people and millions of dollars. I get press releases from some of these programs, usually touting the latest advancement in the state of basic research. I'm in awe of their capabilities, and they seem to be making serious progress toward solving some mighty big problems. The trouble is that they are so much caught up in big problems, and the mind-set is so oriented toward big solutions, that they have trouble seeing the small problems and (hopefully) small solutions that make up much of our lives. An individual engineer, on the other hand, may well spend timegetting to know a small problemonanintimate level, and find a solution that fits perfectly.

If the engineer then goes on to market the solution, our economy has gained a company that will support one, or five, or fifty people for many years. It may never have profits of a billion dollars a year, but then most of us don't feel weneed quite that much to get by. I'mseeingevidence of more and more people deciding that the income from a small company, coupled with the emotional fringe benefits of running a small company, are more than enough to live on. The dynamic nature of these small companies is crucial to a thriving economy, every bit as important as the stability and power of the huge corporations.

The '90s promise to be a decade of dramatic change. Historically, the decades around the turn of a century are illed with social and technical change, and the turn of a millennium is bound to have enormous psychological effect on most people. The thousands of small companies and individuals working to solve practical problems will give us a technological and economic diversity that will be part of a strong and growing global society. We're in for an exciting ride. I'm glad that I'm here to see it.

FOUNDER/ EDITORIAL DIRECTOR
Steve *Ciarcia*

PUBLISHER
Daniel Rodrigues

EDITOR-in-CHIEF
Curtis Franklin, Jr.

PUBLISHING CONSULTANT
John Hayes

ENGINEERING STAFF
Ken Davidson
Jeff Bachiochi
Edward Nisley

CONTRIBUTING EDITORS
Thomas *Cantrell*
Jack Ganssle

NEW **PRODUCTS EDITOR**
Harv Weiner

CONSULTING EDITORS
Mark Dahmke
Larry Loeb

CIRCULATION COORDINATOR
Rose Mansella

CIRCULATION CONSULTANT
Gregory Spitzfaden

ART & PRODUCTION DIRECTOR
Tricia Dziedzinski

PRODUCTION ARTIST/ILLUSTRATOR
Lisa Ferry

BUSINESS MANAGER
Jeannette Walters

STAFF RESEARCHERS

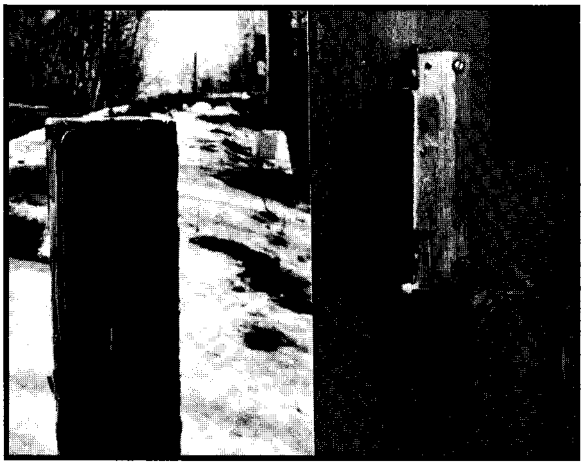
Northeast
Eric Albert
William Curlew
Richard Sawyer
Robert Stek
Midwest
Jon Elson
Tim McDonough
West Coast
frank Kuechmann
Mark Voorhees

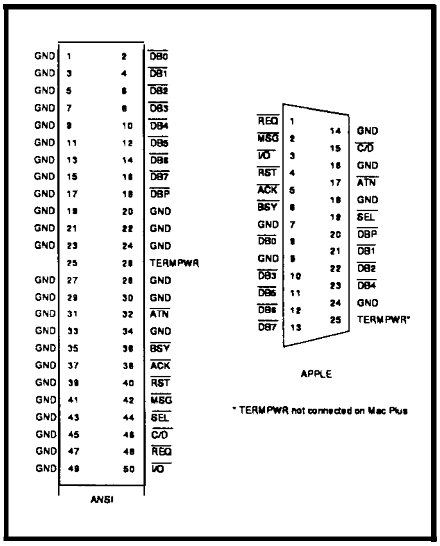
Cover Illustration
by Robert Tinney

CIRCUIT CELLAR INK®
THE COMPUTER APPLICATIONS JOURNAL

In This Issue...

DEPARTMENTS





Circuit Cellar BBS-24 Hrs. 300/1200/2400 bps, 8 bits, no parity, 1 stop bit, (203) 871-1988.

The schematics provided in Circuit Cellar INK are drawn using Schema from Omation Inc. All programs and schematics in Circuit Cellar INK have been carefully reviewed to ensure that their performance is in accordance with the specifications described, and programs are posted on the Circuit Cellar BBS for electronic transfer by subscribers.

Circuit Cellar INK makes no warranties and assumes no responsibility or liability of any kind for errors in these programs or schematics or for the consequences of any such errors. Furthermore, because of the possible variation in the quality and condition of materials and workmanship of reader-assembled projects, Circuit Cellar INK disclaims any responsiblity for the safe and proper function of reader-assembled projects based upon or from plans, descriptions, or information published in Circuit Cellar INK.

CIRCUIT CELLAR INK (ISSN 08968985) is published bimonthly by Circuit Cellar Incorporated, 4 Park Street, Suite 20, Vernon. CT 06066 (203) 875-2751. Second-class postage paid at Vernon, CT and additional offices. One-year (6 issues) subscription rate U.S.A. and possessions $ 14.95, Canada/Mexico $17.95, all other countries $26.95-surface, $38.95 air. All subscription orders payable In U.S. funds only, via international postal money order or check drawn on U.S. bank. Direct subscription orders to Circuit Cellar INK, Subscriptions, P.O. Box 2099, Mahopac, NY 10541 or call (203) 875-2 199.

POSTMASTER: Please send address changes to Circuit Cellar INK, Circulation Dept., P.O. Box 2099, Mahopac, NY 10541.

READER'S INK

Letters to the Editor

AN ISSUE OF ACCURACY

I don't want to start a semantic argument, but "Steve's Own INK" in CIRCUIT CELLAR INK #13 pulled my chain. The proliferation of high-technology Tinker Toys has, at times, caused a lot of grief. The perception that the display of many digits means awesome accuracy is a serious problem.

If I were to calculate the value of π as 2.94159268, the result would be reasonably precise, but totally inaccurate. Of what value is this great precision when few devices can be calibrated to an accuracy greater than 0.01%?

We live in a world of illusion. Please don't perpetuate our ignorance by confusing precision with accuracy. My gross error in the calculation of π is only 6.5%: not very good, but close enough for some applications. I must agree with Steve's plea that common sense must prevail.

Robert C. Woodman
Bay Springs, MS

MAKE SCHEMATICS MORE USEFUL

I have been reading electronics and computer magazines for many years now, both before and after the computer revolution. In the last few years, almost all of these magazines have established bulletin boards for downloading data connected with the projects in the magazine, but I don't think anyone to date has put schematics and printed circuit board layouts from an affordable program on their BBS. All magazines print this information in the pages of their magazine, but due to errors in reprinting, and paper and original printing flaws, this data is sometimes less than perfect.

Why hasn't one of these magazines taken a giant step forward and adopted a software package to do schematics and printed circuit board layouts, from one to four layers? This package would need to be cost effective for most readers, and be priced in the $50–$60 range. If it cost much more, it would be out of reach for many readers. I know that there are packages out there, like Autosketch, that can be purchased **in this price range.**

Perhaps you **could provide a program at a discount for subscribers to the magazine. This could very possibly set an electronic software standard. If the package is accepted by your readers, and they are used to using it, they will probably want to use the same package at their place of work.**

I hope there is one magazine out there that is willing to step into the computer age, rather than just write about it.

Wayne R. Anderson
Smyrna, GA

Letters such as yours help fuel the conflict between what we would like to provide for our readers, and what reality will allow us to provide. We have been discussing, for some time, a way to put schematics on the Circuit Cellar BBS. Problems arise when we try to take into account the variety of computers owned by our readers, and the work habits of our staff.

In order not to slight subsets of our readers, we would need to provide programs for MS-DOS, Macintosh, Amiga, Atari ST, and (heaven help us) some sort of CP/M platform. Next, we would have to take schematics from Schema, which our engineering staff refuses to give up, and port them to the new format.

We are constantly searching for new ways to make CIRCUIT CELLAR INK more valuable to the readers. Increasing the usefulness of the Circuit Cellar BBS is certainly high on our list, but it is unlikely that CIRCUIT CELLAR INK or any other computer magazine will be able to offer a cheap, full-function EE-CAD program in the near future.

ABOUT THOSE PARTS...

Ever since the advent of the IBM PC there seems to have been a steady erosion in the quantity and quality of hardware-oriented books, magazines, and articles. I thought it was all over for serious computer experimenters. One figure carried on and even progressed. I would like to commend Steve Ciarcia **and** CIRCUIT CELLAR INK.

Issue #9 was a gem. Every article was outstanding The two articles on neural networks ("The Adaline Learning Neuron" by Scott Farley and "A Neural Network Approach to Artificial Intelligence" by Christopher Ciarcia) are examples. These articles were better than many books and articles that I have read on the subject.

Lately I have been finding it more difficult to build some of the construction articles. The "high-end" chips are hard to find in small quantities, if at all. One distributor even told me I was ineligible to buy anything. Am I missing something?

Alan Land
Pittsburgh, PA

I am not much on writing "fan mail" but I just wanted to tell you that I think CIRCUIT CELLAR INK is great. I have worked on computers since "way before BYTE" and was very glad to see, once again, a magazine devoted to the serious hardware folks. I am also a "pro" (whatever that means), and am pleased to tell you that I found more useful ideas and information in my first issue than in a year's worth of several other magazines I get.

As far as feedback goes, I like what you're doing just the way you are doing it. I would appreciate an on-line or disk-based cumulative index. An index that I could search by keyword or combinations would be a great help.

Another interesting service you might consider is printing data sheets on new and interesting ICs. "Silicon Update" addresses much of this need, but it sure would be nice to have a tear-out sheet that you could put into a standard three-ring notebook. I would suggest getting some of your parts-house advertisers to stock the "chip of the month."

Finally, you've inspired me: Please send an Author's Guide. I have an idea or two I'd like to submit.

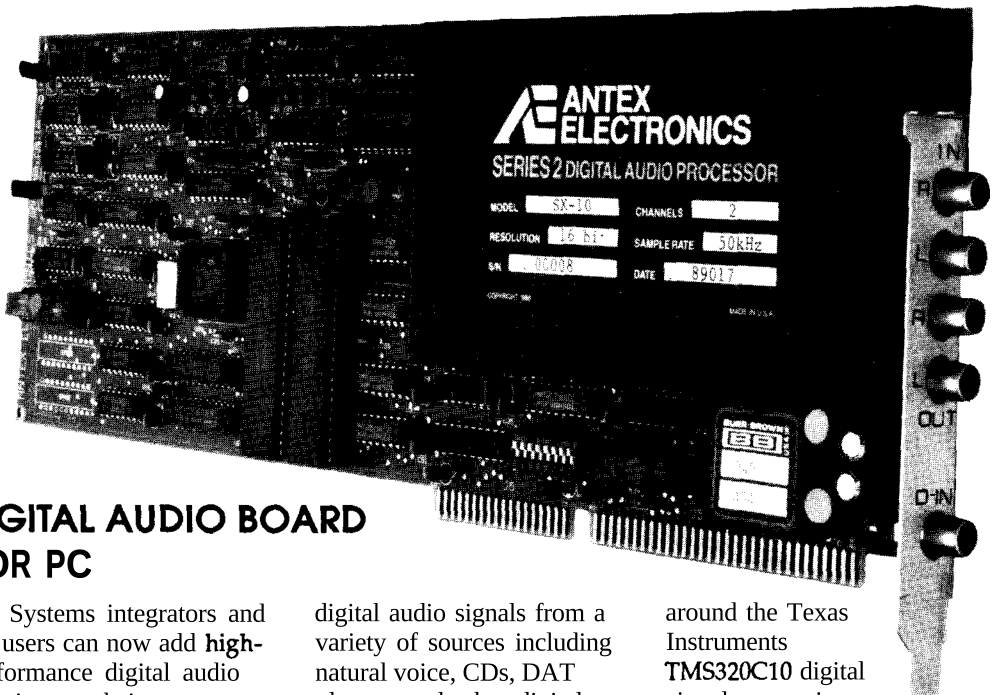
Carl K. Zettner
San Antonio, TX

Thanks to both of you for your kind words. It's nice to get an occasional pat on the back.

Nothing is morefrustrating than not being able to get apart you need. Wearegoing *to ty to address this by printing sources along with construction articles. There will be some exceptions, but it should generally be possible for individuals to purchase parts for all of our projects in single quantities.*

Anyone can get a CIRCUIT ***CELLAR INK*** *Author's Guide by writing and requesting one. The address is:*

Circuit Cellar INK Author's Guide
4 Park Street
Vernon, CT 06066



DIGITAL AUDIO BOARD FOR PC

Systems integrators and PC users can now add high-performance digital audio functions to their systems and application programs with a new board from Antex Electronics Corporation. The Series 2/Model SX-10 digital audio processor features multichannel ultra high fidelity, direct-to-disk sound sampling and reproduction for IBM PCs and compatibles.

The unit is designed to fit into an expansion slot of any IBM AT, PS/2 Model 30, or any compatible 286/386 computer and allows the user to receive both analog and digital audio signals from a variety of sources including natural voice, CDs, DAT players, and other digital devices. The SX-10 can digitize two audio channels, converting the stereo sound into digital input that can be stored on a hard disk or CD ROM. Once stored, the user can manipulate the audio data and perform mixing editing, and archiving tasks. The SX-10 can record and playback simultaneously, so "overdubbing" using a PC becomes a simple process.

The Series 2/SX-10 is a full-length board designed around the Texas Instruments TMS320C10 digital signal processing chip running at 25 MHz. Sampling rates are software programmable, and can range from 6.25 kHz to 50 kHz in 100-Hz steps. Resolution is 16 bits, and audio bandwidth is 20 Hz to 20 kHz. The board also allows for 4:1 ADPCM (Adaptive Differential Pulse Code Modulation) data compression for decreased disk storage requirements.

An on-board digital input interface allows the SX-10 to be connected directly to CDs, DATs and other digital sources. Programmers can achieve direct-to-disk recording and playback by using an optional Series 2 driver to call the SX-10 from a high-level programming language such as QuickBasic, Pascal, Turbo Pascal, and C. An editing program, PCMEDIT, is also available to allow viewing and manipulating up to three audio files on-screen.

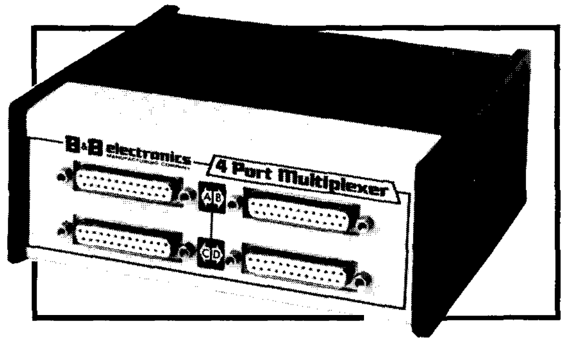
The SX-10 requires an IBM AT or higher with a 1-to-1 interleave factor disk controller, a hard disk with a maximum 28-millisecond access time, and DOS 2.0 or greater. A special daughterboard is also available to allow PCM digital output to optical disk drives, DAT machines, and other digital recording devices.

The price of the SX-10 is $1995.00 and the optional daughterboard is $450.00. A one-time fee for the software driver is $750.

Antex Electronics Corp.
16100 South Figueroa Street
Gardena, CA 90248
(213) 532-3092

Reader Service #190

FOUR-PORT MULTIPLEXER



Combining the signal and handshaking lines from four different RS-232 cables and sending them up to 4000 feet on a single cable is possible with the four-port multiplexer, Model 232FPM, from B & B Electronics. At the far end, another 232FPM separates them into the four different cables. Each port of the 232FPM supports two data lines (TXD and RXD) and four handshaking lines (RTS, CTS, DTR, and DSR) and is wired as a DCE port. A typical application would be to connect a small cluster of terminals and printers located up to 4000 feet away from their host computer.

The 232FPM also features a built-in loopback mode to test for installation problems. It automatically falls into the loopback mode if the two-pair interconnection wiring is broken, or if the power is off at the far end. The interconnecting wire should be a two-pair twisted telephone cable for best results.

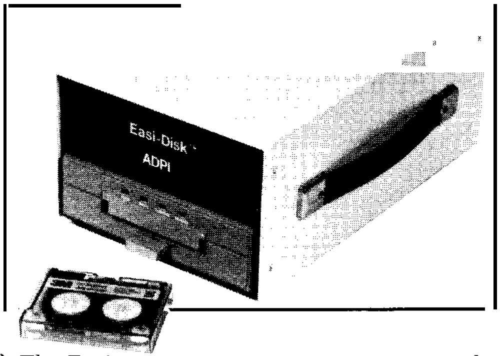
The 232FPM can handle baud rates up to 9600 bps with any combination of bits, parity, and so on. Converters are available to change the DCE port configuration to DTE. These DCE-to-DTE converters cross RS-232 connector pins 2 to 3, 4 to 5, and 20 to 6 and 8.

The Model 232FPM sells for $149.95 including power supply. The Model 232DTE DCE-to-DTE convertor sells for $15.95.

B & B Electronics
4000 Baker Road
P.O. Box 1040
Ottawa, IL 61350
(815) 434-0846

Reader Service #191

"PLUG AND RUN" PORTABLE BACK-UP SYSTEM



A portable hard disk back-up system that operates through the RS-232 serial port of any MS-DOS PC has been announced by Analog & Digital Peripherals Inc. (ADPI). The Easi Tape is a self-contained, 4"H x 6"W x 12.25"D mini-cartridge tape system that emulates a large floppy disk. No special interface cards are required, so the unit is truly a "plug-and-run" device that can be moved between computers as required.

Through the use of an included proprietary MS DOS compatible software driver, Easi Tape accepts standard DOS commands, such as COPY and XCOPY, and can back up 32 megabytes of data. Reed Soloman ECC error checking is incorporated for virtually error-free back-up.

The Easi Tape system can also be used in data logging applications to provide 32 megabytes of storage over RS-232, GPIB-488, RS-422, RS-485, 20-mA current loop, or 8-bit bidirectional parallel interfaces. The unit can be controlled either manually or from the host, so it can accommodate both "smart" and "dumb" devices. Baud rates from 110 to 38.4k are available, and the unit can be used for CAD/CAM archival or active storage.

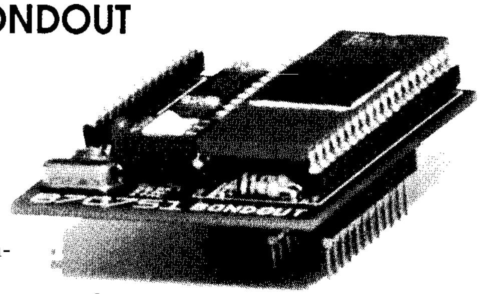
Other applications with MSDOS-based computers include: disk capability for single-board computers, downloading and uploading part information for CNC machines, direct reading and writing of 5.25-inch formats on 3.5-inch-based laptops, and external storage for hand-held PCs.

The Easi Tape system is available with case and power supply for $1295.00. The system with LED display and manual controls is available for $1495.00, and the system with an LCD display and manual controls is available for $1595.00. All prices reflect single quantities, and multiplequantity discounts are available.

Analog & **Digital Peripherals, Inc.**
251 South Mulberry St.
P.O. Box 499
Troy, OH 45373
(513) 339-2241

Reader Service # 192

87C75 1 BONDOUT BOARD



A special version of the Signetics 87C751 microcontroller, called a "bondout," has been incorporated into a device to aid in writing and debugging software. The 87C751 Bondout Board from Parallax Inc. plugs into a socket normally occupied by a DIP 87C751. The bondout is functionally equivalent to the 87C751, but its program resides in an external EPROM rather than in the microcontroller itself. The Bondout Board accepts a 2764 or 27128 as the external EPROM.

The external EPROM allows faster and less costly programming and erasing than with an actual 87C751. The external EPROM Board can also be used with a ROM Emulator, such as the Parallax 2764 ROM Emulator, to reduce normal programming, testing, and erasing cycle time from 2040 minutes to two seconds or less.

The Bondout Board provides a 12-MHz clock and power-on reset. The Parallax ROM Emulator, if used, plugs into the board's EPROM socket and connects to the parallel port of an IBM PC or compatible. When used with the ROM Emulator, the system features command line software, which can be run from batch files, for automatic downloading after assembly; a full-screen editor for program modification; and a tristate reset output to restart the target system after downloading.

The Parallax 87C751 Bondout Board is available for $219. The Bondout Board and 2764 ROM Emulator may be purchased together for $348.

Parallax, Inc.
6200 Desimone Lane, #69A
Citrus Heights, CA 95621
(916) 721-8217

Reader Service #193

UNIVERSAL CROSS-ASSEMBLER

A table-based cross-assembler that compiles programs for many different target processors on any MS-DOS computer has been announced by Universal Cross-Assemblers. Version 2.00 of the Cross-16 Meta-Assembler allows the user to assemble source code from over 20 different microprocessors, microcontrollers, and digital signal processors, using the original manufacturer's mnemonics. The program reads the assembly language source file and a corresponding assembler instruction table, and writes a list file and an absolute hexadecimal machine file in binary, Intel, or Motorola formats. This hex file can then be downloaded to most EPROM programmers, EPROM emulators, and in-circuit emulators.

The two-pass (a third pass if a phase error occurs) assembler supports arithmetic operators and integer constants identical in form and precedence to the ANSI C programming language, as well as several common assembler conventions. Informative error messages identify the exact row and column in which syntax errors occur, and are compatible with many programming editors. Processor families include: 180x, 6502, HD64180, 680x, 8048, 8052, 8085, 8096, COP400, COP800, NSC800, TMS32010, TMS370, TMS7000, Z8, Z80, and Z180.

The Cross-16 User's Manual includes full directions for writing new, and modifying the existing, processor tables. Since many new processor instruction sets are merely supersets of one of the supplied processors, this can be as simple as adding lines to an existing table. This feature prevents the assembler from becoming obsolete. The manual also includes an example source file for each processor on disk.

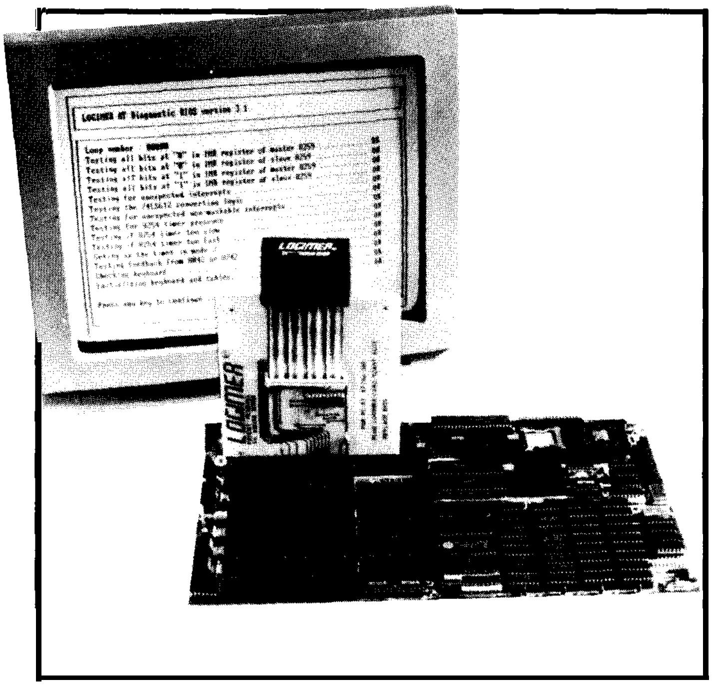
The Cross-16 will run on any system that uses MSDOS Version 2.0 or later, at least 256K of RAM, and a 3.5" 720K or 5.25" 360K floppy disk drive. The program is not copy protected. The suggested retail price for the Cross-16 Meta-Assembler is $99.00, including airmail shipping and handling.

Universal Cross-Assemblers
P.O. Box 6158
Saint John, N.B. Canada E2L 4R6
(506) 847-0681

Reader Service #194

NEWPRODUCTNEWSNEVVPRODUCTNEWS

IN-CIRCUIT DIAGNOSTIC SYSTEM FOR PC



A diagnostic system designed to facilitate the troubleshooting and repair of IBM PC/XT, AT, 80286, 80386, and compatible system boards has been announced by Total Power International Inc. The LOGIMER system consists of a hardware/firmware add-on board that contains diagnostic codes. It comes with three ROM chips and plugs into any expansion slot. The ROM chips, which are installed in place of the existing ROM BIOS chips, contain a program that makes more than a thousand tests in less than one minute, and displays setup instructions and error messages on-screen.

In the event of a computer screen malfunction, a two-digit alphanumeric display on the card will display hexadecimal error codes. The user's manual and supplementary diskette provide additional diagnostic information.

LOGIMER has the capability for detecting intermittent breakdowns. It will still initialize both displays, perform its diagnostic tests, and relay useful information about the system under test with many intermittent controller, timer, and memory chips. It also can carry out loop tests to allow testing during burn-in, and its results may be output to a printer or screen.

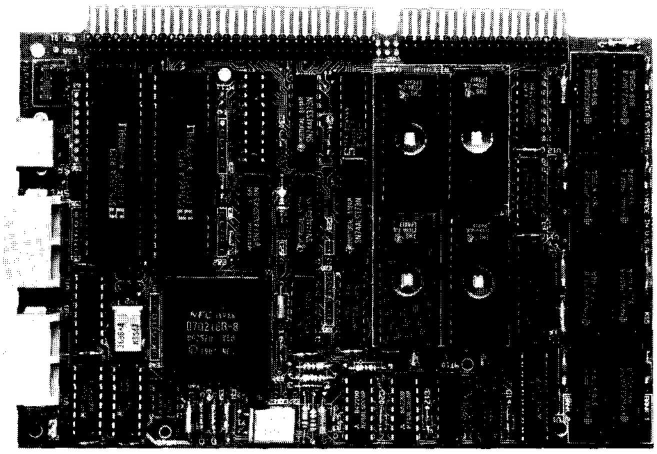
The LOGIMER card can locate the exact number of a defective chip, and shows defective RAM chips on a screen error map to allow easy replacement. It performs complete memory diagnostics including EMS memory up to 16 megabytes, and locates up to 70% of real breakdowns on the motherboard.

The LOGIMER card is priced at $399.00.

Total Power International, Inc.
418 Bridge Street
Lowell, MA 01850
(508) 453-7272

Reader Service # 195

ROM-BASED CPU CARD



A controller card from Kila Systems makes diskless stand-alone or embedded applications easy to design. The KS-5 is a ROM-based CPU card that is configured for an IBM PC/AT bus. Using a passive backplane and off-the-shelf PC/AT-compatible cards, a user can run existing applications and MSDOS directly from EPROM. MS DOS takes up 83K of the 256K total ROM space, leaving 173K available for application programs.

The card features the NEC 70216 (V50) CMOS CPU running at 8 or 10 MHz with zero wait states. From 512K to 1 megabyte of dynamic RAM is available, but usable RAM excludes the ROM space, which can be up to 256K. Five RS232 serial ports are available at programmable baud rates up to 38.4 kilobaud. The card's 98-pin edge fingers are designed for insertion into a passive backplane. A 98-pin I/O expansion connector with the same pinout as the PC/AT bus is available for piggyback cards to interface printers, floppy drives, SCSI, and keyboard.

The card is 4.5" wide and 6.9" long and draws 400 mA at 5 volts. An on-board converter provides ±12 volts. The V50 also features a power-down mode, and an all-CMOS version, which draws only 200 mA, can run off of a battery or solar cell.

The base price of the KS-5 is $299.00 and significant quantity discounts are available.

Kila Systems
655 Hawthorne Avenue
Boulder, CO 80304
(303) 444-7737

Reader Service #196

Correction:

In CCINK Issue #13 "New Product News." the telephone number for **Macrochip Research Inc.** was incorrect.

The correct number is (214) **242-0450**.

We are sorry for any inconvenience this may have caused.

Letters to the INK Research Staff

VISIBLE INK

answers: clear and simple

MYSTERY CHIPS

Recently, I was given several 28-pin ICs with the following printed on them:

i wp9083511
s70099
u6250248s

They all have a UV PROM-type window on them in addition to the writing. I would like to know if you can tell me what they are.

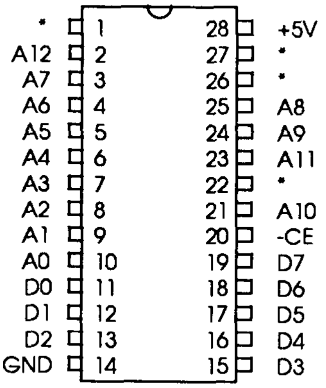
I would also like to know if there is software available for an AT-compatible 80386 computer that performs math operations like the HP-28C calculator does.

P.L. Robertson
Suffolk, VA

You obviously ***have some*** *house-marked EPROMs, but the markings don't provide any indication of* the size or *family. You'll need to do a little electrical exploring to determine how* ***they*** *work; there are not that many choices, so a few evenings should suffice.*

Unless you've got something really weird, the EPROMs may have ***anywhere from*** *64K to 512K bits, with corresponding type numbers from 2764 through 27512. The* smaller ones *are* ***8K-byte*** *devices, while the largest have* ***a*** *full 64K bytes.* ***Fortunafely,*** *they* **all** *usemoreor less* ***thesamepinout,so*** *fhereareonly a handful* of pins ***to*** *test.*

The generic pinouf is:



The pins marked with asterisks have thesefunctions:

Pin	*2764*	*27128*	*27256*	*27512*
1	*VPP*	*VPP*	*VPP*	*A15*
22	*-OE*	*-OE*	*-OE*	*-OE/VPP*
26	*n/c*	*AZ3*	*A13*	*A13*
27	*-PGM*	*-PGM*	*A14*	*A14*

Because those chips are most likely *already programmed, you can wire up a test circuit to read out the data and see what you'vegot. The EPROMs require an address to select one of the* stored bytes, plus chip enable ***(-CE)*** and ***output*** enable ***(-OE)*** control *lines toactivate the infernal circuifyand* ***output*** *drivers. Depending on which chip you've* ***got,*** *you must provide* ***any-*** wherefrom ***13*** to 16 address lines; the least-significant line is ***A0*** *and the most-significant one will be either* ***A12, A13, A14,*** *or* ***A15.***

Wire *up power and ground to pins 28 and 14, respectively. The remaining inputs can come from DIP switches with* ***10K-*** *ohm pull-up resistors; wire the* ***DIPswitchesso*** *that theyground* the inputs when closed. You can monitor the outputs ***(D0*** *through* ***D7)*** *on* ***LEDs*** *connected to* ***+5V*** *through* ***1.5K-ohm*** *resistors; the* ***LEDs*** *will be dim with only 2* ***mA,*** *but you won't need* to wire *up* driver circuits. Remember that the ***LEDs are*** *OFF when the* ***outputs*** *present a logic* ***1*** *or are disabled!*

Both ***the-OEand-CEpins*** *must begrounded fogefanydafa* onfheoufpuf ***pins,so that's a good place to start.*** Makesurefhaf *the four "odd" pins* ***(1, 22, 26,*** and ***27)*** *are high, then vay the other address lines to see whether you'regeffingany data on* ***the*** outputs. A blank EPROM will have FF hex stored in ***every*** *location, so if the* ***LEDs*** *remain* **off** *that's the most likely reason.*

If the ***LEDs*** *blink in interesting patterns, you're in luck! Now* start testing the *"odd" pins* to see ***if you*** get different data *foreachposifion;if fheoufpufs don't changewhen* ***oneof the pins*** *is low, if's not an address line. Simple trial and error will hone* in *on* the right answer fairly quickly.

Once you know the EPROM size, you can erase them and fy programming new data. Space doesn't let us go info the details here, but once you *know* the EPROM family there are *only a few choices for the programming method. Data sheets for some known EPROMs (available from the usual mail-order* suppliers) *and* a little head-scratching will show you that the ***programming*** can *be done quite easily;* you *can probably wire up a simple circuit using your '386 clone and a few latches!*

The2764 EPROM programmer in the DDT-51 project may serve as inspiration; that was written up in volume VII of "Garcia's Circuit Cellar." You will need afew moreaddress bits if your EPROMs are bigger, the programming voltages may be different, and the programming algorithms are much more intelligent, but it's all possible with software.

As far as an HP-28C emulator for your '386 clone goes, Curt rummaged around in his pile of diskettes and came up with a public-domain program that will appear on the Circuit Cellar BBS long before you read this. Dial us up and t y it out!

LONG-DISTANCE X-10

A couple of questions about the X-10 system:

The Heath-Zenith #SL-5320 outputs an X-10 signal when its infrared detector is activated. If the detector is at a neighbor's house, I want to receive the signal at my home (neighborhood watch scheme).

How far back up the power line will communications take place? My guess is up to the first power transformer.

Please verify that a 0.01-µF 240-V capacitor connected across the two "hot" lines of the usual house wiring system will allow signals on one leg to operate equipment on the other leg as well.

Bob Fabris
San Jose, CA

You aren't alone in wondering just how far "up the power line" your X-20 signals will carry, but you may be the first person we've encountered who actually wants your neighbor to receive your codes!

There is no simple way to predict whether the X-20 carrier will make it to a given point in your own home, let alone out to thepoleand back to another building. Znfact, we have all had an X-IO outlet or switch controller mysteriously "stop working" for a few hours, days, or weeks, then start up again as though no thing was wrong.

The rule of thumb is that the signals will not pass through the distribution transformer on the pole, so if your neighbor's house is on a different transformer you "can't get there from here" no matter how much you want to.

A 0.01-µF capacitor presents about 150 ohms of reactance at ZOO kHz, which may be a little high for houses with very low line impedance, so you may need a 0.1-µF cap to punch the signal between the circuits. In any event, the capacitor voltage rating should be two or three times the expected line voltage, so a 450- or 650-volt AC capacitor is the minimum you should use. A 250-volt capacitor doesn't give you much leeway when the power company jacks up the voltage by 10 percent!

If you build such a unit, you should include a small safety fuse and mount the unit in a grounded or well-isolated box. Unlikemostprojects, thisonemaykill you (or a bystander) if you aren't careful about the details.

A FEW WORDS FROM THE STAFF

Dear Readers:

We get *a lot of letters **like** this:*

I'm having trouble getting my project running and need your help. I am **using a** 32-bit microprocessor with 45 interrupt sources, an ADC and DAC updated by **interrupt-**driven firmware, and DMA circuitry for my **custom-de**signed hard disk controller. My tools include a soldering iron, diagonal pliers, and a continuity checker, but I do not have an oscilloscope or a logic probe.

***C'mon,** folks, we are not magicians here! Just as you* wouldn't attempt to *play pro football without a helmet and full pads, you shouldn't expect to debug high-speed digital electronics without **proper** tools.*

For example, if that microprocessor doesn't start up, you're *going to have to verify that the clock is running, that the control signals arefunctional, that the bootstrap EPROM is delivering the right data, and so forth. You don't need a $20,000 logic analyzer **todo** that, **but*** you *will **needat leastagoodoscilloscope...**a logic* probe won't cut *the* mustard.

*Understand, there's quite a lot of electronic experimentation you can do with **rudimentay tools.** Above acertain level of complexity, however, you simply have to make the commitment* to *obtaining theproper tools. This fact is not unique to **electronics:*** Whether you're *designing a* single-board *controller, **build**ingan **Early** American dining room suite, or preparing aCJ-7 to run the Baha 1000, when you progress **from** "foolingaround" to serious, professional-quality projects you have to have the right tools for the job.*

*Admittedly, **a good** scope* will set you backabout a kilobuck, *but that's the cost of **entry** to the game. We don't want to discourageyoufrom experimenting with electronics, but **wealso** don't want to encourage you to waste your time tying to do something that simply can't be done.*

In Visible INK, the Circuit Cellar Research Staff answers microcomputing questions from the readership. The representative questions are published each month as space permits.

Send your inquires to: INK Research Staff
c/o Circuit Cellar INK
Box 772
Vernon, CT 06066

All letters and photos become the property of CCINK and cannot be returned.

IRS

201 Very Useful
202 Moderately Useful
203 Not Useful

Build A Simple SCSI-to-Anything Interface

Jim MacArthur

Take Advantage of the Spec to Simplify Your Designs

The SCSI (Small Computer System Interface) bus, as envisioned by the ANSI committee, is a parallel bus intended to connect small computers to intelligent peripherals. Because of SCSI's high degree of "politeness" (error recovery procedures, message passing, etc.), it is impossible to design a peripheral which fully implements the SCSI spec without some kind of microcontroller. This fact, plus the complexity of the spec itself, has kept hackers from taking full advantage of the bus.

However, it is possible to design a circuit which can interface SCSI to virtually anything, using as few as eight ICs (buffers and gates only) and no microcontroller. The trick is to implement enough of the spec so that the circuit doesn't interfere with "legitimate" SCSI devices, while flagrantly violating the rest of the spec in the name of hardware simplicity.

Of course, there is a tradeoff: It is necessary to write a special SCSI driver for thecomputer tocommunicate with the interface circuit. Because the circuit uses a very simple protocol, however, writing the driver is far less difficult than writing SCSI firmware for an intelligent peripheral.

The first part of the article will give a quick overview of the SCSI spec. Then follows a description of the basic hardware, which interfaces SCSI to a generic 8-bit data-address bus. From this circuit, one can design an interface to virtually any parallel bus or controller IC. Possibilities include interfaces to the PC bus, STD bus, Metrabus, CAMAC, A-bus, ARCnet, A/D converters, GPIB, RS-232, Centronics ports, and so on. The remainder of the article will present a simple driver to interface a Macintosh to the circuit.

THE SCSI INTERFACE

SCSI is an 8-bit parallel bus with 18 signal lines, one termination power line, and 31 ground lines, for a total of 50 lines. It is typically implemented as a 50-conductor flat ribbon cable, or as 25 twisted pairs. The maximum specified length is six meters. In the misguided interest of saving space, Apple removed 25 ground lines and implemented the SCSI interface as a DB-25 connector, thereby restricting the

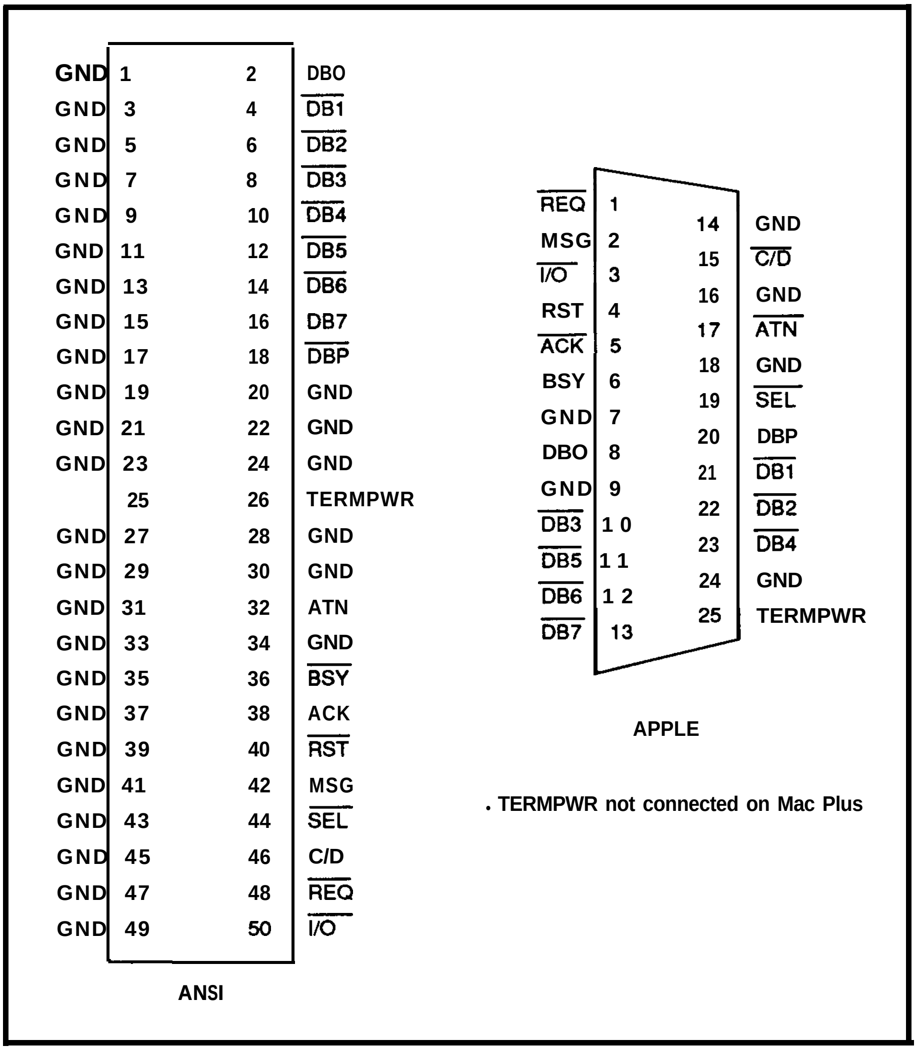


Figure 1 —*The* **SCSI specification** *calls for a 50-pin* **connector** *with* 18 *signal* **lines. one termination** *power line,* **and31 groundlines. In designing the Macintosh, Apple removed 25 ground lines,** *saving space* **in** *the connector, but limiting the maximum length for a reliable cable.*

DB0\–DB7\, DBP	8 data **lines** plus odd parity.
BSY	Busy, asserted by initiator or target to gain control of the bus. Whoever is asserting BSY owns the bus.
SEL\	Select, asserted by initiator to establish communication with a target.
MSG	Message, asserted by target to indicate message phase.
CID\	Command/Data, asserted by target to indicate command phase.
I/O	In/Out, asserted by target to indicate data direction is toward initiator.
ATN\	Attention, asserted by initiator as a request to send a message to the target.
RST	Reset, asserted by initiator to reset the bus.
REQ	Request, asserted by target to start transfer of one byte of data.
ACK	Acknowledge, asserted by initiator to complete transfer of one byte of data.

Table 1—*The 18 SCSI signals. A nineteenth signal. TERMPWR, supplies +5V to the pull-up* **resistors in the terminators.**

maximum length for reliable operation to six feet or so. The differences in the **pinouts** for the two implementations are shown in Figure 1.

Signals are driven by open-collector drivers that must be able to sink 48 **mA.** All signals are active low. When not being driven (the SCSI **spec** uses the **term "asserted"),** signalsarepulled high by terminators at both ends of the bus. The terminators consist of a 220-ohm resistor to **+5V** (or TERMPWR) and a 330-ohm resistor to ground for each signal. In addition to their role of **"deasserting"** signals, the terminators reduce ringing on the bus by absorbing signals that reach the end of the cable, rather than reflecting them.

SCSI allows up to eight devices to be physically attached to the bus. Each device is assigned a unique bus ID from 0 to 7. The SCSI **spec** divides devicesintoinitiatorsand targets. With few exceptions, host computers are initiators and peripherals are targets. Although SCSI **was** carefully designed to allow multiple initiators, the vast majority have only one. Therefore, our first simplifying assumption is that there is only one initiator on the bus.

The 18 SCSI signals are shown in Table 1. A nineteenth "signal" is TERMPWR, which is connected to the **+5V** supply (through a rectifier1 of each device on the bus. It supplies power to the pull-up resistors in the terminators.

Starting from a Bus Free phase **(BSY** and **SEL** deasserted), an initiator arbitrates for control of the bus by asserting **BSY** and the data bus line corresponding to its ID. After 2.2 **µs,** the initiator inspects the data bus. If no data lines higher than its own are asserted, it gains control of the bus by asserting SEL\. Otherwise, it **deasserts** BSY\ and waits for the next Bus Free phase.

Once on the bus, the initiator chooses which device it wishes to communicate with by asserting the data bus with the logical OR of its ID and that of the target. It then deasserts **BSY\,** thus entering the Selection phase. When a target sees **SEL** and its ID\ asserted and **BSY** deasserted, it responds by asserting **BSY\.** The initiator completes the selection by deasserting **SEL\.** The target now owns the bus.

The target then manipulates the phase lines **MSG\,** C/D\, and I/O\ to send and receive data, commands, and messages to and from the initiator. When the command is finished, the target gets off the bus by deasserting BSY\. We won't get into exactly what the target does once it has the bus, because that's the part of the spec we're going to ignore.

THE HARDWARE

Now let's look at the problems of designing a device that only partially conforms to the SCSI spec. The first problem is that we want our interface to be invisible to the initiator when it is running its normal SCSI drivers. This is important because the host computer often initializes the bus by selecting every device ID to see what's out there. If it selects a device that requires a special driver, the initiator will probably hang. Our solution is to modify the selection protocol so that our interface will respond to selection only if the initiator asserts SEL\ and the target's ID, but not the initiator's ID. In order to simplify the hardware, we assume that the bus has only one initiator, and that it is located at ID 7.

Once our target has gained control of the bus by asserting BSY\, we are free to play with the SCSI signals any way we want, with the following exceptions: BSY\ must remain asserted, and SEL\ and RST\ must remain deasserted. As long as these rules are followed, there is nothing the initiator or target can do on the bus that will affect the other targets. That gives us a pretty free rein with the remaining six control signals.

Before we go hog-wild, we should consider any constraints imposed by the initiator's hardware. On the Macintosh, and in many other small computers, the SCSI interface is handled by a single-chip SCSI controller, the NCR 5380. While it can assert all six control lines, it can't assert them all at the same time. When configured as an initiator, it can only assert ATN\ and ACK\. When configured as a target, it can only assert MSG\, C/D\, I/O\, and REQ\. Because target mode gives us twice as many lines to play with, we will reconfigure the initiator as a target when talking to our interface.

The 5380 uses the I/O\ line to control the direction of its transceivers, so I/O\ is forced into service as a write/read line. The initiator is pretending to be a target, so "in" now means towards our interface. Asserting I/O\, therefore, indicates a command which writes to the interface. The phase lines MSG\ and C/D\ are natural candidates for the addresses, and REQ\ is used as the strobe line. This leaves ATN\ free as an interface-driven interrupt line. ACK\ can be used asaninterface-drivendata ready or wait line.

Figure 2 illustrates these techniques with an interface to a generic bus with eight data lines and eight address lines (expandable to 16). The selection process is implemented by one-of-eight selector U3, along with U6b and U7. When the initiator wants to select the interface, it asserts SEL\ along with the interface ID, but *not* its own ID (7), and then deasserts BSY\. When this occurs, U7c will go high, setting the S-R flip-flop formed by U7a and U7b. This sets BUSY high, which causes open-collector NAND buffer U8 to assert BSY\ on the SCSI bus. Note that U8 must be a 74S-series part in order to properly drive the bus. The initiator then deasserts SEL\, completing the selection.

U5 decodes the REQ\ line into eight strobe signals according to the states of MSG\, C/D\, and I/O\. All of the even-numbered strobes (I/O\ asserted low) are writes, and all of the odd strobes are reads. Strobes 7 and 6 are the primary read and write strobes. Strobe 4 writes the address into octal flip-flop U4. Strobes 3 and 2 can be used as high-byte strobes in 16-bit systems. Strobe 0 can be used to latch eight additional address lines.

Strobe 5 ORed with RST\ resets the BUSY flip-flop, removing the interface from the SCSI bus. The capacitor on the RST\ line prevents glitches

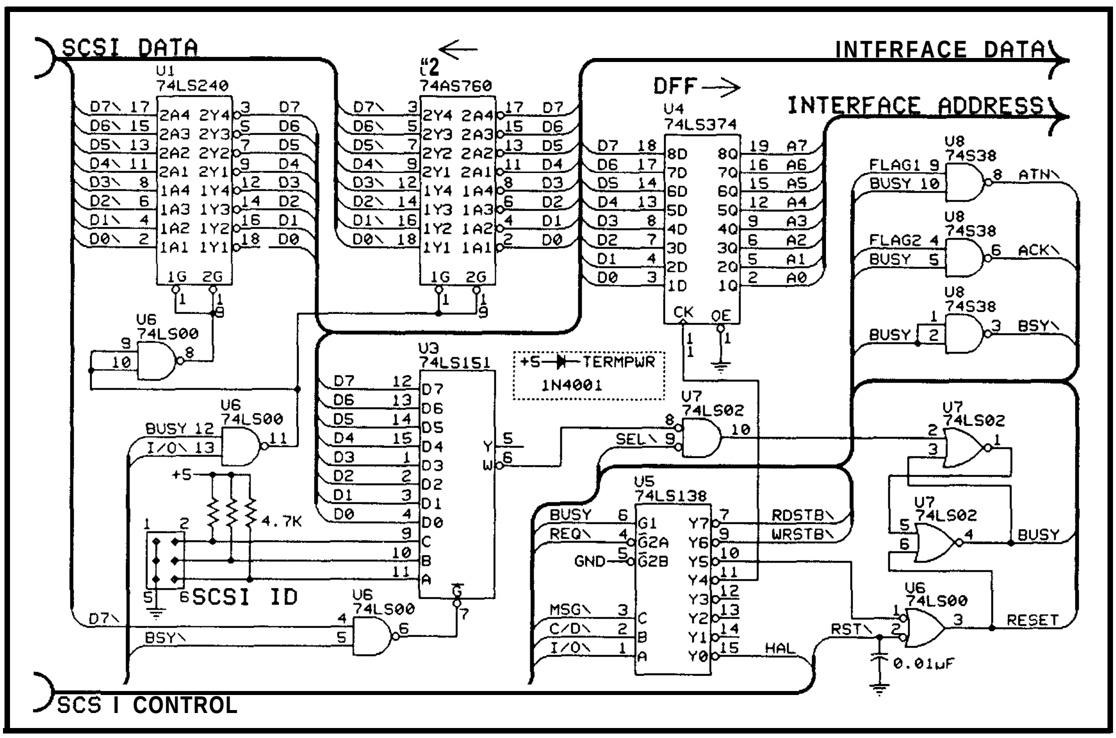


Figure 2—*This circuit interfaces SCSI to a generic bus with eight data lines and eight address lines (expandable to 16). The design trades simple hardware and custom software for the more intelligent hardware and 'standard' software of traditional SCSI.*

```
The following defines are the addresses of the internal registers
of the NCR 5380 SCSI controller in the Mac Plus, SE, and II.

#define   SCSIBase68000  0x580000
#define   SCSIBase68020  0x50F10000

#define   RdData         0x00
#define   WrData         0x01
#define   RdInitCmd      0x10
#define   WrInitCmd      0x11
#define   RdMode         0x20
#define   WrMode         0x21
#define   RdTargCmd      0x30
#define   WrTargCmd      0x31
#define   RdStatus       0x40
#define   RdBusStat      0x50

;Subroutine scuzzy_select selects the target specified by D1.
;Once this routine is called, you must: not perform any "normal"
;SCSI operations until you call scuzzy-deselect.
;On entry, D1 contains expanded target ID, e.g., if ID = 2,
; D1 = 00000100.
;A0 contains base address of 5380.
;On return, D0 contains error code:
;0 = no  error,  1 = SCSI bus occupied, 2 = target not responding

scuzzy_select:
    move.b    #0x0,WrTargCmd(A0)      ;Set bus phase to free.
    move.b    #0x40,WrMode(A0)        ;Set target mode.
    btst.b    #6, RdStatus(A0)        ;Bus should be free, BSY off
    beq       @busfree                ;Jump if bus free
    move.w    #1,D0                   ;Someone else is on bus, set
    bra       @restore_scsi           ; error code to 1 and return.

busfree:      ;Nobody will reselect, bus is all ours.
    move.b    #0x0,WrTargCmd(A0)      ;Set bus phase to free.
    move.b    #0x1,WrInitCmd(A0)      ;Assert data bus.
    move.b    D1, WrData(A0)          ;Put target ID on bus.
    move.b    #0x5,WrInitCmd(A0)      ;Assert SEL
    move.w    #10,D0                  ;Set up D0 as timeout counter

select_lp:
    btst.b    #6,RdStatus(A0)         ;Look for BSY asserted
    bne       @selected               ;Jump if it is
    dbra      D0,@select_lp           ;If not, loop up to 10 times

timeout:                              ;If timed out, set error code
    move.w    #2,D0                   ; to target not responding.
    bra       @restore_scsi           ;Return

selected:
    move.b    #0x0,WrInitCmd(A0)      ;Deassert SEL, data bus
    rts                               ;Return. Target on SCSI bus


;Subroutine write-address updates the interface address register
; with the 8-bit address passed in D0.
;On entry, A0 contains base address of 5380.

write_address:
    move.b    #0x3, WrTargCmd(A0)     ;I/O & C/D on, MSG off
    move.b    #0x1, WrInitCmd(A0)     ;Assert data bus
    move.b    D0,WrData(A0)           ;Address out to scsi bus
    move.b    #0xB, WrTargCmd (A0)    ;Assert REQ
    move.b    #0x3,WrTargCmd(A0)      ;Deassert REQ
    move.b    #0x0,WrInitCmd(A0)      ;Deassert data bus
    rts


;Subroutine  write-data writes the 8-bit data in D0 to the
; interface.
;On entry, A0 contains base address of 5380

write_data:
    move.b    #0x1, WrTargCmd(A0)     ;I/O on, C/D & MSG off
    move.b    #0x1,WrInitCmd(A0)      ;Assert data bus
    move.b    D0,WrData(A0)           ;Data out to scsi bus
```

(continued)

Listing 1 *—The five assembly routines shown here are the special drivers required to compensate for the simple hardware of this interface. While written for the 68000 in the Apple Macintosh, the software should be easily transported to any system using a 5380 for the SCSI interface.*

```
        move.b    #0x9,WrTargCmd(A0)      ;Assert REQ
        move.b    #0x1,WrTargCmd(A0)      ;Deassert REQ
        move.b    #0x0,WrInitCmd(A0)      ;Deassert data bus
        rts

;Subroutine read data reads the B-bit data from the intexface
; to DO.
:On entry, A0 contains base address of 5380

read_data:
        move.b    #0x0,WrTargCmd(A0)      ;Deassert I/O, C/D, MSG
        move.b    #0x8,WrTargCmd(A0)      ;Assert REQ
        move.b    RdData(A0), DO          ;Data in from scsi bus
        move.b    #0x0,WrTargCmd(A0)      ;Deassert REQ
        rts


;Subroutine scuzzy-deselect removes the interface from the
; scsi bus,  and returns the bus to normal operation.
;On entry, A0 contains base address of 5380
;On return,  DO contains error code:
; 0 = no error, 3 = target stuck on bus.

scuzzy_deselect:
        move.b    #0,D0                   ;Set return error to 0
        move.b    #0x2,WrTargCmd(A0)      ;I/O & MSG off, C/D on
        move.b    #0xA,WrTargCmd(A0)      ;Assert REQ
        move.b    #0x2,WrTargCmd(A0)      ;Deassert REQ
        nop                               ;Give BSY time to
        nop                               ; get off the bus
        nop
        nop
        btst.b    #6,RdStatus(A0)         ;BSY still asserted?
        beq       @restore scsi           ;Jump if it is not
        move.b    #3,D0  -----            ;Target stuck on bus
 restore_scsi:
        move.b    #0x0,WrInitCmd(A0)      ;Restore 5380 registers
        move.b    #0x0,WrMode(A0)
        move.b    #0x7,WrTargCmd(A0)
        move.b    #0x0,WrData(A0)
        rts
```

Listing 1—continued

on RST\ (the bane of SCSI) from resetting the interface. To avoid bogging down RST\, do not use a higher value than 0.01 µF. If a power-on reset signal is available, it should also be ORed with RST\ and Strobe 5, so that the interface doesn't crash the SCSI bus when it is turned on.

I/O\, gated with BUSY, controls the bus direction by enabling and disabling U1 and U2. The 74LS240 is a good choice for bus receiver because of its hysteresis (note that the ALS240 doesn't have hysteresis.) As for drivers, keep in mind that you need to sink 48 mA, and that either open-collector or tristate drivers may be used. If you can't find any 74AS760s, try 74S240s.

Two application-specific user flags are available to serve as interrupt or ready lines. U8b and U8c gate the flags with BUSY, and drive ATN\ and ACK\ on the SCSI bus.

One possible modification to this circuit would be to pull U5 pin 3 high, buffer MSG\ through the remaining gate of U7, and use it as an address line. This comes in handy, for example, when interfacing to a 12-bit ADC with an 8-bit data bus and an address input line. By sending MSG\ to that address input, one can extract both bytes from the ADC by simply doing two data reads, one with MSG\ high, and one with MSG\ low, which is faster than updating the address latch with each read.

THE SOFTWARE

As mentioned earlier, this ludicrously simple hardware design comes at the expense of having to write a special driver to interface to it. However, even this isn't as bad as it sounds. Listing 1 shows the five assembly language routines: `scuzzy_select`, `write_address`, `write_data`, `read data`, and `scuzzy_deselect`. The subroutines were written for the Macintosh, but it should be easy to port them over to other machines which use a 5380 for the SCSI interface.

In order to keep things simple, we will make one more assumption about the SCSI bus: targets are not allowed to disconnect from the bus until they have completed their commands. This means that when the bus is free, the only device that will be arbitrating for it is the initiator, so arbitration is unnecessary. This saves a few lines of code in *scuzzy* `_select` and speeds up access to the interface. The current Macintosh SCSI driver does not support disconnection, so this shouldn't present a problem for most users. If it does, the user must add arbitration code before selecting the interface. The other subroutines can stay the same.

The drivers do not support the two application-specific flags. The flags can be monitored by reading the 5380's Bus and Status Register (RdBusStat). ACK\ is bit 0 and ATN\ is bit 1. In both cases, "1" means the bit is asserted.

SCSI TEST BOX

The easiest way to test SCSI hardware and software is with a SCSI test box, such as the one shown in Figure 3. This is simply a collection of debounced switches and LEDs, but you will be amazed at how handy it will be. It can be used to test both the interface hardware and the driver software, and with a bit of practice, you will find yourself whizzing through SCSI commands in a matter of seconds. A word of advice: don't skimp on the switches. Use high-quality toggle or rocker switches, or your fingers will wear out after the first dozen commands.

I hope that this article has helped to strip away some of the mystery surrounding SCSI. It is my fond wish that experimenters will soon be subjecting SCSI to the same kind of abuse they've heaped on the IBM PC's printer port. Once you've mastered the art of scuzzy interfacing, you will never again think of the Macintosh as a closed architecture. ✣

REFERENCES

Inside Macintosh Volume IV
Apple Computer, Inc.
20525 Mariani Avenue
Cupertino, CA 95014
(408) 996-1010

NCR 5380-53C80 SCSI Interface Chip Design Manual
NCR Microelectronics Division
1635 Aeroplaza Drive
Logic Product Marketing
Colorado Springs, CO 80916
(303) 596-5612

Small Computer System Interface (SCSI), ANSI X3.131-1986
American National Standards Institute
1430 Broadway
New York, NY 10018

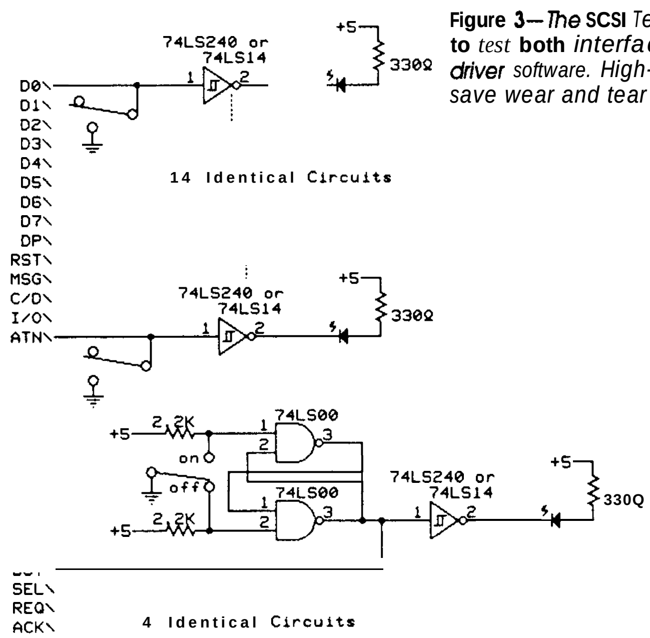


Figure 3—*The* **SCSI** *Test Box can be used* **to** *test* **both** *interface hardware and driver software. High-quality switches will save wear and tear on the user's fingers!*

Jim "Squint" MacArthur is a data acquisition engineer at Acoustic Technology, Boston, Mass. His nonprofessional interests include all forms of musical expression except indoor bagpipe music.

IRS

204 Very Useful
205 Moderately Useful
206 Not Useful

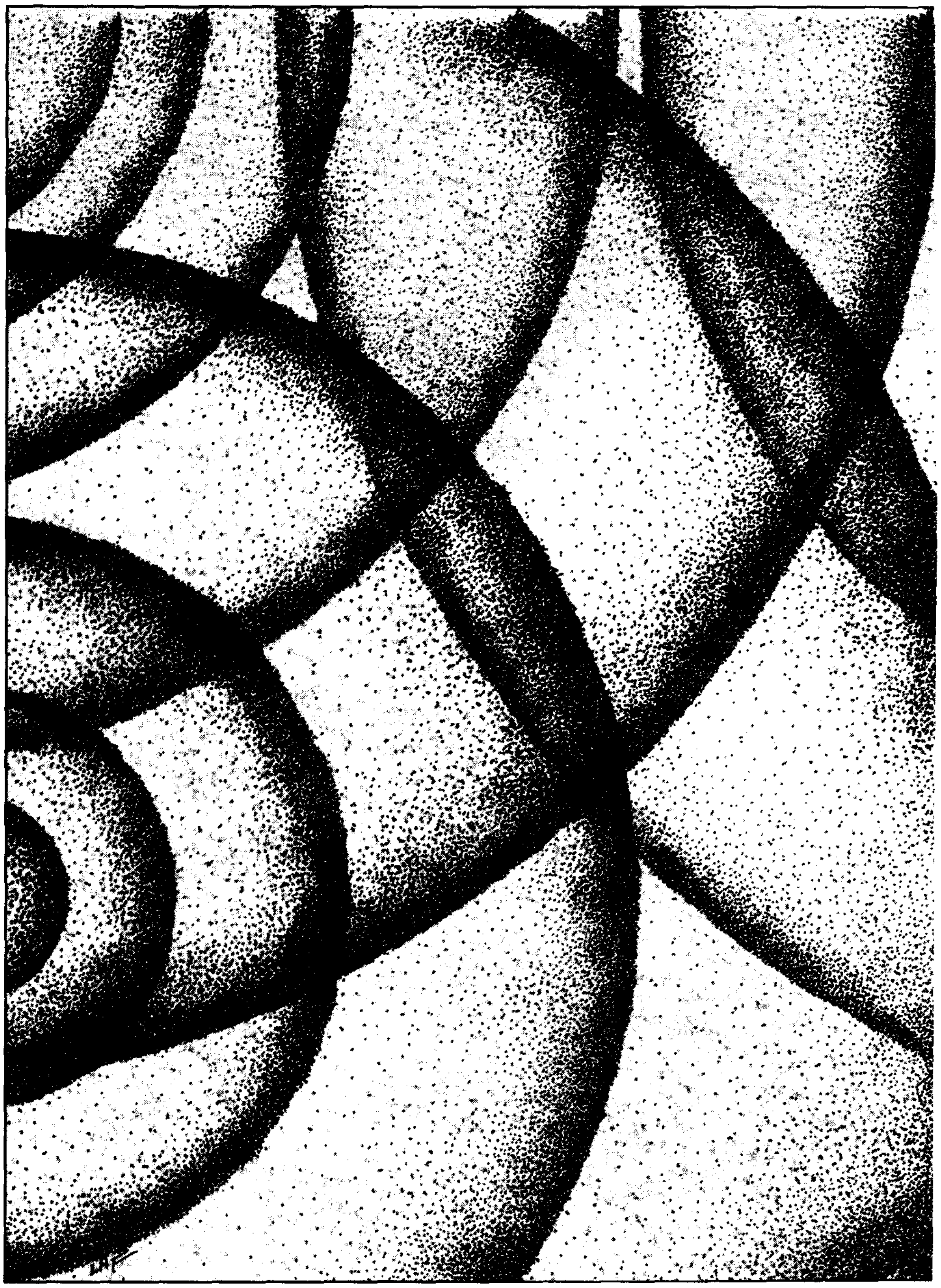
Computer-Generated Holographic Images

Using a PC to Generate Affordable Holograms

[Editor's Note: *This article is a practical tutorialonmethods togeneratea* hologram using an *MS-DOS* computer *with a VGA screen, a 35mm camera, and a HeNe laser (for viewing* the *hologram).*

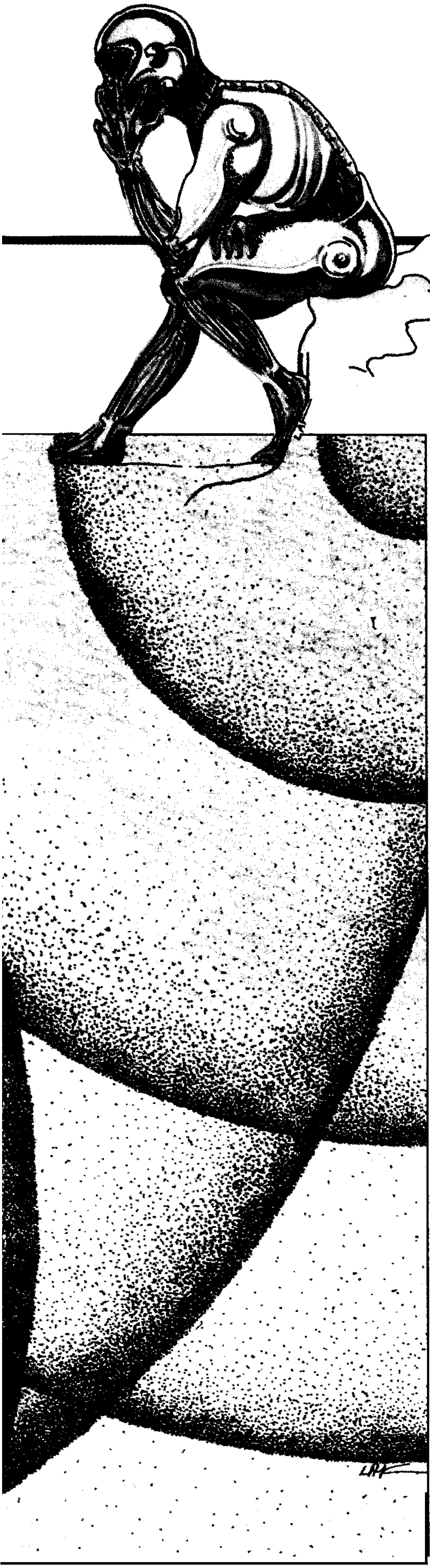
Dale Nassar has written a large manuscript covering laser basics, light *theo y, and the fundamentals of general holography. This article, while a practical tutorial, is an excerpt from the larger work.*

If you would like to purchase Dale's complete work, send $7.50 to: Computers and Holography, 4 Park St., Vernon, CT 06066.1



Holographyisaphotographic process which, unlike ordinary photography, does not record an image of the scene photographed, but encodes the emanating light rays themselves. The resulting optical record is called a hologram and can instantly reconstruct the recorded light rays. Holograms produce an illusion of the original scene in three-dimensional space that is remarkably life-like.

The beautiful images created by this unique recording process are made possible by the coherent light of the laser. However, because a hologram can be considered an array of many bits of information, I decided to investigate the practicality of computerized hologram synthesis. In this article I will demonstrate, without complex analysis, how holographic synthesis can be accomplished in the computer room with no special optical materials or holographic lab. The laser's role in conventional hologram formation is completely emulated by



FEATURE ARTICLE

Dale Nassar

computer, and in some very crucial situations the computer outperforms the laser.

The synthesizing method I use is straightforward and is designed to be easily understood and inexpensively applied with standard photographic and computer equipment. On a more advanced level, a parallel processing environment also lends itself to the application as the holographic bits are mutually independent.

THE SINUSOIDAL GRATING

A thorough understanding of the process of optical interference can easily be had by assuming light to be made up of sinusoidal waves of energy (hence the expression "light waves"). Figure 1 illustrates a sinusoidal waveform and the key elements of its structure as defined in physical optics.

It is important to be aware of the fact that light waves are traveling waves; that is, the contour of the waveform of Figure 1 should be considered as moving toward the right at the speed of light. To get a mental picture of what this means, consider a particle on the time axis in Figure 1 that is allowed to move only vertically in response to the amplitude of the passing light wave. Then the effect the wave has on the particle is a very rapid sinusoidal vertical motion (vibration) about a fixed point on the horizontal axis. It is obvious that the frequency of a light wave is extremely high, visible light has a frequency of the order of 10^{14} Hz (100,000 GHz).

These important quantities arc related by the very simple (and obvious) expression $f=c/\lambda$, where f is the frequency in Hz, c is the velocity of the wave in m/s (3×10^8 m/s for light) and λ is the wavelength in meters. The period of the wave, T, is the reciprocal of the frequency. T represents the time required for one wavelength to pass a given point. Mathematically, the energy of a wave is a measure of i ts intensity, which is proportional to the square of itsamplitude. Thisencrgy is what does the work responsible for exposing photosensitive film.

When two plane waves meet at the surface of a film, as shown in a cross-sectional view in Figure 2a, the interference **pattern** recorded consists of a series of parallel line fringes (in the diagram the lines are perpendicular to the page). This is called a photographic grating and appears as in Photo 1. Figure 2b depicts the same situation but with a larger angle between theinterferingbeams. As illustrated, the effect of increasing the angle between the two beams causes the fringe spacing to become finer. Figure 3 is a graph of theamplitude transmission across the surface of the grating of Photo 1. There are no abrupt changes in the transmission-the variation is sinusoidal with the frequency of the waveform representing the spatial frequency of the grating.

In Fourier analysis it is shown that a wave with very sharply changing shape such as a square wave can be broken **down** into many sinusoidal components, while a sinusoidal wave is the purest formpossible. In the case of the abruptly changing amplitude transmission of the grating, the result

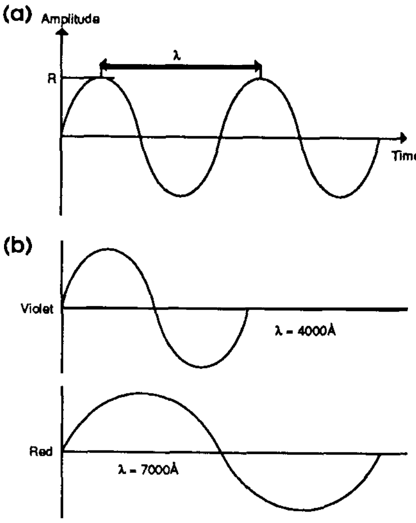


Figure 1—*Part (a)* **shows** *a* **basic waveform** *of light,* **while** *(b) shows relative* **wavelengths of the extremes of the** *visible spectrum.*

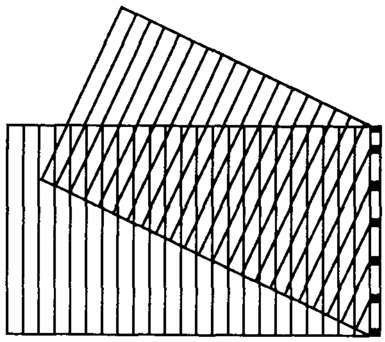


Figure 2a—*When two plane waves meet, an interference pattern consisting of a series* **of** *parallel line fringes is recorded.*

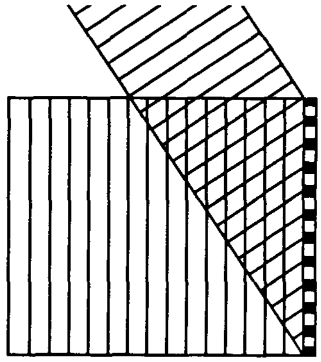


Figure 2b—*A larger angle between the* **Interfering** *beams causes the fringe spacing to be finer.*

is **many** orders of diffracted beams. Each diffraction order consists of two beamsdeflected at equal angles measured above and below the zero-order (straight through) beam. The angle of deflection θ of the diffracted beam is calculated from the standard grating equation:

$$D = \frac{\lambda}{\sin(\theta)} \qquad (1)$$

where D is the fringe spacing and λ is the wavelengthinvolved. On the other hand, the sinusoidal grating produces only one diffraction order.

THE ZONE PLATE: HOLOGRAM OF A POINT

From a holographic point of view, an object consists of many tiny surface points or resolution elements. When light is reflected from such an object onto the film, each resolution element of the object can be treated as if it were a point source of light generating a coherent spherical wavefront. Figure 4a is is a hologram of a basic entity—a resolution element (smallest resolvable point) of the object. Let's define the axis of the system as the line passing through the object point and center of the film. Symmetry **exists** around this axis, and the microscopic pattern recorded on the film will have the form of concentric circles as shown in Photo **2.** Notice that the fringe spacing is relatively coarse at the center of the system, but becomes finer, approaching one wavelength, as the waves move radially outward from the center of the film. This pattern of alternately light and dark circular fringes is called a zone plate and is the general appearance of a hologram of a single point.

Figure 4b shows what happens when theprocessed filmisilluminated. The fringes diffract the light waves as if they were coming from the location of the point source, forming a virtual image of the point. A set of converging waves forming a real image of the point on the opposite side of the hologram is also formed. If this were the actual object wave used in the recording process in place of the diverging point source, exactly the same interference pattern would have resulted. The certain amount of error present in the system is desired to give the mathematical points physical dimension.

The first holograms were made in 1948 (12 years before the invention of the laser) by Dr. Dennis Gabor of the Imperial College of London with the light from a mercury arc lamp which had a coherence length of only about 0.1 mmand a bandwidth of about 1 Å, which is low coherence by the standards of the laser. Because of the poor sources of coherent light available at the time, these were on-axis type holograms and the object was restricted to two-dimensional transparencies with opaque lettering. These conditions greatly reduced the coherence requirement. The light was shined directly through the transparency onto the film. The light passing through the clear areas served as the reference

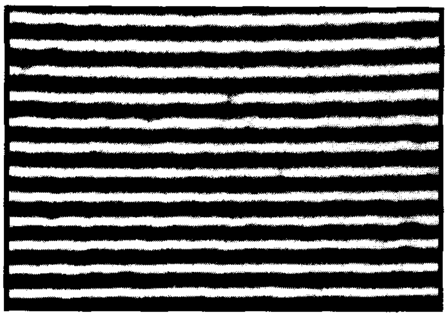


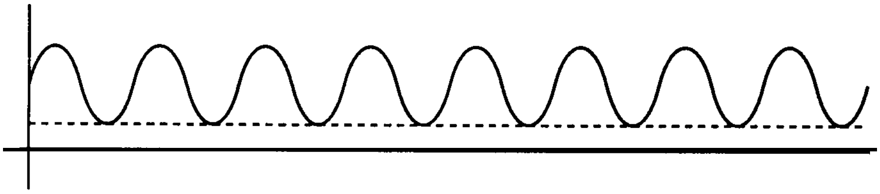
Photo 1 -When *recorded on film. the interference pattern shown above is called a photographic grating.*

beam and the light diffracted by the edges of the lettering served as the object beam. At this time the concept of off-axis holography was unknown. Around 1961 Emmett Leith and Juris Upatneiks of the University of Michigan, in an attempt to separate the real and virtual images of Gabor's hologram, made off-axis holograms with the gas laser. The discovery of holography, or wavefront reconstruction as the technique was called at the time, earned Gabor the Nobel prize in physics in 1971—he died in 1979.

THE FASCINATING FRESNEL

A Fresnel zone plate has a striking similarity to the interference pattern of the hologram of a single point. We use the properties associated with the Fresnel zone plate in many of the calculating procedures required to producecomputer-generated holograms. In deriving the structure of a Fresnel zone plate we make use of Huygen's principle which simply states (and can be proven) that each point on a wavefront may be regarded as a new source of secondary wavelets (of the

Figure 3—*The frequency of the amplitude transmission across the surface of the grating in Photo 1 represents the spatial frequency of the grating.*



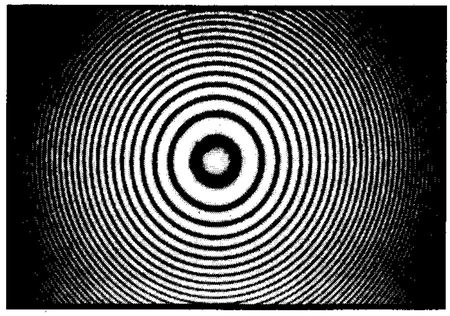
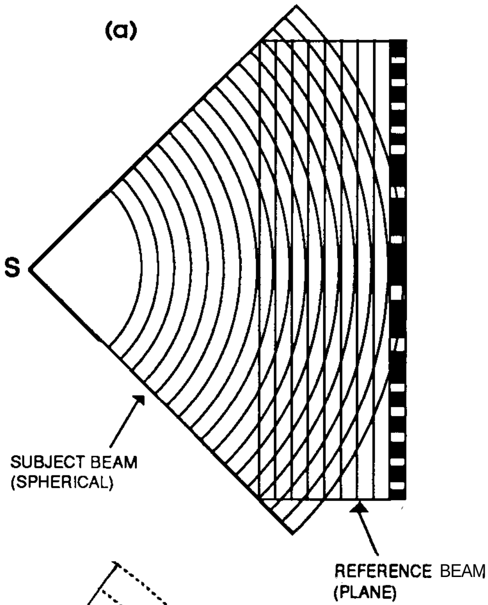
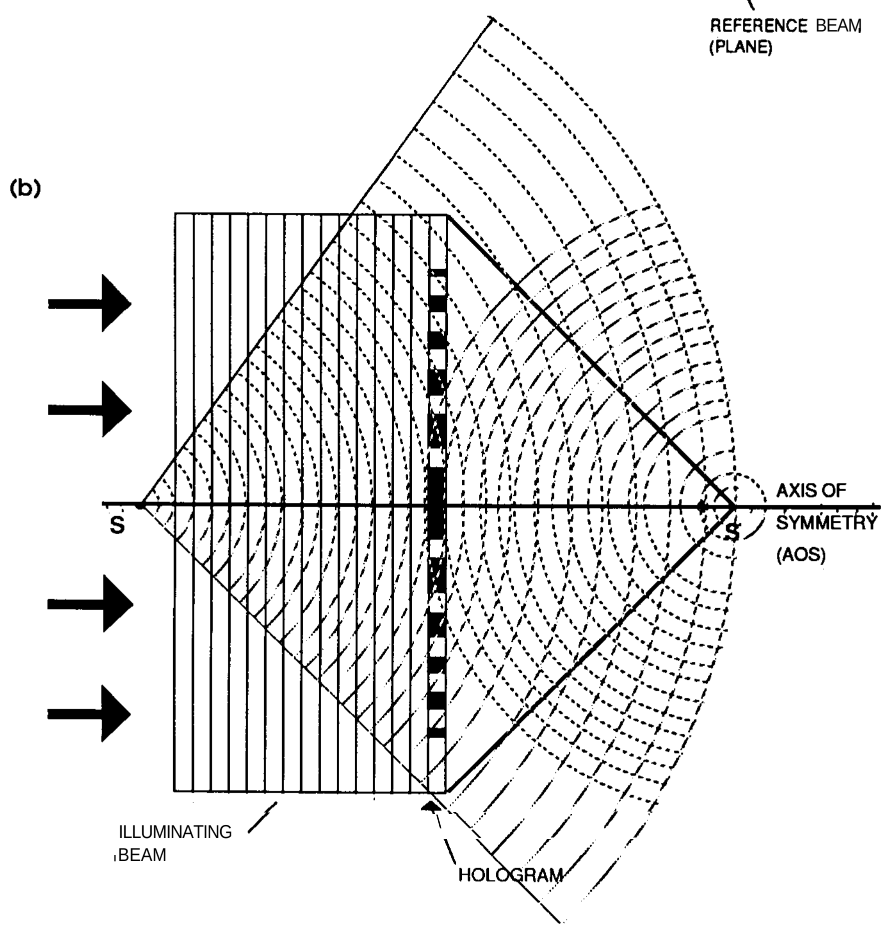


Photo 2-A *hologram of a point consists of concentric circles on the film.*

Figure *4-_(a) A hologram of a single point consists of alternately light and dark circular fringes and is called a zone plate. (b) When the processed hologram is illuminated, a virtual image of the original point is formed.*





same wavelength) and the interaction of these wavelets is responsible for interference effects observed. Figure 5a illustrates the principle. Here a plane wave illuminates an opaque screen with a pinhole in it. The pinhole acts as a new source of spherical waves as shown by the segments of circular arcs. The small circle represents a secondary wavelet of the spherical wavefront. The amplitude of this secondary wavelet is not the same in all directions but varies according to:

$$A = \frac{1}{2}(1 + \cos a) \qquad (2)$$

where A is amplitude and a is the angle at which the radiating amplitude is to be calculated. This equation is known as the obliquity factor. The obliquity factor has a maximum value of 1 which occurs when $a = 0$, corresponding to the direction of travel of the source. At 90 degrees the obliquity factor gives a value of 1/2 and at **180 degrees** the obliquity factor is zero indicating that, as shown in Figure 5b, there is no wave in the backward direction. Figure 5c is a polar graph of the amplitude and intensity distribution as predicted by the obliquity factor. It follows from Equation 2 that the intensity of the secondary wavelets is given by

$$I = A^2 = \frac{1}{4}(1 + \cos \alpha)^2$$

In this respect we can ignore the light source once the coherent wavefront isdefined at the diffracting aperture(s). The Fresnel zone plate is a patternof concentric transparent and opaqueringsdesigned tofocusabeam of plane wavefronts incident upon it much like a magnifying lens. To derive some very important properties of the Fresnel zone plate we will divide a plane wavefront into the various zones as shown in Figure 6. We can assume that the entire flat surface of **theplane** wavefrontconsistsof tiny point sources of light, each emitting spherical wavelets. Now consider some point *P,* illuminated only by light from the wavefront, located a **distance** *f* **past** the wavefront as illustrated. We now will divide a portion of the wavefront into zones such that there is a maximum concentration of light produced at point P. This is done by allowing only the portions of light emitted from the wavefront to reach *P* that would interfere constructively with any other light from the wavefront reaching P. Referring to Figure 6, consider the perpendicular from *P* to the plane wavefront. The intersection with the wavefront is denoted by 0. We now divide the plane wavefront into series of circles of radii *r1,* r2, r3,. . .,*rn,* centered at 0 such that each circle is a half wavelength further **from** *P* than the preceding one. We now can see that the circles, beginning with the innermost circle, are at distances

$$(f+\frac{\lambda}{2}),\ (f+\frac{2\lambda}{2}),\ldots(f+\frac{n\lambda}{2})$$

from *P*. The phases at *P* of any secondary **wavelets** from any given circular zone will not differ by more than **180°** (one half wavelength). If we go from one zone to the next, the amplitude of the wave reaching *P* changes sign. Therefore, if we block every other zone, only constructive interference will occur at *P* (considering only the portion of the wavefront encountering the zone plate). The result of thisconstructionist Fresnel zone plate. It does not matter if we start by blackening the center zone or **not, as long as the pattern alternates**—remember, intensity is proportional to the square of the amplitude.

This discussion should also suggest that the performance of a hologram is unaffected if its dark and light areas are interchanged. This is indeed the case-a hologram does not produce negatives. It is informative

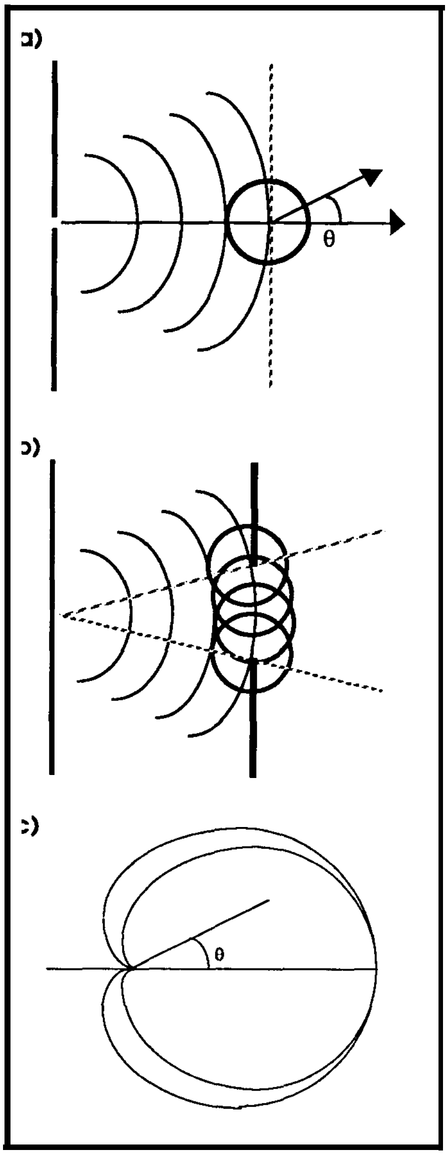


Figure 5—*(a) As a plane wave illuminates an opaque screen with a pinhole, the pinhole acts us a new source of spherical waves. (b) At 180°, the obliquity factor is zero. indicating that there is no wave in the backward direction. (c) The amplitude and intensity distribution as predicted by the obliquity factor is plotted on a polar graph.*

to look at some actual magnitudes involved with the zone plate. For only the first zone, it can be shown that the intensity at *P* is increased four times. This is rather surprising for the case of an opaque center zone since this implies that there should be a bright spot in the center of the shadow cast by an opaque circular obstacle. This is indeed the case and the placement of such a single disk increases the intensity at *P* by four. For a zone plate with only 20 zones, the intensity at *P* is increased by **400!** Looking at the **diagram** of Figure 7, we see that a series of right triangles are formed. For the triangle formed by the Nth radius, we have, from the Pythagorean theorem:

$$\left(f + \frac{n\lambda}{2}\right)^2 = f^2 + r_n^2$$

$$f^2 + nf\lambda + \frac{n^2\lambda^2}{2} = f^2 + r_n^2$$

Eliminating λ^2 since this value is negligible for light:

$$r_n^2 = nfh \qquad (3)$$

We now have a simple formula for constructing a Fresnel zone plate with desired properties. For example, if the above-mentioned zone plate of 20 zones were to focus the light from a He-Ne laser at 10 cm, the entire zone plate would have a radius of only about 1 mm. If we solve the zone plate equation for f, the primary focallength, a very convenient and useful formula is obtained:

$$f = \frac{r_n^2}{n\lambda} = \frac{r_1^2}{\lambda} \qquad (4)$$

We will make much use of this relationship. Consideragainthepoint *P* when the plane wave is encountering a single opaque zone of radius *r1*. As previously stated, the intensity is four times as great as that which results from the plane wave alone. Now if the radius of the opaque disk is expanded to cover the first two Fresnel zones, theintensityat Pdrops toalmost zero. If we continue this process of increasing the radius of the opaque disk, the intensity at *P* goes through a series of maxima and minima as the number of zones included according to the formula becomes even or odd. The result is the same if an open aperture is used allowing only increasing circular areas of the plane wavefront to emerge while blocking the rest.

Because of the abrupt changes between transparency and opacity in the Fresnel zone plate, there will be regions of secondary concentrations of light. A series of secondary foci along the axis between the primary focal point and the zone plate are readily observed if a white card is moved along the area. These foci are fainter

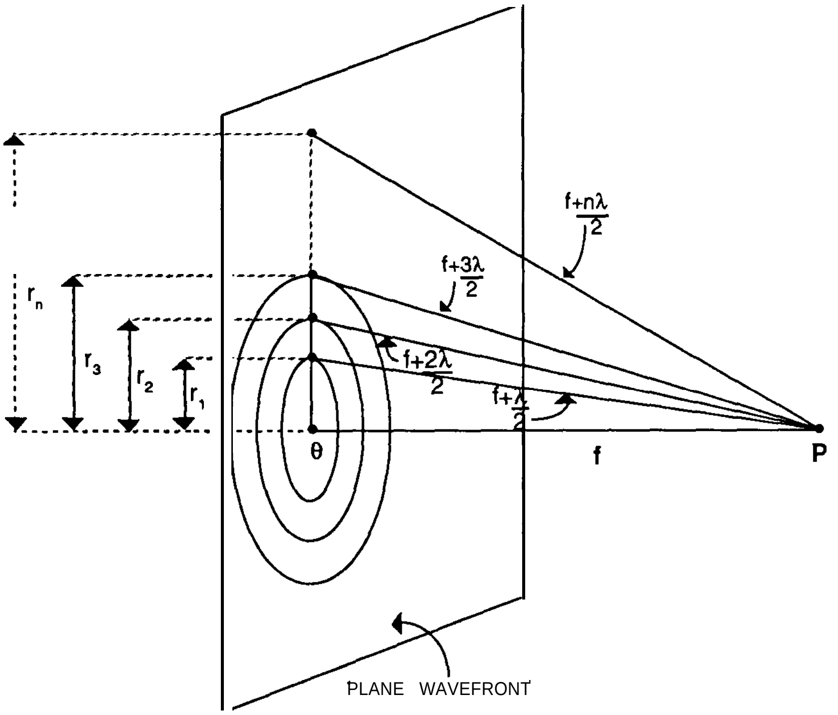


Figure 6—*The Fresnelzone plate is a pattern of concentric transparent and opaque rings designed to focus a beam ofplane wavefronts incident upon it much likea magnifying lens.*

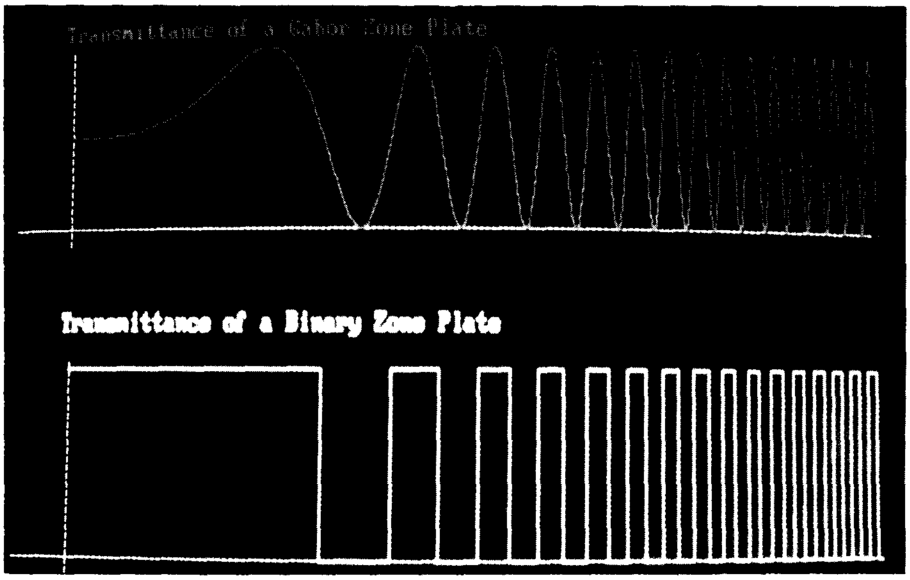


Photo 3—*The amplitude transmissions of a Gaborzone plate (a) and a Fresnelzone plate (b) are shown here.*

than the point at *P* and progressively diminish as the zone plate is approached. They are found at distances f/3, f/5, f/7,... If you give the above discussion a little extra consideration you will realize that these secondary foci are produced by single zones acting in groups of 3, 5, 7,...

There also exist various light concentrations at points off the axis of the Fresnel zone plate. The mathematical analysis of these off-axis maxima and minima are very complex, the results of which verify the presence of concentric circular fringes centered on the axis. These secondary foci do not occur in the holographic zone plate. The sinusoidal variations in the opacity of fringes causes cancellation of these higher order diffractions (by superposition of the secondary wavelets). Photos 3a and 3b illustrate respectively the amplitude transmission of a Gabor zone plate and a Fresnel zone plate. A sinusoidal zone plate is also called a Gabor zone plate. It should be interesting to compute the concentrations of light produced by a Gabor zone plate by using as parameters the secondary wavelets, obliquity factor, and sinusoidal transmission of the film.

To make a distinction between diffraction and interference, diffraction refers to the situation whena very large number of tiny wavelets of a wavefront, such as the Huygen secondary wavelets, are summed (integrated) to produce a the pattern while, interference refers to the interaction (simple addition) of a smaller number of beams. Briefly, the hologram interference pattern will be calculated by summing all of the sinusoidal waves emitted from each point of the object and calculating the resultant phase and amplitude at the hologram surface and then assigning either transparency or opacity at that point. This summation is done for each point.

PRELIMINARY CONSIDERATIONS

The procedure used to produce the computer-generated holograms willconsistofthefollowingthreesteps:

1) An optical interference pattern of a mathematically represented scene is computer calculated by digital approximation. This interference pattern is of the type produced by an off-axis holographic recording process.

2) This pattern is then drawn on a computeroutputdevicesuchasaCRT screen or plotter producing a monochromatic output.

3) The pattern is then reduced photographically thus becoming a transmission hologram designed to be viewed with laser light.

Consideration of the standard recording process of an off-axis transmission hologram reveals some difficulties that will be encountered in reproducing the process by artificial means. We have seen that this procedure produces an extremely fine interference pattern. Specifically, Equation 1 implies that, when the angle between the two recording beams approaches 60°, the fringe spacing approaches that of the wavelength of the light involved, which for a He-Ne laser corresponds to about 1600 line-pairs / mm. Such large angles are required because it is desirable for the off-axis scene to be near the film, thus permitting a large angular viewing range. This necessitates an extremely high resolution film such as the Kodak 649F Spectroscopic emulsion which is capable of resolving a maximum of 7000 line-pairs / mm. A 4- x 5-inch hologram with a maximumdeflection angle of 60” will be required to record about 132 billion dots.

The hologram also records a gray scale. If 64 levels of gray are assumed, the effective data content exceeds 8 trillion (8,000,000,000,000). (These calculationsareveryconservative. To prevent aliasing errors, the resolution in each direction should be at least doubled, and preferably quadrupled.) The size of the hologram is significant because the observer moves his head during viewing to exploit parallax. The dimensions of the hologram should thus be considerably larger than the separation of one’s eyes.

Now consider the time required to numerically calculate the data content for a hologram of a typical (small) object consisting of 1 million resolution elements. This means that there would be 1,000,000 calculations required foreachof the 132billionpoints of the hologram. Although the laser would produce this data instantly on a photographic emulsion (the highest information storage material known), the process would take a computer, working at a rate of one million calculations per second, over 4000 years!

The calculation time of the computer-generated holograms will be decreased by reduction of the following four parameters:

Hologram size-The holograms will be no larger than the standard frame size of the popular 35mm film (36 mm x 24 mm).

Quantify of resolution elements of subject-The subject will be a simple geometrical shape such as a circle consisting of only a few pixels.

Angles between object and reference beam—The maximum angle here will be minimized for a given fringe spacing of the hologram.

Gray scale—There will be no gray scale. The hologram will consist of only transparent or opaque areas. There is a very mysterious and little known property of holograms that is of great significance in this application: The gray scale of the subject is independent of the gray scale of the hologram. One may deduce that if the hologram is of binary form, then the reconstructed image must also be binary in nature. This is not the case. The reconstructed image may have a continuous gray scale regardless of the binary nature of the film. Any level of brightness that is assigned to any pixel in the recording process is stored in its relative proportion in the wave summation over the entire hologram area.

HARDWARE CONSIDERATIONS

Holographic patterns will be drawn using the following three types of output devices:

A standard VGA display of 640 x 480 dot resolution and hologram resolution (640 x 427).

A pen plotter with an effective plot area of 864 mm x 546 mm and 0.1-mm resolution with a 0.3-mm tip diameter pen giving a hologram resolution of 2880 x 1820 (819 mm x 546 mm).

A multihead laser plotter with an effective plot area of 24" x 18" and 0.5-mil (0.0005") resolution giving a hologram resolution of 48000 x 32000 (24" x 16").

The effective plot area of each device is shown in parentheses in order to obtain a width-to-height ratio of that equal to the standard 35mm film frame (3:2). This clipping represents a significant time savings when the reduced pattern is to be of maximum size (36 mm x 24 mm).

I used technical pan film to photograph the holographic interference patterns since this film is readily available and can resolve up to 400 line-pairs/mm at various contrast levels. This film is also ideal for applications involving a He-Ne laser as it has a high sensitivity to light in the red portion of the spectrum. Ektagraphic HC slide film might also be used because of its 750 line-pair/mm resolution.

REDUCTION METHOD

Because we are creating a hologram artificially by imitation of the actual process on a large scale and then reducing it for illumination with the light from a He-Ne laser, we must consider what effect reduction of the pattern as well as reduction of the recording wavelength has on the final image. Keeping in mind that in the plotting process, a large wavelength (imaginary, of course) can be associated with the recording pattern, we will look at the effect, on the reconstructed image, of reducing the hologram pattern as well as the reconstructing wavelength.

To simplify matters for illustration, suppose that the object consists of a single point *A* as illustrated in Figure 7. We know the values of the focal length (f) and of the wavelength of the imaginary waves (λ_i) forming the hologram since we assign these values (of course, this point is never actually produced optically since the hologram was created on a large scale by artificial means and no light of such a large wavelength exists). Given the imaginary wavelength and the distance to the focal point, we have from Equation 4 the size of the hologram (r_1 will suffice). Now suppose it is desired to reduce the hologram by a factor of four with no distortion of the image point. When a single point is involved distortion would be a position of the point relative to the hologram that was not in the same ratio as the reduction. If the hologram is reduced by four, then the point should become four times closer to the hologram which is now four times smaller on each edge (the area reduction is 16) if no distortion is present. If the hologram is reduced by four and the focal point is calculated with the original larger wavelength, then the focal length becomes too short and, generally, the reproduced image is distorted.

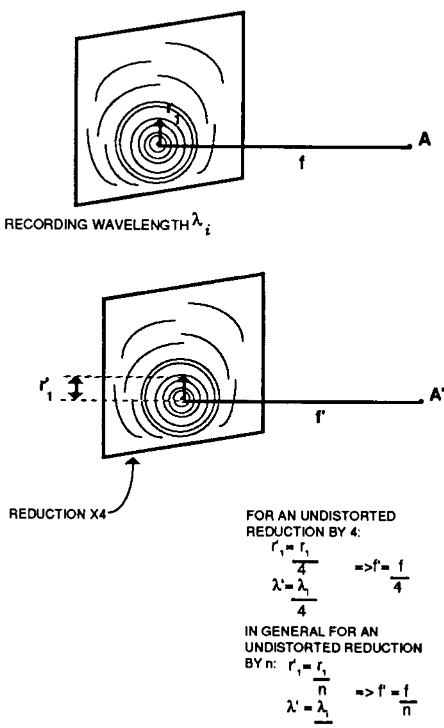


Figure 7—*To reduce a hologram pattern and preserve a distortion-free image, the reduction of the illuminating wavelength should be equal to the reduction of the hologram pattern.*

Now if we reduce the wavelength by four and illuminate the reduced hologram, no distortion is present. Therefore, to reduce a hologram pattern and preserve a distortion-free image, the reduction of the illuminating wavelength should be equal to the reduction of the hologram pattern.

SAMPLING THE DIGITAL HOLOGRAPHIC PATTERN

To illustrate how sampling is to be done, consider the pen plotter with 0.1-mm resolution and a 0.3-mm pen tip. The maximum spatial frequency (line-pairs/mm) that can be plotted here is 1.67 line-pairs/mm. This is effectively the maximum sampling rate that the pen plotter can perform on the hologram pattern. In order to prevent aliasing in the full-scale hologram pattern, all of its spatial frequency components should be larger than 0.83 line-pairs/mm (half the maximum spatial frequency of the plot). We are assuming a photographic reduction of 22.75 (in the case of a 35mm camera) which will result in a holographic resolving power of 18.88 line-pairs/mm. The effective plot area of the paper is 0.864 m x 0.546 m, and if the larger side is to be oriented horizontally, we have, from Equation 1, a maximum diffraction angle of 0.685°.

Referring to Figure 8, if a hologram is to be formed without exceeding the maximum spatial frequency, then all object points must be confined to the shaded region in the illustration (the image points are actually confined to a cone-shaped volume defined by the solid of revolution of the hologram and the two angled lines about the axis of symmetry). In the illustration, I used an antialiasing factor of 2.5 instead of 2 (which is only the lower limit) to allow for error. Any points located outside of this area such that larger angles are produced between the interfering beams will produce errors in the pattern. Photo 4a shows a zone plate calculated point-by-point (I call this a digital zone plate) and plotted in a raster type-motion of the pen (not by drawing concentric circles). The resolution of the pattern exceeded that of the plotter toward

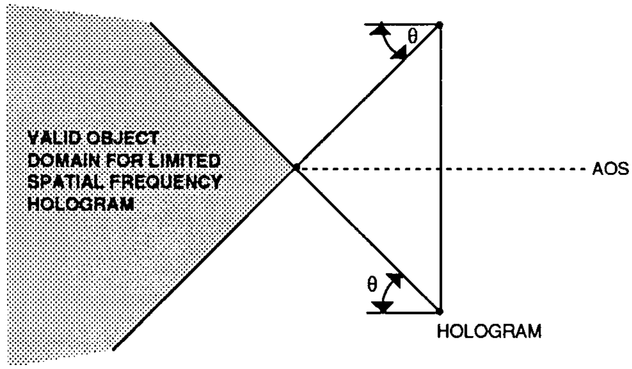


Figure 8—*If* **a hologram is to** *be* **formed without** *exceeding the maximum spatial frequency, then all object* **points** *must be confined to the shaded region.* ***Any points*** *located outside this area such* **that** *larger angles are produced* **between** *the interfering beams will produce errors in the pattern.*

the edges of the plot. Notice how the jagged edges arrange themselves to produce several families of secondary zone plates. When this pattern is illuminated with laser light, each of these secondary zone plates has a focusing effect, and the error emerges as a matrix of concentrations of light (secondary foci) about the primary center focal point as shown in Photo 4b. Photos 4c-4e illustrate zone plate formation on a VGA screen with antialiasing factors of 1, 2, and 3. As can be seen, the aliasing error decreases (less secondary zone plate contrast) as the antialiasing factor increases. From this data, I decided to use an antialiasing factor of 2 for the computer-generated holograms.

SYNTHESIZING A HOLOGRAM

Let's start our example by calculating the interference pattern for a hologram of a computer-generated curve (a three-leaved polar rose). Here, the situation for the VGA display (640 dots horizontally by 480 dots vertically) is used. These displays are usually about 10" x 7.5". Because the width/height ratio is lightly different than that of a 35mm camera, when photographing with a 36-mm x 24-mm viewfinder, simply fill the area vertically. Ideally there will be a small vertical strip along one edge of a full-scale (36 mm x 24 mm) hologram, but all pixels will be used. The spatial frequency of a VGA display is about 1.26 line-pairs/mm. We will divide this value by 2.5 to prevent error: 1.26/2.5 = 0.504 line-pairs/mm. I will conservatively round this value to 0.5 in actual calculations.

The hologram plane is defined in a three-dimensional Cartesian coordinate system as illustrated in Figure 9. The origin of the system has coordinates (x,y,z)=(0,0,0), with the coordinate signs assigned as implied by the drawing. The hologram plane coincides with the xy plane with its top edge on the +x axis and upper-left corner at the origin. Note that +x is to the right and +y isdownward to match the native coordinate system of the computer display. The +z direction is

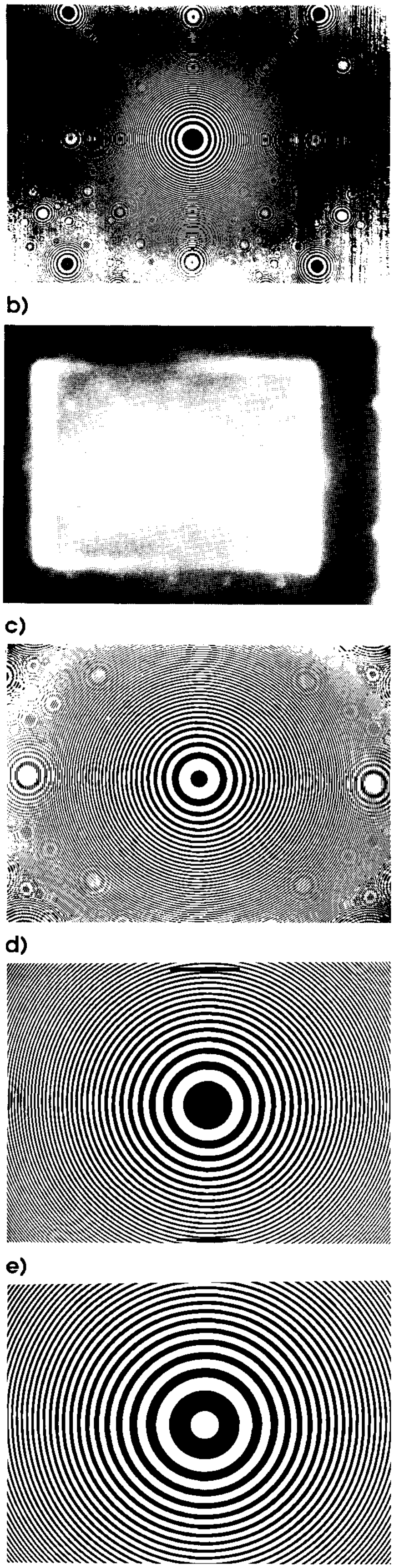


Photo 4—*(a) A typical zone plate. (b) The maxtrix of dots are errors caused by secondary zone plates. (c)-(e) Zone plates with antialiasing factors of 1, 2, and 3.*

toward the rear looking through the hologram plane. The coordinates of the lower right corner of the hologram extent have VGA coordinates (639,479). The screen size photographed will be 10" (0.254 m) by 7.5" (0.192 m). The position, in three-dimensional space, of the lower-right corner of the hologram is located at (0.284,0.192,0). Let's give the rose a radius of 0.1 m and let it consist of about 41 (I incremented the full polar rotation of pi by0.075) equally spaced pixels of equal intensity (more on intensity assignments shortly).

We will now consider how close the reconstructed image can form from the hologram plane. If the entire screen of 0.254 m x 0.192 m is to be reduced as to just fill a 36-mmx 24-mm film frame then the reduction factor is 8, as shown in Figure 10a. This will result in the 0.500 line-pairs/mm of the plot to become about 4 line-pairs/mm in the final hologram. We also know that the synthetic wavelength should be eight times that of the reconstruction (helium-neon laser) wavelength, or 5.0624 x 10^{-6} m. Equation 4 tells us that the maximum angle that the hologram can deflect the reconstruction beam is about 0.146". This corresponds to a minimum object distance (on the plot scale) of about 63 m if the maximum hologram radius is, using the diagonal, 6.25". Notice that an additional 39.52 m must be added to the minimum object distance if the entire 0.1-m rose is to be recorded. Since it would be desirable for the image to be somewhat closer to the hologram, let's look at a reduction twice as great.

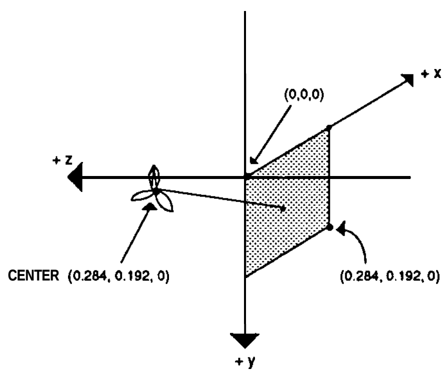


Figure 9—*The hologram plane is defined in a three-dimensional Cartesian coordinate system and is shown as the shaded region.*

A reduction of 16 (Figure 10b) corresponds to a synthetic wavelength of 1.0125 x 10^{-5} m and a maximum deflection angle of 0.29". This brings the minimum large-scale image distance down to 31.5 m. However, we now have only one quarter of the hologram area as before (when photographed, the image seen through the camera's viewfinder should occupy one quarter of the area). After reviewing the parameters for reductions by 24 and 32, I decided to use 16.

When writing the graphics routines, it would be much more convenient to work in units of pixels rather than meters-since 0.254 m corresponds to 640 pixels (for the VGA), we have the relationship 2520 pixels/m. Now we can simply multiply any meter value by 2520 to work directly in graphics mode.

The polar equation of the three-leaved rose is:

$$r = a*\cos(3*t)$$

where r is the dependent variable, f is the independent variable, and a is the radius of the rose. The transformation from polar to Cartesian coordinates is accomplished with the following equations:

$$x(t) = h+r*\cos(t)$$
$$y(t) = k+r*\sin(t)$$

where (h,k) is the center of the rose.

We could tilt the figure out of the xy plane by adding a sinusoidal function *z(t)*. However, because in this example the hologram is relatively course and the object is small and distant from the hologram plane, this tilt would not be very noticeable in the reconstruction.

Listing 1 is a simple QuickBASIC program that will allow the user to define and edit a holographic image before it is processed by the program of Listing 2, which draws the ready-to-photograph holographic interference pattern on the high-resolution graphics screen. Remember that more points in the image means longer drawing time. With a math coprocessor in a 286 machine running at 12 MHz, the rose plot takes about 12 hours to complete. Without the coprocessor, it takes several days.

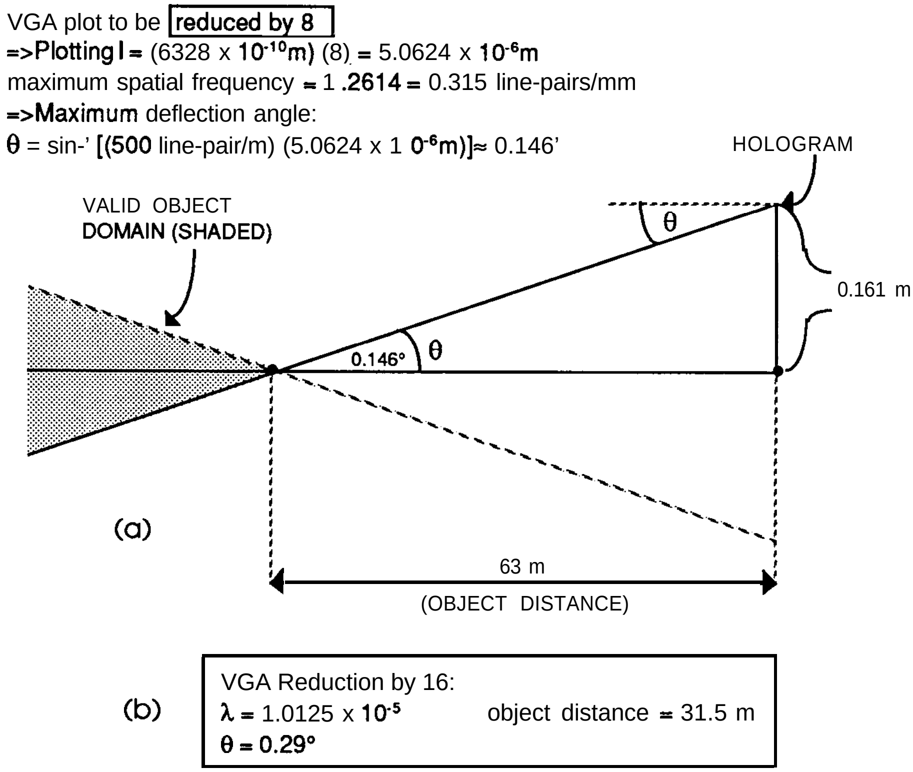


Figure 10—*The amount of reduction used when making the hologram affects how close the reconstructed image will be to the hologram plane. Part (a) shows a reduction factor of 8, while (b) shows a factor of 16.*

```
SCREEN 12: COLOR 4: pi = 3.1416: a = 12.5

 FOR t = 0 TO 1 * pi STEP 0.02
     r = a * COS(t * 3)   '** PLACE ANY FUNCTION HERE **
      x = 320 + r * COS(t): y = 240 + r * SIN(t)
     PSET (x, y)
 NEXT t

DO: LOOP WHILE INKEY$ = ""
```

Listing 1—*This QuickBASIC program allows the user to define and edit a holographic image before it is processed.*

In defining the interference pattern, each point of the rose is considered to be emitting light, thus illuminating the hologram plane with radiation of the synthetic wavelength. Each point emits light with a specified initial phase and reaches each point of the hologram plane with a specific phase. For each point in the hologram plane, the sum of the waves from each point of the circle is calculated and the point is assigned to be either transparent or opaque depending upon the result of the summation. There is also a phase value present at the hologram plane. For simplicity, I assigned a phase value of zero at the hologram plane and let all points on the rose start emission with a sine wave (0 initial phase angle and increasing amplitude) of unit amplitude. At the hologram plane opacity was assigned if the wave summation at that point was greater than or equal to zero and transparency otherwise. The holographer can choose any trigger level desired. Also, different points of an object can be assigned various amplitudes for a proportional intensity in the reconstructions.

Another consideration in the reconstruction of the image of a binary hologram is the formation of extraneous images due to higher order diffractions. From the experiments performed here, these higher order diffractions were so dim that they were unnoticeable. However, Equation 1

```
5 DEFDBL A-Z
10 SCREEN 12: pi = 3.141592653589793#: s = 0: CLS
20 l = 0.0255145: pz = 129528: h = 320: k = 240: a = 252
30 FOR x = 0 TO 639
40 FOR y = 0 TO 479
50 FOR t = 0 TO 1 * pi STEP 0.075
55 r = a * COS(3 * t)
60 px = h + r * COS(t): py = k + r * SIN(t)
70 d = (((px - x) ^ 2) + ((py - y) ^ 2) + ((pz) ^ 2)) ^ 0.5
80 phase = (2 * pi / l) * d
90 s = s + SIN(phase)
100 NEXT t
110 IF s >= 0 THEN COLOR 7: PSET (x, y): GOTO 130
120 COLOR 0: PSET (x, y)
130 s = 0
135 NEXT y
140 NEXT x
150 DO: LOOP WHILE INKEY$ = ""
200 END
```

Listing 2—*This QuickBASIC program draws the ready-to-photograph holographic interference pattern on the high-resolution graphics screen.*

may be used to calculate the diffraction angles of the second-order images in the reconstruction to determine an object size (or fringe spacing) to ensure distortion-free images.

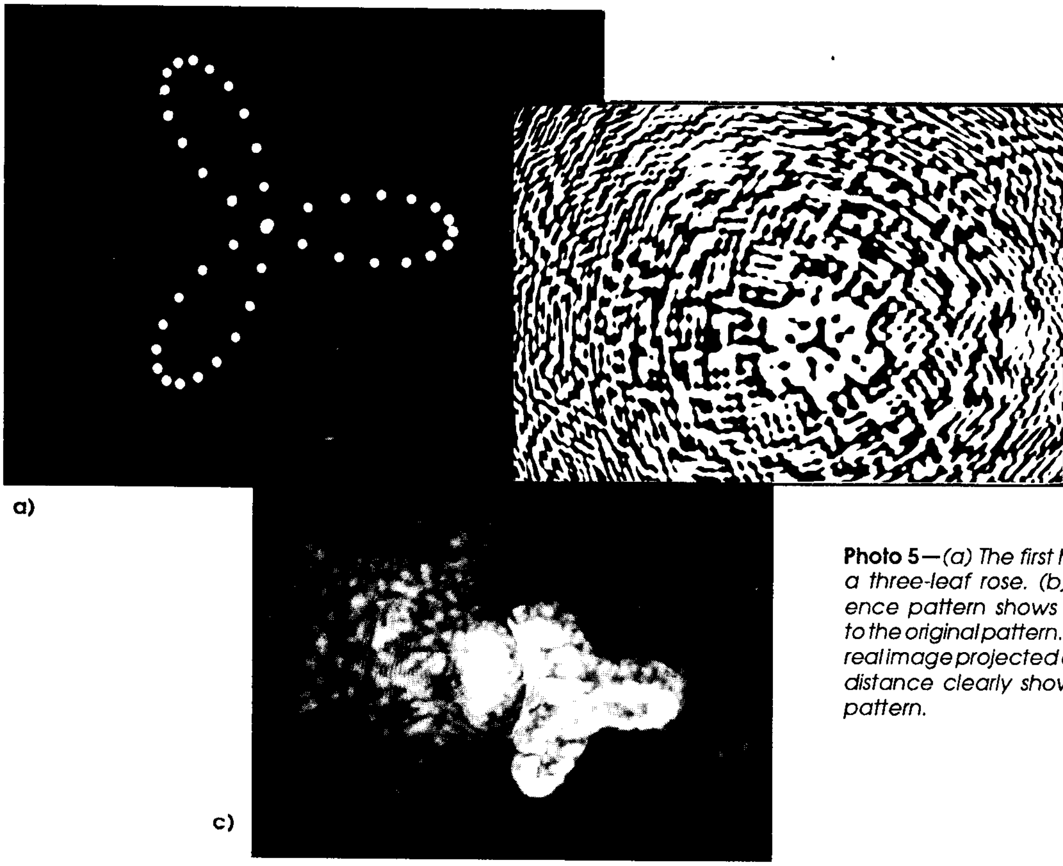
VGA-RESOLUTION HOLOGRAMS AND RECONSTRUCTIONS

Photo 5a shows the subject of the first hologram and Photo 5b is a photograph of the actual interference pattern. Photo 5c illustrates the appearance of the reconstructed real image projected at the predicted focal distance.

Observation of the pattern of the on-axis hologram has a somewhat fuzzy outline of the rose, and one may think that the image is formed by light rays passing straight through the film-this is not the case as can be easily shown in several ways. First, the pattern is not sharp and can't produce the observed bright points of light by projection. Secondly, if the image is viewed between the hologram and the focal point, only a blur appears.

For a more dramatic illustration, I cut the hologram into quarters and the full image was reproduced in each piece. To illustrate this further, I plotted a small offset segment of the hologram having none of the rose-shaped outline (Photo 6). When illuminated, this (now offset) hologram fully reproduces the rose as illustrated in Photo 5c. Also, the distant virtual image of the rose can be seen by looking through the hologram toward the illuminating laser. The hologram shows off-axis properties with less than one-third of VGA resolution!

Photo 6 can be photographed and used as a hologram. If this is done, the final image making up the fringes should be 0.32" x 0.25". It should be photographed with a light background which will become a dark outline for



a) b) c)

Photo 5—*(a) The first hologram is based on a three-leaf rose. (b) The actual interference pattern shows vague resemblance to the original pattern. (c) The reconstructed real image projected at the predicted focal distance clearly shows the intended rose pattern.*

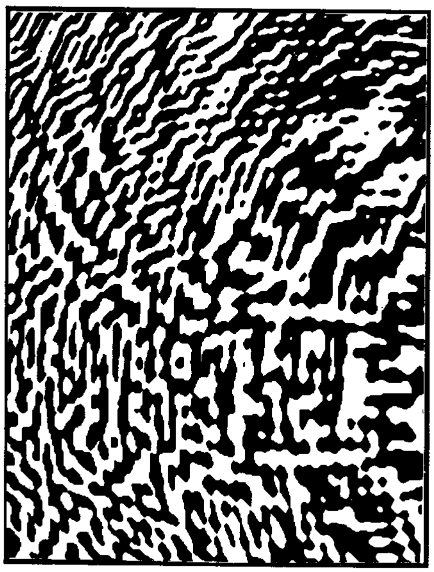
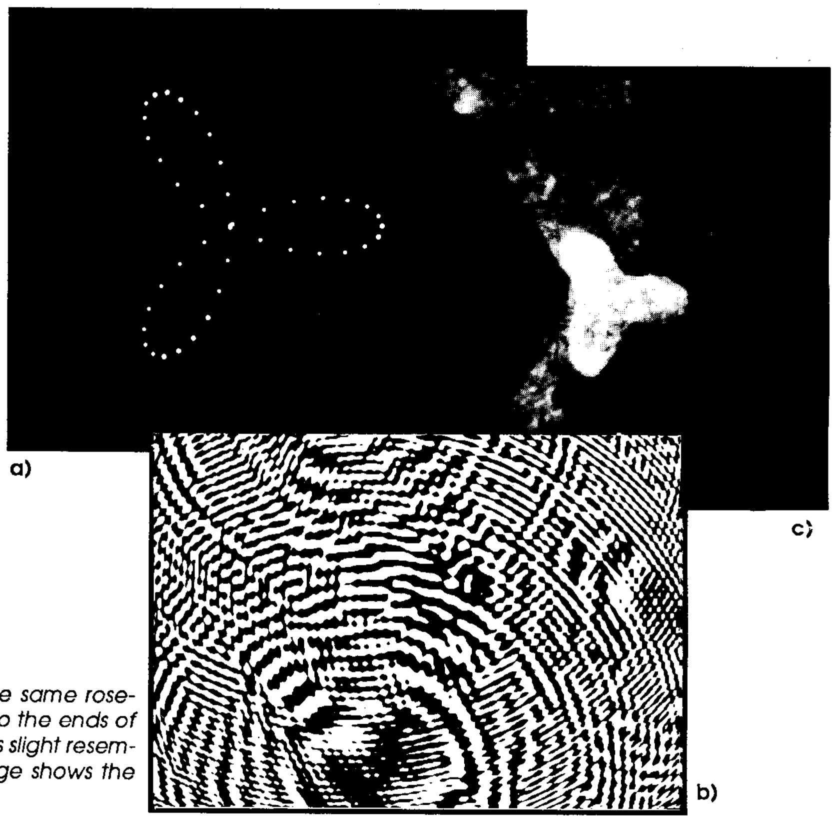


Photo 6—*Plotting a small offset of the interference pattern shown in Photo 5b eliminates any hint of the rose-shapped outline. When i//u-minuted, the projected real image is identical to that in Photo 5c.*



a) c) b)

Photo 7—*(a) The second holographic subject is the same rose-shapped pattern with larger amplitudes assigned to the ends of the petals. (b) The interference pattern again shows slight resemblance to the original. (c) The projected real image shows the expected bright regions on the ends of the petals.*

blocking extraneous laser light when viewing the hologram. It appears that any type of black-and-white film can be used for the VGA holograms.

I also plotted the negative of the pattern simply by reversing the pixels (turning blank pixels on and turning lighted pixels off). The negative produced results identical to those just illustrated.

For the next hologram, I assigned a higher brightness to a few of the pixels making up the rose. Pixels making up three small segments of the rose were assigned larger amplitudes (all others are normally unity). Photo 7a illustrates the rose with several brighter pixels. Photo 7b shows the resulting holographic pattern and Photo 7c is its reconstruction. The sets of two, three, and four bright pixels were assigned amplitudes of three, four, and five, respectively.

The experiments presented here show that in holography the requirementsof coherence, stability, and high-resolution recording media are not as strict as many people believe.

THE FUTURE

Holographic display devices may have a bright future. The nature of such a device will operate on drastically different principles than the CRT (pixels emit incoherent light), probablyusinga sort of supercomputer (parallel processor) to calculate an interference pattern in a reasonable span of time. There will alsobe a great change in the manner in which graphics images are created on these devices. Obviously, no image is drawn on any surface-only an interference pattern. The entire image will be present or not-nowhere in between. A feasible construction would be a display matrix consisting of liquid crystal "shutters" that could be toggled open or closed. The matrix would be illuminated by spread (no eye hazard) laser light from behind. The shutters need not cover the entire display surface since their purpose is only to direct light. Several cluster arrangements, each consisting of shutters of close spacing, would be suitable.

Full-color holographic displays can be produced by use of illuminating lasers of the primary colors. Color holograms are calculated by assigning each pixel three synthetic wavelengths, each corresponding to one of the primary colors. The waves would be of amplitude characteristic of the object point. The monochrome interference pattern is then capable of producing the scene in full color. ✣

I would like to give special thanks to the following people for their patience and assitance in this project: May Lou Nassar, Maria Palmer, Joe Lombardo, and Charles Palmer III.

Dale Nassar has a B.S. in physicsfrom Southeastern Louisiana University (SLU). His hobbies include gymnastics (university team) and springboard diving.

IRS

207 **Very Useful**
208 Moderately Useful
209 Not Useful

Modulating Laser Diodes

The *Search For the Perfect Driveway Sensor*

FEATURE ARTICLE

Steve Ciarcia

It all started when I bought one of those Radio Shack infrared doorway detectors and tried to use it across my driveway to tell when so proaches the house. It's not that I don't like surprises, it's just that any good home control system should be aware of its perimeter property as well. At least, that's what I tell people who see the array of sensors and devices lining the driveway. In actuality, most of it is now an electronic graveyard containing the remnants of many attempts to make a no-fault organic/inorganic driveway sensor. Let me explain.

The Circuit Cellar **driveway sensor** *farm includes IR motion, modulated and* unmodulated *interrupted beam. and pneumatic sensors.*

Unlike the average colonial or saltbox house you'd expect in New England, I live in **one** of those strangely shaped California contemporaries more befitting a reclusive personality. It affords spacious living and I don't have to see any neighbors. Unfortunately, I can't see my BOO-foot driveway either. When I go down in the Circuit Cellar and turn on the stereo I might as well be on another planet as far as anybody pounding on the front door is concerned.

A half dozen years ago circumstancesmademe reevaluate ignorance of above ground events. One time while I was buried (figuratively) in a project in The Cellar, a large truck pulled into the driveway and dumped 14,000 lbs of crushed stone. Later, another truck dumped 8 cubic yards of top soil. I just happened to go upstairs in time to see a third truck backing in with a load of landscaping timbers.

I practically had to throw my body across the hood of this truck to stop it from being dumped next to the other piles. Believe me, it was a real fight. From the driver's perspective I was in the wrong, of course; after all, there were two piles of stuff already on the driveway. This had to be the right place.

Shortly after that I decided the only way to avoid similar situations in the future was to apply the typical Ciarcia response: massive intervention of electronic countermeasures. To keep a closer eye on ground level and perimeter events, I installed a closed circuit video systemand put monitors in strategic locations. If I heard something or wanted to check on outside conditions without leaving my desk, I merely looked at a monitor and switched to an appropriate camera.

As a refinement to the system, I installed devices in the driveway and around the perimeter that triggered control events when they sensed people or cars coming down the driveway.

Without resorting to esoteric "military budget" solutions involving strain gauges under the pavement and hidden microphones with DSP "signature-detection" electronics in the bushes, or low-light-level CCD cameras with video pattern recognition, I decided to attempt a more economical perimeter intrusion detector (I am working on a video digitizer/ DSP analyzer "seeing" detector as the ultimate solution to this problem but that is a future project).

A relatively simple combination of infrared motion or beam sensor and a magnetic coil sensor mounted in the same physical location seemed to offer a quick solution. A large steel vehicle (inorganic) passing the sense point would trigger both the infrared and magnetic sensors (the magnetic sensor is a coil of wire under the driveway with the electronics of a classic metal detector; simple but effective). A person (organic) passing the sense point would trigger only the infrared sensor. Simple binary logic and you have both a people sensor and a vehicle sensor.

Being a pessimist, I expected to have a real problem building the underground metal detector. Fortunately, I found a off-the-shelf magnetic field "vehicle" sensor in a Sporty's catalog (Clermont Ave., Batavia, Ohio 45103, 800-543-8633) for $159 which I promptly installed. Since it worked like a charm with little modification (provided you don't drive in on a lawn mower), I focused my attention on the easy chore of making an infrared driveway sensor. After all, how difficult could it be? Go down to your local Radio Shack and buy one of those infrared door entry sensors and mount it across the driveway? Or, how about a motion detector like the ones that trigger outside lights?

TRUTH IN ADVERTISING

The first thing you should know about practically all low-cost commercial infrared sensors is that they're basically good for nothing when used outdoors. Spend a little time watching one of those IR motion detectors when it is raining and see how many times it triggers falsely. Or, drive a car that has been sitting at ambient temperature a few hours past a sensor while the engine is still cold (what car?). Of course, if the only repercussion is that the porch lights are left on for a couple extra minutes, we hardly notice. Even when the light stays on continuously we frequently don't care. (While I had some success with narrowly aimed IR motion detectors, I was never satisfied with the number of false-positive errors).

In a world of real control applications, however, if such a detector triggered several lighting control actions in the house or opened/closed the garage door, you'd notice false triggering and errors immediately.

Indoor-use interrupted-beam sensors hold a little more promise. With a little work they can be mounted in an environmentally conditioned enclosure (heated and cooled) and perfectly aimed at an appropriate reflector across a driveway. Unfortunately, in my experience, they lack the signal-to-noise discrimination to deal adequately with the dynamics of bright daylight and seasonal changes. Adding black tubular sun shields in front of the lens extended the useful operating period, but there were times when the ambient light completely swamped the reflected signal and there was no output (false-negative error>.

The *Circuit* **Cellar** *laser driveway sensor* **consists of** *an IR laser* **transmitter on the** *garage (right) and a receiver on a post 100 feet away (left).*

Instead of more efficient mirrors, and so on, the solution to increasing signal-to-noise ratio is to eliminate the passive reflector and actively transmit a bright modulated beam from the opposite side of the road. This is, in fact, the design basis of most industrial IR beam sensors. Unlike the indoor-use units that have 20–30-foot ranges using a reflector at the opposite sense point, industrial interrupted beam sensors generally use separate IR transmitters and receivers and can have ranges of hundreds of feet.

A CHALLENGE IN THE MAKING

Creating both a sufficiently bright IR source (signal) and a discriminating receiver which works under all ambient lighting conditions (noise) is no easy task, but it was a challenge. I'd like to say that all my attempts were successful, but they weren't. Distance, point-to-point elevation differences, and weather all played a part in making a simple proposition into a monumental task (have you ever tried to accurately aim a light beam, that you can't see, at a target 160 feet away?).

It would be a waste of article space if I didn't tell you that I was ultimately successful. I did indeed create a brute force industrial-strength infrared laserbeam sensor suitable for use across hundreds of feet of open space in a hurricane. Along the way, I devised a couple of useful circuits for expanding the capabilities of the low-cost off-the-shelf IR beam sensors should you havelessdemandingapplications. For example, while the 20–30-foot indoor Radio Shack units are not suitable for outdoor use, their indoor range canbe extended to 200–300 feet using an active IR transmitter instead of the passive reflector. This make them potentially suitable for grocery stores, warehouses, long garages, and so on.

EXTENDING THE RANGE OF LOW-COST IR SENSORS

The typical single-unit IR beam sensor consistsof a modulated IR LED and lens which directs a focused IR beamacross the area of detection (this beam may diverge to a 3-foot circle at 20 feet). At the detectionend point we place a reflector that reflects the modulated beam back to another lens which focuses the received signal onto a photodiode. The receiver circuit analyzes this signal for intensity and proper modulation frequency and closes a relay contact if it is correct. When the signal beam is present the relay closes; when the beam is interrupted, it opens. Such systems are effective until you reach a round-trip distance where the light reflected back

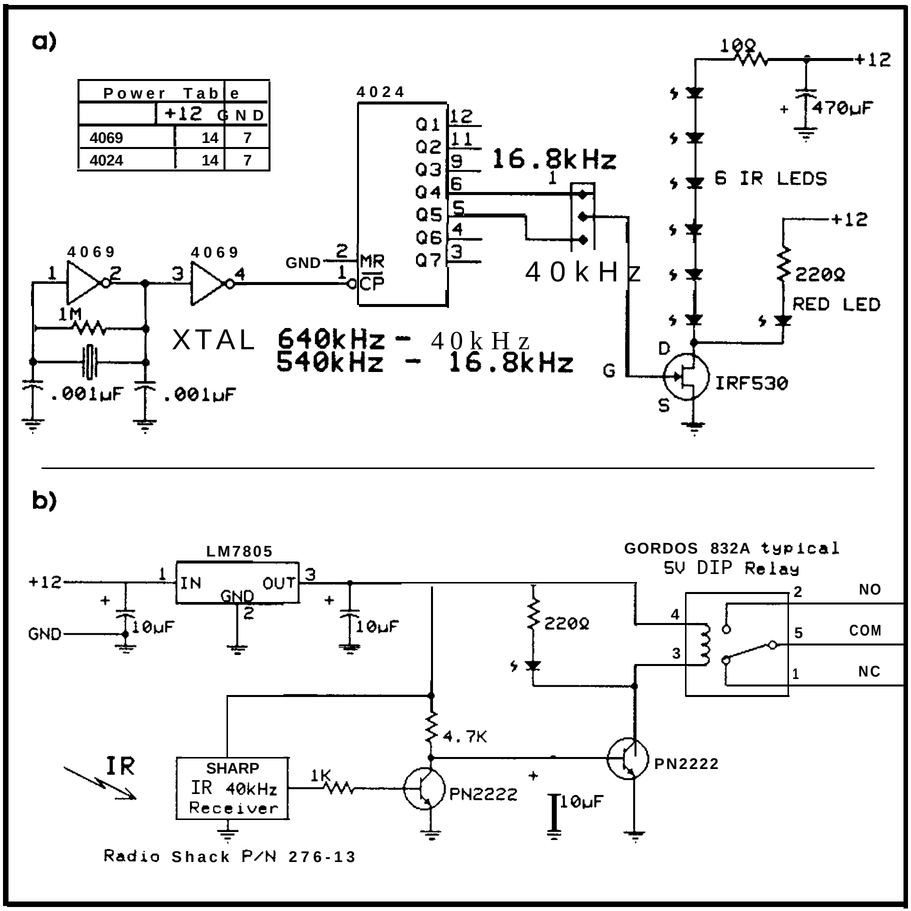


Figure 1—*(a) A powerful modulated IR transmitter can be built with two chips and a handful of discrete parts. (b) A companion receiver is easily made with a Sharp 40-kHz IR receiver module.*

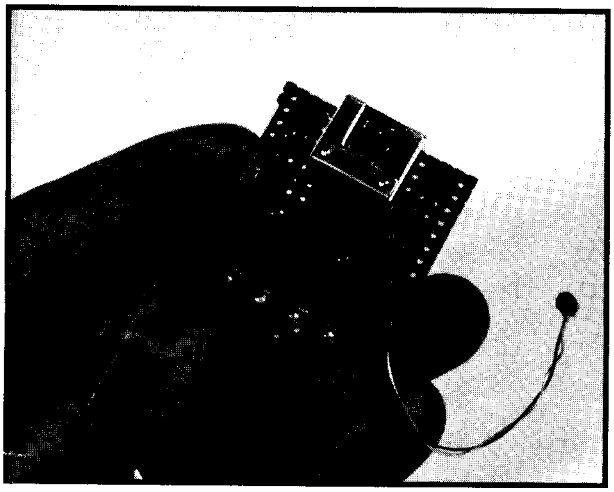


Photo 1—*The modulated IR receiver unit fits neatly on a small piece of perf board. An LED provides a visible indication of when the relay is energized.*

is too low to be accurately sensed above the background light levels.

Unbundling the transmitter and receiver extends the usable range. Since the receiver is tuned to a specific modulation frequency, it makes sense that any source of this frequency would trigger the relay in the same way as a reflected beam would. Further, if this substitute source could be collimated into a tight beam it could actually be located hundreds of feet from the receiver and still trigger the relay if it were the right modulation and sufficient intensity. When you consider how little light actually comes back from an internal LED reflecting off something 20 feet away it is no wonder that substituting a modulated IR source makes a big difference. This difference can be translated two ways: a bright detection point source allows extended range for a given receiver sensitivity, or it can increase the reliability of detection at shorter distances by improving the signal-to-noise ratio with lower receiver sensitivity.

Radio Shack sells two off-the-shelf IR beams sensors. One (model #49-307) is a stand-alone unit with an internal 115-VAC power supply, buzzer, and relay. The second unit (model #49-551) is smaller and requires an external 12-V supply. Interestingly, while these units serve the same functions and are sold by the same vendor, they operate at different modulation frequencies. The first one uses 40 kHz while the second operates at 16.8 kHz.

My experiments indicated that these two units are very selective and don't have a lot of tolerance for out-of-band signals. Even a very bright IR source only 400 Hz off frequency will be ignored. Because they are so selective, simple RC or 555-type oscillator circuits are generally inadequate—especially if the application involves large ambient temperature swings. The proper design for long-term operation should incorporate a crystal-controlled frequency reference.

Figure 1 and Photo 1 show a simple high-power infrared LED modulator with crystal reference. The two-chip circuit typically operates on 12 V but will tolerate 6-14-V supplies. To select a 16.8-kHz modulation frequency you use a 540-kHz ceramic resonator and the Q4 pin on the CD4024 binary divider chip. To select a 40-kHz modulation frequency you use a 640-kHz ceramic resonator and the Q5 output pin. The resulting clock frequency is used to switch the gate of a power FET and six series-connected IR LEDs (the red LED is there just to tell you something is happening).

With each diode dropping about 1.7 V and the IRF530 having an on resistance of about 0.18 ohms, the IR LEDs are being pulsed with about 200 mA. This is very bright and will work in most ambient conditions.

LASERS: LIGHT SHOW OR COMMUNICATION?

It's one thing to be challenged, it's quite another to succeed. While the circuitry just described might sound adequate for virtually any applica-

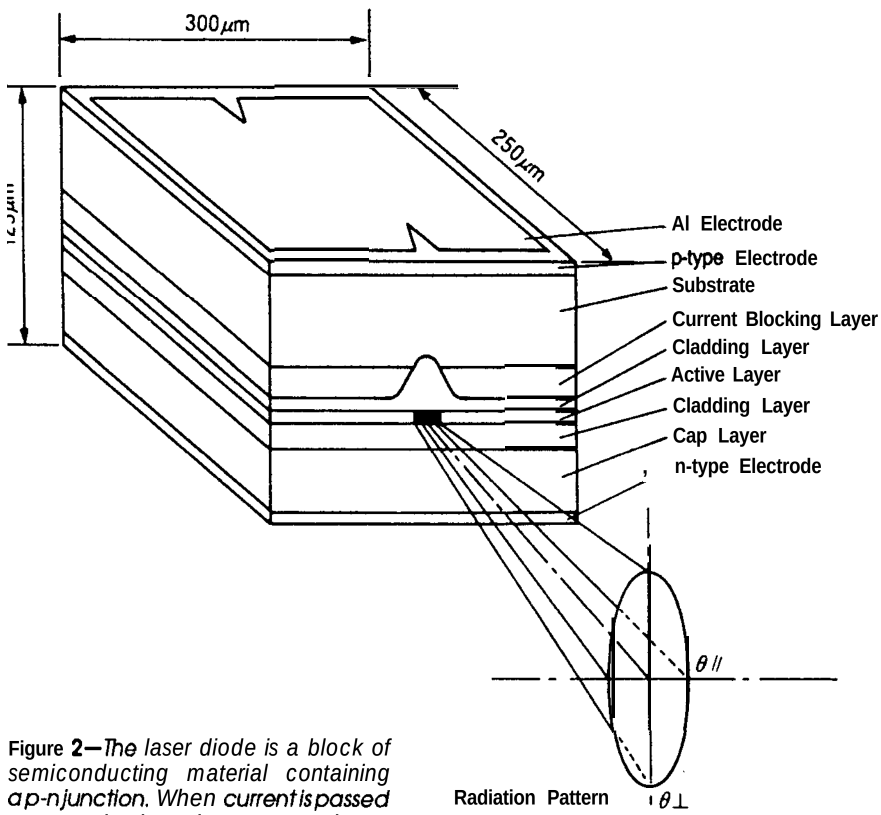


Figure 2—*The laser diode is a block of semiconducting material containing a p-n junction. When current is passed through the junction, energy is released as light and heat.*

tion, the realities of signal-to-noise in my specific outdoor application is a bear. Out there every leaf, snow flake, and rain drop becomes a potential false-positive detection error.

Using the previous circuit only solved part of the problem. Bundling the six LEDs together and adding a 2-pound, 4-inch-diameter lens provided an error-free signal for the 15 feet across one part of the driveway. Unfortunately, I still had a one-hundred-foot span to cover that seemed an insurmountable objective.

Rather than jack hammer the pavement installing pressure transducers, or stake out a pit bull with a walkie-talkie, I concluded that the only high-powered light source that could go the distance and stay collimated would be a laser. Initially I had rejected them because of the cost and complexity of use, but I was getting desperate. A simple experiment convinced me that this was indeed the answer.

Since I had done a previous article on modulating a He-Ne laser, I had all the necessary experimental hardware at hand. As dusk approached, I mounted the laser on a tripod next to the garage and aimed it at a front-surface mirror about 100 feet away. The mirror in turn reflected the beam along another sensing path, through a kitchen window, onto the lens of the receiver (considering that the environment and weather played such a role, if it could be detected through the window then I wouldn't have to deal with heated enclosuresand other weatherproofing headaches). I connected a table lamp through the receiver relay so that I could monitor results from outside.

Turning the laser on cast a faint red spot on the mirror and window pane but the laser still projected a bright spot that covered the whole receiver lens. Even after a distance of about 150 feet the spot was only about three inches in diameter. I turned on the modulation oscillator and slowly increased the input frequency to the laser. As I passed 39.6 kHz, the table lamp came on. Using my hand to interrupt the beam, it shut off in response. Success at last?

Well beyond dusk now, I walked back toward the house to check the arrangement once again. Satisfied that I might have solved the problem at last, I glanced over my shoulder toward my handiwork. Egad! It looked like a carnival light show! Red laser beams projected all over the place.

The bright red beam extended across the driveway from the laser to the mirror. The mirror reflected from its front surface, back surface, edges, dust, scratches, you name it. It could hardly have been more noticeable if I had mounted a red flood light in place of the mirror. Where the beam passed through the kitchen window, the situation was repeated. The various reflected beams were much weaker but in the darkness they were quite pronounced and had the appearance of multiple starbursts.

Well, it was obvious that this intrusion detector had a few bugs. The laser had the proper intensity and sufficient signal-to-noise ratio, but

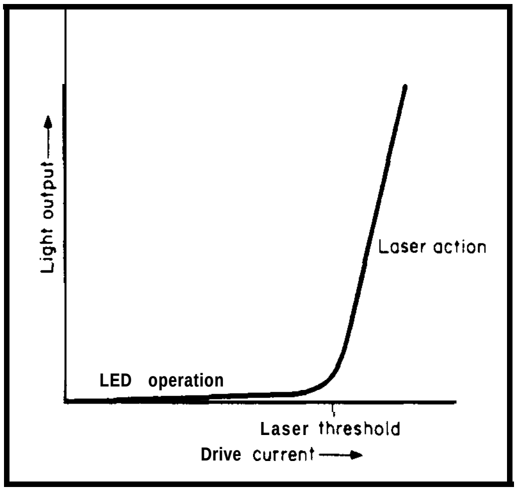


Figure 3—*The laser diode's drive current must be carefully regulated near the laser threshold so it is high enough to be above the threshold, but low enough to avoid burning out the diode.*

using a visible laser was impossible. What seemed barely visible in daylight was now quite spectacular in darkness. Certainly no semiintelligent door-to-door salesman would allow himself to be detected by a system that he could merely step over or around. Should he be closely followed by some eco-phreak protesting laser irradiation of the neighborhood and innocent children, I could also have a bunch of crazy radicals on my hand. I had to be more clandestine about such activities. Invisibility was the only answer.

INFRARED LASER DIODES

The more cost-effective invisible lasers emit in the infrared range. The most common types of these are carbon dioxide gas (CO_2) and LED solid-state lasers. CO, lasers tend to be a bit large and on the powerful side. Of course, if one's intention is deterrence as well as detection, setting fire to a salesman's sample case as he passes through the beam will certainly do that. All humor aside, however, CO, lasers are much too costly and dangerous for anything less than high-heat applications or starwars. IR laser diodes are really the better way to go.

This was not a new revelation, unfortunately. I would have used a laser diode instead of an IR LED in the first place if I had had more experience with them. Laser diodes have been around for years for use in fiber-optic communications, but they have been extraordinarily expensive and complicated to modulate. The ones I had previously used were pulse-mode-only devices that were hardly applicable. Fortunately, improved manufacturing techniques and higher production volumes have changed the situation entirely. With millions of lasers in CD players and printers, it's possible to find continuous-output devices for as low as $5 apiece from surplus houses.

DIODE DYNAMICS

Laser diodes are similar in structure to standard LEDs. The laser diode is a block of semiconducting material containing a p-n junction just like an LED (Figure 2). When current passes through the junction, energy is released as light and heat. The color of the emitted light depends on the band-gap energies of the materials used, and the amount of waste heat depends on the conversion efficiency. Continuous output double-heterojunction laser diodes, typically used in CD players, are made from GaAs or GaAlAs. They radiate at 720 to 900 nm. New InGaAsP and InP lasers emit in the range of 1150 to 1600 nm.

From a structural standpoint, LEDs and laser diodes are very similar. The only important difference is that in a laser diode two of the edge faces (or facets) are cleaved and coated so that they reflect part of the generated light back into the semiconductor. The reflected light stimulates the emission of other photons. The stimulated emission also tends to produce the strongest amplification at narrow wavelengths rather than across a broad range like an LED.

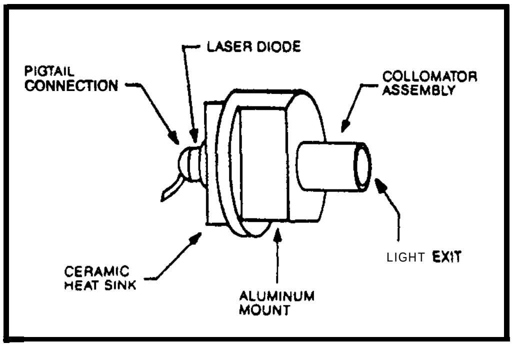


Figure 4—*While the actual laser diode is quite small, the heat sink and collomator assembly are necessary for proper operation.*

Of course, there is a price to be paid for all this. Laser diodes operate at much higher drive currents than LEDs. At low currents, LEDs and laser diodes behave the same. At slightly higher currents, the laser diode becomes a superluminescent diode but does not have sufficient gain to produce laser oscillation. Only when the operating current is equal to or greater than the laser diode's "threshold current" will the diode operate as a laser. Figure 3 shows a typical threshold curve.

This threshold curve is very important to laser diode operation and figures prominently in driver circuit designs. The threshold point of a laser diode is a dynamic value which depends greatly on case temperature. Special care should especially be taken at low temperatures. A laser diode that takes 80 mA to operate when the temperature is 60°C might only take 50 mA at 0°C for the same light output. Unless the current is reduced as the temperature decreases, excessive drive current will burn up the diode.

The lifetime of a laser diode decreases sharply with higher operating temperature and output power. Increasing the operating temperature causes threshold current to rise and efficiency to drop. The lost efficiency creates more heat again. Left unchecked, such a condition can cause thermal runaway of the laser diode.

To facilitate proper operation, laser diodes incorporate an integral photodiode (usually mounted on the rear facet) which directly monitors the light output. Using this photodiode in a closed-loop controller allows a system to set a specific light intensity regardless of ambient temperature. In combination with a simple heat sink to remove excess heat during operation, virtually all worry of thermal runaway is eliminated.

Once you've got the diode running you might think that a tight beam of light is automatically emitted from the laser's quartz window and you can project a spot on the wall, right? Guess again. Like an LED, the beam spreads out in an elliptical pattern from the diode. For the typical IR laser diode, the beam spreads out at a 35" angle perpendicular to the junction and 10" in the plane of the junction.

Just like regular LEDs, laser diodes rely on external optics and housings to redirect the light. Laser diodes generally use stacked lenses called a collimator to tighten the beam into circular form and concentrate the energy. In my experience, trying to use a laser diode without a collimator is like running a car without gas: it has a lot of potential but goes nowhere. When you purchasea laserdiodemake sure it has a collimator if your intention is to have a laser "beam." Like the laser diode used in my prototype demonstrates (Figure 4), this collimator and diode combination need not be particularly large.

MODULATING LASER DIODES

Diode lasers require no warm up and can be modulated directly by varying the drive current. However, whatever the modulation technique, the laser's specified operating envelope must be maintained below its maximum operating limits.

Figure 5 is the schematic of the closed-loop laser diode modulator pictured in Photo 2. The laser diode is a 780-nm 5-mW Sharp LT022MC laser diode with attached collimator which I bought from Meredith Instruments for $15. Its threshold current is nominally 50 mA at 25°C. The IR3C02A chip is a special closed-loop laser diode controller chip made by Sharp. It includes all the current averaging, driver, and comparator circui ts needed to operate and protect a laser diode.

The IR3C02A chip operates on +5 V and -5 V. Because my particular control system uses a common 12-V bus to power remote peripherals, I added power supply circuitry to the modulator that allows it to operate from a single 9-24-V supply input. This supply voltage is regulated down to +5 V with a 7805 three-terminal regulator and inverted with an ICL7660 DC-to-DC converter to produce -5 V.

The IR3C02A provides both monitoring and feedback control to the laser diode and provides up to 170 mA of drive current. The laser diode is connected between pins 1 and 2 with a series current-limiting resistor. The laser's photodiode monitor is connected between pins 2 and 3.

When current is turned on, the controller chip ramps the power to the laser diode. As power is applied, the laser's output is compared to a preset maximum power level determined by the settingof R1. If theoutputexceeds the setting, the chip automatically lowers the current to compensate.

External modulation is applied through a separate transistor which provides additional drive current to the laser. The collector resistor limits this additional drive current to be within protectable limits. The base of the transistor connects to any TTL-level modulation source. You can use the 16.8-kHz/40-kHz oscillator circuit from Figure 1 to make an interrupted-beam sensor or you can send any form of digital data.

SEEING IS BELIEVING

The good news about using a modulated infrared laser as an interrupted beam sensor is that you can use it for hundreds of feet. The bad news is that it is impossible to aim without considerable effort.

Twenty-foot reflected-beam sensors project a very large irradiation pattern. A little trial and error holding the reflector until you happen upon the right "spot" to trigger the beam is a relatively easy task. At 20 feet, simple eyeballalignment techniquesarequite adequate.

Using an infrared laser at a distance of 200 feet is quite another matter. Commercial units usually incorporate sighting scopes for direct line-of-sight alignment. Like sighting a hunting rifle, one simply sets the receiver in the crosshairs of the transmitter'sscopeand tumson the switch. Unfortunately, once we add any kind of an angled reflector into the laser beam's path, visible alignment is complicated by an order of magnitude.

Still, like the visible He-Ne laser, there is no real substitute for seeing

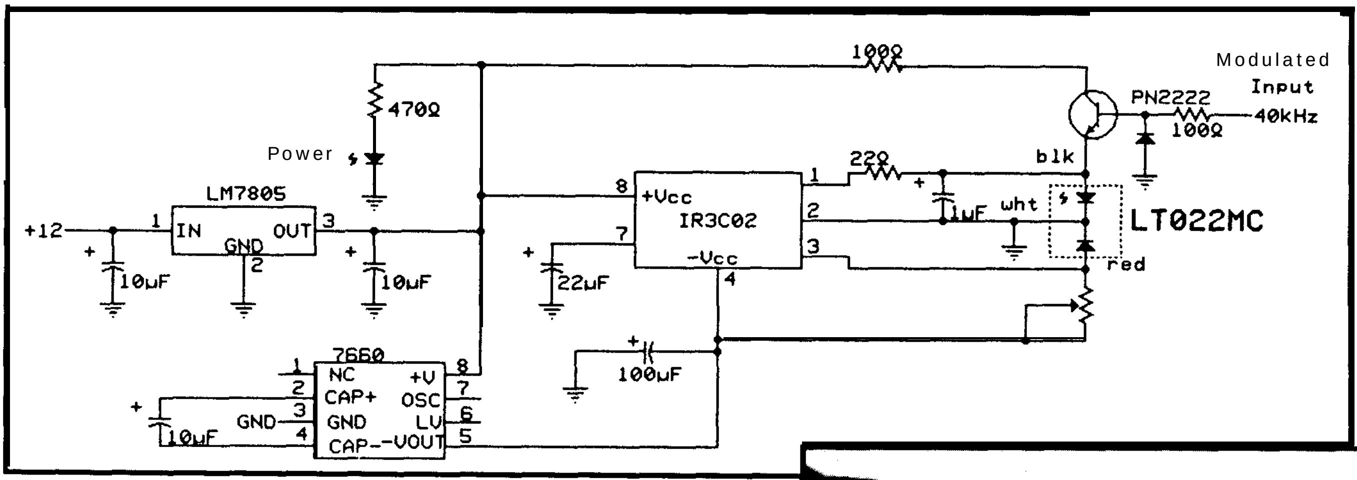


Figure 5—The *closed-loop laser diode modulator circuit uses a Sharp chip designed specifically for the purpose.*

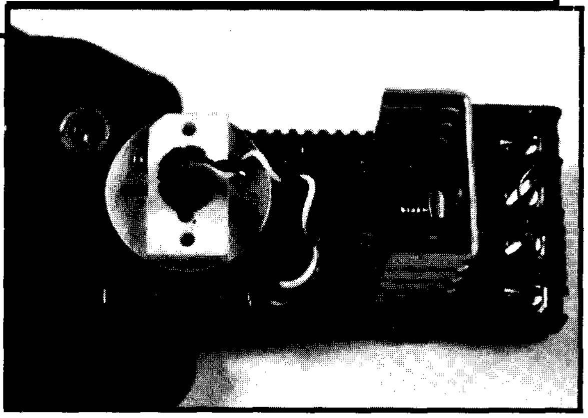


Photo 2—*The LT022MC laser diode can be seen on the left side* **of the prototype of the** *circuit in Figure 5. The LM7805 is attached to the heat sink.*

where the beam is actually going. Are we receiving the true incident beam or a secondary reflection? Is the receiver set in the center of the projected spot or at the edge? Just how bright of a signal is this after bouncing off two reflectors anyway? How do I know I don'thavesomedangerousreflections pointed directly at my eyes while I'm working on this? Questions like this can only be answered if we actually "see" the infrared laser beam.

While infrared radiation is invisible to the humaneye,itisquitevisible to "electronic eyes." A video camera or camcorder which incorporates a CCD (charge-coupled-device) video sensing element "sees" infrared light the same as visible light. If you aim a CCD camera at a blank wall and point the average hand-held IR remote control at it, it will appear like you have a flashlight aimed at the wall. In fact, these invisible sources of light look surprisingly bright once you can see them!

The correct and safe way to experiment with and align an IR laser device is to do it entirely with a CCD camera. While wearing IR safety glasses, view the operating laser on a video monitor. That way you can see exactly where it's going and measure relative intensity as well.

Of course, this can all appear a bit Rube Goldberg to the neighbors. In the dark of night I loaded the video monitor, camera, tools, and assorted test gear into a wheel barrow with a 200-foot extension cord. I attached the transmitter to the side of the garage and pointed it toward a piece of white posterboard about 100 feet away. Using the CCD camera, I easily centered the three-inch spot on the mirror. Using the cardboard again allowed me to find and position the beam at a convenient location on the side of a post. I centered the receiver in the beam pattern and the relay instantly pulled in. Success!

OUT OF ISOLATION

Unfortunately, a good design tends to multiply. One little laser on thecomerofthegaragehasbeenjoined by one across the front deck and one between the house and small garage. Next, I suppose1 have to add one from the small garage to the big garage, one across the rear deck and, I suppose, across the other driveway entrance (What? I failed to mention that one?). Monitoring the perimeter could get a little out of hand. Eventually I could have this giant web of infrared sensing energy encircling the whole place.. .

Enough! See what happens when you make a little LED flasher that works!

Inactuality, thisinterrupted-beam sensor design is only part of an integrated network of environmental monitoring and control. I view these sensors as positive verification that other more esoteric devices are actually working correctly. It is quite true that a web of IR beams is a total solution in itself but this involves a lot of hardwiring and physical placement of sensors. Considering that the latter is done primarily with a posthole digger, I continue to look for less strenuous sensing alternatives.

being down in the Circuit Cellar is no longer like being in a hole (actually, this cellar is better than most people'slivingrooms). Icanseewhat's goingonoutinthedrivewaybyglancing at a video monitor with automatic camera switching (see CIRCUIT CELLAR INK issues 1 and 2 for the design of this video multiplexer). My next design objective is to totally dispense with the perimeter sensors and use a processor with the cameras to directly do motion sensing and pattern recognition.

Yes, I know that such systems supposedly exist. I already have a

$5000 Sony video motion detecting system installed that I am experimenting with, and there are many PC-based units offered as well. Unfortunately, like the IR beam sensors I just finished discussing, these units won't work outside either. The dynamics of outside lighting causes too many errors.

So, I have an infrared laser/magnetic loop sensor that tells me that something is there. The next objective is for the control system to decide who or what it is. Stay tuned; it's only a matter of time!

CAUTION

If you choose to experiment with lasers, especially invisible ones, I can't caution you enough about safety. Admittedly, the 5-mW unit I chose to use poses little hazard unless placed against your eyeball, but I approached the whole technology with respect. Since many of the laser diodes available to experimenters come via the surplus market, they often come without complete manufacturer specs or guarantees. In fact, most are sold as 5-mW-type, 10-mW-type, and so on, with virtually **no** real specifications or ideas of what they may be using as substitute parts. At the very worst, it's quite possible to end up with a wolf in sheep's clothing. Applying 85 mA current to an LT022MD laser diode typically produces about 3 mW output. If you apply the same current to an LT015MD laser diode (which comes in the same physical package) it produces about 30 mW (and can go as high as 50 mW)! Who knows what's in the bag? Be careful! ✣

Steve Ciarcia (pronounced "see-ARE-see-ah") is an electronics engineer and computer consultant with experience in process control, digital design, and product development.

IRS

210 Very Useful
2 11 Moderately Useful
2 12 Not Useful

SOURCES

Sharp IR3C02A
Available prepaid from Marshall Industries. Call (800) 522-0084 between 9:00 A.M. and 8:00 P.M. EST for pricing and delivery.

Lasers and laser diodes
Sharp Corp.
Electronic Components Division
Sharp Plaza
Mahwah, NJ 07430
(201) 529-8757

Gott Electronics
2227 DuFour Ave.
Redondo Beach, CA 90278
(213) 370-6287

Meredith Instruments
6403 N. 59th Ave.
Glendale, AZ 85301
(602) 934-9387

Timeline, Inc.
1490 W. Artesia Blvd.
Gardena, CA 90247
(800) 872-8878
(213) 217-8912

Digital Signal Processing

PART 2

Dean McConnell

DSP Applications with the TMS320C25

In the previous article, we discussed fundamental Digital Signal Processing concepts. Concepts were illustrated with BASIC language routines running on a PC/XT with EGA graphics. These demonstration programs implemented DSP functions such as digital filtering, correlation, and Fourier transforms. We also showed that common analog functions (comparators, peak clippers, rectifiers, and so on) could be conveniently performed in DSP software and, thus, replace analog circuit components.

For real-time digital signal processing, however, a PC alone is not capable of performing the high-speed sum-of-products operation required by digital filters and correlation.

Many excellent microprocessor-like DSP chips have been unveiled by various manufacturers such as Texas Instruments, Analog Devices, AT&T, and Motorola. The Motorola 56001, a 24-bit 10.25-MIPS DSP chip which has two separate X and Y data memories, is featured within Steve Jobs' NeXT computer. The ADSP-2100 DSP chip (manufactured by Analog Devices) provides 10 MIPS of 16-bit processing power and an assembly language instruction set which is almost like an HOL (high-order language). The AT&T DSP16 is a fixed-point DSP chip (like all of the previously mentioned processors) while the DSP32 is a 32-bit floating-point DSP chip. The Texas Instruments TMS320C25 DSP chip is an enhanced CMOS version (40 MHz) of the popular TMS32020 chip, TI's second-generation member of the TMS320 family. TI's latest offering, the TMS320C30, is a 32-bit, 33-MFLOPS floating-point DSP which recently went into full production.

Out of all the possible DSP chips on the market, I chose the TMS320C25 as the basis of a PC-hosted DSP card design. The TMS320C25 offered me the following benefits:

The TMS32010, (the first generation of the TMS320 family) is source code compatible with newer generations of the TMS320 family, which allowed me to reuse my existing TMS32010 routines.

TI provides extensive support via hotline, newsletters, BBS, and published applications articles and reports. An excellent reference is "Digital Signal Processing Applications with the TMS320 Family, Vol. I" (available from Prentice Hall, Englewood Cliffs, NJ).

The TMS320C25 provides more than enough performance for voice band DSP experiments (10 MIPS). It is compatible with the TLC32044 Analog Interface Chip, a 14-bit D/A converter and a 14-bit A/D converter (with built-in lowpass antialiasing filter) which interfaces with the C25 via a high-speed serial port. The TMS320C25's 128K-word total address space provides plenty of room for user applications, and its price is moderate as far as DSP chips go (about $80).

AN OVERVIEW OF THE TMS320C25

The TMS320C25 DSP is a 16-bit fixed-point microprocessor optimized for digital signal processing operations. Key features of the TMS320C25 include:

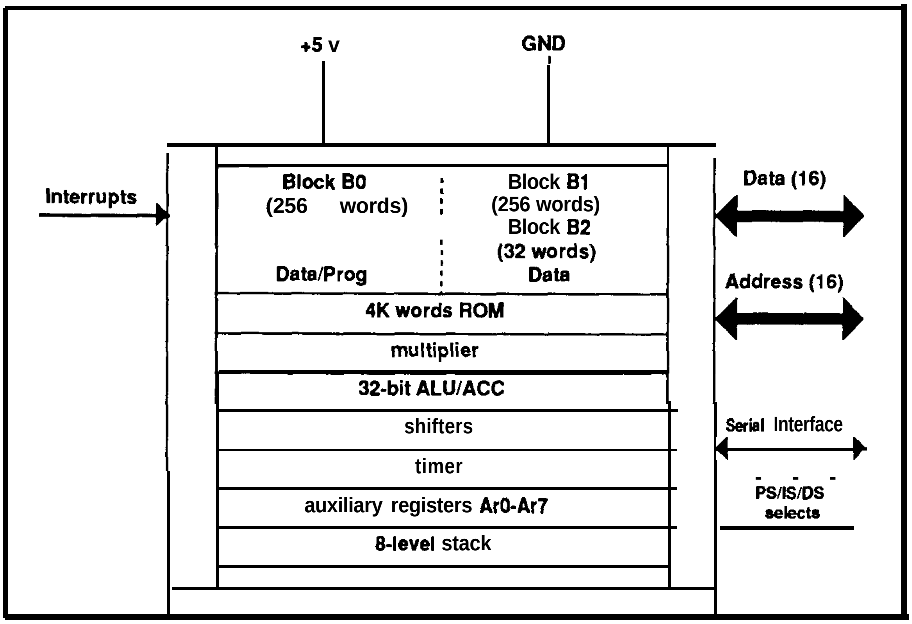


Figure 1—*Key features of the TMS320C25 include on-board high-speed RAM (used* **in digital filtering** *operations),128K-word* **data/program** *space,* **and a synchronous** *high-speed serial port.*

- 100-ns instruction cycle
- 544-word programmable on-chip data RAM
- 128K-word total data/program memory space
- 32-bit ALU/accumulator
- 16x16-bit parallel multiplier with 32-bit product
- repeat instructions for efficient memory and processor use
- block moves for data/program management
- on-chip timer
- eight auxiliary registers
- eight-level hardware stack
- sixteen input and output ports
- 16-bit parallel shifter
- wait states for slower off-chip devices
- serial port for ADCs, CODECs, or high-speed communications
- global data memory interface
- source code compatibility with TMS32010, first-generation DSP
- instructions for adaptive filtering, FFTs, and extended-precision arithmetic
- CMOS technology

Features such as single-cycle multiply/accumulate, 32-bit arithmetic unit, large auxiliary register file, and fast on-chip RAM, make the device a natural choice for digital signal processing applications. However, due to the large address spaces, multiple interrupts, wait states, timer, serial port, and multiprocessor interface, the TMS320C25 is also a good choice for high-speed general-purpose micro designs. Let's examine the key elements of the TMS320C25 architecture, as shown in Figure 1.

MEMORY MAP

The TMS320C25's memory map has its program memory separate from data memory, thus implementing a standard Harvard architecture. A total of 64K words of program memory, 64K words of data memory, and sixteen 16-bit I/O ports are addressable. Data memory is partitioned into 128-word pages. There are 544 words of on-chip 0-wait-state RAM, 256 of which (Block B0) can be configured as data memory or program memory, selectable under software control.

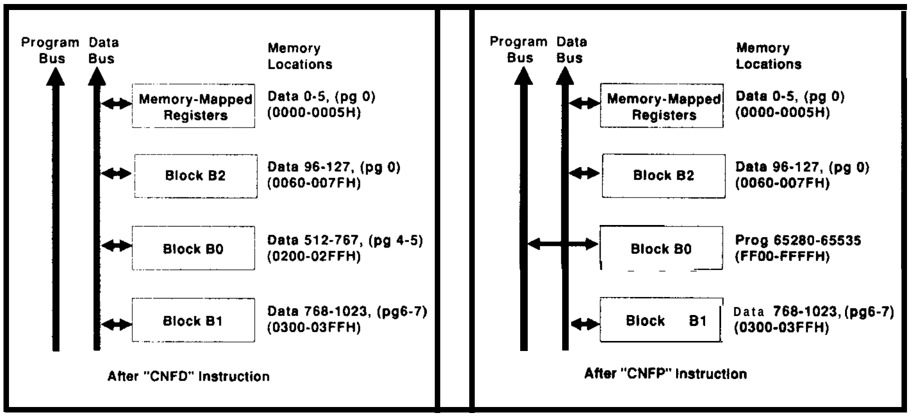


Figure 2—*The on-board 256-word block B0 RAM can appear in the data map or written program space through the software instructions* CNFD *and* CNFP.

Blocks B1 and B2, the remaining 288 words of on-chip RAM, are always configured as data memory. Block B2 consists of 32 words at locations 96–127 in page 0. The first five words are reserved for on-chip memory-mapped registers, and include:

- DRR, the serial port receive register
- DXR, the serial port transmit register
- TIM, the 16-bit timer register
- PRD, the 16-bit period register
- IMR, the interrupt mask register
- GREG, the global memory allocation register.

The remaining memory locations of Block B2 (locations 6–95) are reserved by TI. Block B1 resides in data memory from 0300H to 03FFH or pages 6-7. Block B0, when configured as data RAM (via the CNFD instruction), appears in pages 4-5 (0200H–02FFH). When configured as program memory (via the CNFP instruction), Block B0 appears at FF00H-FFFFH in program memory space (see Figure 2). The on-chip RAM is used in conjunction with special multiply/accumulate instructions to perform the convolution operation required by digital filtering.

ON-CHIP TIMER

The TMS320C25 provides a memory-mapped 16-bit timer register and a 16-bit period register. The timer is a continuously clocked down counter which is fed by CLKOUT1. For a 40-MHz TMS320C25 DSP, CLKOUT1 is 10 MHz. The period register, PRD, holds the starting count for the timer. A timer interrupt (vector location 0024H) occurs when the count goes to zero. The timer is reloaded with the PRD value after a decrement to zero. By programming the PRD register from 1 to 65,535, a timer interrupt, TINT, can be generated at intervals ranging from 200 ns to 6.5536 ms on a 40-MHz TMS320C25.

ALU/ACCUMULATOR/MULTIPLIER

The 32-bit accumulator is split into two 16-bit segments for storage in data memory: ACCH (high 16 bits) and ACCL (low 16 bits). The SACL dma instruction stores the low accumulator in the designated data memory address while the SACH dma instruction stores the high accumulator in data memory. Shifters at the output of the accumulator allow a left shift of from zero to seven bit positions. The TMS320C25 utilizes a 16 x 16-bit two's complement multiplier which can compute a 32-bit product in a single machine cycle. Two multiply/accumulate instructions (MAC and MACD) provide the basis for sums-of-products operations used in filtering.

AUXILIARY REGISTERS

Eight auxiliary registers are used for indirect addressing of data memory or for temporary data storage. The auxiliary register pointer (ARP) selects one of the eight auxiliary registers. Subsequent indirect addressing instructions will use the auxiliary register pointed to by ARP. For ex-

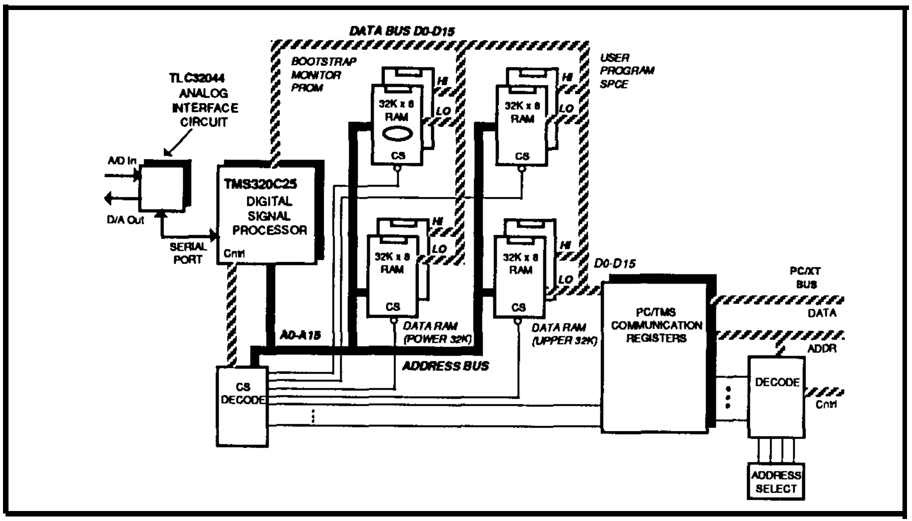


Figure 3-A typical TMS320C25 system design incorporates EPROM, RAM, and a TLC3204 for analog interfacing.

ample, if the contents of ARP is 2, then theinstruction **SACL** *+ will store the lower 16 bits of the accumulator in the data memory address pointed to by AR2 followed by an incrementing of the AR2 auxiliary register.

SERIAL PORT

The TMS320C25 includes an on-chip full-duplex synchronous serial port capable of operation to 5 MHz. Uses of the serial port can include high-speed synchronous data links, interprocessor communication, and connection to serial interface A/D converter chips such as the TLC32044. A typical TMS320C25 design is shown in Figure 3.

TLC32044 ANALOG INTERFACE CIRCUIT

TheTLC32044 AIC (Analog Interface Circuit) simplifies the task of providing an antialiasing lowpass filter, a sample-and-hold circuit, a 14-bit A/D converter, and a 14-bit D/A converter. All of the AIC functions are packaged within a single 28-pin DIP or an LCC which runs off of ±5 volts and connects to the TMS320C25 via the 6-line serial port interface. When an A/D sample is ready, the 14-bit sample is transmitted from the AIC to the TMS320C25 causing a serial receive interrupt, RINT. The received word is then read through the Data ReceiveRegister (location0 in thedata

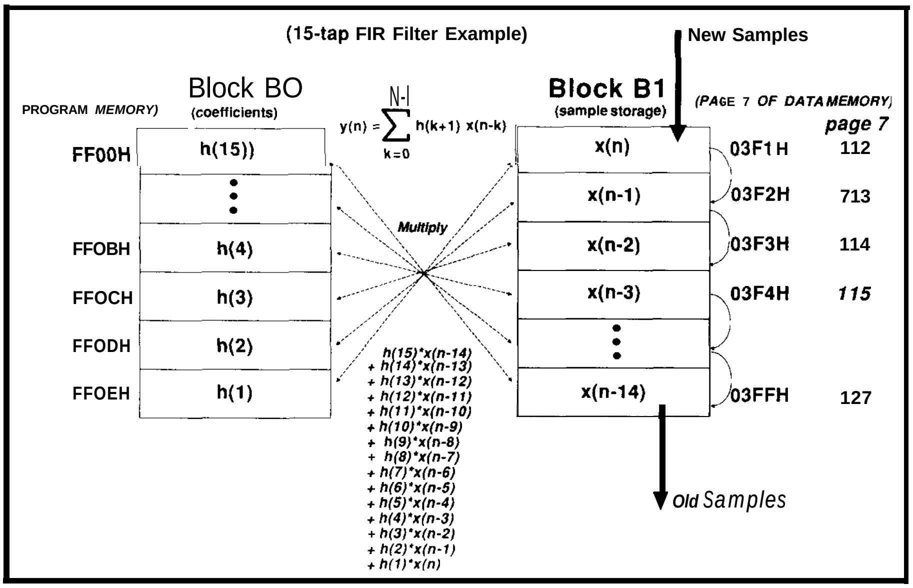


Figure 4—Digital filtering is made easy with the RPTK and MACD instructions. Input samples are placed in block B1, while coefficients are placed in block B0.

FIXED 16-BIT FRACTIONAL ARITHMETIC FOR THE TMS320C25

The first step for implementation on the TMS320C25 is to convert the floating-point numbers to a 16-bit two'scomplement representation. In two's complement notation, the largest positive number is 7FFFH (which represents +0.999969 in Q15 format) and the most negative number is 8000H (which represents -1). To convert a floating-point coefficient, for example –8.996965363621712E–003 (h(1) in the low-pass filter example), to 16-bit two's complement Q15 representation, we multiply by the maximum value available in 15 bits (32,768). The resultant product, -294.8, is two's complemented to get FED9H. Shown in the tables below are the lowpass and highpassfloating-point coefficients from the first article's FIR filter examples and their 16-bit two's complement counterparts used by the TMS320C25. It is worth noting that 16-bit numbers have a minimum resolution of 0.000031, so we must round any floating-point coefficient to thenearest millionths. In very high-precision digital filters, truncation of coefficients leads to Gibb's phenomenon, a "rippling" effect seen in the stop band of the frequency response.

FILTER COEFFICIENTS
FLOATING-POINT AND FIXED 16-BIT REPRESENTATION

LOWPASS COEFFICIENTS: (312.5 Hz @ 10000 Hz Fs)

COEFFICIENT NUMBER	FLOATING-POINT VALUE	16-BIT REPRESENTATION (2'S COMPLEMENT)
h(1)	-8.996965363621712E-003	FEDSH
h(2)	7.864067330956459E-003	0102H
h(3)	3.349926695227623E-002	044AH
h(4)	6.534869968891144E-002	085DH
h(5)	9.889715164899826E-002	0CA9H
h(6)	0.1295655647516251	1075H
h(7)	0.1499759534597397	1312H
h(6)	0.15625	1400H
h(S)	0.1489759534597397	1312H
h(10)	0.1265655647516251	1075H
h(11)	9.889715164899826E-002	0CA9H
h(12)	6.534869968891144E-002	085DH
h(13)	3.349926695227623E-002	044AH
h(14)	7.864067330956459E-003	0102H
h(15)	-8.996965363621712E-003	FED9H

HIGHPASS COEFFICIENTS: (1250 Hz @ 10000 Hz Fs)

COEFFICIENT NUMBER	FLOATING-POINT VALUE	16-BIT REPRESENTATION (2'S COMPLEMENT)
h(1)	8.996965363621712E-003	0127H
h(2)	-7.864067330956459E-003	FEFEH
h(3)	-3.349926695227623E-002	FBB6H
h(4)	-6.534869968891144E-002	F7A3H
h(5)	-9.889715164899826E-002	F357H
h(6)	-0.1265655647516251	EF8BH
h(7)	-0.1489759534597397	ECEEH
h(8)	0.64375	6C00H
h(9)	-0.1469759534597397	ECEEH
h(10)	-0.1295655647516251	EF8BH
h(11)	-9.889715164899826E-002	F357H
h(12)	-6.534869968891144E-002	F7A3H
h(13)	-3.349926695227623E-002	FBB6H
h(14)	-7.864067330956459E-003	FEFEH
h(15)	8.996965363621712E-003	0127H

memory map). To output to the DAC, the desired 14-bit code is written to data memory location 1 of page 0, DXR, the serial port transmit register.

FILTERING WITH THE TMS320C25 DESIGN

Now, let's see how the TMS320C25 can be programmed to perform real-time digital filtering. Recall the 15-tap Finite Impulse Response (FIR) filter from Part 1 of this article, in CIRCUIT CELLAR INK #13. It required a "sum-of-products" operation of the current and 14 most recent past input samples multiplied by 15 special coefficients (each sum-of-products operation was required with each new sample input).

The TMS320C25 provides a repeat instruction (RPTK N-1) which, when coupled with a MACD instruction, performs a looped sum of products of coefficients (placed in Block B0) with data samples stored in Block B1. In the case of the 15-tap filter, the argument to the RPTK instruction

```
;*****************************************************************
;               FINITE IMPULSE RESPONSE FILTER PROGRAM
;
;              For the TMS320C25 Digital Signal Processor
,
; This program implements a 15-tap FIR Low-Pass filter. It
; assumes that a TLC32044 Analog Interface Circuit is connected
; to the TMS320C25's serial port.  A receive interrupt (RXINT)
; occurs with every new A/D sample and is read through the
; receive serial port register,  DRR (loc 0 of data page 0).
; Outputs to the D/A converter are performed through writes to
; the serial data transmit register,  DXR (location 1 of data
; page 0).
;*****************************************************************
,
; PAGE 7 VARIABLES
XN      EQU     113          ; STARTING LOCATION IN PAGE 7 WHERE
                             ;  INPUT SAMPLES ARE STORED
                             ;  X(N-1)=LOC 114 . ..X(N-14)=LOC 127
YN      EQU     112
,
LENGTH  EQU     14
BOCOFS  EQU     OFFOOH       ; BEGINNING OF BLOCK BO FILTER COEFF.
                             ;  h(15)=LOC FFOOH...h(1)=LOC FFOEH
TMPO    EQU     112          ; used for temporary storage
,
FCOFS:
        ; FIR FILTER COEFF. FOR A 0.03125 Fs Low-Pass Filter
        ; N = 15 (15 taps)
        ; Q15 FORMAT (FIXED 16-BIT TWO'S COMPLEMENT)
        DW      OFED9H, 0102H, 044AH, 085DH   ; hl-h4
        DW      OCA9H, 1075H, 1312H, 1400H    ; h5-h8
        DW      1312H, 1075H, OCA9H, 085DH    ; h9-h12
        DW      044AH, 0102H, OFED9H          ; h13-h15
,
START:  SOVM                 ; SET OVERFLOW (SATURATION) MODE
        LDPK    0            , SELECT PAGE 0
        LALK    OEOOH        ; INITIALIZE
        SACL    TMPO         ; STO TO
        LST     TMPO         , OEOOH
        LALK    03FOH        ; INITIALIZE
        SACL    TMPO         ; ST1 TO
        LST1    TMPO         ; 03FOH
        LDPK    7            ; SELECT PAGE 7
        ZAC
        LARP    0
        LRLK    0,03FFH      ; POINT ARO TO B1
        RPTK     LENGTH      ; CLEAR INPUT BUFFER
        SACL    *-
        CNFD                 , MAKE BLOCK BO DATA RAM
        LRLK    0,0200H      ; POINT TO BO DATA RAM ADDRESS
        RPTK    LENGTH       ., READ COEFFICIENTS INTO BO
        BLKP    FCOFS,*+     ; LOAD FILTER COEFFICIENTS
        CNFP                 ; NOW, CONFIGURE BO AS PROGRAM MEMORY
        LDPK    0            , SELECT DATA PAGE 0
        LACK    020H         ; ALLOW ONLY TXINT
        SACL    IMR          ; STORE IN INTERRUPT MASK REGISTER
        EINT
        ; SET UP TLC32044 TO A 10-KHZ SAMPLING RATE
        LALK    3
        SACL    DXR          ;SIGNAL SECONDARY COM
        LALK    32028        ; set Fs TO 10280 HZ
        IDLE                 ; WAIT HERE UNTIL TXINT
        ZAC
        EINT
        IDLE                 ; WAIT HERE UNTIL TXINT
        EINT
        LALK    3
        SACL    DXR          ; SIGNAL SECONDARY COM
        LALK    1400H        ; 16638
        IDLE                 ; WAIT HERE UNTIL TXINT
        ZAC
        IDLE                 ; WAIT HERE UNTIL TXINT

        LACK    011H         ; ENABLE RXINT ONLY
        SACL    IMR          ; STORE IN INTERRUPT MASK REGISTER
        EINT
LOOP:   B       LOOP
TINT:   SACL    DXR
        EINT
        RET
```

(continued)

Listing 1 —*FIR.ASM illustrates a 15-tap FIR lowpass filter implemented with a TMS320C25 DSP* and a TLC32044 AIC.

would be set for (15 − 1), or 14, iterations. Let's look at Figure 4 for an illustration of how the input samples and FIR coefficients must be stored in memory for the filter to work properly. First of all, the input sample buffer must be stored in Block B1, with the oldest sample (greater than 15 delays old) stored at the top of the block, or at location 127 of page 7. $X(n)$, the current input sample, is stored at (127 – 15), or location 112, of page 7.

The coefficients are placed within the main body of a filter program in the form of data statements. These statements are assembled and placed in program memory. Before the filter can operate, we must place the filter coefficients located in program memory into Block BO. We do this by configuring Block BO as data RAM with the CNFD instruction. Then we move the block of coefficients stored in program memory to the Block BO data RAM locations with the block move statement, BLKP. After moving the coefficients, we configure BO as program memory appearing at locations FFOOH-FFFFH with the CNFP instruction.

Refer to Listing 1, FIR.ASM, which illustrates the 15-tap FIR lowpass filter implemented with a TMS320C25 DSP, TLC32044 AIC design (the TMS320C25 is the CPU and the TLC32044 is the A/D and D/A converter). [Editor's Note: Software for *this article is available for downloading from the Circuit Cellar BBS or on Software On Disk #14. For downloading and purchasing information, see page 84.]* The lowpass filter coefficients we used in the last article's BASIC implementation used floating-point numbers. For the TMS320C25, however, the coefficients must be in a 16-bit two's complement representation. For a description of this format, see "Fixed 16-bit Fractional Arithmetic" on page 51.

As each new sample arrives from the A/D converter, we execute the RPTK 14 and MACD FFOOH,*- instructions. The MACD FFO OH, *- instruction performs the following sequence for each pass:

- *the previous product is added to the accumulator

```
RINT:   LDPK    0          ; SELECT DATA PAGE 0
        LAC     DRR        ; READ INPUT SAMPLE FROM A/D
        LDPK    7          ; SELECT DATA PAGE 7
        SACL    XN         ; STORE LATEST SAMPLE IN XN
        LRLK    1,03FFH    ; POINT AR1 TO BOTTOM OF BLOCK B1
        LARP    1
        MPYK    0          ; CLEAR P REG
        ZAC                ; CLEAR ACCUMULATOR
        RPTK    LENGTH     ; N - 1 FILTER STAGES
        MACD    BOCOFS,*-  ; MULTIPLY FILTER STAGES BY COEFFICIENTS
        APAC               ; ACCUMULATE FINAL PRODUCT
        SACH    YN, 1      ; STORE OUTPUT IN YN
        LAC     YN,2       ; GET OUTPUT OF FILTER IN ACCUM
        LDPK    0          ; SELECT DATA PAGE 0
        ANDK    0FFFCH     ; MASK OFF BITS 0-1 FOR AIC
        SACL    DXR        ; OUTPUT TO D/A
        EINT
        RET
        END
```

Listing 1-continued

- the data memory location pointed to by the selected auxiliary register is multiplied by the program memory location specified within the MACD instruction (FF00H starting address)
- the program memory pointer is incremented (FF01H, FF02H, etc.)
- the data memory location pointed to by the selected auxiliary register is copied into the next higher data memory location (*x(n-13)* is copied into *x(n-14)*)
- the selected auxiliary register is decremented (the "*-" within the instruction performs this function)

When the MACD instruction is placed after a RPTK N- 1 instruction, the sequence shown above is repeated N times. As discussed in the previous article's FIR filter example, this sequence is repeated 15 times. When placed within the RPTK loop, the MACD instruction requires a single cycle (100 ns at 40 MHz). So, for a 15-tap filter, the sum-of-products loop takes only 1.5 µs! To check out the operation of the filter, you could sweep a function generator signal (approximately 2 V p-p from DC to 5 kHz) into the ADC input while viewing the output (from the D/A converter) on a dual-trace oscilloscope.

SINE WAVE GENERATION

The high-speed nature of the TMS320C25 makes it a natural for the generation of sinusoidal waveforms. Using a ROM look-up table which holds 128 entries of sine values from 0 to 360 degrees, we can generate a sine wave of any frequency (within the audio band) by stepping through the table at a constant rate with a constant step size. Of course, the accuracy of the sine wave generated depends on the number of points stored in the table. But for our purposes, 128 table entries will do.

The frequency of the sine wave depends on the time interval between output samples and the step size with which we index into the sine table. To simplify things, we will use a constant output rate of, say, 10 kHz. The frequency of the output waveform will then depend only on the step size. The step index must wrap around the end of the table in a "modulo 128" fashion. See Listing 2, DIAL.ASM, an adaptation of "Precision Digital Sine-Wave Generation with the TMS32010" by Domingo Garcia, Texas Instruments (found in "Digital Signal Processing with the TMS320 Family, Volume I"). *Alpha* is a variable which serves as a modulo 128 counter/index which cycles through the sine table. *Delta,* the step size, is added to *Alpha* each time the routine is called. In this adaptation, the sine wave generation subroutine will be called via a timer interrupt set for an 8-kHz rate. The sine table entries are in two's complement representation ranging from +1 to -1.

The program shown in Listing 2 generates a standard telephone dial tone, which is the sum of 350-Hz and 440-Hz sinusoids. ALP350 and DEL350 are variables used in the

```
;*************************************************************
; DIAL.ASM
; This program generates a dial tone (350 Hz + 440 Hz).
; It is an adaptation of: "Precision Digital Sine-Wave
; Generation with the TMS32010" by Domingo Garcia, Texas
; Instruments found in "Digital Signal Processing
; Applications with the TMS320 Family, Volume 1."
;
; This program also demonstrates the use of the timer
; interrupt.  We want a timer interrupt to occur every
; 125 microseconds (an 8-kHz rate).   The PRD, period
; register, determines the number of clock cycles which
; are counted down prior to a timer interrupt.
;    For a 40-MHz TMS320C25, PRD ---> 1250,
;    For a 36-MHz TMS320C25, PRD ---> 1125,
;    For a 20-MHz TMS320C25, PRD ---> 625,
;     to yield an 8,000-Hz timeout rate.
;
; The tone variables, Ton350 and Ton440, determine the
; frequency of the resulting sine waves by the
; formula:  tone = desired frequency/0.25
; For example:  Ton350 = 350Hz/0.25 = 1400
;      and,     Ton440 = 440Hz/0.25 = 1760
;
;*************************************************************
;CONSTANTS
PRDVAL  EQU     1125
TON350  EQU     1400        ; value for 350 Hz
TON440  EQU     1760        ; value for 440 Hz
;PAGE 0 VARIABLES
DEL350  EQU     96
ALP350  EQU     97
SIN350  EQU     98
TEMP    EQU     99
MASK    EQU     100
OFSET   EQU     101
ALP480  EQU     102
DEL480  EQU     103
SIN480  EQU     104
TMPO    EQU     105
STRT:   LDPK    0
        LALK    07FFFH
        SACL    MASK
        LALK    SINE
        SACL    OFSET
        ZAC
        SACL    ALP350
        SACL    ALP480
        LALK    TON350
        SACL    DEL350
        LALK    TON440
        SACL    DEL480
        LALK    OE00H           ;SET STATUS 0 REG
        SACL    TMPO
        LST     TMPO
        LALK    03F0H           ;SET STATUS 1 REG
        SACL    TMPO
        LST1    TMPO
        ZAC
        SACL    TEMP
        LACK    1
        SACL    TMPO
        LALK    PRDVAL          ;SET PRD REGISTER
        SACL    PRD             ;FOR A 8KHZ TIMEOUT RATE
        LACK    008H            ;INTERRUPT MASK FOR
        SACL    IMR             ;TIMER INTR ONLY
        EINT
LOOP:   B       LOOP
TMRINT: LAC     ALP350,8
        SACH    TEMP
        LAC     TEMP
        ADD     OFSET
        TBLR    SIN350
        LAC     ALP350
        ADD     DEL350
        AND     MASK
        SACL    ALP350
        LAC     ALP480,8
        SACH    TEMP
        LAC     TEMP
        ADD     OFSET
```

(continued)

Listing 2—*DIAL.ASM generates a standard telephone dial* **tone, which is the** *sum of 350-Hz and 440-Hz sinusoids.*

generation of the 350-Hz sine wave while ALP 440 and DEL440 are used in the generation of the 440-Hz sine wave. TONE1 and TONE2 are constants to which the Delta variables are initially set, respectively. At an 8-kHz sampling rate, the constant equals the desired frequency multiplied by four. For example, the constant for 350Hzis 350*4 or 1400. Just prior to sending the output to the D/A converter via the Sacl DXR instruction, the two sine waves are scaled and added together. Connecting the DAC output to an oscilloscope would show how the two signals add together to form an enveloped waveform. Or, you could connect theoutput to an audio amplifier/speaker (such as the Radio Shack 277-1008) to hear the dial tone.

GOING FURTHER

We could continue with the sine wave table look-up theme to include DTMF generation, busy tone, ringback tone, error tone, and many others. All it takes is a few calculations of the Delta table index constants and timing loops. For instance, a busy tone is the **sum** of 480-and 620-Hz sine waves pulsed on/off at a half-second rate. Ringback is the sum of 440 Hz and 480 Hz waves turned on for one second and turned off for three seconds.

With constants having been calculated for mark/space frequencies along with the proper bit period timing, one could develop a simple FSK modulator. At each bit boundary, a mark or space frequency would be selected depending on the next bit's value. Inherent in the look-up table scheme, a transition from a mark to a space (or vice versa) results in a change of the table step size, providing a phase-continuous waveform at bit boundaries. Phase-continuous FSK switching allows for better signal processing on the opposite end of the modulation process (demodulation). An analog FSK modulator which switches between two free-running oscillators (one for space, one for mark) doesn't provide phase continuity since the two oscillators are asynchronous with each other.

```
        TBLR    SIN480
        LAC     ALP480
        ADD     DEL480
        AND     MASK
        SACL    ALP480
        LAC     SIN350
        ADD     SIN480
        ANDK    0FFFCH
        SACL    DXR
        EINT
        RET
;**********************************************************
; SINE - 128 points of a sine wave in Q14 2's complement
;        format
;**********************************************************
SINE:   DW      0000H,0324H,0646H,0964H,0C7CH,0F8DH,1294H
        DW      1590H,187EH,1B5DH,1E2BH,20E7H,238EH,2620H
        DW      289AH,2AFBH,2D41H,2F6CH,3179H,3368H,3537H
        DW      36E5H,3871H,39DBH,3B21H,3C42H,3D3FH,3E15H
        DW      3EC5H,3F4FH,3FB1H,3FECH,4000H,3FECH,3FB1H
        DW      3F4FH,3EC5H,3E15H,3D3FH,3C42H,3B21H,39DBH
        DW      3871H,36E5H,3537H,3368H,3179H,2F6CH,2D41H
        DW      2AFBH,289AH,2620H,238EH,20E7H,1E2BH,1B5DH
        DW      187EH,1590H,1294H,0F8DH,0C7CH,0964H,0646H
        DW      0324H,0000H,0FCDCH,0F9BAH,0F69CH,0F384H,0F073H
        DW      0ED6CH,0EA70H,0E782H,0E4A3H,0E1D5H,0DF19H
        DW      0DC72H,0D9E0H,0D766H,0D505H,0D2BFH, 0D094H
        DW      0CE87H,0CC98H,0CAC9H,0C91BH,0C78FH,0C625H
        DW      0C4DFH,0C3BEH,0C2C1H,0C1EBH,0C13BH,0C0B1H
        DW      0C04FH,0C014H,0C000H,0C014H,0C04FH,0C0B1H
        DW      0C13BH,0C1EBH,0C2C1H,0C3BEH,0C4DFH,0C625H
        DW      0C78FH,0C91BH,0CAC9H,0CC98H,0CE87H,0D094H
        DW      0D2BFH,0D505H,0D766H,0D9E0H,0DC72H,0DF19H
        DW      0E1D5H,0E4A3H,0E782H,0EA70H,0ED6CH,0F073H
        DW      0F384H,0F69CH,0F9BAH,0FCDCH
        END
```

Listing 2-continued

ADAPTIVE FILTERING

In our last TMS320C25 application, we will examine a fascinating form of filters which actually adjust themselves in real time. This form of filter is called an Adaptive Filter. It is given the name "adaptive" because it adapts or changes in real time to minimize an error attribute. In the case of the example that follows, the filter adapts its coefficients to form a bandpass filter which tracks periodic (sinusoidal) signals seen at its input.

One use for an adaptive filter is the cancelation of unwanted periodic signals (periodic noise). For instance, an annoying 60-cycle hum corrupting an audio signal, background motor noise drowning out a speaker's voice over a telephone line, or an enemy CW (continuous wave) jammer burying a weak friendly signal are all examples of **where** adaptive filters could be applied.

You might ask why a notch filter couldn't be used? Well, notch filters would be fine as long **as** the frequency of the interfering noise is known beforehand and doesn't vary with time. However, when the frequency is unknown and/or varies with time, the adaptive filter provides the best solution.

In Figure 5, a 15-tap adaptive filter is **shown.** Notice the arrow through each of the fifteen coefficients indicating variable adjustment. Each time a new input enters, the filter performs its basic sum-of-products operation yielding a new $y(n)$ output which is subtracted from $x(n)$ to provide an error, $e(n)$. With the error, adaptation constant (Mu), and past x values, the coefficients, $h1$ through $h15$, are updated.

An implementation of a 15-tap Least Mean Square (LMS) adaptive filter, ADAPT.ASM, can be found on the Circuit Cellar BBS. Very quickly after start of execution, this program adjusts its coefficients to form a bandpass filter matching periodic spectra seen at its input. The output of the adaptive filter, $e(n)$, is equal to $x(n)-y(n)$, the output of the bandpass subtracted from the input sequence. To test the filter, one could feed a sine wave into the ADC input and watch the output of the DAC on an oscilloscope. Notice that after rapid readjustmentsofthegenerator'sfrequency, the DAC output diminishes to a low level in a matter of milliseconds. The adaptive filter automatically adjusts itself to attenuate any sinusoid seen at its input.

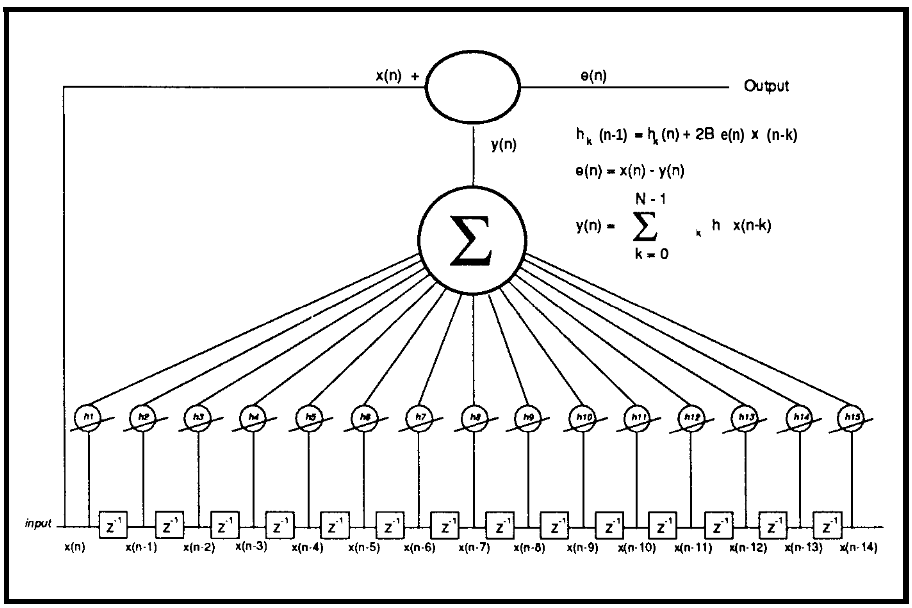


Figure 5-*A sample application is this adaptive filter which adjusts its coefficients on the fly to cancel sinusoidal signals at its input.*

AND INTO THE FUTURE

In this article, we have provided application examples featuring the Texas Instruments TMS320C25 Digital Signal Processor. Other exciting areas in the field of digital signal processing include speech voice coding (vocoding), speech synthesis, speech recognition, signal demodulation, and spread spectrum communications, to name a few. I believe that in the future, digital signal processing will replace many traditional analog circuit functions with software. DSP will also make its way into "smart," cost-effective sensors which will not only detect the movement of a warm body into or out of a room, but will determine if the object is a dog, cat, child, or adult. With this type of detailed sensor information, a master home computer/controller could determine if turning on or turning off room lights is appropriate.

In the near term, general-purpose DSP will likely be applied to primarily low-frequency audio band applications. As future generations of DSPs and A/D converters increase in speed and performance, digital signal processing will find widespread acceptance beyond the audio range and into the VLF, LF, and HF RF bands.❖

To facilitate experimentation in DSP applications, "DXP-25," a PC-based DSP development system is now being offered through:

InteliHome
571 Responsive Way
McKinney, TX 75069
(214) 548-8503
Fax: (214) 548-1521

The $529 DXP-25 package includes a full-length PC/XT DSP card based on a 36-MHz TMS320C25 DSP design, a resident monitor in EPROM (with associated PC host monitor control program), 8K words of l-wait-state user program RAM, 8K words of l-wait-state data RAM, an 8-bit PC/DXP-25 communication port, an 8-byte read/write I/O register block for PC/DXP-25 communication, a TLC32044 (14-bit ADC and DAC with programmable sampling rates up to 19.2 kHz), a TMS320C25 assembler, download utilities, and demonstration programs. Options include O-wait-state RAM, expansion RAM (up to 128K bytes data RAM and 64K bytes program RAM), and a FIR filter design program.

Dean McConnell is a Senior Software Engineer for Rockwell International, Richardson, Texas, where he is involved with development of real-time computer-controlled communication systems for the military.

IRS

2 13 Very Useful
2 14 Moderately Useful
215 Not Useful

BASIC Radioactive Randoms

FIRMWARE FURNACE

Ed Nisley

Generating random numbers is a perennial-computer magazine topic, but the articles always seem to discuss the linear congruential or software shift register implementations. While pseudo-random (pronounced "fake random") numbers may be OK for computer science types, Real Engineers get Real Random Numbers by timing nuclear disintegrations with a Geiger-Müller detector. When you want random numbers, why settle for less than the best?

The problem has **always been** that you had to build a radiation detector from scratch, adapt one from the surplus market, or pay far too much for a heavy-duty chunk of electronics with far more features than you need. A few months ago I saw the RM-60 Micro Roentgen Radiation Monitor from Aware Electronics. It is a Geiger-Müller tube that connects to a PC's parallel or serial port, with the circuitry drawing power from a single interface pin.

The RM-60 costs under $100 in onesies and comes with some reasonably well-done software that turns your PC into a Geiger counter, strip-chart recorder, and data logger. All that is well and good, but as soon as I saw the gadget I realized that Real Random Numbers were now within reach! All I needed was an RTC52 and a little firmware.. .after all, who wants to tie up a PC to count fissions? *[Editor's Note: See "From the Bench" in issue #8 of* CIRCUIT CELLAR INK *for the design of the RTC52 single-board controller.]*

Although the RM-60 is interesting, I thought this would provide a good excuse to explore grafting assembly language programs onto the BASIC-52 interpreter. Measuring time intervals is a common industrial problem and I have never met anyone who wrote a BASIC program that ran fast enough.

I'll start with a pure BASIC implementation and wind up with a BASIC language extension devoted to returning random numbers. Along the way you'll learn more about the innards of the BASIC interpreter than ever showed up in the manuals.

BASICALLY RANDOM

Regardless of how trivial the hardware interface appears, you should always write a simple test program to make sure that the bits are actually arriving at the right ports. Figure 1 shows the interface between the RM-60 and an RTC52 board; while you can certainly use different computer hardware, it's hard to mess up three color-coded wires.

The RM-60 produces a downgoing 75–90-µs pulse each time it detects a radioactive decay particle. It is sensitive to alpha, beta, and gamma particles, but the output pulse is identical for all three because the detector tube operates in Geiger mode. The maximum count rate is thus over 10,000 counts per second...at which point you have more than just a radon problem in your basement.

The background radiation in my office provides about 10-15 counts per

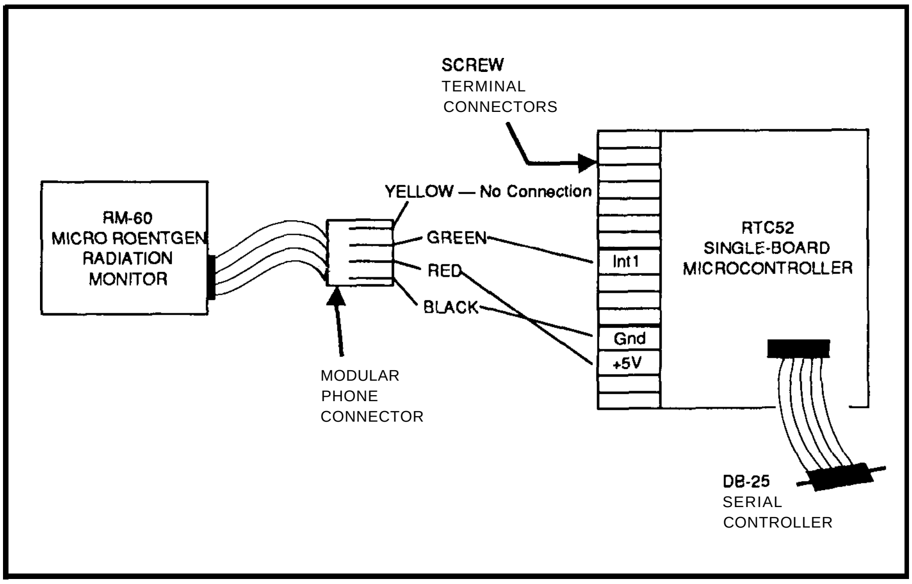


Figure 1 *--Interfacing the RM-60 Micro Roentgen Radiation Monitor to a microcomputer is as easy as three wires.*

Listing 1 -This *BASIC-52 program, BASRAND.BAS, measures the time interval between pulses on the INT1 input pin. Because BASIC requires about seven milliseconds to handle each interrupt, it cannot resolve short time intervals.*

```
100 rem - set up things
110  t0=0: time=0: clock1
120 s0=10 : rem scale factor
130 onex1 9010
200 rem - reporting loop
210 if t0=0 goto 210
220 print using(##.###),t0,"|",
222 s1=t0*s0 : if s1>70 then s1=70
224 print spc(s1),"*"
226 t0=0
230 goto 210
9000 rem - beginning INT1 pulse
9010 time=0
9012 if t0=0 then onex1 9110 : port1=1
9014 reti
9100 rem - ending INT1 pulse
9110 t0=time
9112 onex1 9010 : port1=0
9114 reti
```

minute, with the time between counts ranging from 100 µs to over 45 seconds: after all, nuclear disintegrations are random events. The firmware should be able to cope with the shortest possible interval, which, as we will see, can be a problem with pure BASIC code.

I chose the INT1 input because BASIC-52 has an ONEX 1 statement that branches to a specific line number when a pulse occurs on the INT1 pin. The BASRAND program shown in Listing 1 uses a pair of interrupt routines to measure the interval between two successive pulses; the one at Line 9000 sets the TIME value to zero when the first pulse occurs, while the one at Line 9100 records the TIME value at the second pulse.

Figure 2 shows BASRAND's output with the RM-60 sitting on my desk. Because the code requires two pulses for each time interval, it can record at most half the intervals. While BASIC-52 is a fast interpreter, it lacks the speed for timing millisecond-length events. The BASIC interrupt response time is about 2 ms and the interrupt handlers require about 5 ms each. BASRAND cannot detect pulses occurring during those times and will return incorrect values.

The RTC52 board has an LED on the 8052's TO output (also known as Port 3, Bit 4). While it would be comforting to blink the LED when the program detects a pulse, the RTC52 was not designed to allow control of that bit from a BASIC program. Instead, BASRAND pulses the low-order bit of Port 1 (P1.0) which I watched on an oscilloscope.. .rest assured that the remaining programs will blink the LED!

CALLING THE COUNTER

BASRAND's peccadilloes may be acceptable in some applications, but there are others where returning the wrong answer can be fatal. The solution is to capture the pulses using an assembly language routine.

The CALL statement provides a convenient way to invoke an assembly language routine from BASIC. CALL takes one argument, which may be an integer from 0 through 127 or any other unsigned 16-bit quantity. The interpreter converts a value in the first group into an address from 4100h through 41FEh and branches to that location; values in the second group are treated as exact branch addresses. BASIC does not look before it leaps, so it is your responsibility to have the assembly language code in place before CALLing it.

CALLRAND.BAS, shown in Listing 2a, uses the first method to CALL three routines. CALL 0 and CALL 1 turn that silly LED OFF and ON, respectively, so the BASIC program can indicate when it gets a pulse without yourneedinganoscilloscope. CALL 2 accesses the code that measures the pulseintervals. The CALL entry points between 3 and 127 are not used.

The BASIC-52 CALL statement will not accept a variable as the parameter, so you must code an integer. I favor enumerated constants rather than obscure digits (CALL LEDON would be easier to decipher than CALL 1), but in this case it makes little difference.

The method you use to put your assembly language code at the right addresses dependson your assembler and linker. The Avocet AVMAC51 macro assembler includes "segments" thatcanbestartedatspecificaddresses; other assemblers use simple ORG statements to set the address. Despite the bad reputation hardware segments

```
 1.740|                 *
 4.025|                                        *
 1.380|             *
 3.850|                                      *
 1.535|               *
 1.540|               *
 4.135|                                         *
 2.310|                       *
11.415|                                                                                                              *
 2.950|                             *
 7.715|                                                                             *
 1.010|          *
 1.860|                  *
 2.865|                            *
 4.735|                                               *
 0.895|        *
 9.860|                                                                                                  *
 4.395|                                           *
 5.265|                                                    *
 2.330|                       *
 0.985|         *
 0.995|         *
 2.020|                    *
 1.315|             *
 5.535|                                                       *
 3.350|                                 *
 3.615|                                    *
 7.350|                                                                         *
 9.630|                                                                                                *
 1.390|             *
 2.450|                        *
 4.630|                                              *
```

Figure 2— *BASRAND displays the measured time interval and also converts the value info a 'bar chart." The intervals have a resolution of 5 milliseconds.*

have gained from the IBM PC's convoluted programming, software segmentation is actually not a bad idea.

As you can see from CALLRAND. A5 1 in Listing 2b, much of the source establishes the correct addresses for code and variables in the BASIC interpreter. The critical addresses are the three AJMPS starting at address 41OOh in the EPROM_4K segment. Because BASIC expects to find successive CALL entry points at 4100, 4102, 4104, and so on, the two-byte AJMP instruction is just what's needed.

Of course, there is no reason why the three routines couldn't be accessed using CALL 0, CALL 10h, and CALL 20h, which would put the entry points at 4100, 4120, and 4140, respectively and eliminate the need for any JMPS at all. You could also use three-byte L JMPs by CALLing only even-numbered routines at the cost of one unused byte between successive LJMPS.

The CALL_0 and CALL_1 procedures are only a few bytes long, as they use the 8052's SETB and CLR bit-banging instructions to twiddle the LED output. The listing omits the latter routine to save space.

CALL _2 uses two utility routines: Wait Input stalls until the input bit from the RM-60 matches the FO flag and returns with interrupts disabled, while CatchTime converts the (frozen) BASIC TIME value into a 16-bit integer. The assembly code records the time at which two successive pulses occur, leaving it to the BASIC program to convert the values into a time interval.

BASIC-52 provides a T IME function that returns the current time in 5-millisecond increments. The CLOCK 1 and CLOCK 0 statements start and stop the clock, respectively, and the current time is recorded in three bytes in the 8052's internal RAM. One byte counts from 0 to 199, with one count every 5 ms, while the remaining two bytes are a 16-bit counter of whole seconds.

To get a consistent 16-bit value from those three bytes, the code multiples the seconds counter by 200, adds the fractional seconds, and uses the low-order 16 bits of the result. The result can time up to 327.68 seconds

```
100 rem- setup
110 call 0         :rem turn LED off
120 s0=10          :rem chart scale factor
130 clock 1        : rem start the clock
1000 rem - capture starting 6 ending times
1010 call 2
1020 call 1        :rem turn LED on while thinkin
1030 pop t2,t1     : rem t2 = stop, tl = start
1040 t0=t2-t1      :rem t0 = elapsed time
1050 if t0<0 then t0=t0+(64*1024):
                       rem fix 16 bit wrap
1060 t0=t0/200     :rem convert back to seconds
2000 rem - print a chart line
2010 print using(##.###),t0,"|",
2020 s1=t0*s0 : if s1>70 then s1=70
2030 print spc(s1),"*"
2100 call 0         : rem turn LED off again
2110 got0 1000
```

listing 2a—*This* **code,** *CALLRAND.BAS, CALLs three assembly-language routines to time the pulses* **and turn an** *LED on and off. BASIC is better at formatting the output, while assembler handles the timings and bit-banging with* **aplomb.**

```
;--------
; Define segments

            DEFSEGBAS RAM,START=$0000,ABSOLUTE,CLASS=XDATA
            DEFSEGBAS-ROM,START=$OOOO,AESOLUTE,CLASS=CODE
            DEFSEGEPROM_4K,START=$4000,ABSOLUTE,CLASS=CODE
            DEFSEGCALLCODE,START=$4200,CLASS=CODE
            DEFSEGINT_RAM1,START=$00,ABSOLUTE,CLASS=DATA
            DEFSEG INT_RAM2,START=$30,ABSOLUTE,CLASS=DATA

;--------
; BASIC entry points & constants

            SEG    BAS_ROM
            ORG    $30
BasicOps    EQU    $           ; BASIC operations entry point
PUSHR2RO    EQU    154         ; push R2:RO to arq stack

<<< some code omitted >>>

;--------
; BASIC internal RAM variables

            SEG    INT_RAM2
            ORG    $47
TimeMS      DS     1           ; 47 - TIME value, 5 ms counter
TimeSecHigh DS     1           ; 48 - TIME value, sec (high byte)
TimeSecLow  DS     1           ; 49 - TIME value, sec (low byte)
MS-TICKS    EQU    200         ; number of TimeMS ticks per sec

I· - -
; BASIC CALL vectors
; Must use 2-byte AJMP instructions to fit vector spacing!

            SEG    EPROM_4K
            ORG    $4100
            AJMP   CALL 0      ; 0 - turn LED OFF
            AJMP   CALL-       ; 1 - turn LED ON
            AJMP  CALL-2       ; 2 - return next random interva

;. -
; Turn LED OFF

            SEG    CALLCODE
CALL-0      PROC
            SETB   LED         ; LED OFF when output is high
            RET
CALL-0      ENDPROC

<<< CALL 1 is similar code that turns the LED ON >>>

;--------
; Return next random interval

            SEG    CALLCODE
CALL-2      PROC
```

(continued)

listing 2b—*The* **BASIC CALL interface requires assembly code and variables** *at predefined addresses. You can specify where your code will wind up by using segments; this program, CALLRAND.A51, uses Avocet's AVMAC51 macro assembler syntax.*

```
;- ensure no input pulse right now...

            SETB FO         ; ON = inactive
            CALL WaitInput
            SETB EA         ; turn interrupts back on again

;- wait for starting pulse and capture current time

            CLR  FO         ; OFF = active
            CALL WaitInput
            CALL CatchTime
            SETB EA         ; allow interrupts now
            MOV  A,#PUSHR2R0
            LCALL BasicOps

;- make sure pulse has gone away

            SETB FO         ; ON = inactive
            CALL WaitInput
            SETB EA

;- wait for ending pulse and capture time again

            CLR  FO         ; ON = inactive
            CALL WaitInput
            CALL CatchTime
            SETB EA         ; allow interrupts now
            MOV  A,#PUSHR2R0
            LCALL BasicOps
            RET
CALL_2      ENDPROC
;------
;<<< WaitInput routine omitted because it's pretty obvious >>>
;<<<  it returns with interrupts disabled to allow time capture
>>>
;------
; Capture current time in R2:R0
; Convert time to linear count as
;  (seconds * counts/second) + counts
; Interrupts must be OFF to ensure that time doesn't change!

CatchTime   PROC
            MOV  A,TimeSecLow ; form  low partial product
            MOV  B,#MS_TICKS
            MUL  AB
            MOV  R2,B       ; save it
            MOV  R0,A
            MOV  A,TimeSecHigh ;  form high partial product
            MOV  B,#MS_TICKS
            MUL  AB
            ADD  A,R2       ; combine high part of low partial
            MOV  R2,A
            MOV  A,TimeMS   ; combine with ms counter
            ADD  A,R0
            MOV  R0,A
            MOV  A,R2
            ADDC A,#0
            MOV  R2.A
            RET
CatchTime   ENDPROC
            END
```

Listing 2b—*continued*

before wrapping, and the wraps are easy to detect in BASIC.

Although some assembly routines (such as the LED control procedures) don't return any data, ones like CALL 2 pose a problem. BASIC stores variables as 6-byte packed BCD floating-point numbers, quite different from the one- or two-byte integers with which you are familiar. How do you convert between formats?

The solution is the BASIC-52 Argument Stack, which BASIC uses during expression evaluation. The interpreter parses a statement such as A=B+5 by pushing the values of variable B and constant 5.0 onto the A-stack, calling a routine that adds the top two elements, then popping the result into the location of variable A. The PUSH and POP statements give you direct control over the contents of the A-stack, and utility routines accessible from assembly language allow you to POP and PUSH floating-point numbers, integers, and suchlike.

The assembly code PUSHes the time at which the starting and ending pulses occurred onto the A-stack, using a BASIC-52 routine that converts a 16-bit value into an equivalent floating-point number. The times appear to BASIC as values ranging from 0 through 65535.0, and the difference of two such floating-point numbers can range from +65535.0 through -65535.0 with no trouble: a timer wrap produces a negative value, which the code corrects by adding 64K.

The whole process of capturing the current time, converting it to a 16-bit value, and PUSHing it onto the A-stack takes about 1 ms, most of which is spent PUSHing. While this is much better than 7 milliseconds for the BASRAND interrupt handler, CALL-RAND can still miss a pulse while it's busy and report the wrong time interval.

Before going into the third program, a digression is in order: how do you get assembly language routines working in the first place? After all, BASIC's built-in debugging facilities are useless for CALLed routines. The answer involves some software, some hardware, and some firmware.

ADDRESSEE UNKNOWN?

The first step, of course, is to coerce your assembler into producing a hex file with instructions at the right addresses. Whether you use segments, as shown in Listing 2b, or ORGS, as you might with another assembler, the end product is a hex file you burn into an EPROM and stick into a socket on your 8052 board.

But, more likely than not, when you turn the power on and run that first version, the CPU will lock up and die. How do you find out what's wrong?

The first, and cheapest, line of defense is a software simulator that runs on your PC, reads the hex file (and perhaps the linker's MAP file), and simulates the action of the 8052 CPU while running your code. Because the simulator displays all of the internal registers, RAM, and SFRs, you can be very confident that your program will actually do what you intend. And, because the simulator doesn't depend on the actual 8052 hardware, your program can't crash the system by doing something really stupid.

Simulatingyourcodeisone thing, but how can yoube sure that you have all the interfaces between it and the BASIC interpreter correct? The answer is easy, but not obvious: simulate the whole BASIC interpreter along with your (teeny) program!

The key is to get the interpreter code as a hex file that you can feed into the simulator. To do so, you can actually extract the code directly from whatever processor chip you've got. Without going into the grim details, this program fragment will dump the entire contents of the internal ROM on the console:

```
FOR i=O to lfffh :
   PHO.CBY(i) : NEXT i
```

Of course, you must wrap a bit of code around that core to output the data in standard Intel hex format, then capture the console output to a PC disk file, but the process is not particularly difficult and serves as a good learning experience. Hint: you need to write a routine that converts a variable into two or four hex digits and displays the results on the console. As you'll find out, the PH 0 statement prepends a blank and appends an "H" character, making it useless for precisely formatted output.

And a caveat: the interpreter is both copyrighted and **mask registered,** so you need to be careful about making duplicates.

Anyhow, once you have loaded both BASIC-52 and your code into the simulator and fired it up, you can, no kidding, type BASIC code directly into the (simulated) serial port! It runs rather slower than real time, but works precisely like the real system. You can set code breakpoints and single-step right through the interface. After you have done this a few times, you can be quite sure that your code will work the first time in real hardware.

Al though you can bum your hex file into an EPROM, you may want to use an EPROM emulator instead. I've been using the Parallax 2764 EPROM emulator for a few months and find it to be an invaluable tool. You've seen the picture in ads, but the real charm of the device is that it uses surface-mount components on a circuit board that is only slightly larger than the 2764 EPROM it replaces.

The board connects to the parallel port on your PC through a modular phone cable. The EM64 program transfers a hex file through the cable to the emulator hardwareand-poof!-your program is available to the 8052 as if you had burned an EPROM. EM64 has several other useful tricks up its sleeve, but the hex file loader gets the most use. The gadget is an example of clean, simple, dedicated design that does one job and does it well.

It would be nice to have a full in-circuit emulator to provide hardware breakpoints, true single-stepping,and suchlike when the program is run-

Listing 3a— *EXTRAND.BAS. A cleaner way to add functions to BASIC-52 uses the built-in language extension feature, as in Line 200. This example adds a single keyword, but more complex code can have additional arguments that may include complete BASIC expressions.*

```
100 rem - setup
110 s0=50
200 rem - get the next interval, spin if not
      available
210 random
220 pop t0
230 if t0 < 0 got0 210
240 t0=t0/1000 : rem convert to milliseconds
300 rem - print a chart line
310 print using(##.###),t0,"|",
320 s1=t0*s0 : if s1>70 then s1=70
330 print spc(s1),"*"
400 rem - back to the top
410 got0 200
```

ning on the real hardware, but I find that the combination of a software simulator, an EPROM emulator, and some careful coding work nearly as well for the problems I need to solve.

KEY WORDS

But, for all of that, CALLRAND can still miss an occasional pulse while it's pushing values on the A-stack because two pulses may occur faster than the code can absorb them. The solution is to put all of the pulse timing into an assembly routine and leave the console output to the BASIC program, which is just what EXTRAND.BAS in Listing 3a does.

If you take a close look at Line 210 in EXTRAND.BAS, you'll see the keyword RANDOM, which you won't find in your BASIC-52 manual. RANDOM, as you might guess, puts the next random time interval from the assembly language routine on the A-stack, ready for the POP in Line 220.

Admittedly, the CALL interface from CALLRAND.BAS would work as well for this code, but I find that the slight extra effort required to integrate new functions right into the language pays off handsomely. RANDOM makes much more sense than an opaque CALL 2, particularly after a few months on a different project.

Although the BASIC keyword interface is somewhat more complex, the bulk of the code in EXTRAND.ASM is due to its improved capabilities. EXTRAND records times to the nearest millisecond, can't miss an input pulse, and blinks the LED automatically... what more can you ask?

TIME FOR A BUFFER

Unfortunately, the full listing for EXTRAND.ASM won't fit in the pages Curt gives me, but you can download it from the BBS. **[Editor's *Note:*** *The software for this article is available for downloading from the Circuit Cellar BBS, or on Software on Disk #14. See page 84 for information on downloading and ordering all the software from this issue.]* For our purposes here, there are two sections of interest: the interrupt routine that captures the pulse timing and the BASIC keyword interface.

Before that code can do anything useful, though, the code must reprogram some of the 8052's hardware. BASIC-52 passes control to a "user reset" after a power-on reset, so EXTRAND.ASM can set the serial bit rate, set Timer 0 to count 1-millisecond ticks, define a handler for External Interrupt 1 that gets control on down-going edges, and perform other housekeeping functions with no trouble at all.

Because Timer 0 runs at five times the normal rate, EXTRAND.ASM also includes an interrupt handler to increment the variables BASIC uses for the TIME and CLOCKx functions, but only every five ticks. In effect, the BASIC program behaves as though CLOCK1 were always in effect, so you can't turn off the clock without disturbing the EXTRAND interrupt handler. In addition, EXTRAND increments Elapsed, a separate 16-bit counter, every millisecond.

The code shown in Listing 3b gets control whenever the RM-60 detector produces an output pulse. Ext1Handler does only three things: copy the current contents of Elapsed into a ring buffer, clear Elapsed to

```
; PSW is pushed by BASIC interrupt handler code
                SEG EXTCODE
Ext1Handler PROC
                CLR LED              ; mark an incoming pulse
                PUSH ACC             ; save bystanders
                PUSH B
                PUSH DPH
                PUSH DPL
                ORL PSW, #$18        ;select bank 3
;--- capture current time in ring buffer
                JB   RingFull,L?done; skip if full
                MOV  DPH,RingPage  ; get ring head pointer
                MOV  DPL,RingHead
                CLR  EA              ; ensure time stands still
                MOV A,ElapsedHigh ; transfer to ring
                MOVX @DPTR, A
                INC DPTR
                MOV A,ElapsedLow
                MOVX @DPTR,A
                INC DPTR
                MOV ElapsedHigh,#0 ; reset counters
                MOV  ElapsedLow,#0
;--- update ring head pointer & check for full
                MOV A,DPL
                CJNE A,#LOW RINGSIZE,L?newhead
                CLR  A               ;  reset to start of buffer
L?newhead       MOV RingHead,A
                CLR  RingEmpty       ; have at least one entry!
                CJNE A,RingTail,L?nohit  ; head = tail means ring full
                SETB RingFull        ;  so mark it
L?nohit         SETB EA              ; timer ticks OK now
;--- return to interrupted code
L?done          POP DPL              ; restore bystanders
                POP DPH
                POP B
                POP ACC
                POP  PSW             ; restore this one
                SETB LED             ; turn off marker
                RET1
Ext1Handler ENDPROC
```

Listing 3b—*External* interrupt *1* handler routine.

zero, and update the **ring buffer pointers**. As I've discussed ring buffers in previous columns, there isn't much to say about this function. One "gotcha" is that BASIC catches External Interrupt 1 first and pushes the 8052 PSW register onto the stack before calling `Ext1Handler`, so what may appear as an unbalanced POP just before the RET is actually a required statement.

The ring buffer is 256 bytes long, so it can hold up to 128 two-byte time interval entries before overflowing. `Ext 1Handler` simply **discards interrupts** that occur when the ring is full. Because `EXTRAND.BAS` has no trouble keeping up with the average pulse rate the buffer never fills up.

The code in Listing **3c** removes entries from the ring buffer and returns them on the BASIC argument stack. It uses the BASIC routine that converts a 16-bit integer into a floating-point number and pushes the result, which can return values from zero through **64K.** If the ring is empty, `GETRANDOM` **uses the** PUSHFPC routine to push the floating-point value -1 .O directly onto the stack.

```
;--------------
; Return next random number from ring

           SEG   EXTCODE
GetRandom  PROC

;--- extract the current ring entry to R2:R0
           JB    RingEmpty,L?empty   ; ring empty?
           CLR   EA                  ; can't change ring now
           MOV   DPH,RingPage
           MOV   DPL,RingTail
           MOVX  A,@DPTR
           MOV   R2,A                ; R0.2
           INC   DPTR
           MOVX  A,@DPTR
           MOV   R0,A                ; R0.0
           INC   DPTR
;--- tick ring tail pointer & check for empty
           MOV   A,DPL
           CJNE  A,#LOW RINGSIZE,L?newhead
           CLR   A                   ; reset to start of buffer
L?newhead  MOV   RingTail,A
           CLR   RingFull            ; at least one empty slot
           CJNE  A,RingHead,L?nohit ; head = tail: ring empty
           SETB  RingEmpty           ;  so mark it
L?nohit    SETB  EA                  ; OK to add new entries!

;--- push R2:R0 on BASIC's Argument stack
           MOV   A,#PUSHR2R0
           LCALL BasicOps
           SJMP  ToBasic

;--- ring is empty, so push return value = -1
L?empty    MOV   DPTR,#Minus1
           CALL  PushFPC
           SJMP  ToBasic

;--- step BASIC's text pointer to the end of the line
ToBasic    MOV   DPH,TextPtrHigh     ; set up BASIC text pointer
           MOV   DPL,TextPtrLow
L?nextchar MOVX  A,@DPTR             ; at CR?
           CJNE  A,#<CR,L?notcr
           SJMP  L?goback
L?notcr    CJNE  A,#COLON
L?notcolon                           ; at colon?
           SJMP  L?goback
L?notcolon INC   DPTR                ; nope, continue
           SJMP  L?nextchar
L?goback   MOV   TextPtrHigh,DPH     ; restore BASIC text ptr
           MOV   TextPtrLow,DPL
;--- if in command mode, restart the interpreter's READY mode
           JB    CmdMode,L?cmd
           RET                       ; simple RET in RUN mode
L?cmd      MOV   A,#COMMANDMODE
           LJMP  BasicOps            ; LJMP rather than LCALL
GetRandom  ENDPROC

;--------------
; Push floating-point number at DPTR in code space to Argument
;  stack.  Number is stored with exponent at first byte to
;  simplify addressing.

PushFPC    PROC
           MOV   A,ArgSP             ; make room on arg stack
           ADD   A,#-FPNUMSIZE
           MOV   ArgSP,A
           MOV   R0,A                ; ptr to strt of stack slot
           MOV   R1,#0               ; byte offset & counter
           MOV   P2,#HIGH ArgStack   ; page in Extexnal RAM
L?xfer     MOV   A,R1                ; byte offset
           MOVC  A,@A+DPTR           ; get byte from code space
           MOVX  @R0,A               ; drop onto stack
           DEC   R0                  ; tick target addr downward
           INC   R1                  ; tick offset
           CJNE  R1,#FPNUMSIZE,L?xfer; do for all bytes
           RET
PushFPC    ENDPROC
```

Listing 3c—*GETRANDOM is used to remove entries from the ring buffer and return them to BASIC.*

NEW WORDS

Unlike most computer languages, BASIC-52 allows you to add new keywords to perform application-specific tasks. The interface between the BASIC interpreter and your code is quite clean, but you have to get all the steps correct or it just won't work right.

You can think of the process as an intelligence test for your code: during the power-on reset, the BASIC interpreter makes sure that your code exists and can respond to requests. After it's qualified, BASIC skips the tests and callsitdirectly. Figure 3 diagrams everything you need to know to set this up; the source code is available on the BBS.

BASIC needs to know two things about each new keyword: the ASCII text of the keyword and the addressof the code that carries out the function. Both of these are stored in tables, so, once you get a single keyword functioning, it's easy to add more. There is an upper limit of 16 new keywords, but each of those can provide multiple functions.

`GetRandom` doesn't require any parameters, but your routines can examine the rest of the BASIC line and extract other information. BASIC includes functions to interpret expressions and return a value, so the BASIC source can include any valid BASIC variables, constants, or expressions.

If your new keyword needs to return a value, you can push it on the stack or store it in a known address above `MTOP`. Of course, if your routine has only "side effects" it doesn't have to return anything at all.

After pushing the result, `GetRandom` must adjust BASIC's text pointer variables to the end of the source code line and return control to the interpreter. These functions are required at the end of the routines that add new keywords to the BASIC-52 language, so you'll have to include them in your code regardless of what else it does.

Now, here's another gotcha. According to the manual, "when MCS BASIC-52 enters the command mode it will examine code memory location 2002H." Guess what happens if the initialization code detects a valid BASIC program in EPROM?

Exactly right.. .because the interpreter doesn't enter command mode, it never calls your routines that enable the new keywords, so the keywordsare not valid, so BASIC-52 reports a SYNTAX ERROR at your first new keyword!

Fortunately, the solution is trivial: if you set that bit during your reset code (in `ColdStart,` forexample), the extensions will be enabled. Your reset code must check for the program in EPROM and set the BASIC program pointers appropriately.

During startup

Code address 2002 = 5A?
No-extensions are not in effect

CALL routine at 2048
If it sets bit 25.5 hex, extensions are in effect

During BASIC execution

CALL routine at 2078
Sets DPTR to address of token/keyword table:

```
DB  10h        ; first token value
DB  "RANDOM"  ; first new keyword
DB  0          ; separator
DB  11h        ; second token value
DB  "ANOTHER";  second new keyword
DB  0FFh       ; table terminator
```

If keyword is found in table,
CALL routine at 2070
Sets DPTR to address of routine vector table:

```
DB  GetRandom;  address of RANDOM routine
DB  DoAnother;  address of ANOTHER routine
```

CALL address at table entry 2*(token value-10h)

Figure 3—*BASIC-52* keyword *extensions*

OVERLAID CODE

EXTRAND .A51 uses vectorsnear 2000h and 4000h, so it should require 16K bytes in code space. However, because it uses only a few hundred bytes, all of the BASIC-52 vectors and code will fit into a single 8K EPROM as long as it responds to both address ranges. This trick comes in handy if you have only an 8K EPROM emulator, as I do.

On the RTC52 boards, you need to install the jumpers that specify a 2764 EPROM, then two jumpers that select address ranges 2000h and 4000h. The EPROM is selected for both ranges, so references to 2002h and 4002h refer to the same address and return the same value. As the EPROM must be in code space, only socket U8 fills the bill; U9 holds an 8K or 32K RAM in data space.

The Parallax ROM Emulator software includes an "offset" value that specifies the EPROM's **starting address** in the target system. The program transfers only the hex values within 8K bytes of that starting address, so you must make two passes through the file with two different "offset" values. If you needed to do this on a regular basis, it would make sense to tinker up some conditional assem-

Tick	Program		Comments
1	BASRAND	0.005	Radium watch on top of RM-60 detector tube
2	CALLRAND	0.005	Radium watch on top of RM-60 detector tube
3	EXTRAND	0.001	Radium watch in front of RM-60 detector window
4	EXTRAND	0.001	Background radiation

1	2	3	4
0.765	0.020	0.037	9.213
0.165	0.500	0.020	4.947
0.870	0.130	0.017	3.226
0.625	0.495	0.057	0.397
2.310	0.780	0.259	6.255
1.060	0.540	0.016	6.921
0.295	0.530	0.142	4.721
0.735	0.170	0.051	1.187
0.535	0.315	0.024	1.362
0.255	0.030	0.011	17.738
1.505	0.830	0.033	3.768
1.140	0.945	0.207	1.092
1.015	0.890	0.042	1.229
0.755	0.100	0.031	2.535
0.905	0.320	0.005	3.114
0.525	0.365	0.011	10.772

Figure 4—*Results of the programs described in the text. The time intervals are in seconds. The downloadable file includes 1000 numbers in each set.*

bly statements to shuffle the segments around and get a single 8K hex file with all the code in the "wrapped" addresses.

IS IT RANDOM OR IS IT HOOEY?

Volume 2 of Knuth's *The Art of Computer Programming* contains an extensive summary of tests to separate random number generators into the good, the bad, and the awful. As he points out, the tests are mostly of use to people who are determined to put someone else's generator in the latter category...

I collected four sets of 1000 consecutive random numbers using the three programs under various conditions and ran a few simple tests using the Excel spreadsheet (see Figure 4). The raw data are recorded in a file available for downloading from the Circuit Cellar BBS; you can run more extensive tests if you feel the need to discredit my technique!

A histogram shows that the random intervals are not uniformly distributed; there is a definite peak around an "average" value,but you'll find all values betweenzero and huge. The "average" value depends on the radiation level, so dropping a radium watch on the detector will certainly change the characteristics of the random numbers.

From what little I remember from my courses in probability theory, the curve resembles a Poisson distribution. A quick thumb check of the text confirmed a suspicion I've had for awhile: I was considerably smarter in my younger days. If any of you have current experience with this stuff, do, please, take a look at the data and see if it makes any sense!

However, one thing is clear from my observations: unlike the fake random number generators you find in the other articles, Real Radioactive Randoms won't repeat no matter how long you wait... and that counts for something!

Ed Nisley is a member of the Circuit Cellar INK engineering staff and enjoys making gizmos do strange and wondrous things. He is, by turns, a beekeeper, bicyclist, Registered Professional Engineer, and amateur raconteur.

IRS

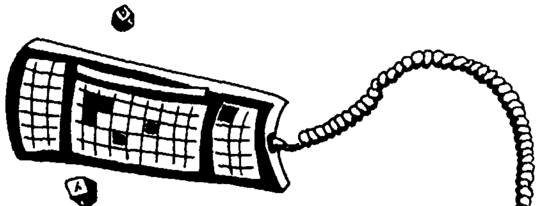
216 Very Useful
217 Moderately Useful
218 Not Useful

FROM THI BENCH

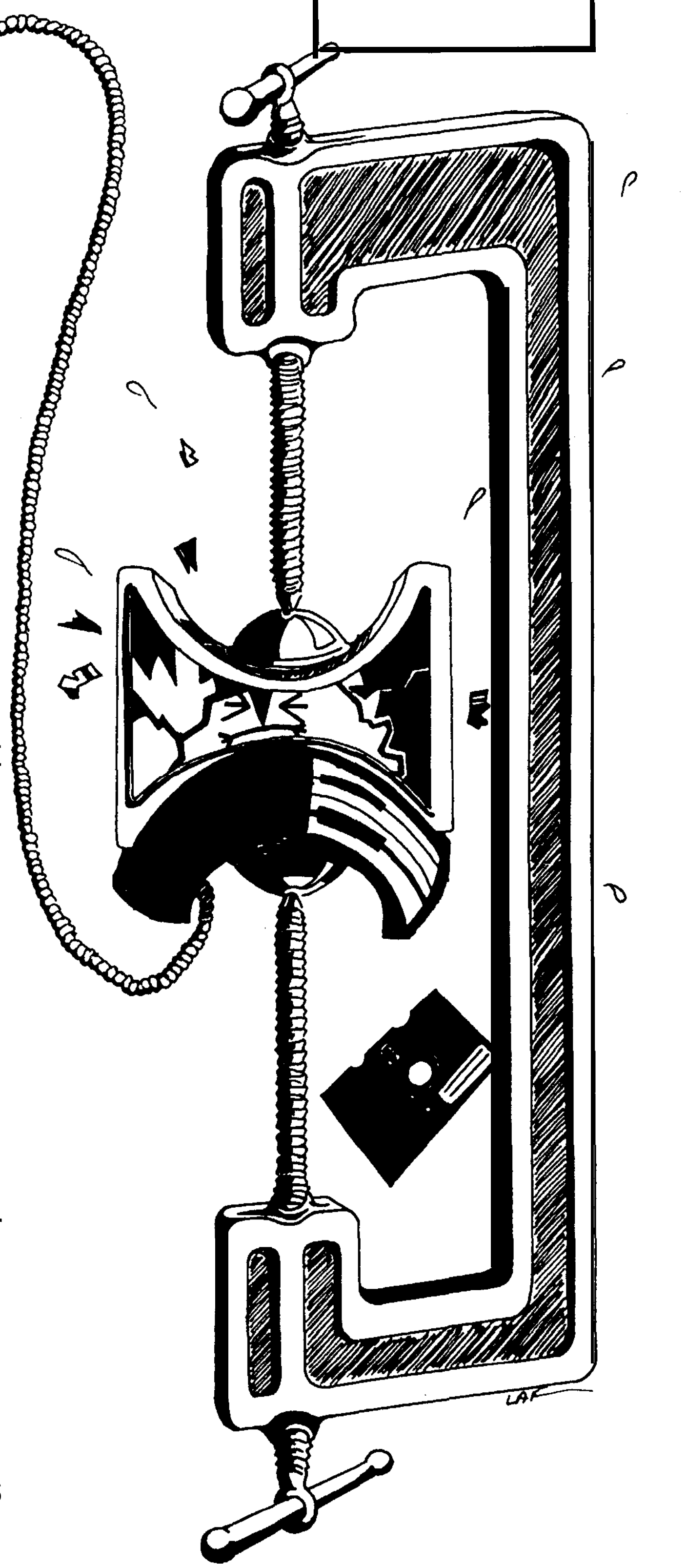
Jeff *Bachiochi*

Honey, I Shrunk the...

New Uses Abound for the Smallest AT-Clone Yet

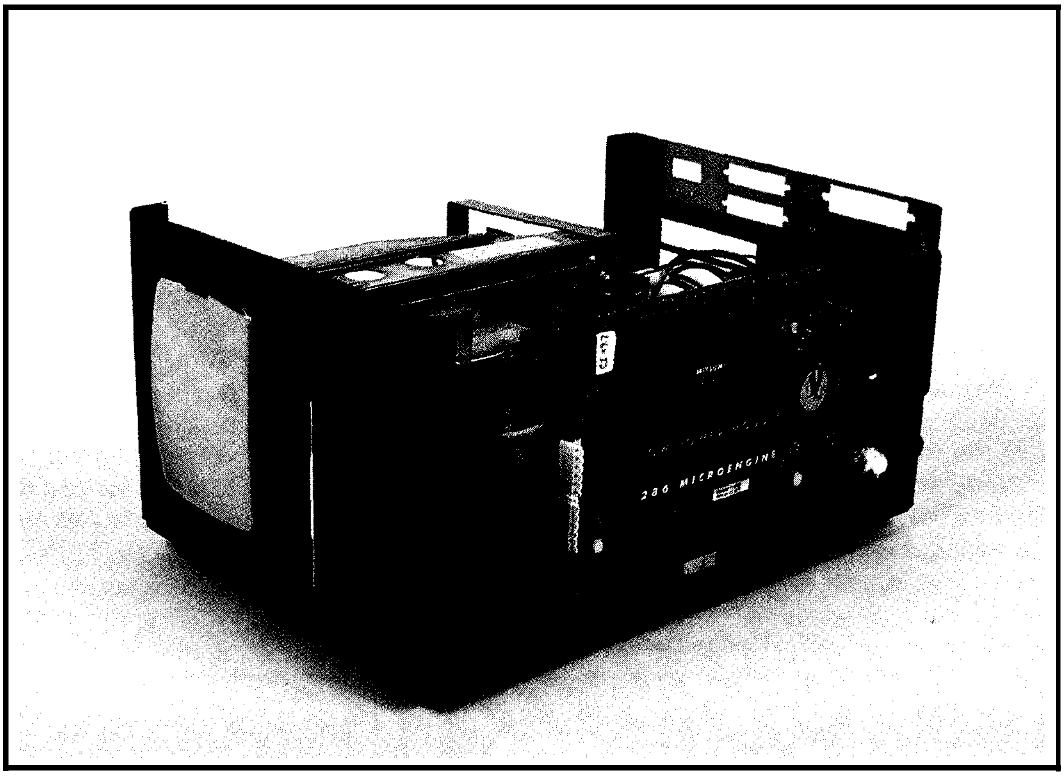


“Don’t you dare bring another piece of equipment into this house!” I knew those were the words lwould hear. I wouldn’t even chance it— Unless, I could somehow.. . ‘Naw, that couldn’t possibly work, or could it?” I mumbled to myself. It certainly was worth a try. Let’s see, I’ll start with this old Model I, The covering of dust which had settled like a blanket on my old friend was an indication of the last time I’d seen the TRSDOS sign-on message. “Look at all these single-sided diskettes,” I thought. “I could put ten of these on one high-density floppy now.”



I called to one of my sons for help, “Dan, help me carry this stuff out of here, OK?” I instructed him to use the front door and load the car. After complaining that the car was much closer to the back door, I explained there was a method to my madness. ‘If you use the front door Mom will see you.” ‘So what?” Dan argued. “She’s sure to wonder what we're doing," I answered back. "You see," I continued, "I don't think she's ever seen any equipment going out! It's sure to spark her curiosity."

I returned to the scene of the yet-to-be-committed crime and viewed the now-empty void, nestled right between the XT and the oscilloscope. This was practically the only free space left in the house. You see, we are about to add on to this crackerbox, since space is at such a premium.



The Home Monitoring System is made up of a Mitsumi 286 Microengine, ATI EGA Wonder, Seagate ST-02 floppy/hard disk controller, 3.5' floppy drive, 3.5' hard drive, Metrabyfe RS-485 serial interface. and an Audiotronics 5-inch monitor. all in an Integrand enclosure.

"This old vacuum should do the trick," I thought to myself. As I switched it on, a puff of something blew out the back, like starting a car that was badly in need of a ring job. The roar was deafening. I went right to work cleaning up what was left behind. You know the kind of stuff I'm talking about: bits 'n pieces of the cosmos. Even God couldn't explain how it got there.

It didn't take long for the curiosity to build to a level of investigation. "Wh...on... hmmf," I heard as my wife poked her head around the corner. I powered down the 01' sucker and asked "What?" as a final puff blew out of the vacuum, like a dying gasp. 'You killed it," she gleamed, certifying the need for a new one.

I knew this was it, time for that costly phrase. "Yeah, I **guess** you can start shopping around for a new one." Her face paled, as if hypnotized. "By the way," I added, "This is where I'm going to install our new Home Monitoring System." "That's nice," I heard her say, but I knew she meant, "Where's the checkbook?" We were both happy now.

ONE FOR ALL AND ALL IN ONE

The Home Monitoring System will take years to complete. In fact, it is one of those projects that just keeps on growing. Adding the extra wiring necessary for home monitoring is much easier to accomplish during the framing stage of building. Now is the time to prepare, before construction of the addition actually begins. Did I say addition? This is more like adding a house to an already existing garage.

Cramming things into small spaces has become the norm at our house. So naturally I'd like a home monitoring system which is compact yet powerful enough to allow software development and program maintenance. Rummaging through my flea market leftovers was easy; not much was left. It seems it's getting tougher and tougher to find good junk, er, equipment nowadays. I did find a 6-inch monitor in the heap, but it's not even composite video. "It'll need horizontal and vertical syncs," I thought. "Humph, no enclosures. Wait, the TRS-80 I just removed had two full-sized floppy drives mounted in an external box."

I carefully smuggled the appropriate equipment back into the house. Upon ripping out the old drives, I smiled as I noticed its built-in power supply. "Look at that, the monitor even fits." Now with a four-slot PC-style passive backplane, yeah, and 3.5-inch drives, that's it.

That's not it. There is no room for any full-length PC **boards** like the OEM-286. Rats! *[Editor's Note:The OEM-286 (CCAT) was presented in 'Garcia's Circuit Cellar"* in *the September/October 1987 issues of BYTE.1*

THE BIG SQUEEZE

Intel entered the miniaturized computer market in 1988 with their WILDCARD-88. An 80C88 PC/XT-compatible motherboard on a small 2x4-inch form factor. Unfortunately, DRAM and DRAM drivers were left off, making it fairly useless without a considerable amount of additional system design. The module uses a high-density 68-pin SIMM socket as an expansion connector.

Could size reduction of this scale be accomplished using today's technology to produce a complete 80286 engine?

ENTER, THE JAPANESE-SPONSORED AMERICANS

Here's a role reversal for you: A Japanese company, Mitsumi Electric, creating a California-based subsidiary, Mitsumi Technology Inc. or MTI, to employ Americans experienced in computers, software, and telecommunications. Their job: to develop new technology and products for the U.S. marketplace.

Upon visiting Japan, MTI was enlightened by the parent company's ability to produce miniaturized electronics. With this information, they set out to shrink the size of today's popular powerful computers into a package which could be embedded into industrial and consumer devices.

HYBRID SUBSTRATE FALLOUT

Integrated hybrid circuits consist of multiple semiconductor chips placed on the same substrate. This is similar to surface mounting chips without all the plastic and leads which normally surround each device. The final substrate, which can contain many individual chips, is then enclosed as one hybrid circuit. Each individual semiconductor chip, as it stands prior to bonding, can only be guaranteed for a 95% reliable yield. That's a 5% rejection rate. Not great, but it's only the beginning. The worst-case scenario would be something like this (assuming a 5% rejection rate): Build 100 substrates. If each substrate has one device on it you end up with five bad ones. If each substrate has two devices and the 10 possible bad devices are all on different substrates, you end up with 10 bad substrates out of 100. If each substrate has 20 devices on each and all bad devices are on different substrates, all the substrates will be bad. Notice how the problem compounds itself. These devices are not like ICs where you can simply take the bad ones out of their sockets and replace 'em.

In order to achieve a better yield, pretested devices are necessary. Since pretested devices are also prepackaged, their size is limited by the number of leads and the lead spacing. Fifty-thousandths of an inch lead spacing is standard on most surface mount components. That's twenty leads per inch, or 68 leads in a l-inch-square 80286 processor.

To produce products without using the hybrid approach at the substrate level, some special techniques are needed. In addition to multilayer glass epoxy boards using double-sided surface mounting techniques, special packaging to house some of the standard VLSI chips must be developed. This reduces package size but retains pretest and replacement criteria. With the new package size, some VLSI chips now require only one quarter of the

CIRCUIT CELLAR INK'S

ADVERTISER'S INDEX

Reader Service Number	Advertiser	Page Number
101	Alligator Technologies	50
102	Alpha Products	11
103	American Advantech	33
104	Andratech	77
105	Ariel Corp.	16
106	Avocet	C2
107	Berry Computers	41
108	Big Bang Software	80
109	Binary Technologies	35
110	Cabbage Cases	63
111	Catenary Systems	40
*	Ciarcia Design Works	86
*	Ciarcia's Circuit Cellar	68
112	Circuit Cellar	67
113	Circuit Cellar	65
114	Computerwise	63
115	Cottage Resources	31
116	Covox Inc.	77
117	Dalanco Spry	47
118	Davis Instruments (DIGITAR)	14
119	Dycor Industrial (DA/M)	64
120	Emerald Microware	57
121	Engineers Collaborative	37
122	Express Circuits	30
123	F&W Communications	45
124	Gott Electronics	27
125	Grammar Engine	80
126	GTEK	C4
127	Hazelwood	43
128	High Resolution Technologies	37
129	Information Modes	87
130	Innotec	7
131	Introl Corp	29
132	Komputerwerk	87
133	Laboratory Microsystems	84
134	Link Computer Graphics	47
135	Logical Systems	35
136	LS Electronics	83
137	LS Electronics	87
138	Magnum Opus	87
139	Meredith Instruments	13
140	Micro Dialects, Inc.	74
141	Micro Digital	72
142	Micro Resources	84
143	Micromint	17
144	Micromint	61
145	Micromint	69
146	Ming Engineering (Electronic 1 2 3)	7
147	Needham Electronics	81
148	Nisley Micro Engineering	87
149	NOHAU Corp	81
150	Paradigm Systems	75
151	Parallax, Inc.	51
152	Parks Associates	21
153	PC Boards	43
154	Peripheral Technology	74
155	PseudoCorp	32
156	R & D Electronics	75
157	Real Time Devices	19
158	Ryle Design	87
159	Sierra Systems	C3
160	Silicon Alley	26
161	SIMAR, Inc.	68
162	Timeline	4/5
•	Tinney	69
163	Traxel Laboratories Inc.	14
164	Universal Cross Assemblers	72
165	University of Pittsburgh	87
166	Unkel Software	53
167	URDA, Inc.	87
168	Witts Associates	87
169	Wytec	69

HAJAR ASSOCIATES
NATIONAL ADVERTISING SALES REPRESENTATIVES

NORTHEAST
Lisa D'Ambrosia
Tel: *(617)* 769-8950
Fax: *(617)* 769-8982

MID-ATLANTIC
Barbara Best
Tel: *(201)* 741-7744
Fax: *(201)* 741-6823

SOUTHEAST
Christa Collins
Tel: *(305)* 966-3939
Fax: *(305)* 985-8457

MIDWEST
Nanette Traetow
Tel: (708) 789-3080
Fax: (708) 789-3082

WEST COAST
Barbara Jones & Shelley Rainey
Tel: (714) 540-3554
Fax: (714) 540-7103

SUBSCRIPTION PROBLEMS?

If you have problems with your subscription (delayed or missing issues, change of address, or questions on renewals), call the Circuit Cellar INK Subscriber Service Line at (914) 628-0885 or write:

Circuit Cellar INK
Subscriber Service Dept.
P.O. Box 2099
Mahopac, NY 10541

IRS

INK Rating Service

How useful is this article?

At the end of each article and some features there are 3-digit numbers by which you can rate the article or feature.

Please take the time to let us, at Circuit Cellar INK, know how you feell our material rates with you. Just circle the numbers on the attatched card.

previous board space. Additional space is gained by using gate arrays.

Within three years of the initial investigations of manufacturing techniques, MTI released the "286 Microengine" shown in Photos la and lb. The Microengine contains all the major components of an 80286 motherboard, including 512K of DRAM, compressed into a 4 x 2.6-inch module. What's most interesting is that it is totally manufactured in the U.S. using a ZyMOS AT chipset and Award's AT BIOS. Although another BIOS and chipset may be used in the future, the end product will always be electrically compatible with the original module.

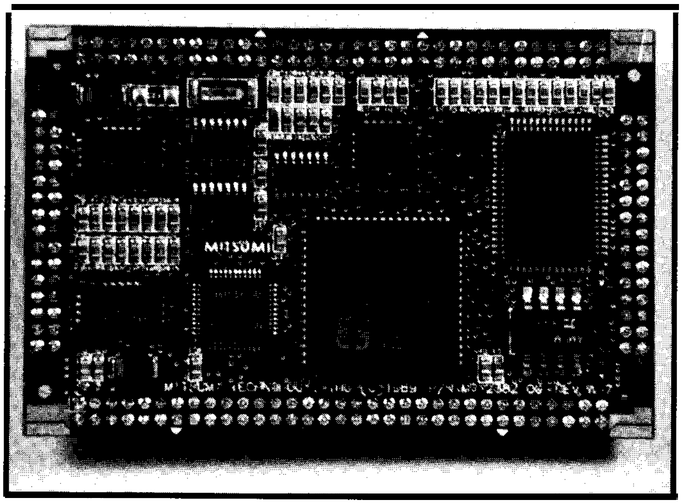


Photo 1 a—*The top side of the 286 Microenglne shows the 80286 processor, half of the ROM BIOS, DRAM parity bits, and assorted glue logic.*

Photo la shows the top surface of the 286 Microengine. The 80286 CPU is mounted on the top side of the module. It is the largest chip because it is used in its standard J-lead package. To the right of the processor is the only other component used in its original package: the digital delay line. Above the delay line, in the upper right corner of the board, is a custom VLSI chip holding the odds and ends logic of the system, such as RAS and CAS generation. It uses 0.040" lead spacing. To the left of this VLSI and above the CPU is a standard surface mount 512x8 PROM. Three standard TTL surface mount packages and the 32.76-kHz crystal for the real-time clock function reside to the upper left of the CPU. To the immediate left of the CPU is one of the BIOS ROMs mounted in special packaging with 0.033" lead spacing. On the extreme left of the module are two 256Kx1 surface-mount DRAM memory devices. These are used as the parity bits for the system's two 256K-byte banks of DRAM.

Photo lb shows the bottom surface of the Microengine. The two banks of DRAM are located on the far left. Four 256Kx4 surface mount DRAMs comprise the 512K bytes of system memory. The second BIOS ROM is the upper chip located next to the DRAMs. The Award BIOS includes the setup routines necessary for configuring the system without the need for a setup diskette. Below the second BIOS ROM is the 8042 keyboard controller, again in a special package to reduce real estate. Next to these devices are what has to be the most impressive job of packaging I've ever seen. The ZyMOS POACH chipset, which consists of four VLSI chips, contains the logic functions of all the microprocessor peripherals on a standard AT motherboard. POACH 1 includes two 8259As (master and slave programmable interrupt controllers), an 82284 (clock generator and ready interface), a 6818 (real-time clock), an 82288 (bus controller), plus miscellaneous control and interface logic. POACH 2 includes an 8284A (clock generator), an 8254 (programmable interrupt timer), a 74LS612 (memory mapper), two 8237s (byte and word DMA controllers), plus refresh, timing, and parity check logic. The two remaining POACH chips are identical. POACH 3 is a buffered interface chip capable of driving the AT-style expansion bus. One chip is used for the data and associated control lines and the other for the address and associated control lines. On a standard AT clone motherboard, these four chips take up about nine square inches of board space. Using MTI's special packaging, these chips take up about two square inches, mostly due to super dense, 0.020" lead spacings. The lead spacing (and space between leads) here is 0.010". It gives me a sense of scale when I remember that a human hair is about 0.001" in diameter.

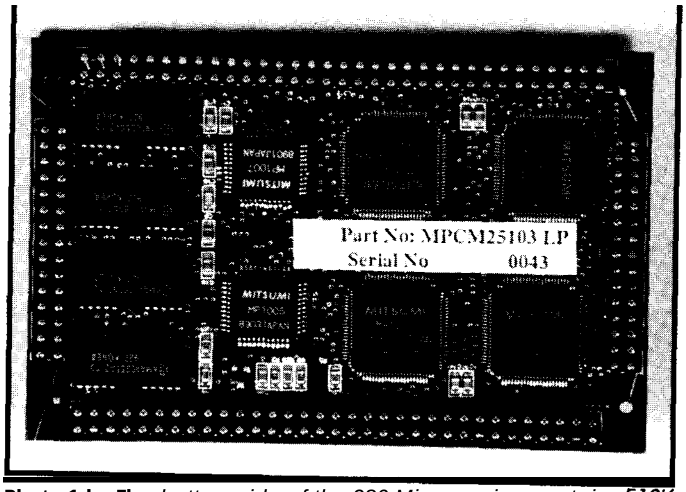


Photo 1 b--The *bottom side of the 286 Microengine contains 512K bytes of DRAM. the second half of the BIOS, the keyboard controller, and supporting POACH chips.*

The 286 Microengine has a total of 192 pins, giving the designer access to all of the internal buses as well as the connections needed for AT simulation, such as AT bus, keyboard, speaker, and so on. The module can be soldered directly to your embedded controller circuit board or plugged in using two 2x15 and two 2x33 standard strip sockets. This is the same spacing as standard 0.1-inch-on-center square-pin headers. The entire module is enclosed

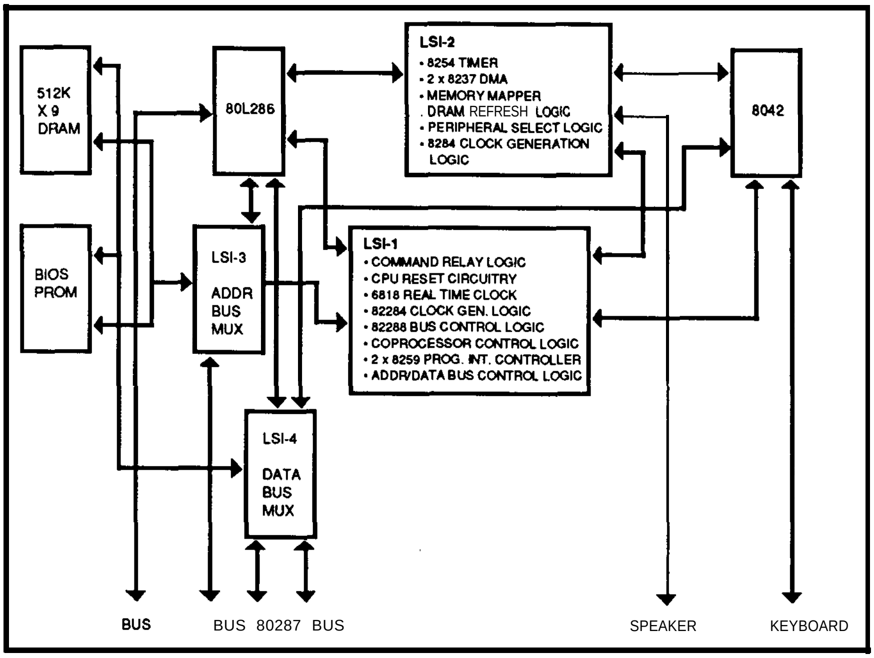


Figure 1 *-A block diagram of the Microengine 286 shows that the module contains all the necessary components to put a complete AT motherboard on a half-size expansion card.*

within a metal container to contain EMI and offer some heatsinking to the 80286 processor. Figure 1 shows a block diagram of the 286 Microengine.

To help engineers feel comfortable with the 286 Microengine, the MTI engineering team has a half-size Evaluation Card. This card contains AT-bus edge connectors and the necessary jumpers and clocks to simulate a complete IO-MHz 512K l-wait-state AT motherboard. The half-size card can drive an AT passive backplane just like the full size OEM-286 processor board.

Applications for the 286 Microengine such as PC-based workstations, laboratory control, and data acquisition equipment will be springing up left and right. Modules containing other functions such as video, serial, and parallel I/O are already under development. The 286 Microengine is competitively priced at $500 in single quantities. The evaluation card adds an extra $100. The 286 Microengine consumes less than 5 watts at 5 volts and its lower-power sibling uses less than 3 watts.

JUST THE RIGHT SIZE

As you can probably guess by now, I had an immediate application for the 286 Microengine. My choice of enclosures and desire to include an internal monitor capable of displaying EGA (for use with MC-Net; see Photo 2) determined the need. The enclosure is similar to the #2905WA from Integrand Research Corp. The display is an Audiotronics #900946-41, a 5-inch 12-volt monitor. The monitor is attached to an "L" bracket which bolts to the enclosure where a disk drive would normally go. A standard monochrome display card would drive the monitor fine, but I had to use an EGA Wonder card because it can drive a TTL monochrome monitor with the EGA video necessary to run MC-Net. As it stands now, until CEBus becomes a bit more established, the plan is to use MC-Net for the Home Monitoring System.

One of the smallest implementations of a combination floppy/hard disk controller card is the Seagate ST-02. This will support two SCSI hard drives and two floppies (360K/720K/1.2M/1.4M), although nothing larger than 3.5-inch drives fit here. The floppy drive is mounted to the opposite side of the monitor's "L" bracket, which allows the monitor and floppy to be installed as one unit. The bracket is made out of sheet aluminum which makes a good electrical and mechanical connection with the enclosure, eliminating interference between the two devices.

Any serial port card would do as an interface to an RS-485 adapter for MC-Net. However, MetraByte Corp. comes in with one of the smallest serial port cards that has RS-485 output.

The MTI Evaluation card completes the hardware necessary for this transportable AT. Although it probably won't be moved, it will establish a firm base for the Home

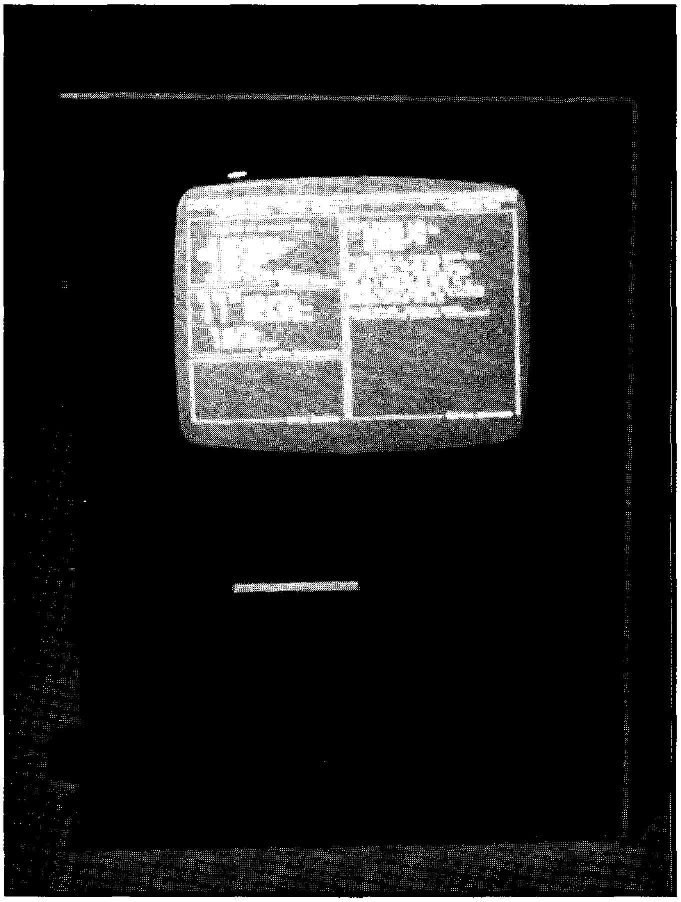


Photo 2—*Until CEBus* **becomes** *a bit more* **established,** *the plan is to use MC-Net for the Home Monitoring System.*

Monitoring System. All of the half-size cards fit easily into the old double disk drive enclosure. Bring on the carpenters!

Now I will be able to run MC-Net and up to thirty-one independent microcontroller nodes to implement the distributed Home Monitoring System. Or, let's see, maybe embed the 286 Microengine module as an RTC286 processor board. This is like playing with building blocks all over again!

Jeff Bachiochi (pronounced "BAH-key-AH-key") is a member of the *Circuit Cellar INK engineering* staff. *His background includes work in both the electronic engineering and manufacturingfields. In his spare time, Jeff enjoys his family, windsurfing, and pizza.*

SOURCES

MC-Net
Micromint, Inc.
4 Park St.
Vernon, CT 06066
(203) 871-6170

286 Microengine
Mitsumi Technology, Inc.
3295 Scott Blvd.
Santa Clara, CA
(800) 980-5400

Wildcard-88
Intel Literature
Dept. W-472
Santa Clara, CA
(800) 548-4725

Enclosures
Integrand Research Corp.
8620 Roosevelt Ave.
Visalia, CA
(209) 651-1203

RS-485 Serial Card
MetraByte Corp.
440 Myles Standish Blvd.
Taunton, MA 02780
(617) 880-3000

ST-02 Hard Disk
Seagate Technology
920 Disc Drive
Scotts Valley, CA
(408) 438-6550

Correction to "Prom the Bench" #13, "Building an LED Moving Message Display":

The following items are available from

Circuit Cellar Kits
P.O. Box 772
Rockville, CT 06066
(203) 875.2751

1. Blank PC board, manual, and demo software on 5.25" 360K PC-format disk. SD-1 .... $49
2. Eight Sprague UCN5801A driver chips. SD-2. . . . . $30
3. Eight LPT2058AE red 5x8 LED array modules. SD-3 . . $50

please add $3 shipping and handling in U.S.; $8 elsewhere.

IRS

219 Very Useful
220 Moderately Useful
221 Not Useful

SILICON UPDATE

Tom Cantrell

Whither Zilog?

A Roller Coaster on the Back of the Z80

The heroes and villains of Silicon Valley sometimes seem to be drawn from the pages of a fairy tale.. .or a soap opera. The only difference is the source of the wealth that drives the intrigue: Instead of the Golden Goose or oil wells, it all starts with sand.

Zilog is just one story in the "Valley of the Heart's Delight." But, over the course of the company's history we get to see all the ups and downs that, like earthquakes, make life around here interesting.

IN THE BEGINNING

Zilog was founded in the mid-70s by a couple of Intel guys (Federico Faggin and Ralph Ungermann) with a better idea. They scratched together some venture capital, made a foundry deal with Mostek and voilà!—the Z80 was born.

The chip was an instant hit. It combined the software popularity of Intel's 8080 with the 5V-only, single-chip (no clock generator, bus controller, or interrupt controller ICs required) advantages of Motorola's 6800.

The company quickly followed on with the necessary peripheral ICs: SIO, PIO,CTC, and so on. These chips were quick winners too. In fact, many who weren't convinced of the merits of the Zilog CPU were often swayed by the capabilities of these peripheral ICs with which it worked. It was a high-tech version of the proverbial tail wagging the dog.

By the end of the 1970s, Zilog had become, along with the much bigger Intel and Motorola, a powerhouse in the microprocessor world.

TROUBLE IN PARADISE

In retrospect, what befell Zilog in the early 1980s was as much bad luck as a bad business decision. At the time, despite the ongoing shipment of millions of 8-bit CPUs, Intel, Motorola, and Zilog all adopted the conventional wisdom that the 8-bit world was dead, and that all existing customers would quickly move to the new 16-bit chips (the Zilog Z8000, Motorola 68000 and Intel 8086). So began the 16-bit wars, a bitterly fought campaignbetween the giants.

Arguably, the 28000 was a good chip. Some say it lost because it was a little late. Others point to the competitors' marketing strength. I think the main problem was the premise that everyone needed a 16-bit chip. In fact, until the emergence of the PC and Mac, there was little 16-bit business for anybody.

Meanwhile, Zilog, under the tutelage of former investor and then owner Exxon, continued to lose focus (and money) dabbling in everything from RAMs to UNIX boxes.

Too late, the company tried to retrench in the 8-bit world with the infamous Z800. The chip was intended to offer a high-performance alternative for existing Z80 customers. Unfortunately, management and personnel turnover had reached the point that completing the design became impossible. New engineers would leave even before they got up to speed on what the previous engineer had done. The final straw was the conclusion late in the game that the Z800 should be a CMOS, not NMOS part. Back to the drawing board again.. .

The mid-80s were not a happy time at Zilog, with semiconductor market slow-downs, little28000 business, management turmoil, and the 2800 faux pas. If the story stopped here, the title of this article might be "Remember Zilog?"

	Package	#I/O Lines	RAM	ROM
Z86C09	18 pins	14 lines	124 bytes	2K bytes
Z86C19	18 pins	14 lines	124 bytes	4K bytes
Z86C30	28 pins	24 lines	236 bytes	4K bytes
Z86C40	40 & 44 pins	32 lines	236 bytes	4K bytes
Z86C90	40 & 44 pins	32 lines	236 bytes	OK bytes

Figure 1 -2ilog's new *CCP chips differ mostly in pin count, number of I/O lines. and amount of on-board memory.*

CONTROLLERS FOR THE MASSES

One semibright spot during the dark days was the development of the Z8 8-bit single-chip computer. As so often is the case for Zilog, the part itself was a competitive, if not arguably superior, design. However, independent of the Z8's merits, Zilog had to swim upstream in the face of the earlier, dominant Intel (8051) and Motorola (6801) single-chips. To the company's credit, they stuck with the Z8 through the tough times and managed to build a respectable business.

Now, coincident with the departure of Exxon, the newly "privatized" Zilog has combined the Z8 architecture with their latest CMOS process to make a lineup of controllers-the Consumer Controller Processors (CCP)—that are notable for their low cost and minuscule power consumption. The 1.6-micron CMOS process is a life saver for Zilog, since it is apparent that one cannot hope to be a major player in the embedded control market without its low power and reliability.

As shown in Figure 1, the differentiation between the CCP family members is largely one of package size, ROM (sorry, no EPROM yet), and RAM (more accurately, register banks). Otherwise, all members share a common set of I/O functions including two 8-bit counter/timers, two analog comparator channels (two pins' voltages compared with a third), and a watchdog timer (which automatically resets the CPU if things don't seem right). A cost-saving feature is the ability to choose between various clock generation alternatives. Less expensive circuits including RC, LC, and ceramic resonator may be used instead of a crystal.

The hallmark of CMOS is low power consumption and the CCPs fill the bill. Active power consumption (Vcc = 5V @ 8 MHz) for the lineup ranges from 10 to 20 mA (more pins translates to more power), while the lowest of the low-power modes (STOP mode) consumes a thousand times less (10–20 µA). Carrying the virtues of CMOS to the extreme, Zilog specs the CCPs to operate with Vcc as low as 2.75 V, which further cuts power consumption and is great for battery-driven applications. However, check to make sure the CCP low-Vcc I/O levels meet your needs.

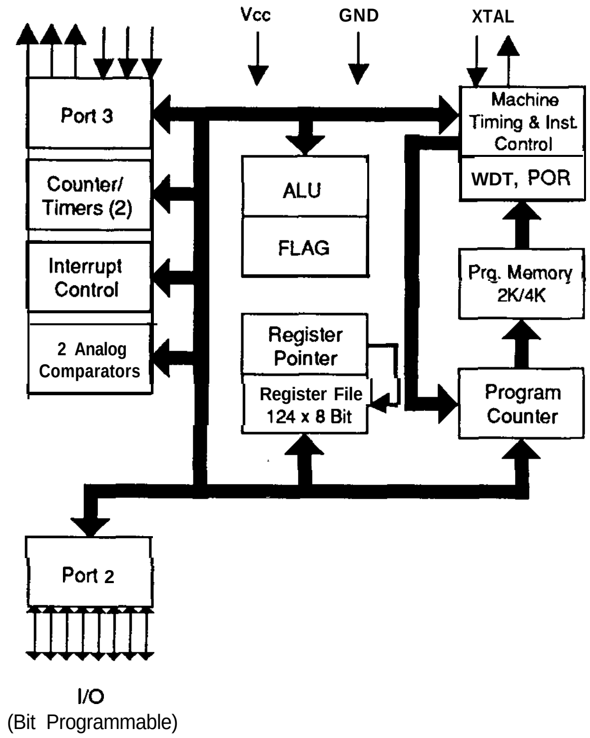
I find especially intriguing the smallest family member (the Z86C09/19 shown in Figure 2). It comes in an 18-pin DIP! Best of all, the parts have tiny prices too: As low as $1.50 in high volumes.

With this packaging and pricing, the CCPs are truly targeted at consumer markets (I mean toasters and toys). Watch out 4-bitters, the CCPs are breathing down your neck.

Z80 DÉJÀ VU...

The latest Zilog offerings come full circle. Will they bring the success of the heady days when the Z80 was king?

The Z84013, Z84C13, Z84015, and Z84C15 are high-integration chips which combine, using Zilog's "Superin-



Pin Identification

Pin No.	Symbol	Function	Direction
I-4	P24-7	Port 2 pin 4, 5, 6, 7	In/Output
5	Vcc	Power Supply	Input
6	XTAL2	Crystal Oscillator Clock	output
7	XTAL1	Crystal Oscillator Clock	Input
8-10	P31-3	Port 3 pin 1, 2, 3	Fixed Input
11-13	P34-6	Port 3 pin 4, 5, 6	Fixed Output
14	GND	Ground	Input
15-18	P20-3	Port 2 pin 0, 1, 2, 3	In/Output

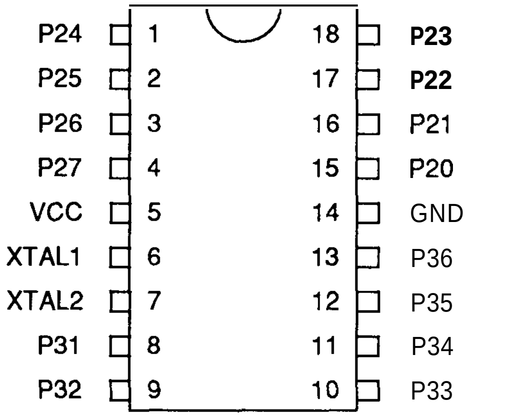
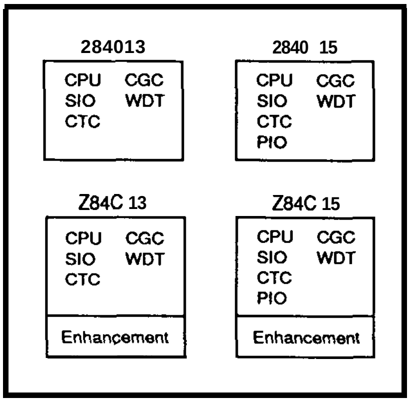


Figure 2—*The Z86C09/19 is a complete 8-bit microcontroller in an Id-pin DIP package.*

Figure 3—*Four of Zilog's new* **high-integration chips include a whole lot** *more* **than just** *a micro-processor.*



tegration" standard cell design technology, the Z80 CPU with other standard peripheral functions. In addition to the quality of their design, there's good news in the fact that they, like the CCPs, take advantage of Zilog's fast (10-MHz) low-power CMOS process. Furthermore, the CPUs exploit modern high pin-count packaging: PLCC (84-pin Plastic Leaded Chip Carrier) and QFP (100-pin Quad Flat Pack).

The advanced design, manufacturing, and packaging skills yield the lineup of chips shown in Figure 3. The four new parts are permutations of two features: PIO (parallel I/O in the "15s") and enhancements (in the "Cs").

Most of the core function-the Z80 CPU, SIO, CTC, and (for the 15s) PIO- work exactly the same as the earlier nonintegrated (and very popular) chips of the same name. In fact, the block diagram (Figure 4) shows that the new Superintegrated chips are just like multichip system designs of old (note the on-chip address, data, and status/control buses) shrunk onto a single die. Besides making the **chip** design easy, another advantage of this straightforward approach is that software drivers written for older multichip systems port easily to the newly integrated equivalents.

Other functions added include a clock generator (CGC) and watchdog timer (WDT). The watchdog timer is the same core that is used in the CCPs. Also like the CCPs, low-power operation modes (IDLE1, IDLE2, and STOP) are provided.

The enhancements embodied in the "C" versions encompass a variety of add-ons such as power-on reset, wait state generator, two chip-select lines, and 32-bit CRC hardware for the SIO. Together, they serve to slash TTL from, and boost performance of, a minimal system. For most applications, the enhancements probably justify the "C" version's 20% higher asking price.

Speaking of prices, the family ranges from $9 (284013) to $13 (Z84C15) in 1000-piece quantities. Not bad at all considering that these days you can blow $100s ($1000s?) for a RISC chip set and still end up short of cache!

BACK TO THE FUTURE...

As we enter the 1990s, Zilog's sales level, profitability, and 8-bit focus are much the same as 10 years ago. A key difference is that now everyone, including Zilog, recognizes 8 bits will live forever. Zilog gets another chance, as if the '80s were just a bad dream.

Make no mistake, it won't be easy for a $100,000,000 company like Zilog to compete with today's billion dollar class giants. In particular, they must have the means and the will to match the leaders' R&D and manufacturing investments. This time around, Zilog had better stay focused on meat-and-potatoes products; no "Z8,000,000 targeted at the projected-to-be-exploding Personal Mainframe market."

Based on the strength of the new CCPs and high-integration Z80s, I say that Zilog will not only survive, but has a good chance of making a **strong** comeback. The story isn't over yet..

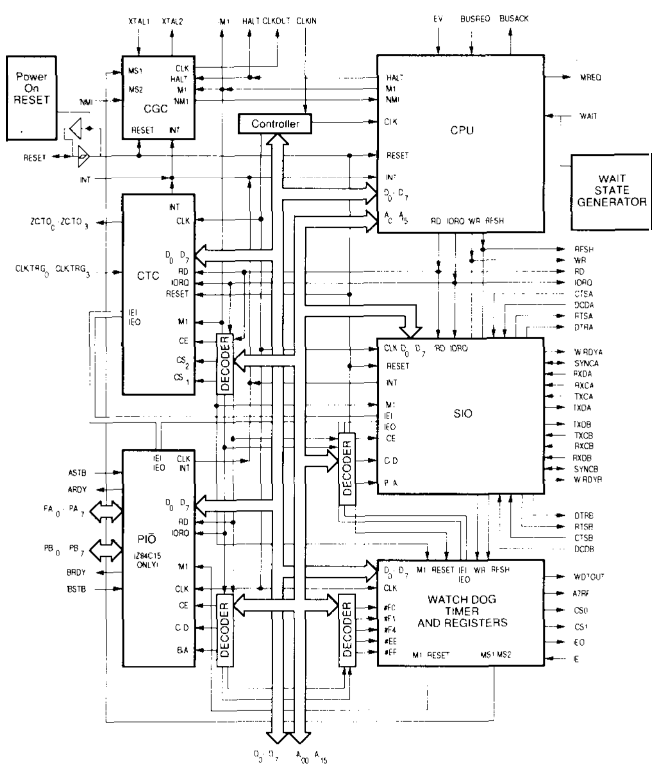


Figure 4—*Zilog's new Superintegrated chips are just like multichip system designs of old shrunk onto a single die.*

Contact

Zilog, Inc.
210 Hacienda Ave.
Campbell, CA 95008-6609
(408) 370-8000

Tom Cantrell holds a B.A. in economics and an M.B.A. from UCLA. He owns and operates Microfuture Inc., and has been in Silicon Valley for 10 years involved in chip, board, and system design and marketing.

IRS

222 Very Useful
223 Moderately Useful
224 Not Useful

CONNEC-TIME

Excerpts from the *Circuit Cellar* BBS

Conducted by ***Ken Davidson***

The Circuit Cellar BBS
300/1200/2400 bps
24 hours/7 days a week
(203) 871-1988
Four Incoming Lines
Vernon, Connecticut

We have just one message thread this issue, so we'll make it a good one. Telephone circuits always seem to be popular, and detecting call-progress tones is something many applications require. There are numerous methods that can be used to accomplish the goal, some better than others.

Msg#:23806
From: DAWN R. BANKS To: ALL USERS

I've been trying to put together a circuit to do telephone call progress monitoring. So far, I've bought a DTMF decoder chip at the local Radio Shack, and that gives me digits. Next, I wired up four 567s to give me dial tone, ringback, and busy. After this, I start running down a rathole. First, I see that if I want to detect that loud "you left the phone off hook" noise, I'm going to have to detect four simultaneous frequencies, which don't seem to overlap anywhere with those that I'm already looking for. After that, I also find that if I want to listen for that "bodo-weep" for an out-of-service phone number, I'm running into still more frequencies to detect.

Before I get done with this, I can see having a whole room full of PLLs, each detecting one of the frequencies, and each incorrectly detecting two or three of the others. The question that this leads up to is: Is there really any better way of doing this? Is there some off-the-shelf DSP chip that I could easily adapt to my needs, or am I completely out of luck here? I would happily accept any other constructive suggestions, including those that indicate that I'm wasting my time completely. Keep it simple, because I'm not really a hardware type, and this project is just me pretending that I am. Thanks.

Msg#:23821
From: NATHAN ENGLE To: DAWN R. BANKS

Call progress is one of those sorts of things that gets harder to do as your requirements get closer to never giving you an error.

Part of the problem if you're trying to design a product that is going to be sold all over the place (I know you're not, but just bear with me) you find that telco central offices tend to use a very wide variety of cadences, and so on for different call progress tones. So trying to squeeze that last 5%-of-the-time error condition out of your design can be really hard.

One tactic that you might want to consider is to back off from the idea of trying to detect the "phone left off hook" tone, and just use a timer to hang up the phone after a given wait. I have to admit that it isn't a lot of good if you're doing something like an automated telemarketing box that has to know when the disgusted party at the other end of the line has cut your sales pitch.

Msg#:23828
From: DAVE EWEN To: DAWN R. BANKS

If this system is based on a PC or something with computational ability, you might consider an ADC to sample the tones and then take the data and perform an FFT to convert to the frequency domain.

Msg#:23854
From: DAWN R. BANKS To: NATHAN ENGLE

Thanks for the ideas. Yes, it will be connected to something with some computing power (a Z8), so the ADC is a point worth considering.

Question: Am I going to get into some kind of hairy FFTs when I'm detecting something that's more than one tone (such as DTMF, dial tone, etc.)? The next question I have concerns call completion; that is, what do I listen for? Do I get a "wink" on the phone line, or is there something more involved? As a related question, how do I differentiate this "call disconnection" signal from the "call waiting" signal? (I ask this because other devices, such as phone answering machines and other RS boxes, seem to have difficulty with this distinction.)

From this, I guess I'll just stick with the basic four-tone decode that gives me dial tone, ringback, and busy. I should probably see a dial tone before I see the "phone off hook" noise anyway. Thanks again (suggestions still happily accepted).

Msg#:23889
From: KEN DAVIDSON To: DAWN R. BANKS

Or you could just use a call-progress-tone detector chip. Teltone makes a few (M-980, M-981, M-982, and M-984). Silicon Systems second sources most of them as the SSI 980, SSI 981, and SSI 982.

Contact the companies and get data for the chips. I think it'll probably be easier than using a DSP to do FFTs or whatever on the incoming signal...

Teltone Corp.
P.O. Box 657
10801120th Ave. N.E.
Kirkland, WA 98033-0657
(206) 827-9626

Silicon Systems, Inc.
14351 Myford Rd.
Tustin, CA 92680
(714) 731-7110

Msg#:23899
From: BOB PADDOCK To: DAWN R. BANKS

Mite1 also makes call progress parts. Try (619) 2763421, (408) 249-2111, or (312) 574-3930, whichever is closer.

If you're trying to find out if the other end of a long-distance call has hung up, maybe you can detect the short 'beep" you hear at the end of the call. It is a 2600-Hz tone, called "line idle." Problem would be to not have voice detected as this tone.

Msg#:23919
From: DAWN R. BANKS To: KEN DAVIDSON

Thanks. This is exactly what I was hoping someone would suggest (just didn't know what to ask for).

Since my sources here in New England for electronic parts seem to be a bit limited, any suggestions as to where I could obtain a couple of these (assuming that I've successfully laid hands on the documentation)? Mail order sources?

Msg#:23930
From: KEN DAVIDSON To: DAWN R. BANKS

It's not likely you'll find either Teltone, SSI, or Mite1 at any surplus mail order place. About the only way to proceed is to contact each company directly and find out where the nearest distributor is, then ask them about getting some.

Msg#:24042
From: ERIC BOHLMAN To: DAWN R. BANKS

Teltone is pretty good about giving out engineering samples; in fact, I was able to get an M-982, which is a pretty good call-progress detector chip. I think they run about $12 in small quantities.

Signals to indicate 1) when a call has been connected and 2) when the party on the other end has hung up are far from standard. Folk wisdom says that the polarity of your line reverses briefly when the other party answers; my experience has been that that's true only if you're on a step-by-step exchange. You also can't count on getting a burst of 2600-Hz at the end of a connection, since that's an artifact of using trunks with SF signalling, and not all connections are routed over such trunks.

One further note: if you're just interested in detecting dial tone, try using a DTMF detector like the SSI 202, but use a 1.3-MHz crystal instead of the usual 3.58. Dial tone should then show up as the "*" output from the detector.

Msg#:24057
From: DAWN R. BANKS To: ERIC BOHLMAN

Thanks for the input. I was just looking at an "IC Master," which seems to be the only reference source I can find for any of this stuff. Obviously, I can get this information from the manufacturer, but I was hoping to find the differences between the 980/981/982/984. From the sound of things, the 982 does pretty much what my set of four 567s does, albeit in a much smaller package, and probably better. (That is, the IC Master said it detected 350, 440, 480 and 620.) The short description for the 984 said it also detected other frequencies, although there wasn't any description of which.

I think I'm going to try to get a couple of 984s. Another intriguing-looking chip is the SSI 20C90, which purports to be a DTMF transceiver with call progress indication. Again, this is all I know about the chip.

The most recent reason I'm so hot on using one of these chips is that although I can detect most of what's advertised with my little breadboard of a DTMF decoder and those four 567s, I can hear one of the chip's oscillators on the phone line it's plugged into. I didn't start hearing it until I started adding the 567s, so I suspect it's just shoddy design on my part, and not the SSI 202 DTMF decoder chip. I have the phone line isolated with a 600/600-ohm transformer (with DC-blocking capacitor on input, and the 567s are behind a 0.1-uF cap on the other side). I'm hoping that a chip designed for this isn't going to bother me with as many of these problems.

As for call termination status: I'm not really looking for a general solution here that'11 work everywhere. In fact, all I'm interested in is something that'11 work on the phone exchange here at home. To that end, I've been meaning to put a scope on the phone line and watch what happens at the end of a phone call.

Msg#:24059
From: ERIC BOHLMAN To: DAWN R. BANKS

You ought to get hold of the databooks from Silicon Systems and Teltone. The SSI 20C90 has only a limited call progress detector; it just gives you an output when energy is present in the 305- to 640-Hz band. I'm not quite sure of the differences between the M-981 and M-982; they seem to be pin compatible. The 984 is optimized for detecting the special tones; it can't discriminate between dial tone, ringback, and busy. Oh, I just found out the difference between the 981 and 982: the 982 can detect 620 Hz, whereas the 981 replaces that channel with 400 Hz.

Msg#:24084
From: DAWN R. BANKS To: ERIC BOHLMAN

Well, it would seem that I'm not going to find a one-stop solution. I had already found the differencesbetween the980, 981, and 982. I was kind of hoping that the 984 would be a superset of the 982, but I guessed wrong. Oh well...982 it is, I suppose.

The Circuit Cellar BBS runs on a IO-MHz Micromint OEM-286 IBM PC/AT-compatible computer using the multiline version of The Bread Board System (TBBS 2.1M) and currently has four modems connected. We invite you to call and exchange ideas with other Circuit Cellar readers. It is available 24 hours a day and can be reached at (203) 871-l 988. Set your modem for 8 data bits, 1 stop bit, and either 300, 1200, or 2400 bps.

IRS

225 Very Useful
226 Moderately Useful
227 Not Useful

SOFTWARE **and BBS AVAILABLE on DISK**

Software on Disk

Software for the articles in this issue of Circuit Cellar INK may be downloaded free of charge from the Circuit Cellar BBS. For those unable to download files, they are also available on one 360K, 5.25" IBM PC-format disk for only $12.

Circuit Cellar BBS on Disk

Every month, hundreds of information-filled messages are posted on the Circuit Cellar BBS by people from all walks of life. For those who can't log on as often as they'd like, the text of the public message areas is available on disk in two-month installments. Each installment comes on three 360K, 5.25" IBM PC-format disks and costs just $15. The installment for this issue of INK (April/May 1990) includes all public messages posted during January and February,1990.

To order either Software on Disk or Circuit Cellar BBS on Disk, send check or money order to:

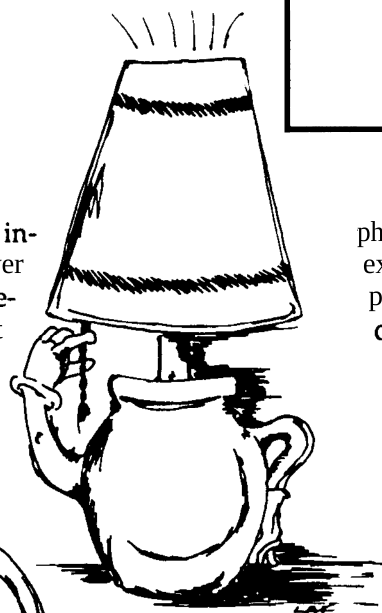
Circuit Cellar INK — Software (or BBS) on Disk
P.O. Box 772, Vernon, **CT** 06066

or use your MasterCard or Visa and call (203) 875-2199. Be sure to specify the issue number of each disk you order.

CEBus Comes One Step Closer to Reality

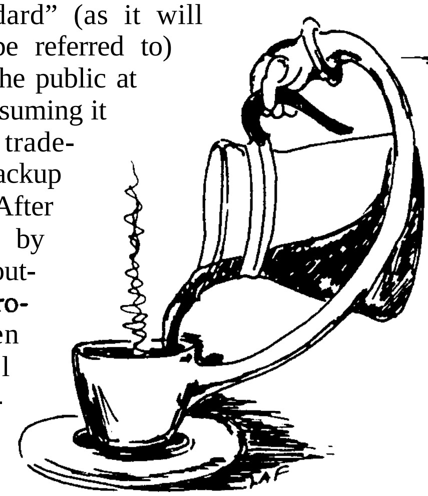
DOMESTIC AUTOMATION

Ken Davidson



Anyone following the home automation industry has no doubt heard of CEBus. Well, after over five years of work, we're finally about to see something official come down the line. I recently got back from the November EIA/CEG CEBus Committee meeting in Sunnyvale and have some interesting events to report.

First of all, we should dispense with all this talk about "CEBus." The "EIA Home Automation Standard" (as it will always correctly be referred to) will be known to the public at large as "Synq" (assuming it gets through the trademark search; the backup is "Harmonet"). After extensive research by both EIA and an outside consultant produced a list of a dozen or so potential names for the standard, EIA settled on this one. No, it doesn't stand for anything. It's easy to say, conveys an aura of high tech with the "Q" at the end, and should be just what the marketing boys are looking for. Please forgive me if I continue to refer to the standard as CEBus for the time being.



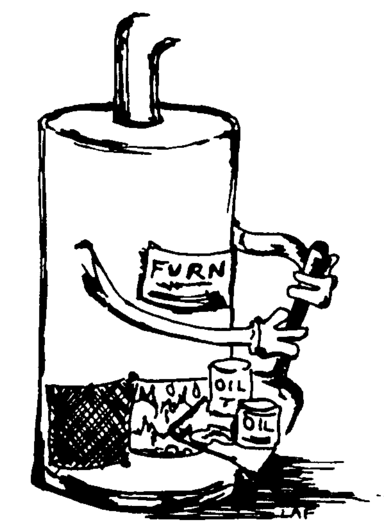
The long-anticipated event, though, was the official release of several portions of the CEBus specification for comment. Before getting into exactly what was released, let me explain EIA's procedure for releasing a specification,

EIA STANDARDS MAKING

When the committee was first put together, several working groups were established to hammer out the details of individual portions of the spec. For example, the language Working Group (LWG) is responsible for the upper network layers including CAL. The Power Line Working Group (PLWG) is responsible for the power line physical layer details. Similar groups exist for the other physical media plus conformance, publicity, and so on.

In charge of all the working groups is the Technical Steering Committee (TSC). The TSC oversees the entire specification-making process, sets policy guidelines, and has the final say in any matter regarding the spec.

Once a working group has a portion of the specification that they feel is ready for public comment, they present it to the TSC for approval. Once approved, the proposed spec is published and goes out for a comment period. During the comment period, anybody who would like to review the proposed spec and make comments on it is free to do so. The intention is to get the spec into the hands of engineers who work for companies that are members of the committee, but haven't necessarily been able to attend meetings. However, anybody who takes the initiative to obtain a copy of the spec may comment. Positive comments are always welcome, but don't affect any final decisions. Negative comments must be accompanied by supporting arguments, and may also include alternative ideas. Negative comments without supporting arguments are ignored. Each comment is acknowledged by the TSC, though they are under no obligation to take action on the comment.



Once all comments have been received and acknowledged, any significant changes that the TSC makes to the spec as a result of the comments are sent out for comment (rather than the whole spec). This cycle continues until everyone is happy with the specification. At that point, the document becomes a interim EIA specification that companies can be comfortable in using to put together product.

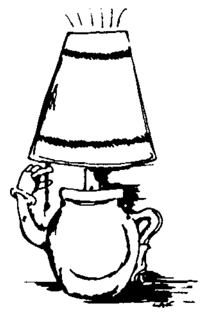
At the November committee meeting, the LWG and PLWG hammered out several last minute details and submitted the power line physical layer (PLBus), the data link layer (which is made up of the node medium access control sublayer [MAC] and the node logical link control sublayer [LLC]), the node network layer, the node application layer, and CAL to the TSC for approval. The TSC gave such approval, so the power line physical layer and all the other network layers have gone out for comment.

The current schedule calls for the initial comment period to close at the end of April, at which time the committee will take a careful look at all the comments and make any necessary changes.

Once the CEBus spec has graduated from being a proposed spec to an interim spec, it will be known as IS60. Once adopted as an official specification, it will be EIA600.

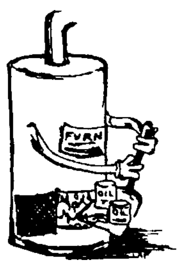
To get your own copy of the proposed CEBus specification and have the opportunity to influence the future of homeautomation, see theinformationboxat the end of the column. EIA is encouraging engineers to scrutinize the proposed spec and make constructive comments. I'm sure they'd love to hear from you.

MORE CEBUS HARDWARE



In issue #10 of CIRCUIT CELLAR INK, I described in my CEBus overview article two hardware implementations of CEBus that are available for engineersinterested in **embedding CEBus** in an upcoming product. Texas Instruments has developed a pair of chips that make implementing a CEBus interface much easier and have a new-generation evaluation board available that uses those chips. The SEM300 (I talked about the SEM200 in the overview article) uses TI's new SN75C080 CEBus controller chip and the SN75081 powerline modem chip to implement a complete PLBus CEBus interface that can be attached to switches, lights, or a processor for smarter control. They also have software available that allowsmonitoringof network traffic, and sending and receiving of packets. Contact TI for more information.

OVERSEAS



In developments overseas, the European community is about to officially announce D^2B to the world. D^2B is an international standard for control and communication for audio/video devices. Expect to see stereos and VCRs starting to show up in the trade shows in the months ahead sporting D^2B interfaces.

WRAP UP

I also just got home from Las Vegas where I attended the Winter Consumer Electronics Show (WCES). Though I didn't see as much as I'd hoped aimed at the home control market, there was enough to be interesting. In the coming months, I'll be going into more detail about what I saw and where the market seems to be headed. And, of course, as soon as we get our hands on some hardware, especially actual chips, we'll be putting it through its paces and showing it to you in these pages. Stay tuned...+

Sources

EIA CEBus Proposed Specification ($35)
EIA Standards Sales Dept.
1722 Eye St. NW
Washington, DC 20006

SEM300
Texas Instruments, Inc.
P.O. Box 809066
Dallas, TX 753804957

Ken Davidsonis themanagingeditoranda member of the Circuit Cellar INK engineering staff. He holds a B.S. in computer engineering and an M.S. in computer science from Rensselaer Polytechnic Institute.

IRS

228 Very Useful
229 Moderately Useful
230 Not Useful

STEVE'S OWN INK

The Home Computer Revolution is Over

I remember the early days of The Revolution. There weren't many small computer systems, there were even fewer magazines about them, but there was an almost evangelical fervor about how these "toy" computers were going to change our lives. Of course, in the years since then, the computers have changed, and our expectations have changed along with them. I'm not going to go into what computers have done for our business lives, but I think it's time for a quick peek at the corpse-littered trail of the home computer market.

In the late '70s and early '80s, you couldn't read a technology-oriented magazine or newspaper article without seeing several paragraphs about how every home would soon have a personal computer occupying a position of honor and importance. Dad would bring home work from the office and keep the family checkbook in order, Mom would search a voluminous recipe database and work budgeting miracles to rival the Pentagon's, and little Buffy would churn out Ph.D.-level term papers and learn with joy, havingreplaced sternMs.Grumpit with the friendly warm glow of her CRT.

Now, we know that, with the possible exception of both Mom and Dad bringing home work from the office, very little of that scenario has come to pass. Most people don't want to complicate their lives with a new system. A four-function calculator provides as much computing horsepower as most folks can cope with, and costs about the same as a box of Twinkies. Generally, the media and general population have written off the home computer along with the home autogyro and personal hovercraft.

It's not that the best of American industry didn't try to make the home computer as much a part of the residential landscape as the television. Apple, IBM, TI, Sinclair, Commodore, Coleco, and others gave it their best shots, but came away without much to show for their efforts (unless you count excess inventory tax credits as "something to show." Listen to any computer industry pundit and they'll tell you: Home computers are about as common as the black-footed ferret. The only problem with the industrygurusisthat they'reabsolutely,irrefutablywrong.

Whileeveryone had theirattentionglued to the "glamour toy" versions of home computers, a serious clandestine movement was beginning. At the same time that families across America were just saying no to computers, they were embracing microwave ovens like long-lost relatives. Nearly all of the microwave ovens carried microcontrollers or microprocessors into the homes of the heartland. Meanwhile, the energy crisis was making more people think about improving their home climate control system. In most cases, better control meant a microprocessor of some flavor. The tide swept onward, with VCRs, televisions, stereos, telephones, coffee makers, and other common household items flooding American homes with little computers. Ever hear of Nintendo? Yes, it's a rhetorical question. Every time a wild-eyed adolescent plugs into the Super Mario Brothers, he or she is booting a computer.

Now that most people are more comfortable with digital electronic technology, marketers are starting to—oh, so cautiously-introduce computer-like appliances. In Japan, there are keyboards and storage devices that plug into Nintendo systems. Several digital televisions would need a minimum of add-ons to start functioning as high-quality data terminals. (If you have trouble with that concept, just remind yourself that Sears owns a large chunk of Prodigy.) More than all of that, however, the trend-watchers are telling us that "home offices," complete with computer, fax machine, photocopier, and shredder, are going to be all the rage in the 1990s.

Thesimpletruthis that, while weal1 hadourhandsfull of microwave popcorn, the "home computer revolution" was fought and won by microprocessors and embedded microcontrollers. The home office, stuffed to the gills with computing and communications gear, wouldn't be possible if millions of people had not become comfortable with the idea of computers through microwaves, video games, and VCRs. The barricades are down: It's time to move forward.

Now if we can just figure out a way to start sneaking home robotics in though the back door.. .

Steve