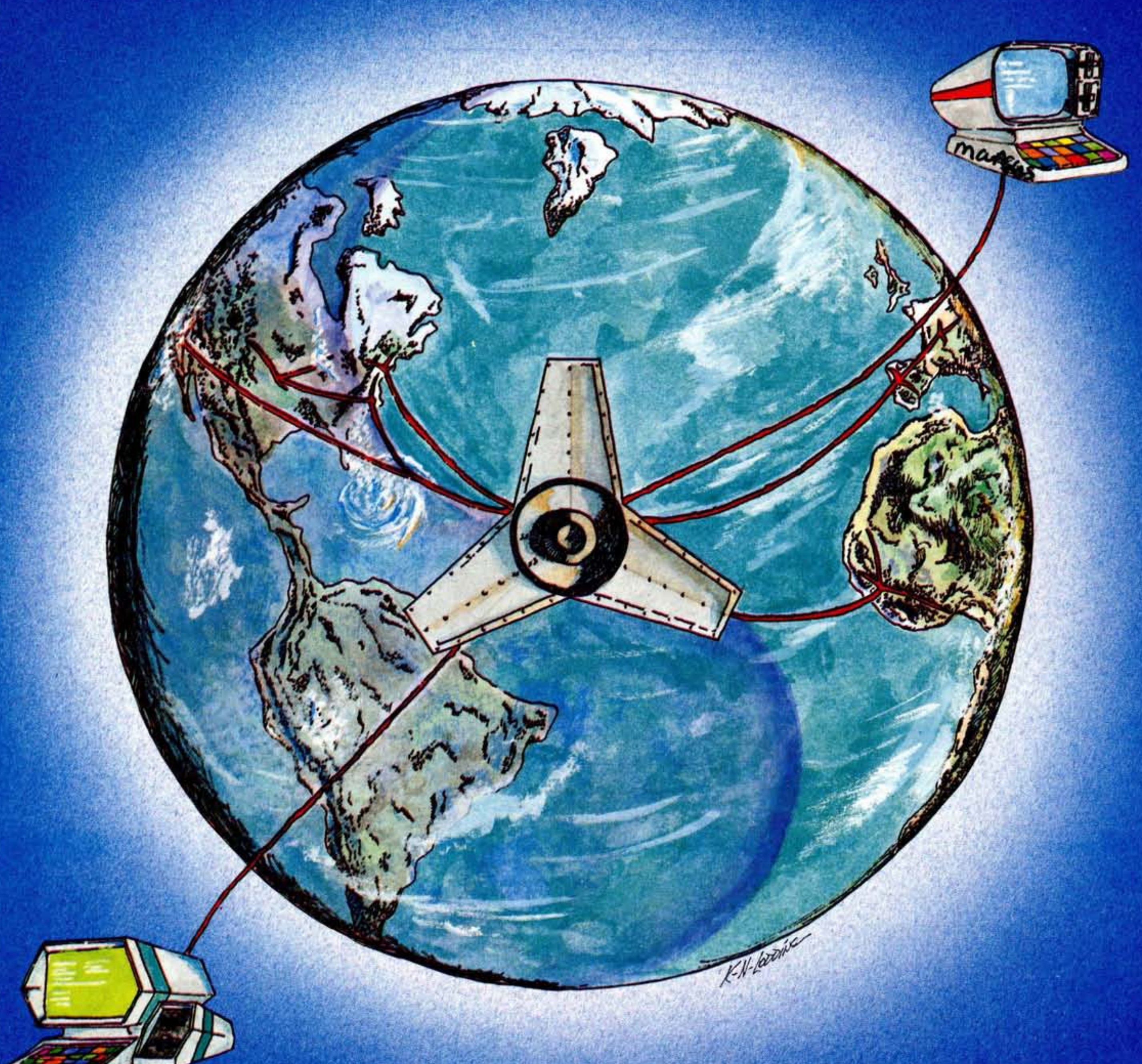
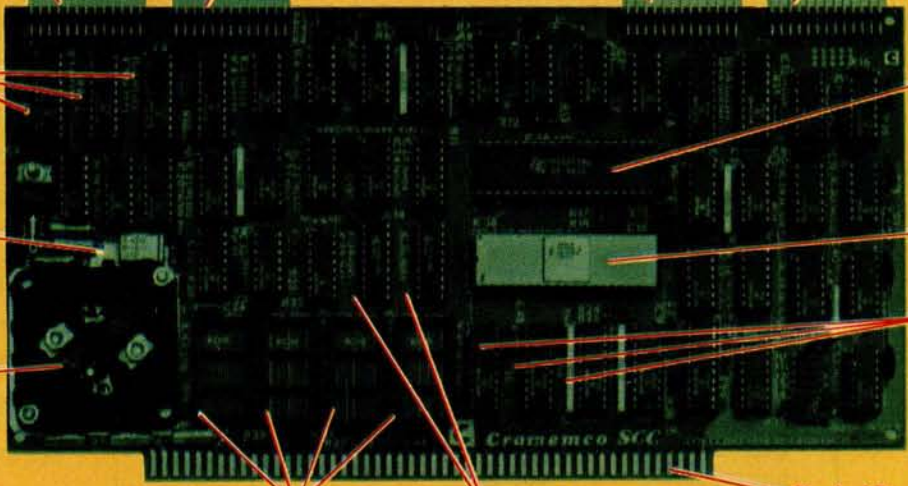
NOVEMBER 1978 VOLUME 3, Number 11 $2.00 in USA
$2.40 in CANADA

BYTE

the small systems journal





The single card computer with the features that help you in real life

COMPLETE COMPUTER

In this advanced card you get a professional quality computer that meets today's engineering needs. And it's one that's complete. It lets you be up and running fast. All you need is a power supply and your ROM software.

The computer itself is super. Fast 4 MHz operation. Capacity for 8K bytes of ROM (uses 2716 PROMs which **can be programmed by our new 32K** BYTESAVER® PROM card). There's also 1K of on-board static RAM. Further, you get straightforward interfacing through an RS-232 serial interface with ultra-fast speed of up to 76,800 baud — software programmable.

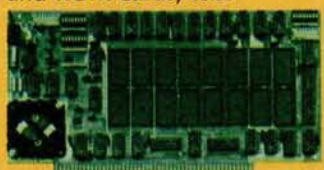
Other features include 24 bits of bi-directional parallel I/O and five on-board programmable timers.

Add to that vectored interrupts.

ENORMOUS EXPANDABILITY

Besides all these features the Cromemco single card computer gives you enormous expandability if you ever need it. And it's easy to expand. First, you can expand with the new Cromemco 32K BYTESAVER PROM card mentioned above. Then there's Cromemco's broad line of S100-bus-compatible memory and I/O interface cards. Cards with features such as relay interface, analog interface, graphics interface, opto-isolator input, and A/D and D/A conversion. RAM and ROM cards, too.

Card Cage



32K BYTESAVER PROM card

EASY TO USE

Another convenience that makes the Model SCC computer easy to use is our Z-80 monitor and 3K Control BASIC (in two ROMs). With this optional software you're ready to go. The monitor gives you 12 commands. The BASIC, with 36 commands/functions, will directly access I/O ports and memory locations — and call machine language subroutines.

Finally, to simplify things to the ultimate, we even have convenient card cages. Rugged card cages. They hold cards firmly. No jiggling out of sockets.

AVAILABLE NOW/LOW PRICE

The Cromemco Model SCC is available now at a low price of only $450 factory assembled ($395 kit).

So act today. Get this high-capability computer working for you right away.

Cromemco
incorporated
Specialists in computers and peripherals
280 BERNARDO AVE., MOUNTAIN VIEW, CA 94040 • (415) 964-7400

Circle 80 on inquiry card.

SEE YOUR LOCAL CROMEMCO DEALER

Anchorage, AK 99502
ALASKA DIVERSIFIED DIGITAL
1700 West Hoteco Avenue
(907) 277-6312

Huntsville, AL 35805
COMPUTERLAND OF HUNTSVILLE
3020 University Drive, N.W.
(205) 539-1200

Jonesboro, AR 72401
BUSINESS COMPUTER SYSTEMS
1122 Huntington Ave.
(501) 972-1800

● Little Rock, AR 72206
COMPUTER PRODUCTS UNLIMITED
2412 South Broadway
(501) 371-0449

Phoenix, AZ 85013
BYTE SHOP OF PHOENIX
24 W. Camelback Rd.
(602) 265-0065

Phoenix, AZ 85021
COMPUTER RESEARCH CORP.
2225 W. Mtn. View Road—No. 8
(602) 943-8719

Tempe, AZ 85281
BYTE SHOP TEMPE
813 N. Scottsdale Rd.
(602) 894-1129

Tempe, AZ 85282
COMPUTERWARE, INC.
214 W. Southern
(602) 968-6312

Tucson, Arizona 85716
BYTE SHOP OF TUCSON
2612 East Broadway
(602) 327-4579

● Carson, CA 90746
SUNSHINE COMPUTER COMPANY
20710 South Leapwood Avenue
(213) 327-2118

Citrus Heights, CA 95610
BYTE SHOP SACRAMENTO
6041 Greenback Lane
(916) 961-BYTE

Dublin, CA 94566
COMPUTERLAND DUBLIN
6743 Dublin Blvd.
(415) 828-8090

El Cerrito, CA 94530
COMPUTERLAND EL CERRITO
11074 San Pablo Avenue
(415) 233-5010

● Fountain Valley, CA 92708
ADVENTURES IN COMPUTING
8756 Warner Avenue
(714) 848-8588

Hayward, CA 94542
COMPUTERLAND OF HAYWARD
22634 Foothill Boulevard
(415) 538-8080

Huntington Beach, CA 92647
ALGORITHM PERSONAL COMPUTERS
7561 Rhine Drive
(714) 751-8080

Inglewood, CA 90302
COMPUTERLAND OF W. LOS ANGELES
6840 La Cienega Boulevard
(213) 776-8080

Lawndale, CA 92060
BYTE SHOP OF LAWNDALE
6508 Hawthorne Boulevard
(213) 371-2421

Lawndale, CA 90260
COMPUTERLAND SOUTH BAY
16720 So. Hawthorne Blvd.
(213) 371-7144

● Loomis, CA 95650
KINGMONT ENTERPRISES, INC.
9900 King Road
(916) 988-0189

Los Altos, CA 94022
COMPUTERLAND LOS ALTOS
4546 El Camino Real
(415) 941-8154

Los Angeles, CA 90025
AMERICAN RECORDER COMPANY
1655 Sawtelle Boulevard
(213) 477-2063

Mission Viejo, CA 92630
COMPUTERLAND SADDLEBACK VLY
24001 Via Fabricante—No. 904
(714) 770-0131

Moraga, CA 94556
ACCOUNTING BY COMPUTER
1161 Larch Ave.
(415) 376-1730

● Mountain View, CA 94040
BYTE OF MT. VIEW
1063 W. El Camino Real
(415) 969-5464

Orange, CA 92667
COMPUTER MART
633-B West Katella Boulevard
(714) 633-1222

Palo Alto, CA 94306
BYTE SHOP PALO ALTO
2227 El Camino Real
(415) 327-8080

Palo Alto, CA 94301
MICROTECH EXPORTS
912 Cowper Street
(415) 328-1712

Pasadena, CA 91101
BYTE SHOP PASADENA
496 South Lake Avenue
(213) 684-3311

Sacramento, CA 95816
MICROCOMPUTER APPLICATIONS SYS
2322 Capital Avenue
(916) 443-4944

San Diego, CA 92111
COMPUTERLAND OF SAN DIEGO
4233 Convoy Street
(714) 560-9912

San Diego, CA 92111
THE COMPUTER CENTER
8205 Ronson Road
(714) 292-5302

San Francisco, CA 94105
COMPUTERLAND OF SAN FRANCISCO
117 Fremont Street
(415) 546-1592

● San Francisco, CA 94103
THE COMPUTER STORE OF S.F.
1093 Mission Street
(415) 431-0640

San Francisco, CA 94118
THE NETWORK
495 Third Avenue—No. 4
(415) 221-1112

San Jose, CA 95124
BYTE SHOP #3 of SAN JOSE
2626 Union Avenue
(408) 377-4685

San Jose, CA 95129
COMPUTERLAND SAN JOSE
1077 Saratoga-Sunnyvale Road
(408) 253-8080

San Jose, CA 95123
THE COMPUTER ROOM
124-H Blossom Hill Road
(408) 226-8383

San Mateo, CA
COMPUTERLAND SAN MATEO
42—42nd Avenue
(415) 572-8080

● San Mateo, CA 94401
COMPUTER TERMINAL
309 S. San Mateo Drive
(415) 347-9894

San Mateo, CA 94402
OMEGA BUSINESS SYSTEMS
1202 W. Hillsdale Blvd.
(415) 341-4200

● Santa Ana, CA 92705
ADVANCED MICROCOMPUTER PROD.
1310 East Edinger
(714) 558-8813

Santa Clara, CA 95051
THE BYTE SHOP OF SANTA CLARA
3400 El Camino Real
(408) 249-4221

Santa Maria, CA 93454
GRANITE BUSINESS SYSTEMS
2222 S. Broadway
(805) 928-2030

Santa Monica, CA 90401
THE COMPUTER STORE
820 Broadway
(213) 451-0713

Sunnyvale, CA 94086
RECREATIONAL COMPUTER CENTER, INC.
1324 South Mary Avenue
(408) 735-7480

Thousand Oaks, CA 91360
COMPUTERLAND 1000 OAKS
171 E. Thousand Oaks Blvd.
(805) 495-3554

Tustin, CA 92680
COMPUTERLAND OF TUSTIN
104 West First Street
(714) 544-0542

● Van Nuys, CA 91411
COMPUTER COMPONENTS
5848 Sepulveda Boulevard
(213) 786-7411

Walnut Creek, CA 94596
BYTE SHOP COMPUTERS OF DIABLO VLY
2989 N. Main Street
(415) 933-6252

Walnut Creek, CA 94598
COMPUTERLAND WALNUT CREEK
1815 Ygnacio Valley Rd.
(415) 935-6502

Denver, CO 80222
COMPUTERLAND DENVER
3422 So. Colorado Blvd.
(303) 759-4685

Englewood, CO 80110
BYTE SHOP OF ARAPAHO COUNTY
3463 South Acoma Street
(303) 761-6322

Fairfield, CT 06430
COMPUTERLAND OF FAIRFIELD
2475 Black Rock Turnpike
(203) 374-2227

Windsor Locks, CT 06096
THE COMPUTER STORE
63 South Main Street
(203) 627-0188

Newark, DE 19711
COMPUTERLAND OF NEW CASTLE CITY
Astro Shopping Center
Kirkwood Highway
(302) 738-9656

Ft. Lauderdale, FL 33334
BYTE OF FT. LAUDERDALE
1044 E. Oakland Park Blvd.
(305) 561-2983

Ft. Lauderdale, FL 33312
COMPUTERS FOR YOU, INC.
3608 W. Broward Blvd.
(305) 581-8945

● Ft. Lauderdale, FL 33334
DATA ADMINISTRATORS, INC.
87 N.E. 44th Street
(305) 491-8600

● Ft. Myers, FL 33901
MICRO COMPUTER SYSTEMS, INC.
12800 U.S. 41 South
(813) 481-3376

Plantation, FL 33317
COMPUTER AGE
1400 S.W. 70th Avenue
(305) 791-8080

● Tampa, FL 33609
MICRO COMPUTER SYSTEMS
144 S. Dale Mabry Avenue
(813) 879-4225

Venice, FL 33595
SARA TECH ELECTRONICS
400 Base Ave. Suite 238
(813) 485-3559

● Atlanta, GA 30340
ATLANTA COMPUTER MART
5091-B Buford Highway
(404) 455-0647

Marietta, GA 30060
EDUCATIONAL DATA SERVICES
79 Oxford Road
(404) 926-5439

Smyrna, GA 30080
COMPUTERLAND MARIETTA
2423 Cobb Parkway
(404) 953-0406

● Haiku, Maui, HI 96708
CAPACITY, INC.
P.O. Box A
(808) 575-2930

Kailua, Oahu, HI 96734
MICROCOMPUTER ENTERPRISES
1490 Mokulua Drive
(808) 261-3281

Arlington Heights, IL 60004
COMPUTERLAND ARLINGTON HTS.
50 East Rand Road
(312) 255-6488

Champaign, IL 61820
BYTE SHOP CHAMPAIGN
1602 S. Neil Street
(217) 352-2323

Chicago, IL 60632
BRONSON & BRATTON, INC.
5161 S. Millard Avenue
(312) 735-6200

● Lisle, IL 60532
COMPUTER & CONTROL AFFILIATES, INC.
4315 Azalea, No. 401
(312) 968-6348

Lombard, IL 60148
MIDWEST MICROCOMPUTERS, INC.
708 South Main Street
(312) 495-9889

Naperville, IL 60540
ILLINI MICROCOMPUTERS
612 East Ogden Avenue
(312) 420-8813

Niles, IL 60648
COMPUTERLAND NILES
9511 N. Milwaukee Avenue
(312) 967-1714

Oak Lawn, IL 60453
COMPUTERLAND OF OAK LAWN
10935 South Cicero Avenue
(312) 422-8080

Rockford, IL 61101
IMPERIAL COMPUTER SYSTEMS
2105—23rd Avenue
(815) 226-8200

Schaumburg, IL 60195
DATA DOMAIN SCHAUMBURG
1612 East Algonquin Road
(312) 397-8700

● Springfield, IL 62704
COMPTEC
1104 West Fayette
(217) 546-7951

● Bloomington, IN 47401
DATA DOMAIN
221 W. Dodds St.
(812) 334-3607

● Fort Wayne, IN 46805
DATA DOMAIN OF FORT WAYNE
2805 East State Boulevard
(219) 484-7611

Indianapolis, IN 46268
DATA DOMAIN OF INDIANAPOLIS
7027 Michigan Road
(317) 251-3139

West Lafayette, IN 47906
DATA DOMAIN OF WEST LAFAYETTE
219 West Columbia
(317) 743-3951

Mission, KS 66202
COMPUTER CENTER
5813 Johnson Drive
(913) 432-2983

Overland Park, KS 66206
PERSONAL COMPUTER CENTER, INC.
3819 W. 95th Street
(913) 649-5942

Lexington, KY 40501
DATA DOMAIN OF LEXINGTON
506-1/2 Euclid Avenue
(606) 233-3346

Louisville, KY 40222
COMPUTERLAND OF LOUISVILLE
813-B Lyndon Lane
(502) 425-8308

Louisville, KY 40220
DATA DOMAIN OF LOUISVILLE
3028 Hunsinger Lane
(502) 456-5242

● Louisville, KY 40206
PRAGMATECH
2310 Mellwood Avenue
(502) 895-1230

Shreveport, LA 71106
MICRO BUSINESS SYSTEMS, INC.
7721 Creswell Rd.
(214) 343-7611

● Burlington, MA 01803
THE COMPUTER STORE
120 Cambridge Street
(617) 272-8770

Concord, MA 01742
ZEPHYR DATA SYSTEMS INC
52 Bradford St.
(617) 369-0643

College Park, MD 20740
INTELLIGENT BUSINESS MACHINES
7338 Baltimore Avenue, Suite 200
(301) 779-7998

● Pikesville, MD 21208
MODULAR SYSTEMS, INC.
4005 Seven Mile Lane
(301) 484-6322

● Rockville, MD 20852
COMPUTER WORKSHOP
1776 East Jefferson
(301) 468-0462

Rockville, MD 20855
COMPUTERLAND OF GAITHERSBURG
16065 Frederick Road, Route 355
(301) 948-7676

Silver Spring, MD 20901
C & M ENGINEERING, INC.
8318 Worth Avenue
(301) 589-2779

● Towson, MD 21204
COMPUTERS ETC.
13-A Allegheny Avenue
(301) 296-0520

Ann Arbor, MI 48104
NEWMAN COMPUTER EXCHANGE
1250 North Main Street
(313) 994-4445

Ann Arbor, MI 48104
THE COMPUTER STORE
310 East Washington
(313) 995-7616

Ann Arbor, MI 48104
UNITED MICROSYSTEMS CORP.
2601 South State Street
(313) 668-6806

● Berrien Springs, MI 49103
THE ABACUS
Route No. 1, Box 193, (Niles Avenue)
(616) 429-3034

Grand Rapids, MI 49508
JEPSAN, GROUP K
4706 Morningside Drive, S.E.
(616) 698-9057

Kentwood, MI 48508
COMPUTERLAND GRAND RAPIDS
2927—28th Street, S.E.
(616) 942-2931

Royal Oak, MI 48073
COMPUTER MART OF ROYAL OAK
1800 West 14 Mile Road
(313) 576-0900

Southfield, MI 48034
COMPUTERLAND SOUTHFIELD
29673 Northwestern Highway
(313) 356-8111

Minneapolis, MN 55404
MINNESOTA MICRO SYSTEMS, INC.
514 Cedar Avenue South
(612) 338-5604

Chesterfield, MO 63017
COMPUTER SYSTEMS CTR ST. LOUIS
13461 Olive Boulevard
(314) 576-5020

Lake St. Louis, MO 63367
DATA DYNAMICS, INC.
1000 Lake Saint Blvd., Ste. 324
(314) 625-2400

Nashua, NH 03060
COMPUTERLAND OF NASHUA
419 Amherst
(603) 889-5238

Cherry Hill, NJ 08034
COMPUTERLAND CHERRY HILL
1442 E. Route 70
(609) 795-5900

Clark, NJ 07066
S-100, INC.
7 White Place
(201) 382-1318

● Iselin, NJ 08830
COMPUTER MART OF NEW JERSEY
501 Route No. 27
(201) 283-0600

Morristown, NJ 07960
COMPUTERLAND OF MORRISTOWN
2 De Hart Street
(201) 539-4077

Reno, NV 89502
BYTE SHOP OF RENO
4104 Kietzke Lane
(702) 826-8080

Buffalo, NY 14150
COMPUTERLAND OF BUFFALO
1612 Niagara Falls Boulevard
(716) 836-6511

● Buffalo, NY 14226
CORSON COMPUTER CORP., INC.
3834 Main Street
(716) 832-0662

De Witt, NY 13214
COMPUTER SHOP OF SYRACUSE
3470 Erie Blvd. East
(315) 446-1284

Hewlett, NY 11577
LOUIS SCHAEFFER
319 Hamilton Avenue
(516) 374-1286

● Hollis, NY 11423
SYNCHRO-SOUND ENTERPRISES
193-25 Jamaica Avenue
(212) 468-7067

Ithaca, NY 14850
COMPUTERLAND OF ITHACA
225 Elmira Road
(607) 277-4888

New York City, NY 10016
BYTE SHOP EAST
130 E. 40th Street
(212) 889-4204

New York City, NY 10016
COMPUTER MART OF NEW YORK
118 Madison Avenue
(212) 686-7923

Rochester, NY 14609
COMPUTER HOUSE, INC.
721 Atlantic Avenue
(716) 654-9238

● Syracuse, NY 13202
MINIMICROMART
1618 James Street
(315) 422-4467

Cincinnati, OH 45409
DIGITAL DESIGN
7694 Camargo Rd. (Madiera)
(513) 561-6733

Cincinnati, OH 45202
21ST CENTURY SHOP
16 Convention Way
Skywalk, Vine & Race
(513) 651-2111

Columbus, OH 43201
MIDWESTERN DIGITAL ELECTRONICS
80 West Lane Avenue, Suite 13
(614) 294-2891

● Dayton, OH 47409
COMPUTER SOLUTIONS
1932 Brown Street
(513) 223-2348

Kent, OH 44240
OHIO MICRO SYSTEMS
233 South Water Street
(216) 678-5202

Mayfield Heights, OH 44124
COMPUTERLAND OF CLEVELAND EAST
1288 Som Center Road
(216) 461-1200

Oklahoma City, OK 73106
HIGH TECHNOLOGY
1611 Northwest 23rd Street
(405) 528-8008

Beaverton, OR 97005
BYTE SHOP OF BEAVERTON
3482 S.W. Cedar Hills Boulevard
(503) 644-2687

Eugene, OR 97401
REAL OREGON COMPUTER CO.
205 West Tenth Street, P.O. Box 52
(503) 484-1040

Tigard, OR 97223
COMPUTERLAND PORTLAND
12020 S.W. Main Street
(503) 620-6170

● Tualatin, OR 97062
CREATIVE SYSTEMS
8101 S.W. Nyberg Road
(503) 638-8406

Allison Park, PA 15101
R. DON KRIGGER
9380 Springfield Drive
(412) 364-4945

Huntington Valley, PA 19006
MARKETLINE SYSTEMS, INC.
2337 Philmont Avenue
(215) 947-6670

King of Prussia, PA 19406
COMPUTER MART OF PENNSYLVANIA
550 De Kalb Pike (Route 202)
(215) 265-2580

Pittsburgh, PA 15237
ELECTRONICS PLACE
7250 McKnight Road
(412) 367-2900

Warwick, RI 02888
COMPUTER POWER, INC.
100 Post Road
(401) 738-4477

● Columbia, SC 29205
BYTE SHOP OF COLUMBIA
2018 Green Street
(803) 771-7824

Memphis, TN 38101
COMPUTER STORE
2162 Courtland Place
(901) 274-1938

Nashville, TN 37206
COMPUTER WORLD
625 Main Street
(615) 244-6094

● Nashville, TN 37211
SURYA CORPORATION
5755 Nolensville Road
(615) 834-5638

● Amarillo, TX 79109
COMPUTER CORNER, INC.
1000 South Georgia St., No. 3
(806) 355-5618

Austin, TX 78757
COMPUTERLAND AUSTIN
3300 Anderson Lane
(512) 452-5701

● Austin, TX 78731
COMPUTERS'N THINGS
2825 Hancock Drive
(512) 453-5970

● Corpus Christi, TX 78411
MICRO SYSTEMS SERVICES, INC.
5301 Everhart, Space H—P.O. Box 6535
(512) 855-4516

● Dallas, TX 75243
COMPUSHOP
211 Keystone Park, 13933 North Central
(214) 234-3412

● Houston, TX 77057
COMPUTERLAND OF S.W. HOUSTON
6439 Westheimer
(713) 977-0909

Houston, TX 77006
COMPUTERTEX
2300 Richmond Avenue
(713) 526-6934

Houston, TX 77036
INTERACTIVE COMPUTERS
7646-½ Dashwood, P.O. Box 36304
(713) 977-7037

● Houston, TX 77098
THE MOS
1853 Richmond Avenue
(713) 527-8008

Lubbock, TX 79412
COMPUTER MART OF W. TEXAS
3508 D Avenue, Q
(806) 765-7134

Richardson, TX 75080
THE MICRO STORE
634 South Central Expressway
(214) 231-1096

Provo, UT 84601
THE COMPUTER WORKS
735 South State Street, P.O. Box N
(801) 373-7522

Alexandria, VA 22304
COMPUTERS PLUS, INC.
678 S. Pickett Street
(703) 751-5656

McLean, VA 22101
COMPUTER SYSTEMS STORE
1984 Chain Bridge Road
(703) 821-8333

Bellevue, WA 98007
COMPUTERLAND BELLEVUE
14340 N.E. 20th
(206) 746-2070

Federal Way, WA
COMPUTERLAND SOUTH KING CITY
1500 South 336th St.
Parkway Center, Suite 12
(206) 838-9363

Seattle, WA 98115
THE RETAIL COMPUTER STORE
410 N.E. 72nd Street
(206) 524-4101

Tacoma, WA 98499
COMPUTERLAND TACOMA
8791 South Tacoma Way
(206) 581-0388

● Appleton, WI 54911
SOUND WORLD, INC.
3015 W. Wisconsin Avenue
(414) 734-7698

Madison, WI 53711
THE MADISON COMPUTER STORE
1863 Monroe Street
(608) 255-5552

Madison, WI 53711
COMPUTERLAND MADISON
690 S. Whitney Way
(608) 273-2020

Milwaukee, WI 53208
MILWAUKEE COMPUTER STORE
4710 West North Avenue
(414) 445-4280

● Onalaska, WI 54650
OMICRON MICROCOMPUTERS
638 11th Ave. North
(608) 783-1869

● Racine, WI 53405
COLORTRON TV
2111 Lathrop Avenue
(414) 637-2003

• INTERNATIONAL •

Carnegie, Vict. Australia
SONTRON INSTRUMENTS
17 Arawatta Street
568-0358

Croydon, Vict., Australia, 3136
COMPUTER ART & EDUCATION
2 Stephen Crescent

Prahran, Victoria, Australia, 3181
GEMINI MICRO COMPUTERS
P.O. Box 99

Sydney, NSW, Australia, 2000
COMPUTERLAND SYDNEY
52-58 Clarence St.
TLX: AA 26657

West Perth, 6005, W. Australia
AUSTRALIAN COMPUTER PRODUCTS
1300 Hay Street
(09) 322-6497

● A-2401 Fishamend, Austria
KREBS GES. M.B.H.
Hainburger Strabe 34
02232/6918223

Calgary, Alberta, Canada
ROBO-TRONICS
509-16 Avenue N.W.
(403) 282-9496

Montreal, Quebec, Can. H3B 3C9
FUTUR BYTE, INC.
1191 Phillips Square
(514) 861-3130

Toronto, Ont., Can. M4G 3B5
COMPUTER MART, LTD.
1543 Bayview Avenue
(416) 484-9708

Toronto, Ont., Can. M4R 1A1
FIRST CANADIAN COMPUTER STORE
44 Eglinton Avenue West
(416) 482-8080

Toronto, Ont., Can. M5V 1Z1
TRINTRONICS, LTD.
186 Queen Street West
(416) 598-0260

Calgary, Alta., Can. T2T 4T9
THE COMPUTER SHOP
3515 Eighteenth Street, S.W.
(403) 243-0301

● Little Paxton, Huntingdon, Cambs.
Cambs., England PE19 4NZ
COMART LIMITED
24A Market Square
0480-215-005

Paris 75017, France
COMPUTER BOUTIQUE
149 Avenue de Wagram

Chiyoda-Ku, Tokyo 101, Japan
BYTE SHOP SOGOH
1-4-6 Sotokanda
03 255-1984

Kashiwa City Chiba Pf. 277, Japan
BELLREY TEC. INC.
5-9-2-Chome Kaga

Tokyo 106, Japan
IIE CORPORATION
4-33, 3-Chome, Roppongi, Minatoku
(03)-585-2333

● Mexico 18, D.F., Mexico
INTELEX, S.A.
P.A. De Los Santos 70
(905) 516-9970

Makati, Rizal, The Philippines
DECISION SYSTEMS CORP.
150 Amorsolo St. & Pasay Rd.
Legaspi Village

● 10052 Stockholm 20, Sweden
DATORISERING KONSULT AB
Fack 2

4053 Basel, Switzerland
EUREX, INC.
Beinwiler Str. 13
061-357069

● 8005 Zurich, Switzerland
COMICRO AG
Badenerstrasse 281

● Hiltrup 4400, Munster, W. Germany
BASIS MICROCOMPUTER VERTRIEB
Von-Flotow-Strabe 5
02501-4800

● 2000 Wedel, Holstein, W. Germany
DIGITRONIC COMPUTERSYSTEME
Bei Der Doppeleiche 3-5
04103-7393

● DENOTES SYSTEMS THREE DEALER

Cromemco
i n c o r p o r a t e d
Specialists in computers and peripherals
280 BERNARDO AVE., MOUNTAIN VIEW, CA 94040 • (415) 964-7400

Circle 80 on inquiry card.

In the Queue

BYTE November 1978
Volume 3 Number 11

Foreground

Background

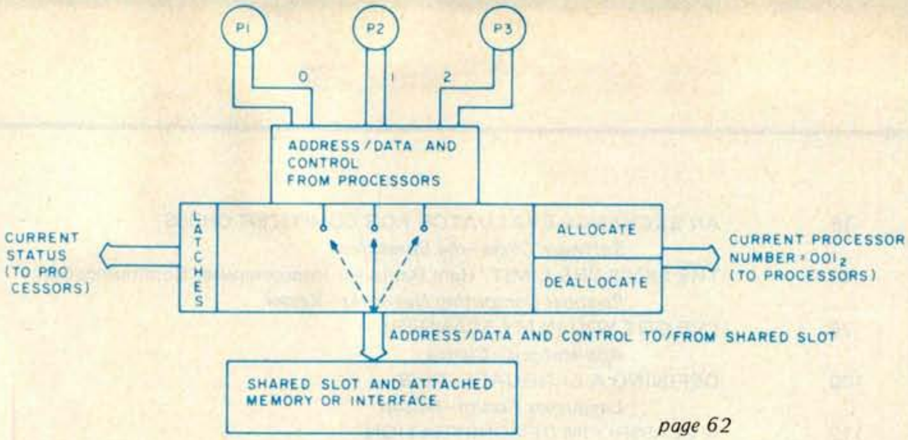
Nucleus

BYTE is published monthly by BYTE Publications Inc, 70 Main St, Peterborough NH 03458. Address all mail except subscriptions to above address: phone (603) 924-7217. Address all editorial correspondence to the editor at the above address. Unacceptable manuscripts will be returned if accompanied by sufficient first class postage. Not responsible for lost manuscripts or photos. Opinions expressed by the authors are not necessarily those of BYTE. Address all subscriptions, change of address, Form 3579, and fulfillment complaints to BYTE Subscriptions, PO Box 590, Martinsville NJ 08836. Second class postage paid at Peterborough NH 03458 and at additional mailing offices—USPS Publication No. 102410. Canadian second class registration No. 9321. Subscriptions are $15 for one year, $27 for two years, and $39 for three years in the USA and its possessions. In Canada and Mexico, $17.50 for one year, $32 for two years, and $46.50 for three years. $25 for a one year subscription by surface mail worldwide. Air delivery to selected areas at additional rates available upon request. $25 for a one year subscription by air delivery to Europe. Single copy price is $2.00 in the USA and its possessions, $2.40 in Canada and Mexico, $3.50 in Europe, and $4.00 elsewhere. Foreign subscriptions and sales should be remitted in United States funds. Printed in United States of America.

Subscription WATS Line: (800) 258-5485

About the Cover

This month's cover by artist Ken Lodding emphasizes the personal computing potential of the OSCAR amateur radio satellites described in Joe Kasser's article on page 48. When Ken isn't doing technology-related art work, he helps design and implement experimental systems for Digital Equipment Corp in Merrimack NH.



page 62

In This BYTE

One of the basic building blocks of any computer chess playing program is the exchange evaluator. Authors Dan and Kathe Spracklen describe the exchange evaluator used in their Sargon program in **An Exchange Evaluator for Computer Chess.** *page 16*

Taylor series expansions are not necessarily the best polynomial approximations for many functions. Fred R Ruckdeschel describes several minimax and rational polynomial approximations for some common functions in **Functional Approximations.** A method for creating polynomial approximations for microcomputers using mathematical tables and large system statistical routines is also discussed. *page 34*

One method of connecting computers into a communications network is by VHF radio stations. To ease some of the physical difficulties encountered with this arrangement, a satellite can be introduced into the system, and then, as Joe Kasser says, **The Sky's the Limit** for personal computer users. *page 48*

One theme of this issue is VHF communication among computers to produce a network. There are other types of networks which are available to computer users. One of these uses a **Distributed Network,** which is described by Glen R Horton. *page 62*

Ciarcia's Circuit Cellar turns to melodrama this month as Steve and Lloyd are confronted by a mysterious stranger. All ends happily, however, with the aid of a computer controlled stepper motor driven infrared and visible light scanner. Read **I've Got You in My Scanner!** *page 76*

One of the more fascinating uses of a computer, teaching courses, is frequently known as computer assisted instruction. George A Gerhold describes one group of people that is actively involved in implementing **Computer Assisted Instruction on a Microcomputer.** *page 90*

As part of the never ending struggle for truth, beauty and the ultimate in high level languages, David Wilson provides readers with a Languages Forum proposal, **Defining a Language: PL/B,** a combination of some of the features of a high level language like BASIC with access to assembly language details when necessary. *page 100*

Controlling a physical system is a natural for the microprocessor: parameters can often be changed in a matter of seconds in software without any mechanical changes to the system. A simple airflow control system is described in Garnet L Hill's article, **A Classroom Demonstration: Controlling a System with a Microcomputer.** *page 112*

The VHF communication network already in existence provides a readily available communication network for the computer experimenter. R E Bruninga describes one system which is up and running in his article **A Multiuser Data Network.** *page 120*

A home computer system can be used for communication among many people. This month Ward Christensen and Randy Suess describe their implementation of a **Hobbyist Computerized Bulletin Board.** *page 150*

For people who are just being introduced to the world of microprocessors, the first steps can seem confusing. For a description of one device which can aid the introduction read W N Hubin's review of the **Heath Microprocessor Training System.** *page 158*

Craig Anderton's simple circuit described in **A Cassette Interface Switching Box for the TRS-80** shows you how to operate the cassette recorder manually and monitor tapes while they are being read in without having to unplug any cables. *page 160*

This month we continue the series **Creating a Chess Player** by Peter W Frey and Larry R Atkin with the first half of **Chess 0.5,** a program written in Pascal by Larry Atkin, who is coauthor with David Slate of the world championship computer chess program, Chess 4.6. *page 162*

In this issue, Kin-Man Chung and Herbert Yuen conclude their series of articles on **A "Tiny" Pascal Compiler** with a discussion of a p-code (pseudocode) to 8080 code conversion program and the needed runtime routines. *page 182*

Editorial

Editors. This is a word of multiple definitions. One definition is implicit in the job description of myself and my associates at BYTE. Another definition is that applied to a class of utility programs which every BYTE reader's computer system has in some form or another. It is this latter definition which provides the subject for this editorial.

What is often ignored is the fact that editor programs make an excellent form of software project, less complex than an interpreter or a compiler, but of sufficient magnitude to be interesting and educational. The problems and characteristics of editors are analogous to those of compilers and interpreters, especially when viewed as tools of software development. Just as compilers beget better compilers, editors can be used to edit better editors.

By Carl Helmers

On the Virtues of Writing Editors

Software projects are some of the most exciting avocational applications of personal computers, applications which emphasize computing as an art form and means of expression of personal tastes. This editorial on the philosophy of software projects in general and text editing in particular was inspired by recent completion of a text editor begun as a spare time project in March of this year. Five months and many evening and weekend hours later it is now mid August and the editor program is working sufficiently well to serve as the primary monitor program and software tool of my homebrew computer system.

Why Write an Editor Program?

There are alternatives to do-it-yourself. In the usual case of purchasing a commercial system from a computer store, an editor program is built into the systems software of the computer. The present day technology of self-contained desktop computers with read only memory systems software (BASIC interpreters) invariably includes primitive line oriented editors built into the computer.

On computers which have assemblers plus high level language facilities like Pascal, C, FORTRAN or COBOL or brand X, one or more forms of more useful text editing programs may be available in order to prepare source language files. This is the configuration found in the typical \$3000+ personal microcomputer or \$10,000+ commerical minicomputer system.

Even hardware homebrewers do not necessarily have to start completely from scratch with editor software. Often a microprocessor chosen for a homebrew system has a software package which can be purchased from one of a number of software vendors. And in the newer processor designs like the Motorola 6809, complete dynamic relocatability of code allows systems software to be sold in address space position independent read only memory chip sets which homebrewers can expect to have access to over the next few years.

But to do-it-yourself is one of the principle reasons why people get involved in intricate avocational pursuits. Why does the amateur wine maker inject yeast into juice and wait for the results when Napa Valley and its industries exist? Why does the amateur pilot learn to fly when professional transportation options are available from the largest airline to the smallest air taxi service? Why make your own astronomical telescope when there exists such a perfection as Questar? And similarly in computers, why make your own text editor, compiler or computer when so many options exist in the marketplace? On a practical or short term basis there is no visible advantage to engaging in such pursuits. But on a long-term scale of personal development, actively applying one's mind and energy is always rewarding.

What Was Needed?

With this spirit and a secondary purpose of improved systems software in mind, I proceeded to think about writing an editor program. What is required in a personal computer system to engage in a project of this complexity? The requirements are essentially identical to the requirements for programming any arbitrary application of the personal computer, from electronic music systems to sophisticated information storage and retrieval systems: tools. Software tools are needed in order to create more software, and software tools presuppose a certain hardware basis.

The hardware components needed for a useful system are available right now from a

Continued on page 68

New from North Star
Double Density Performance at Single Density Prices

Specifications:
S-100 compatible. MFM encoding, 35 tracks with ten 512-byte sectors per track. 179,200 bytes on double density SA-400 and North Star BASIC, DOS, and Monitor included.

For further information, write for full color catalog or contact your local computer store.

The new HORIZON computer and Micro Disk System now record in double density! That means each new Shugart SA-400 minifloppy disk drive accesses 180K bytes of on-line information. All double density HORIZON computers and Micro Disk Systems have a redesigned controller which allows the use of quadruple capacity disk drives as they become available in early 1979. A three-drive North Star System with quadruple capacity disk drives will access over a megabyte of on-line information. But, best of all there's no price increase for double density models.

North Star BASIC and DOS have been upgraded to accommodate the increased capacity and yet run existing programs with little or no change. The new disk system also supports single density, so existing single density diskettes can still be used. Single density SA-400 drives previously purchased with North Star systems can also be used.

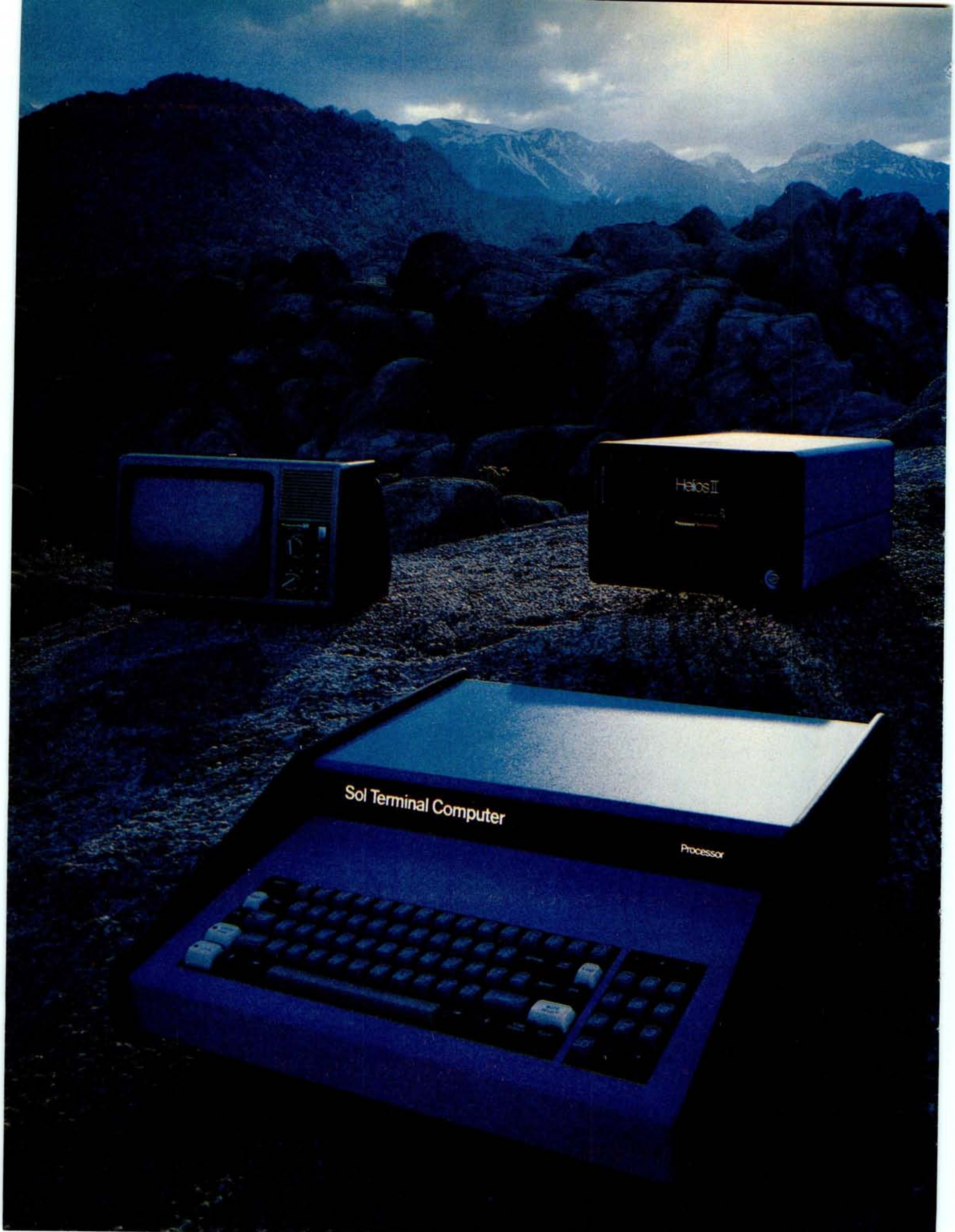
Pricing

HORIZON with one double density SA-400 minifloppy (180K bytes), 16K RAM, Z80A processor and serial I/O port: $1599 kit, $1899 assembled.

MICRO DISK SYSTEM with one double density SA-400 minifloppy, controller board and power regulation: $699 kit, $799 assembled. (Cabinet and power supply $39 extra each.)

NORTH STAR ★ COMPUTERS
2547 Ninth Street
Berkeley, California 94710
(415) 549-0858

Circle 285 on inquiry card.


Helios II
Sol Terminal Computer
Processor

Letters

SCHEMATIC ENCOUNTER

I had a close encounter of the third kind while trying to read and understand the schematics to a Teletype ASR33 – it was just awful. The engineers had their hands on it for some time when it was decided to change it over to a 20 mA loop, and I was the victim elected to do it.

Do any BYTE readers know if there is a book that explains the ASR33 Teletype schematics in understandable terms? I know it took many years to make the simple hard to understand, but maybe someone out there said that enough is enough to this madness.

Yes, I finally got the show on the road, but for awhile I was thinking: why me?

Craig Mueller
10130 Sepulveda Blvd
Mission Hills CA 91345

INTERPRETER INFORMATION WANTED

I have just finished reading Osborne and Associates' *8080A/8085 Assembly Language Programming* book. Now I want to find something that shows how a BASIC interpreter is put together in assembly language. All the software supplied here does not even attempt to explain in anything but vague terms.

The question is, where can I find literature like this? I would like something that would actually run and something not theoretical. I hope you can help.

Hugh Shedd
#8 3419 Portland Av S
Minneapolis MN 55407

Contact Dr. Dobb's Journal of Calisthenics & Orthodontia *for information on what they have published in the past with respect to BASIC interpreters. Their address is POB E, Menlo Park CA 94025.*

PRAISE FOR BITS

I do not write letters to editors often but this is one pleasant exception. About a week ago my daughter ordered a book for my birthday from BITS. The birthday was only a week off and she asked for fast service. Not only did the book arrive exactly one week from the date that she sent her letter, but it was also wrapped in birthday paper with a bow. It arrived the day before my birthday. Both my daughter and I were quite delighted at such a personal touch. I believe that the name of the person

Continued on page 141

BEFORE YOU BUY COMPUTER #1, VISIT #1 COMPUTERLAND

If the truth is that you want a computer . . . then we want to be your computer store.

We're ComputerLand, the #1 computer store chain in the U.S. What's meaningful about that fact is, that ComputerLand has been chosen by more people as having what they've been looking for. And, since you're looking, let us tell you what you'll find, when you visit a ComputerLand store.

You'll find a product line that's continually evaluated to provide you with the widest and best selection in quality, brand name microcomputers anywhere. You'll find an enthusiastic and knowledgeable staff able to interpret all the equipment specifications, in terms of how they apply to you, and in a way you'll understand. You'll find demonstration areas where you can get a firsthand experience of running a computer yourself.

COMPUTERS FOR BUSINESS

You'll find educational materials to give you a total insight into the world of microcomputers.

You'll find a fully equipped service department to provide whatever assistance is required to keep your computer running in top-notch condition. You'll find computer user's clubs to join, where you can share ideas with people as enthusiastic as yourself. And, with each new visit, you'll find excitement—from the people you deal with, the equipment they offer, and from your own ever-growing personal involvement.

Enough about us. How about what computers do. To attempt to describe all the things your computer might do, would be to describe your imagination. So instead, we'll briefly list some of the many things for which small computers are already being used.

In business, the advent of the versatile and compact microcomputer has put the benefits of computing within reach of small companies. With systems starting at less than $6000, the businessman can computerize things like accounting, inventory control, record keeping, word processing and more. The net result is the reduction of administrative overhead and the improvement of efficiency which allows the business to be managed more effectively.

COMPUTERS FOR THE HOME

In the home, a computer can be used for personal budgeting, tracking the stock market, evaluating investment opportunities, controlling heating to conserve energy, running security alarm systems, automating the garden's watering, storing recipes, designing challenging games, tutoring the children . . . and the list goes on.

In industry, the basic applications are in engineering development, process control, and scientific and analytical work. Users of microcomputers in industry have found them to be reliable, cost-effective tools which provide computing capability to many who would otherwise have to wait for time on a big computer, or work with no computer at all.

COMPUTERS FOR INDUSTRY

And now we come to you, which leads us right back to where we started: **If you want a computer, then we want to be your computer store.**

Whether you want a computer for the home, business or industry, come to ComputerLand first. We'll make it easy for you to own your first computer. Because, simply put, we really want your business. When you come right down to it, **that's what makes us #1.**

ComputerLand®

WE KNOW SMALL COMPUTERS

14400 Catalina St., San Leandro, CA 94577 (415) 895-9363 • Franchise Opportunities Available

Why Apple II is the world's best selling personal computer.

Which personal computer will be most enjoyable and rewarding for you? Since we delivered our first Apple® II in April, 1977, more people have chosen our computer than all other personal computers combined. Here are the reasons Apple has become such an overwhelming favorite.

Apple is a fully tested and assembled mainframe computer. You won't need to spend weeks and months in assembly. Just take an Apple home, plug it in, hook up your color TV* and any cassette tape deck — and the fun begins.

To ensure that the fun never stops, and to keep Apple working hard, we've spent the last year expanding the Apple system. There are new peripherals, new software, and the Apple II Basic Programming Manual. And wait till you see the Apple magazine to keep owners on top of what's new.

Apple is so powerful and easy to use that you'll find dozens of applications. There are Apples in major universities, helping teach computer skills. There are Apples in the office, where they're being programmed to control inventories, chart stocks and balance the books. And there are Apples at home, where they can help manage the family budget, control your home's environment, teach arithmetic and foreign languages and, of course, enable you to create hundreds of sound and action video games.

When you buy an Apple II you're investing in the leading edge of technology. Apple was the first computer to come with BASIC in ROM, for example. And the first computer with up to 48K bytes RAM on one board, using advanced, high density 16K devices. We're working to keep Apple the most up-to-date personal computer money can buy. Apple II delivers the features you need to enjoy the real satisfaction a personal computer can bring, today and in the future.

15 colors & hi-resolution graphics, too.

Don't settle for a black and white display! Connect your Apple to a color TV and BASIC gives you instant command of three display modes: Text, 40h x 48v Color-graphics in 15 colors, and a 280h x 192v High Resolution array that lets you plot graphs and compose 3-D images. Apple gives you the added capability of combining text and graphics, too.

Back to basics, and assembly language too.

Apple speaks three languages: fast integer BASIC, floating point BASIC for scientific and financial applications, and 6502 assembly language. That's maximum programming flexibility. And, to preserve user's space, both integer BASIC and monitor are permanently stored in 8K bytes of ROM, so you have an easy-to-use, universal language instantly available. BASIC gives you graphic commands: COLOR=, VLIN, HLIN, PLOT and SCRN. And direct memory access, with PEEK, POKE and CALL commands.

Software: Ours and yours.

There's a growing selection of preprogrammed software from the Apple Software Bank — Basic Finance, Checkbook, High Resolution Graphics and more. Now there's a User Section in our bank, to make it easy for you to obtain programs developed

by other Apple owners. Our Software Bank is your link to Apple owners all over the world.

Alive with the sound of music.

Apple's exclusive built-in speaker delivers the added dimension of sound to your programs. Sound to compose electronic music. Sound to liven up games and educational programs. Sound, so that any program can "talk" back to you. That's an example of Apple's "people compatible" design. Another is its light, durable injection-molded case, so you can take Apple with you. And the professional quality, typewriter-style keyboard has n-key rollover, for fast, error-free operator interaction.

Apple is the proven computer.

Apple is a state-of-the-art single board computer, with advanced LSI design to keep component count to a minimum. That makes it more reliable. If glitches do occur, the fully socketed board and built-in diagnostics simplify troubleshooting. In fact, on our assembly line, we use Apples to test new Apples.

Apple II plugs into any standard TV using an inexpensive modulator (not included).

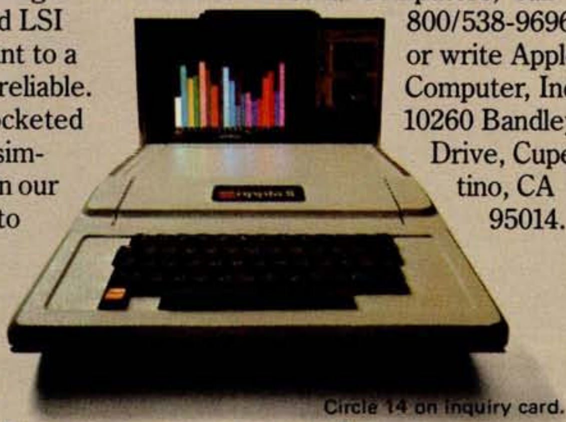
In California, call 408/996-1010.

Apple peripherals are smart peripherals.

Watch the far right column of this ad each month for the latest in our growing family of peripherals. We call them "intelligent interfaces." They're smart peripherals, so you can plug them in and run them from BASIC without having to develop custom software. No other personal computer comes close to Apple's expandability. In addition to the built-in video interface, cassette I/O, and four A/D inputs with two continuously variable game paddles, Apple has eight peripheral slots, three TTL inputs and four TTL outputs. Plus a powerful, state-of-the-art switching power supply that can drive all your Apple peripherals.

Available now.

Apple is in stock and ready for delivery at a store near you. Call us for the dealer nearest you. Or, for more details and a copy of our "Consumer's Guide to Personal Computers," call 800/538-9696** or write Apple Computer, Inc., 10260 Bandley Drive, Cupertino, CA 95014.



Circle 14 on inquiry card.

New from Apple.

Valuable new series of software packages for investors

Now private investors can generate their own stock market reports and perform critical investment analysis instantly with Apple II. Just log your Apple II computer on to Dow Jones' central data bank with powerful Apple software: the Dow Jones Series. The first two of these highly practical programs are available now.

With Apple's **Stock Quote Reporter** program, a local telephone call links you to Dow Jones' continuously updated stock quotes for more than 6000 companies listed on six major U.S. exchanges. Current activity for stocks in the investors portfolio is delivered automatically: ask/open, bid/close, high, low and last prices, and volume traded.

Our **Portfolio Evaluator** enables you to analyze current value of your portfolio, and short- and long-term gain/loss for each stock—or for your entire portfolio.

Cost of Apple's Dow Jones service is a one time contract fee of $25, which includes the Stock Quote Reporter program. An additional $3 charge is made for the first three minutes of any transaction and 50¢ per minute thereafter.

To take advantage of Apple's new financial services, Apple II users need only a communications card, a modem and an ordinary telephone. This equipment, the Dow Jones Series, and a broad selection of other Apple software are now in stock at your local Apple dealer.

Circle 15 on inquiry card.

apple computer®


DYNABYTE DB8/2
POWER ON
RESET
POWER OFF
HALT
DYNABYTE DB8/1
POWER ON
RESET
POWER OFF
HALT
DYNABYTE DB8/4
POWER ON
POWER OFF

An Exchange Evaluator for Computer Chess

Dan and Kathe Spracklen
10832 Macouba Pl
San Diego CA 92124

Three main tasks are basic to computer chess: generation of moves, evaluation of positions and selection between alternatives. Of these three, the central determining factor in the strength of the program relative to the capacity of the host machine is the *evaluation* segment. The reason for this is that any program must come to grips with the task of move generation, and various techniques of "pruning" decision trees are by now widely known. Furthermore, the smaller and slower the host machine, the more importance must be assigned to the evaluation facility. If a search can be carried to a great depth of *ply*, inaccuracies can generally be corrected long before the machine has been committed to a costly line of play. (A ply is a move by one player, ie: half of a complete move involving both players.) On the other hand, if processing limitations prevent a critical exchange from being examined to its conclusion, then not just accuracy but clairvoyance is demanded. Thus an attack evaluator assumes tremendous importance in a microcomputer chess program, much more so than in a large scale machine. But the limitations placed on the programmer of an 8 bit machine make it correspondingly more difficult to achieve this type of predictive power. The ability of Sargon (a chess playing program we wrote in Z-80 assembler language) to accurately forecast the outcome of an exchange has been the greatest single factor in its success.

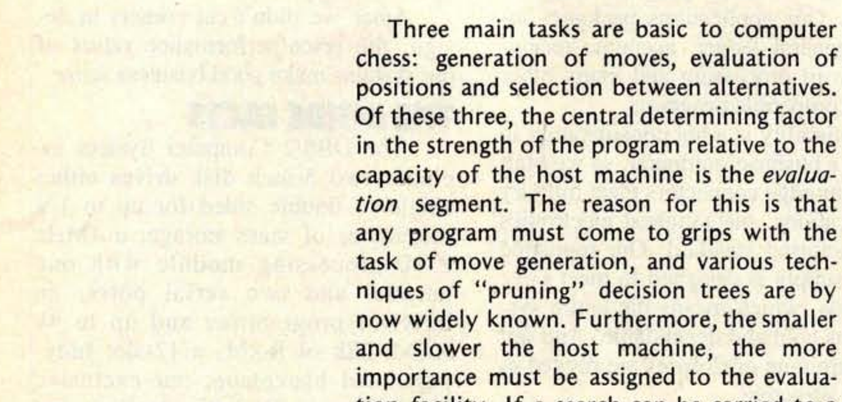


Table 1: Format of the attacker's array, a 14 byte array divided into two sections, seven bytes for White and seven bytes for Black. The first byte of each section contains the number of attackers (or defenders) in the array. The other six bytes contain the values of the pieces participating in the attack under analysis. Since no more than four bits are required per piece, two pieces are stored per byte and the array has a fixed format. The routine that fills the array assigns the first attacker of a given type to the low order four bits of the byte. A subsequent attacker of the same type is added by shifting up the low order four bits and inserting the new attacker.

Some Tactical Considerations

First, consider the capabilities desired of the routine. Assume that the computer is faced with evaluating the board position in figure 1. Black possesses a dangerous passed pawn that White has blockaded with a Knight. White is piling up attackers on the pawn and presently assaults it with King, Queen, and from behind the Queen, a Bishop: a total of three attackers. Black defends with Queen, Rook, and Knight; but the Black Knight is pinned against the Black King by White's Bishop, so Black really only has two usable defenders. Does this mean the pawn is lost? No, consider the order in which the exchange would occur. The King cannot legally capture first and the Bishop is behind the Queen, so the Queen must be the first taker. When Black responds with Rook takes Queen, Black has gained considerable material and is under no obligation to go any further with the exchange. To summarize the subtleties involved, the program must recognize transparent attacks through its own pieces which move in the same direction. It must recognize pins (and partial pins such as a Rook pinned along a rank or file). It must understand the relative values of attacking and defending pieces, and, finally, it must realize that the exchange may be terminated at any point by either side. Pins are a whole topic in themselves, and Sargon's pinned piece routines will not be discussed in any detail. Instead, we shall concentrate on the exchange routine itself, which weighs the relative merits of the battles engaged on the board.

The Data Structures

The basic data structure used by the exchange evaluator is the attackers array. It is a 14 byte area divided into two sections, seven bytes for White and seven for Black. The first byte of each section contains the number of attackers (or defenders) contained in the array. The other six bytes in each section store the piece values of the pieces participating in the attack. Since no more than four bits are required, two pieces are stored per byte, and the array has a fixed format. Table 1 illustrates the arrangement within a section. The routine which fills the array assigns the first attacker of a given type to the low order four bits of the byte. A subsequent attacker of the same type is added by shifting up the low order four bits and inserting the new attacker in its place. The instruction used to implement this is the rotate left digit (RLD) (see figure 2). If a piece attacks from behind the Queen, such as the Bishop in figure 1, it is placed in the high order four bits of the Queen byte. From that position it will not come into play in the attack until after the Queen has captured. It is possible for two Rooks to attack through the Queen. In this situation one Rook is stored behind the Queen and the other in the King byte, pushing him up behind the Rook if he is involved in the attack. (By the rules of chess, the King cannot capture unless all defenders are exhausted, so he is properly placed behind the Rook.)

A note about overflows: the table is necessarily limited in size and is adequate for all the pieces originally on the board. If pawn promotions result in multiple pieces and a table overflow occurs, the excess pieces are ignored in evaluating the exchange.

An Overview of the Exchange Evaluator

The exchange evaluator (XCHNG) operates on a prefilled attacker's array. The array itself is filled by the *attack save* (ATKSAV) routine as attackers are discovered by the *attacker's* routine (ATTACK). The latter two routines are important, and recent changes to them have resulted in a significant improvement in the performance of Sargon, but they are not discussed in this article. The attacker's array describes a specific battle over a given occupied square. The player who occupies the square is the defender and the player with the opposite color is the attacker. The attacker's section is examined for the lowest valued attacker. That piece is compared in value to the piece on the occupied square. If the attacker is lower in value than the defended piece, we know at once that we can win material by capturing that piece. We don't yet know how much, because the piece may have been totally undefended, or it may be that our lower value piece will be captured in return. For example if our Bishop attacks an enemy Rook, we can be sure at least of "winning the exchange" (a phrase chess buffs use to describe trading a Rook for a minor piece, ie: for a Bishop or Knight). But to find out whether the whole Rook is ours for free or if we must give up our Bishop in return, we must toggle the attacker/defender roles, since our Bishop now occupies the square, and run through the analysis again. Of course back when the Bishop was retrieved from the attacker's array, it was also removed, the attack count decremented, and its position filled with zeroes.

The evaluation is not so obvious when the attacker is of higher value than the piece on the occupied square. In this case there are only two situations in which you would want to capture. One occurs when the attacked piece is totally undefended, and the

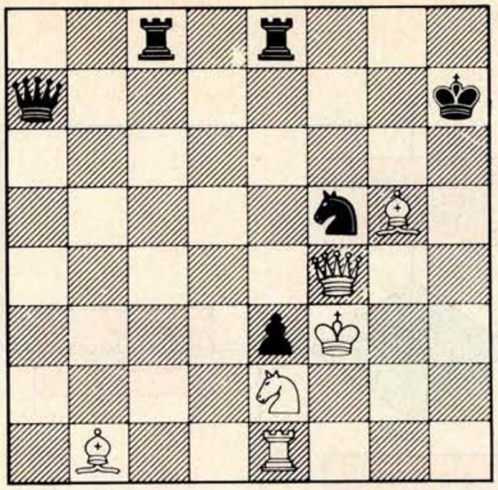


Figure 1: Sample board position. White's Bishop is indirectly attacking the pawn, so the value of the Bishop is stored in the high order four bits of the Queen byte (which is directly attacking the pawn) in the attacker's array. See table 1.

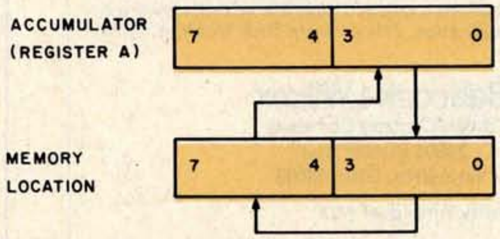


Figure 2: The Rotating Left Digit (RLD) instruction, used to add attackers to the attacker's array.

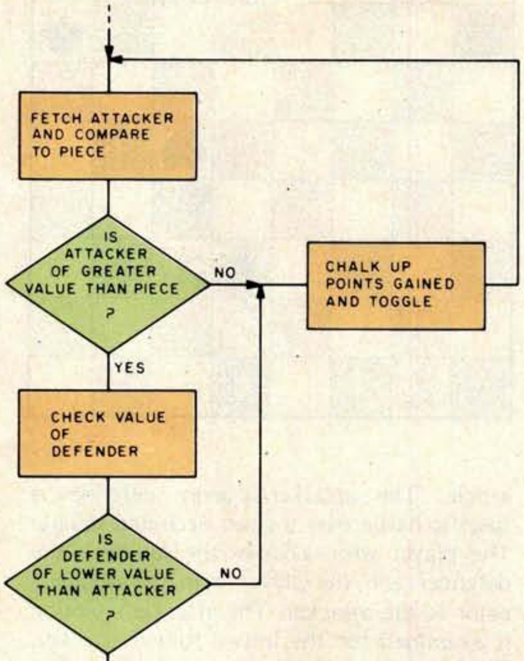


Figure 3: Summary of the flow of exchange evaluation. If the attacker is of the same value as the piece under potential attack, material cannot be lost by swapping, and the piece may in fact be taken for free. To determine the potential for winning material, assume the capture takes place, switch (or "toggle") the roles of defender and attacker and run through the analysis again.

other occurs when the attacked piece is defended by a piece of the same or higher value, and we can back up the attack with yet another attacker. Suppose, for example, our Queen attacks an enemy pawn. If the pawn is completely unguarded, we can, of course, take it for free. We might also want to take it if it is defended by the opponent's Queen and we can recapture with, say, a Bishop which attacks from behind our Queen. But any time the attacked piece is defended by a piece of lower value than the attacker, we can terminate the exchange right there, since it would not be to the advantage of the attacker to continue. For example, if our Queen attacked a pawn that was defended by an enemy pawn, we wouldn't consider making the capture.

If the attacker is of the same value as the piece on the occupied square, we know we can't lose material by swapping, and the piece might be ours for free. So to find out what we stand to gain, we assume the capture takes place, switch (or "toggle") the attacker/defender roles, and run through the analysis again (see the summary in figure 3).

Quantizing the Evaluation

We now have a general plan for the flow of the evaluator. What is needed is a means of quantizing the results and coming up with a points total, the exchange residue, which accurately describes that particular battle. The exchange residue is zero at the onset of the analysis and will be adjusted up or down as the evaluation proceeds. At each iteration the number of points at stake is the value of the piece which currently occupies the square in question. If the analysis calls for a capture on the first iteration, the points at stake are added to the exchange residue. Thus the exchange residue will contain the number of points lost by the initial defender (or, conversely, won by the initial attacker). We will maintain this frame of reference throughout the evaluation. If the analysis requires that attacker/defender roles be toggled, and a capture occurs on the second iteration, the points at stake would be subtracted from the exchange residue. Suppose we again have a situation where our Bishop attacks an enemy Rook. The points at stake are the assumed value of the Rook, and let's suppose we value the Rook at five points. We know the analysis will call for Bishop takes Rook, so at that time the five points for the Rook will be added to the initially zero exchange residue. Then

ULL SIZE FLOPPY DISK $995 COMPLETE!

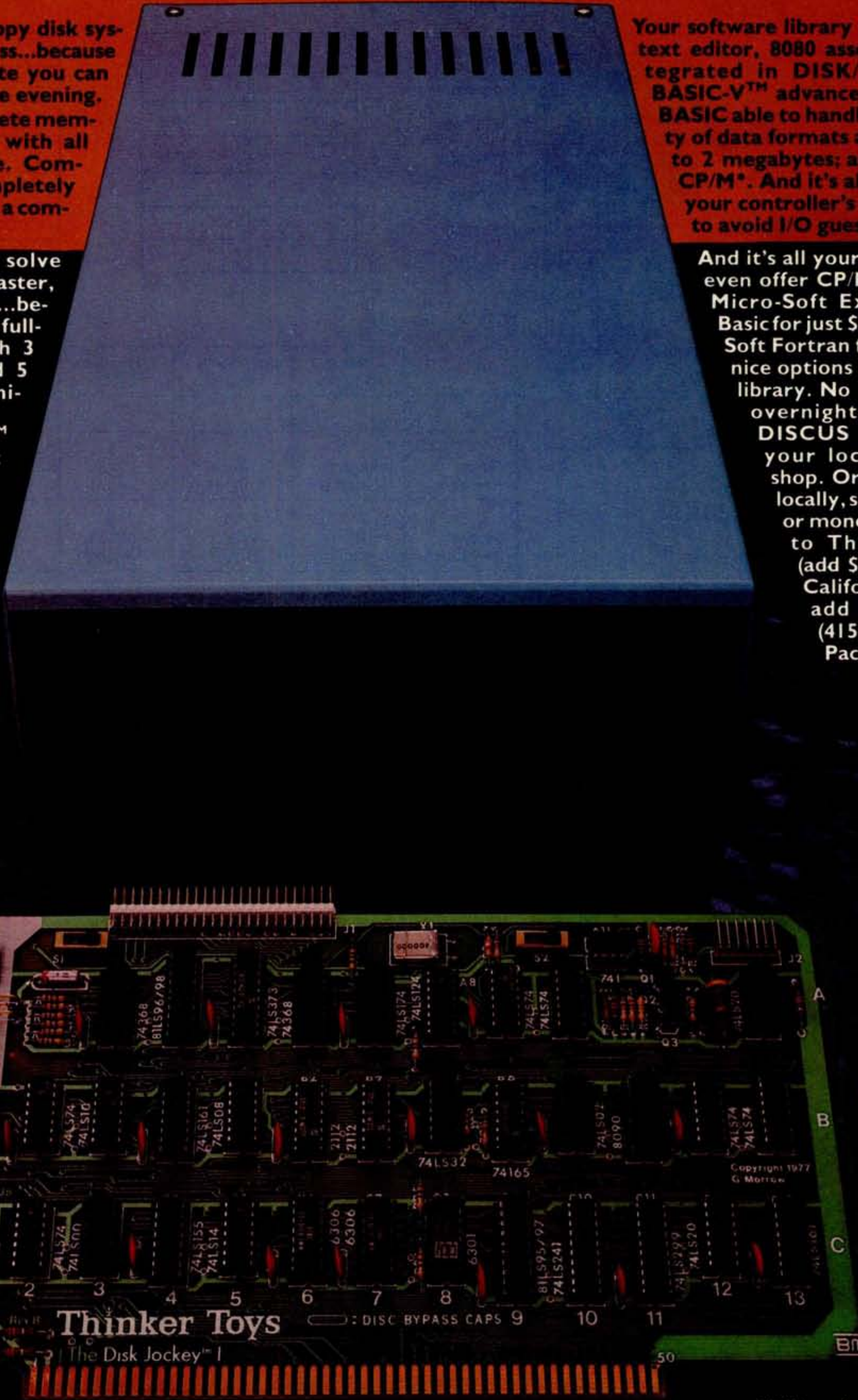
SCUS I™ full-size floppy disk sys-
m is an overnight success...because
s delivered so complete you can
ve it running in a single evening.
r just $995, it's a complete mem-
y system. Complete with all
rdware and software. Com-
etely assembled. Completely
erfaced. And tested as a com-
ete system.

nd you can not only solve
ur memory shortage faster,
u can solve it longer...be-
use DISCUS I™ is a full-
e floppy system with 3
mes the storage and 5
mes the speed of mini-
ppies.

ur $995 DISCUS I™
stem includes a Shugart
0R full-size drive with
wer supply in a hand-
me freestanding cabi-
t, our 8-drive capaci-
S-100 controller
th on-board buffer
d serial interface,
cables and con-
ctors, and all the
ftware you need.

Your software library includes DOS, text editor, 8080 assembler (all integrated in DISK/ATE™); our BASIC-V™ advanced virtual disk BASIC able to handle a wide variety of data formats and address up to 2 megabytes; and patches for CP/M*. And it's all interfaced to your controller's serial I/O port to avoid I/O guesswork.

And it's all yours for $995. We even offer CP/M for just $70, Micro-Soft Extended Disk Basic for just $199 and Micro-Soft Fortran for just $349 as nice options to add to your library. No wonder it's an overnight success! See DISCUS I™ today at your local computer shop. Or if unavailable locally, send your check or money order direct to Thinker Toys™ (add $7 for handling; California residents add tax.) Or call (415) 524-5317, 10-5 Pacific Time.



*CP/M is a trademark of Digital Research.

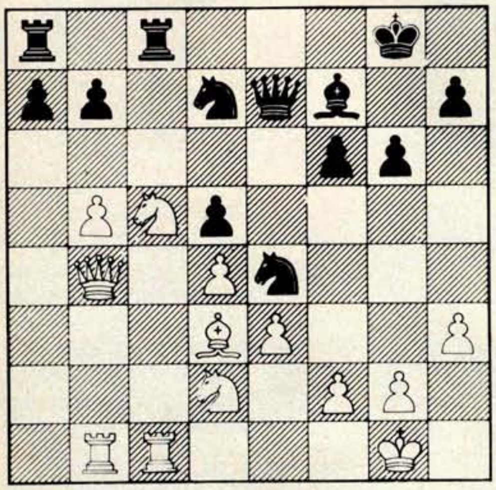
product of Morrow's
icro-Stuff for

Thinker Toys

1201 10th Street
Berkeley, CA 94710

Circle 255 on inquiry card.

BLACK



WHITE

(SARGON)

White to play.

Values used are based on the scale:

Queen	9 points
Rook	5 points
Bishop	3 points
Knight	3 points
Pawn	1 point

Attacker's Array:

	White		Black	
Count	3		2	
Pawns				1 pt
Knights	3 pts	3 pts		
Bishops		3 pts		
Rooks				
Queen				9 pts
King				

	Points At Stake	Attacker's Value	Defender's Value	Exchange Residue	
Onset	–	–	–	0	
First Iteration	3 (Black Knight)	3 (White Knight)	1 (Black Pawn)	3	Black Defends
Second Iteration	3 (White Knight)	1 (Black Pawn)	3 (White Knight)	0	White Defends
Third Iteration	1 (Black Pawn)	3 (White Knight)	9 (Black Queen)	1	Black Defends
Fourth Iteration	3 (White Knight)	9 (Black Queen)	3 (White Bishop)	No Change* 1	White Defends

*Black will not continue at the cost of his Queen.

Figure 4: Analysis of a typical chess battle, in this case at the K4 square, taken from the game of Keres versus Najdorf, International Tournament at Margate, 1939. The associated chart shows how the points at stake, attacker's value, defender's value and exchange residue are altered at each successive iteration.

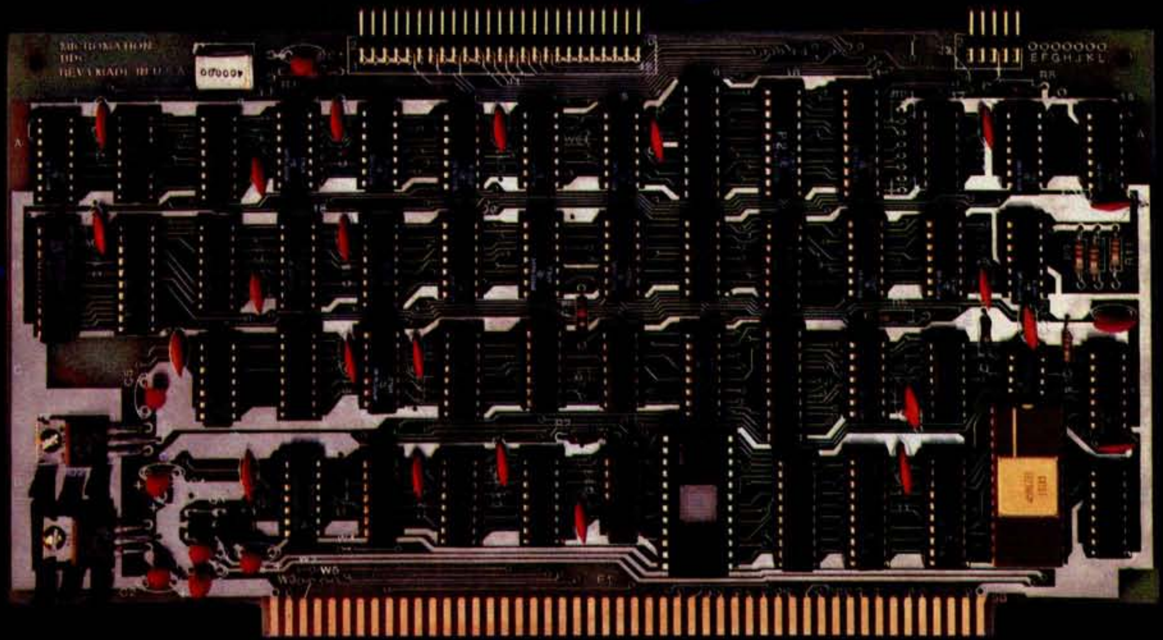
the attacker/defender roles are toggled, and if our Bishop, worth say three points, is recaptured, those three points would be subtracted from the exchange residue leaving a current residue of two points. If the battle continues, on the third iteration the points are again added, and on the fourth subtracted, etc. Figure 4 gives a typical battle and the associated chart shows how the points at stake, attacker's value, defender's value and exchange residue are altered at each successive iteration.

A note on the bounds of the exchange residue is pertinent here. The exchange residue will always be a positive number. This is clearly so, since for it to go negative the attacker would have to engage in an unsound exchange, such as the Queen capturing a pawn defended by another pawn as in a previous example. Such an exchange would be a blunder. We will assume that this won't occur on the part of our opponent, and we will eliminate it from our moves. The exchange residue will also have as a maximum the number of points at stake initially, since the defender will not make a move that will cost more than has already been lost. Thus, $0 \leqslant$ exchange residue $\leqslant$ value of attacked piece.

Programming the Evaluator

Great care is necessary in coding the routine, since it must be executed once for every attacked piece on the board. If we assume that an average of five pieces will be under attack at a time, this means the routine will be executed five times for every board evaluated. Since typically 5,000 to 12,000 board positions will be evaluated by the most recent version of Sargon using a 4 ply search, this means the exchange evaluator may be executed up to 60,000 times in determining a single move. So an inefficiency in execution time as slight as needlessly pushing and popping four registers would be magnified to a total cost of three seconds (assuming a 2 MHz clock) in the time required to process a single move. For this reason chess programmers must quickly become familiar with the relative execution times of their machine's instructions. If the exchange evaluator seems obscure, the

Micromation has done for the S-100 bus what IBM did for the floppy disk.



Reliably doubled capacity.

Double Capacity

The DOUBLER — Micromation's latest advance in floppy disk technology — doubles the capacity of floppy disk systems. Over 500 KBytes are recorded on each side of an 8" disk. This means bigger files for more powerful systems.

Double Speed

Data transfer with the DOUBLER is twice as fast — 500 Kbits per second. And since there is twice as much data on each track, your drive steps only half as much — so your system runs faster than it ever has before!

Increased Reliability

That's right — even better reliability. Why? Because we did it the IBM way. IBM designed 2D formatting — so it has to be reliable. Micromation's innovative, state-of-the-art design incorporates write precompensation electronics and a phase lock oscillator on a single, all digital, S-100 circuit board. So we guarantee the DOUBLER will be more dependable than your present single density controller — and we warantee the DOUBLER for a full year.

Unbeatable Convenience

It couldn't be easier to step up to double density. The DOUBLER operates automatically in either single or double density. Just insert a diskette and you're running properly. You can transfer files between single or double density diskettes without any software or hardware changes — or even operate with one single and one double density diskette.

Installation is a snap. There's a hardware UART on board and the software is all ready to go. An onboard 2708 EPROM contains the bootstrap. There's even jump-on-reset circuitry so you can operate without a front panel. And, of course, we include utilities to format diskettes.

Universally Versatile

The DOUBLER will operate with all industry-standard mini and full-sized drives. And it will work in any 8080 or Z-80 S-100 computer operating at 2 to 4 MHz. The DOUBLER will support up to four double or single headed drives.

Fully Compatible

The DOUBLER is compatible with CP/M* version 1.4. If you have a CP/M* 1.4 system, just add our CBIOS — or you can buy our ready-to-boot version. Install the new controller, connect any terminal to the RS-232 interface, and boot off your new double-sized, double-speed system. You still can use all your old software without any changes.

Completely Affordable

All Micromation products are fully assembled, thoroughly tested, include complete documentation, and are priced for value:

DOUBLER double density controller	$ 495.
MEGABOX dual drive double density system	2,295.
ZEPHER — Per Sci double density system	2,595.
Z-PLUS — MEGABOX 32 KZ-80 computer	4,295.

Available

The DOUBLER is available NOW at your local computer store.

Micromation Inc. 524 Union Street San Francisco California 94133 / 415 398-0289

MICROMATION

Where there's always more in store.

*CP/M is a trademark of Digital Research.

Circle 223 on inquiry card.

Table 2: *Map of the registers used by the exchange evaluator.*

AF	(Various Uses)	Program Status Flags
BC	Defender Count	Attacker Count
DE	Attacker Section Address	
HL	Defender Section Address	
IX	Index to Value Array	
IY		

AF'	Defender	Program Status Flags
BC'	Attacked Piece Value	Flag to Defender Side
DE'		Exchange Residue
HL'		Attacker Value

Table 3: *Map of the BC, DE, and HL registers used by the attacker/defender routines.*

BC	Attacker Count	Defender Count
DE	Defender Section Address	
HL	Attacker Section Address	

blame lies in just such considerations.

Since nearly every register in the Z-80 processor is utilized in the routine, a map is provided for reference in the discussion (see table 2). Although Sargon is coded in Z-80 assembler language using TDL mnemonics, no prior knowledge of the specific instructions is assumed. However, it is assumed that the reader is familiar with some microprocessor assembly language. Two routines are described: XCHNG, which performs the actual evaluation, and NEXTAD, which searches the attacker's array for the next attacker or defender.

Using the Exchange Residue

The exchange evaluator has completed its work once it has returned the outcome of the battle. But the evaluation segment is by no means complete. Information gleaned by analyzing attacks must be blended with data concerning piece mobility, development, total material and any other heuristics included in the program. The total picture is the responsibility of a routine called POINTS, which is not discussed here. But it is useful to see how POINTS makes use of the information returned by XCHNG.

The exchange evaluator must be called to examine every potential battle on the

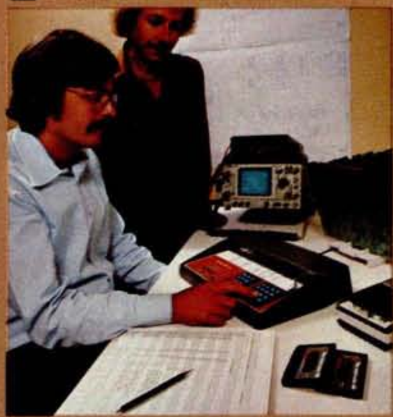
Listing 1: The Sargon exchange evaluator, written in Z-80 assembler language with TDL mnemonics.

Label	Op Code	Operand	Commentary
XCHNG:	LDA	P1	Fetch the attacked piece into register A. The piece includes a color flag in bit 7 (0 for White, 1 for Black) and the piece type in bits 2-0.
	LXI	H,WACT	Load into the HL and DE register pairs.
	LXI	D,BACT	The beginning addresses of the White and Black sections of the attackers array.
	BIT	7,A	Test the color flag bit of the piece and
	JRZ	XC5	skip the XCHG if the piece is White.
	XCHG		Otherwise swap the contents of the HL and DE registers. The result is to produce a pointer to the defender's section of the attackers array in the HL register pair and a pointer to the attacker's section in the DE pair.
XC5:	MOV	B,M	Fetches the byte pointed to by the HL pair into the B register. Fetches the byte pointed
	LDAX	D	to by the DE pair into the A register, then
	MOV	C,A	moves it into the C register. Since the first byte if each section of the attacker's array is the count (see table 1), we now have the total number of defenders in register B and attackers in C.
	EXX		Swap registers BC, DE, and HL for registers BC', DE', and HL', ready to initialize the rest of the data used by the exchange evaluator.
	MVI	C,0	Register C contains a flag which tells when the attacker/defender roles have been toggled. Each time the roles are reversed, register C is incremented. Then by examining bit 0 of C, we can tell which side is being examined. A value of 0 indicates the attacker's side is under consideration, and a value of 1 the defender's side.
	MVI	E,0	Initialize the exchange residue.
	LIXD	T3	T3 is an index by piece type into an array,
	MOV	B,PVALUE(X)	called PVALUE, which contains the point value (the worth) of each type of piece.

Note: A documented source listing of the entire Sargon program is available for $15 from Dan and Kathe Spracklen, 10832 Macouba Pl, San Diego CA 92124.

Hands-on microprocessor experience from day one...

The Instructor 50.



Learning by doing is still the best method of education. And when it comes to learning about the world of microcomputers, you won't find a better method than the Instructor 50.

It's the fast, ready-to-use learning device that immediately provides "hands on" experience for gaining microprocessor knowledge—in your home, office, or in the classroom.

Superior to other microprocessor learning aids, the Instructor 50 is a COMPLETE package—including a **built-in power supply** (50/60 Hz), an **LED prompting display**, and both **functional and hexadecimal keyboards**. You also get **S-100 compatibility** for adding memory and other peripherals. This lets you expand the machine's capability—and your microprocessing applications knowledge. Moreover, you can easily build a program library by recording your own audio cassettes.

The Instructor 50 comes complete with a *Users' Guide*, along with step-by-step instructions for those with no previous microprocessor experience.

Signetics offers one of the broadest choices of microprocessors in the industry. This knowledge stands behind the Instructor 50. When you need to learn about microprocessors, start with Signetics. Start with the Instructor 50. Send for your descriptive brochure today.

We can help you understand microprocessors.

signetics

a subsidiary of **U.S. Philips Corporation**

Signetics Corporation
811 East Arques Avenue
Sunnyvale, California 94086
Telephone 408/739-7700

Circle 317 on inquiry card.

A Solid State Keyboard as Modern as Your Computer

Solid state electronics has moved the computer quickly from the business world into personal uses. Meanwhile, computer keyboards have hardly moved at all.

Now TASA introduces a keyboard as modern as your computer. Don't confuse it with ordinary flex switches. It is fully solid state and self-contained, ready to plug in and use. Since it has no mechanical moving parts, it responds quickly to your touch. And it provides full ASCII coding in TASA's exclusive color-keyed layout that makes it easier to say what you want to say to your computer.

This is the TASA Micro Proximity Keyboard, and it sells for only $49.95. Despite the price, CMOS/LSI integrated circuits make it totally reliable inside. With the sensors behind a shield of polycarbonate—the most rugged plastic ever developed—it is also durable and reliable outside.

The TASA Keyboard contains all the features you would expect in a professional keyboard—shift, shift lock, control functions, and a normal typewriter format.

If you're tired of costly mechanical keyboards and kits you have to assemble, bring your computer up to date the easy way. Plug in a TASA Keyboard. It will never come between you and your computer.

---

the tasa

Micro Proximity Keyboard

$49.95

			The index is loaded into the IX index register and then the value of the piece under attack is loaded into the B register. So register B contains the number of points at stake in this attack.
	CALL	NEXTAD	Getting the value of the next attacker in register A. NEXTAD also sets the zero flag if there are no more attackers.
	RZ		Return if no more attackers.
XC10:	MOV	L,A	Save the attackers value in the L register.
	CALL	NEXTAD	Getting the value of the next defender in register A, and setting the zero flag if no more defenders.
	JRZ	XC18	If no defender, the piece is lost. Go chalk up points gained.
	EXAF		Save the defender by swapping AF and AF' registers.
	MOV	A,B	Move the value of the attacked piece into the A register to then compare its value to that of the attacker. Branch to XC19 if the value of the attacker is not greater than the value of the piece, to chalk up points gained and toggle.
	CMP	L	
	JRNC	XC19	
	EXAF		To reach this point, the attacker must be worth more than the piece under attack. So it is necessary to consider the value of the defender. This instruction swaps A and A' again to restore the value.
XC15:	CMP	L	Compare the value of the defender to the attacker. If the defender is worth less, return. It will not be to the attacker's advantage to continue the exchange.
	RC		
	CALL	NEXTAD	Otherwise get the value of the next attacker.
	RZ		Return if none. If the defender is worth the same or more than the attacker, the exchange should continue, provided there is another attacker available to recapture. Save the new attacker's value in the L register.
	MOV	L,A	
	CALL	NEXTAD	Then find out if there are any more defenders to contend with. If so, jump back to XC15 and repeat the process.
	JRNZ	XC15	
XC18:	EXAF		The exchange is terminated. There are no more defenders. The zero flag is set, so save it by swapping AF and AF'.
	MOV	A,B	Get the value of the attacked piece.
XC19:	BIT	0,C	Test for attacker's or defender's side.
	JRZ	XC20	Skip if on the attacker's side. Otherwise negate the value of the attacked piece. (On successive iterations the value is alternately added and subtracted.)
	NEG		
XC20:	ADD	E	Add the previous exchange residue to the new points won or lost and store the result as the new exchange residue.
	MOV	E,A	
	EXAF		Restore the last defender and the zero flag.
	RZ		Return if there are no more defenders.
	MOV	B,L	The last attacker becomes the new defender.
	JMP	XC10	Move his value into the B register and return to XC10 for another iteration.
NEXTAD:	INR	C	Increment side flag.
	EXX		Swap registers BC, DE, HL for BC', DE', HL', getting the set that contains the attacker and defender counts.
	MOV	A,B	Swap attacker and defender counts.
	MOV	B,C	
	MOV	C,A	
	XCHG		Swap attacker's array pointers. The register map is now as in table 3.
	XRA		Zero the A register and compare it to the attacker count. Go return if there are none.
	CMP	B	
	JRZ	NX6	
	DCR	B	Otherwise decrement the count, since one will be removed from the array.
NX5:	INX	H	Check the next byte of the attacker's array, looking for an attacker.
	CMP	M	
	JRZ	NX5	If not in this byte, go check the next.
	RRD		Otherwise rotate the attacker into the A register. The rotate right digit (RRD) is the reverse of the rotate left digit illustrated in figure 2.
	DCX	H	Decrement HL to back up the pointer. With two attackers stored per byte, the routine will return to the same byte to look for the next one.
NX6:	EXX		Swap registers BC, DE, HL and BC', DE' and HL' back again.
	RET		Return.

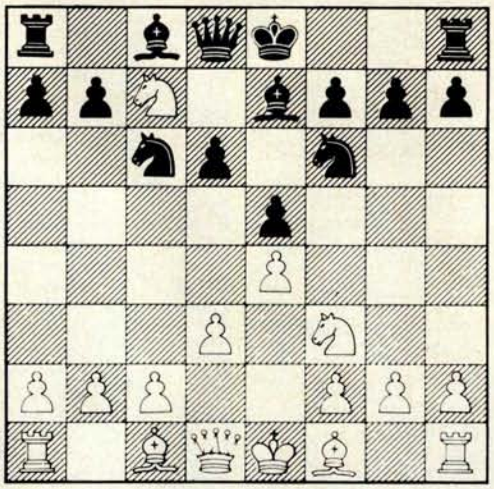


Figure 5: Potential problem arising from the author's evaluation scheme: White's Knight is attacking the Black King and Rook, for which White gains 3/4 of the Rook's value. The Knight is doomed to be captured by Black's Queen, but subtracting the Knight's value from this number still gives White an illusion of material gain. The authors avoided this problem by having the program check to see if the piece that has just moved is subject to capture.

board for a given position. In some of the attacks, the side which has completed a move will have lost points. In others the side about to move will be in danger of losing material. As battles are evaluated one by one, the highest points lost for the side having moved is maintained. This value represents the amount of material this side stands to lose, and it is subtracted directly from the material score. Two scores are maintained for the side about to move: the highest points lost, and the second highest points lost. Both values are saved, because it is assumed that this side will always use its move to rescue its highest value piece. Then only 3/4 of the value is deducted for the loss of the second highest piece, since deducting the entire value would make attacks look as good as captures (see the text box for an example of this procedure). Bonus points are given to each side for additional battles won, but this is still experimental and may not be needed.

One problem that arose with this evaluation scheme was the Knight's tendency to engage in useless forks. In figure 5 we see White's Knight attacking the enemy King and Rook, for which White gains 3/4 of the Rook's value. The Knight is of course doomed to be captured by the Queen and

COSMAC VIP

$249* gets the entire family into creating video games, graphics and control functions. For starters.

COSMAC VIP, the completely assembled, ready-to-operate RCA Video Interface Processor, opens up a whole new world of computer excitement. New challenges in graphics, games and control functions. Yet it's just $249.00.

Easy to buy. And easy to program, thanks to its unique, easy-to-use interpretive language. You get a complete how-to book including programs for 20 games: fun, challenging, and ready to load and record on your cassette.

Simple but powerful.

Built around an RCA COSMAC microprocessor, the VIP is a complete computer system that can grow with you. It has 2K of RAM, expandable on-board to 4K. Plus a ROM monitor, audio tone output to a built-in speaker, power supply, and 8-bit input and output ports for control of relays, sensors, or other peripherals.

Soon RCA will offer options for color graphics and 256 tone sound generation. An optional auxiliary keyboard will open up an exciting world of two-player games.

Take the first step now.

Check your local computer store or electronics distributor for the VIP. Or contact RCA VIP Marketing, New Holland Avenue, Lancaster, PA 17604. Phone (717) 291-5848.

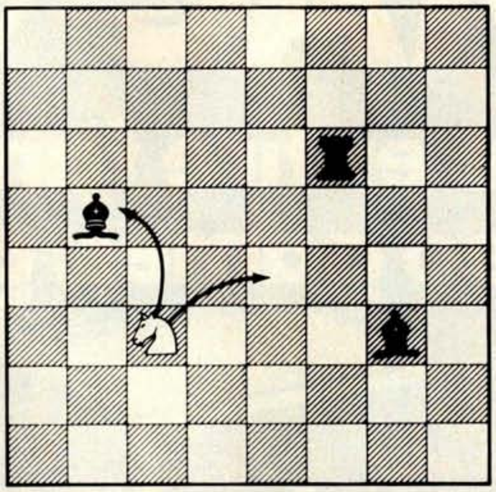
*Suggested retail price. Does not include video monitor or cassette recorder.

The fun way into computers.

RCA

Circle 322 on inquiry card.

BLACK



WHITE

(SARGON)

Making an Immediate Capture More Attractive Than an Attack

In the diagram, the program (White) has two attractive moves: capture the Bishop, or move the Knight so that it simultaneously attacks the Black Rook and the second Bishop. The program assumes that in the latter case Black will protect the Rook (the more valuable of the two pieces) by moving it away. The decision then reduces to one of capturing the first Bishop or simply attacking the second Bishop. In order to insure that the capture takes place, the program assigns 3/4 the normal value to the second Bishop so that the capture looks more attractive. The drawback to this technique is that it precludes the possibility of intentionally avoiding an immediate capture for strategic purposes; this would require a much more complicated program, of course.

can never carry out its threat, but subtracting the Knight's value still gives White an illusion of material gain. Sargon avoids this problem by checking to see if the piece that has just moved is subject to capture. If so, we assume that the side about to move can escape both attacks. The attack with the highest points lost is ignored completely and the attack with the second highest points lost is moved up in its place.

Current Limitations and Future Developments

The problem of the Knight fork as just discussed is only one of a whole set of difficulties. Pinned pieces, the overworked piece, discovered attacks and other motifs can all occur dynamically during play of a board position, but are difficult to evaluate statically. Attacks of this nature are second order attacks and are not considered in the exchange evaluator we described. There are eight possible second order attacks (see table 4). The first group are the discovered attacks and the second group are the transparent attacks. If all of the second order attacks could be taken into account, the evaluation would be much improved. Currently work is being done to accomplish this. Ultimately, of course, the entire board should be considered as a single complex battle. How close to this ideal can static evaluation progress? At what point does static evaluation begin to take more time than the look-ahead itself? Where will compromises in the evaluation be least harmful? Currently in the field of computer chess there is a tendency to downplay the importance of look-ahead in future developments. Has look-ahead reached a dead end? Will it be replaced by a Sargon-like exchange analysis? These are open questions.■

Table 4: Second order attacks. This type of attack, including pinned pieces, overworked pieces, discovered attacks, and so on, is not considered in the exchange evaluator described in this article.

Group	Type	Description
Discovered Attacks	W1 → B1 W2	W1 attacks B1. If B1 moves, W1 defends W2.
	W1 → W2 W3	W1 defends W2. If W2 moves, W1 defends W3.
	W1 → B1 B2	W1 attacks B1. If B1 moves, W1 attacks B2. (PIN)
	W1 → W2 B1	W1 defends W2. If W2 moves, W1 attacks B1.
Transparencies	W1 → B1 → W2	W1 attacks B1. B1 attacks W2. W1 defends W2 through B1.
	W1 → W2 → W3	W1 defends W2. W2 defends W3. W1 defends W3 through W2.
	W1 → B1 → B2	W1 attacks B1. B1 defends B2. W1 attacks B2 through B1. (PIN)
	W1 → W2 → B1	W1 defends W2. W2 attacks B1. W1 attacks B1 through W2.

North Star Systems Go To Work

More and more, you see the North Star HORIZON computer at work: in business, research, and education. Its high performance qualifies the HORIZON for demanding professional applications. Over 10,000 users during the past two years have proven that North Star hardware has the reliability for day-in, day-out computing. The HORIZON is now a serious candidate for any small system installation.

SOFTWARE IS THE KEY TO HORIZON MATURITY

North Star BASIC and DOS have been used to develop hundreds of commercial program packages. These packages establish that North Star software has the completeness and convenience necessary for serious program development. Because of the many independent vendors offering software using North Star BASIC and DOS, the HORIZON owner now has the widest selection of software in the microcomputer industry! Software available includes: word processing, general ledger, accounts payable/receivable, mailing list processing, inventory and income tax preparation. Program development systems such as assemblers, debuggers, editors, PILOT and FORTRAN are also available.

EXPAND YOUR HORIZON

The basic HORIZON computer includes a Z80 microprocessor, 16K bytes of RAM memory, an I/O interface and one Shugart minifloppy disk drive. The HORIZON can be expanded to 60K bytes or more of RAM, three disk drives, and three I/O interfaces. Performance can be enhanced by the addition of the North Star hardware floating point board. Also, S-100 bus products from other manufacturers may be used to expand the HORIZON.

For more information, contact your local computer store.

NORTH ★ STAR
COMPUTERS

2547 Ninth Street
Berkeley, California 94710
(415) 549-0858

Considering a Microcomputer?

Be Sure to Check Out the Product Offerings of the World's Largest Full Line Microcomputer Company.

All Ohio Scientific machines come with microcomputing's fastest full feature BASIC-in-ROM or on-Disk for instant use.

Challenger I Series

Economical computer systems that talk in BASIC.
Ideal for hobbyists, students, education and the home.

	Minimum Configuration	Base Price
Superboard II — World's first complete system on a board including keyboard, video display, audio cassette, BASIC-in-ROM and up to 8K RAM	4K RAM	$ 279
Challenger IP — Fully packaged Superboard II with power supply	4K RAM	$ 349
Challenger IP Disk — Complete mini-floppy system expandable to 32K RAM	16K RAM	$1190

Challenger IIP Series

Ultra high performance BUS oriented microcomputers for personal, educational, research and small business use.

	Minimum Configuration	Base Price
C2-4P — The professional portable	4K RAM	$ 598
C2-8P — The world's most expandable personal machine for business or research applications	4K RAM	$ 799
C2-4P Disk — The ultimate portable	16K RAM	$1464
C2-8P Single Disk — Ideal for education, advanced personal users, etc.	16K RAM	$1738
C2-8P Dual Disk — Most cost effective small business system	32K RAM	$2597

Challenger II Serial Interface Series

Same great features as Challenger IIP Series for those who have serial terminals: small business, education, industry.

	Minimum Configuration	Base Price
C2-0—Great starter for users with a terminal	4K RAM	$ 298
C2-1 — Great timeshare user accessory; cuts costs by running simple BASIC programs locally	4K RAM	$ 498
C2-8S — Highly expandable serial machine, can add disks, etc.	4K RAM	$ 545

Challenger III The Ultimate in Small Computers

The unique three processor system for demanding business, education, research and industrial development applications.

	Minimum Configuration	Base Price
C3-S1 — World's most popular 8″ floppy based microcomputer	32K RAM dual floppys	$3590
C3-OEM — Single package high volume user version of C3-S1	32K RAM dual floppys	$3590
C3-A — Rack mounted multi-user business system directly expandabe to C3-B	48K RAM dual floppys	$5090
C3-B — 74 million byte Winchester disk based system. World's most powerful microcomputer	48K RAM dual floppys	$11,090

OHIO SCIENTIFIC also offers you the broadest line of expansion accessories and the largest selection of affordable software!

Compare the closest Ohio Scientific Model to any other unit you are considering. Compare the performance, real expansion ability, software and price, and you will see why we have become the world's largest full line microcomputer company.

I'm interested in OSI Computers. Send me information on:
- ☐ Personal Computers
- ☐ Small Business Computers
- ☐ Educational Systems
- ☐ Industrial Development Systems
- ☐ I'm enclosing $1.00 for your 64-page small computer buyer's guide.

Ohio residents add 4% tax.

Name ______________________

Address ______________________

City ______________________

State ______________ Zip ________

Phone ______________________

OHIO SCIENTIFIC
1333 S. Chillicothe Road
Aurora, Ohio 44202
(216) 562-3101

Circle 290 on inquiry card.

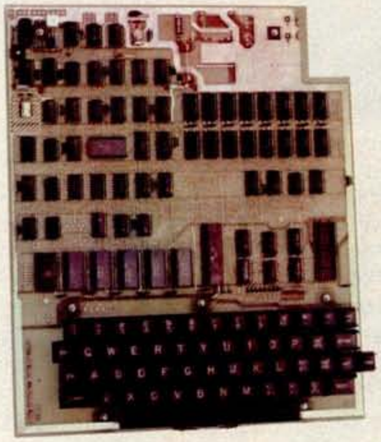
The Age of Affordable Personal Computing Has Finally Arrived.

Ohio Scientific has made a major breakthrough in small computer technology which dramatically reduces the cost of personal computers. By use of custom LSI micro circuits, we have managed to put a complete ultra high performance computer and all necessary interfaces, including the keyboard and power supply, on a single printed circuit board. This new computer actually has more features and higher performance than some home or personal computers that are selling today for up to $2000. It is more powerful than computer systems which cost over $20,000 in the early 1970's.

This new machine can entertain your whole family with spectacular video games and cartoons, made possible by its ultra high resolution graphics and super fast BASIC. It can help you with your personal finances and budget planning, made possible by its decimal arithmetic ability and cassette data storage capabilities. It can assist you in school or industry as an ultra powerful scientific calculator, made possible by its advanced scientific math functions and built-in "immediate" mode which allows complex problem solving without programming! This computer can actually entertain your children while it educates them in topics ranging from naming the Presidents of the United States to tutoring trigonometry all possible by its fast extended BASIC, graphics and data storage ability.

The machine can be economically expanded to assist in your business, remotely control your home, communicate with other computers and perform many other tasks via the broadest line of expansion accessories in the microcomputer industry.

This machine is super easy to use because it communicates naturally in BASIC, an English-like programming language. So you can easily instruct it or program it to do whatever you want, *but you don't have to.* You don't because it comes with a complete software library on cassette including programs for each application stated above. Ohio Scientific also offers you hundreds of inexpensive programs on ready-to-run cassettes. Program it yourself or just enjoy it; the choice is yours.



Ohio Scientific offers you this remarkable new computer two ways.

Challenger 1P $349
Fully packaged with power supply. Just plug in a video monitor or TV through an RF converter to be up and running.

Superboard II $279
For electronic buffs. Fully assembled and tested. Requires +5V. at 3 Amps and a video monitor or TV with RF converter to be up and running.

Standard Features

- Uses the ultra powerful 6502 microprocessor
- 8K Microsoft BASIC-in-ROM
  Full feature BASIC runs faster than currently available personal computers and all 8080-based business computers.
- 4K static RAM on board expandable to 8K
- Full 53-key keyboard with upper/lower case and user programmability
- Kansas City standard audio cassette interface for high reliability
- Full machine code monitor and I/O utilities in ROM
- Direct access video display has 1K of dedicated memory (besides 4K user memory), features upper case, lower case, graphics and gaming characters for an effective screen resolution of up to 256 by 256 points. Normal TV's with overscan display about 24 rows of 24 characters; without overscan up to 30 X 30 characters.

Extras

- Available expander board features 24K static RAM (additional), dual mini-floppy interface, port adapter for printer and modem and an OSI 48 line expansion interface.
- Assembler/editor and extended machine code monitor available.

Interested in a bigger system? Ohio Scientific offers 15 other models of microcomputer systems ranging from single board units to 74 million byte hard disk systems.

ORDER FORM

Order direct or from your local Ohio Scientific dealer.

☐ I'm interested. Send me information on your:
☐ Personal Computers ☐ Business Systems
☐ Send me a Superboard II $279 enclosed
☐ Send me a Challenger 1P $349 enclosed
☐ Include 4 more K of RAM (8K Total) $69 more enclosed

Name ______________________

Address ______________________

City __________ State __________ Zip __________

Payment by: BAC (VISA) ____ Master Charge ____ Money Order ____

Credit Card Account # ______________________

Expires ________ Interbank #(Master Charge) ________
Ohio Residents add 4% Sales Tax

TOTAL CHARGED OR ENCLOSED ________
All orders shipped insured UPS unless otherwise requested. FOB Aurora, OH

OHIO SCIENTIFIC
America's Largest Full Line Microcomputer Company
1333 S. Chillicothe Road • Aurora, Ohio 44202 (216) 562-3101

The C3-S1

by Ohio Scientific

Possibly the world's most popular floppy disk based microcomputer.

Since its introduction in August, 1977, the Challenger III has gained tremendous acceptance in small business, educational and industrial development applications. Thousands of C3-S1's have been delivered and today hundreds of C3-S1 demonstrator units are set up at computer retailers around the country.

Why has the Challenger III become so successful in the fiercely competitive microcomputer industry? Here are just a few of the possible reasons.

- The Challenger III is the fastest microcomputer in BASIC (see "BASIC Timing Comparisons," *Kilobaud*, October, 1977, where Ohio Scientific out benchmarks all competitors).
- The Challenger III is the only computer system with a 6502A, 6800 and Z-80 offering the programmer all popular micros for maximum versatility.
- The C3 is backed by the largest base of systems level software for any microcomputer system including:

  For the 6502A:
  - Microsoft 6 and 9 Digit BASIC
  - Assembler Editor
  - Word Processor
  - OS-65D Development DOS
  - OS-65U End User DOS with Extended BASIC
    - For Floppys
    - Winchester Hard Disks
    - Multi-users (Level 2)
    - Distributed Processing (Level 3)

  For the 6800:
  - Floppy DOS
  - Assembler Editor

  For the Z-80:
  - Floppy DOS
  - Microsoft Disk Extended BASIC
  - Microsoft FORTRAN
  - Microsoft COBOL
  - Macro Assembler and Editor
  - And Much More

- The C3 supports OS-65U, the ultra high performance "virtual data memory" DOS for floppys and hard disks which makes complex file structures like multi-key ISAM easy to use.
- The C3 is backed by a large library of applications programs and can make use of the tremendous amount of BASIC programs offered by independent suppliers and publishers because it uses Microsoft BASIC, the standard of the industry. Complete turnkey and custom business packages are available for the C3 from most OHIO SCIENTIFIC DEALERS.
- The C3 electronics and software are available in alternate mechanical configurations for special applications including the C3-OEM for volume users and the C3 letter series (C3-A, C3-B) which are optimized for use with hard disks.
- C3 systems are always delivered ready to use with 32K static RAM, dual floppys for 500K bytes of on-line storage and an RS-232 port strappable from 75 to 19,200 baud all *standard* in the minimum configuration.
- C3 systems offer the greatest expansion capability in the microcomputer industry. The C3 series supports OHIO SCIENTIFIC'S full line of over 40 expansion accessories. The maximum configuration is 768K bytes RAM, four 74 million byte Winchester hard disks (CD-74), 16 communications ports, real time clock, line printer, Word Processing printer and numerous control interfaces.
- C3 systems have phenomenal performance-to-cost ratios. The C3-S1 base price with 32K RAM, dual floppys, RS-232 port complete with 8K BASIC and DOS is under $3600 and expansion accessories are comparably priced. For example, the CD-74, 74 million byte Winchester disk complete with interface and OS-65U operating system at about $6000.

The C3 series is quite possibly so successful because it offers the highest hardware performance, best software support, most versatility and greatest expandability in the microcomputer systems market at nearly the lowest price in the industry.

For more information, contact your local OHIO SCIENTIFIC DEALER or the factory at (216) 562-3101.

OHIO SCIENTIFIC

1333 S. Chillicothe Road • Aurora, Ohio 44202

Circle 290 on inquiry card.

The C3-B
by Ohio Scientific

The world's most powerful microcomputer system is far more affordable than you may think.*

STANDARD FEATURES:

- 74 million byte Winchester technology disk drive yields mainframe class file access speeds and capacity.
- High level data file software makes high performance file structures like multikey ISAM easy to use.
- Triple processor CPU with 6502A, 6800 and Z-80 gives the programmer the best of all worlds in performance and versatility.
- The included 6502A based extended disk BASIC by Microsoft out-benchmarks every micro available, including 4 MHz Z-80 and LSI-11 with extended arithmetic.
- 48K of high reliability static RAM is standard.
- High density 8″ floppys provide program and data mobility from machine to machine.
- Completely integrated mechanical system with UL-recognized power supplies; continuous duty cycle cooling; modular construction and rack slide mounted subassemblies.
- Based on a 16 slot Bus-oriented architecture with only 7 slots used in the base machine.
- Directly expandable to 300 megabytes of disk, 768K of RAM in 16 partitions, 16 communication ports, plus console and three printers.
- C3-B's have been in production since February, 1978, and are available now on very reasonable delivery schedules.

The C3-B was designed by Ohio Scientific as the state of art in small business computing. The system places its power where it's needed in the small business environment; in the data files. The C3-B's advanced Winchester technology disk, coupled with its smart controller and dedicated high speed memory channel, gives the C3-B data file performance comparable with today's most powerful maxi-computers.

Yet, the C3-B costs only slightly more than many floppy only computers but offers at least a thousand times performance improvement over such machines (50 times storage capacity multiplied by 20 times access speed improvement).

But what if your business client cannot justify starting with a C3-B? Then start with Ohio Scientific's inexpensive C3-S1 floppy disk based system running OS-65U. When he is ready, add the CD-74 big disk and directly transfer programs and files from floppy to big disk with *NO* modifications.

That's upward expandability!

*Rack as shown above complete with 74 megabyte disk, dual floppys, 48K of static RAM, OS-65U operating system and one CRT terminal under $13,000.

Multiple terminal systems with printers and applications software are priced in the mid-20's.

OHIO SCIENTIFIC
1333 S. Chillicothe Road • Aurora, Ohio 44202
(216) 562-3101

Circle 290 on inquiry card.

Functional Approximations

Fred Ruckdeschel
773 John Glenn Blvd
Webster NY 14580

Many BASIC interpreters presently in use have limited function calculation capability. Almost all interpreters larger than 4 K bytes have built-in routines for determining square roots, logarithms and exponentials, as well as some trigonometric functions like sine and cosine. Several interpreters also provide inverse trigonometric functions via the arctangent function.

However, many minicomputer and microcomputer systems do not have such software. Also, some of the newer microprocessors presently do not have much support software. I have recently been developing software for use with the North Star Micro-Disk system; specifically, working with North Star BASIC, version 6, release 2. The North Star software contains sine and cosine functions but no inverse trigonometric routines. This deficiency led to a cursory investigation of series approximations which very quickly showed that Taylor series expansions are *not* generally optimum for computer use.

Taylor Series Expansion

In general, any real, continuous function f(x) having defined derivatives may be expressed as a polynomial expansion about a reference point x_o with the longhand representation:

$$f(x) = f(x_o) + f^{(1)}(x_o)(x-x_o) + f^{(2)}(x_o)(x-x_o)^2/2! + \ldots \quad (1a)$$

Observe that $f^{(n)}(x_o)$ is defined to be the *n*th order derivative of f(x) evaluated at x equal to x_o. The shorthand equivalent of equation (1a) is:

$$f(x) = \sum_{n=0}^{\infty} f^{(n)}(x_o)(x-x_o)^n/n!$$

or:

$$f(x) = \lim_{n\to\infty} f_n(x) \quad (1b)$$

where:

$$0! \equiv 1$$

$$f^{(0)}(x_o) \equiv f(x_o)$$

$$f_n(x) \equiv \sum_{m=0}^{n} f^{(m)}(x_o)(x-x_o)^m/m!$$

Equation (1b) is in a form which is convenient in terms of BASIC's FOR-NEXT loop evaluation, which may account for its popularity with computer programmers.

Many programmers who are faced with approximating a function use a Taylor series expansion along with a convergence test such as:

$$|f_n(x) - f_{n-1}(x)| < \epsilon. \quad (2)$$

Presumably when the difference between the approximation using n terms and that using n−1 terms is less than ϵ, the accuracy of $f_n(x)$ in approximating f(x) is better than ϵ. This assumption can be grossly in error depending on the function being evaluated. Slowly converging series often present problems in this respect.

The reasons for not indiscriminately using equations (1) or (2) may be demonstrated by considering the sine and arctangent functions. The Taylor series expansion around the zero reference point (known as the MacLaurin series) for the sine function is:

$$\sin(x) = x - x^3/3! + x^5/5! - \ldots \quad (3a)$$

or:

$$\sin(x) = \sum_{n=0}^{\infty} (-1)^n x^{(2n+1)}/(2n+1)!\,. \quad (3b)$$

If we apply equation (2) as the test for accuracy, we have

$$\frac{x^{(2n+1)}}{(2n+1)!} < \epsilon. \quad (4)$$

If x is one radian (approximately 57°) and the required accuracy is 10^{-8}, then equation (4) indicates that seven terms are required in the series for the specified accuracy. The test for residual may be done more correctly in principle by noting that the series is uniformly convergent, having terms alternating in sign. Thus the absolute error of the approximation is less than the absolute value of the last term included.

Therefore we again conclude that seven terms are needed for 10^{-8} accuracy in approximately calculating the sine of one radian. In the case of alternating series having terms which monotonically decrease in absolute value, equation (2) is applicable.

If, instead, we are interested in the sine of −1, the accuracy test would again call for seven terms. However, this time, because the signs of the terms do not alternate, a ratio test would have to be applied to correctly examine the residual, yielding the same result for the required number of terms. In general, if the series is rapidly converging, equation (2) is an adequate test for accuracy.

So far we have assumed that the computer is perfect in terms of roundoff error. In some eight decimal place accuracy interpreters and compilers, the computer rounds off to the eighth decimal place by consistently rounding down or up. Quite often the direction is down, since this corresponds to simple truncation. In those cases the ensuing error in calculating n terms in a series expansion is on the order of $\pm (n/2) \times 10^{-8}$. (If the software is sophisticated enough to round to the nearest value, the error becomes $\sqrt{n} \times 10^{-8}$ for eight bits.) For the sine expansion example given above, the expected truncated series accuracy of 10^{-8} would be reduced to approximately 10^{-7} because of roundoff error. It is shown in the next section that a different series expansion for sine can be used which contains only five terms and which gives an error of less than 10^{-8} before roundoff, along with a generally better answer when roundoff is considered.

So far the observation is that the error test, equation (2), leads to a correct estimate for the required number of terms in the case of the Taylor series approximation for sine. There is an approximation in which only five terms are sufficient, which is described later. These are not very exciting conclusions, largely because the Taylor series expansion for sine in the first quadrant is very rapidly convergent and thus quite adequate. Another important function, arctangent, is not nearly as quickly convergent in some regions of its limited convergence interval. The MacLaurin series expansion for arctangent is

$$\arctan(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \ldots$$

$$= \sum_{n=0}^{\infty} (-1)^n x^{(2n+1)}/(2n+1)$$

$$(-1 < x \leq 1). \qquad (5)$$

This expansion has difficulty converging near x equal to positive or negative one. When x equals negative one the series diverges, although a finite answer ($\pi/4$) exists, whereas at x equal to one the series converges very slowly to $\pi/4$, or 45°. Only the region between 0° and 45° (0 and $\pi/4$ radians) need be used for the expansion, since trigonometric identities exist for extending the inversion to other regions (eg: $\arctan(x) = \pi/2 - \arctan(1/x)$ for $x \geqslant 0$). The technique of *range* reduction for improving accuracy will be considered in a later section.

Consider the use of equation (5) near an x value of one. Using equation (2) to determine the number of terms required for 10^{-8} accuracy, we get a value of 5×10^7 for n. Applying the alternating sign convergence test also leads to an n value of 5×10^7. In either case, this series expansion for arctangent is unusable. Even if the computer were fast enough so that the approximation could be calculated in an acceptable length of time, the ensuing roundoff error would be prohibitive. It is shown in the next section that there exist other series expansions which are better than (in terms of relative error) 4×10^{-8} and which have only eight terms. The series presented are operable either over the range $1 \leqslant |x|$ (note the equality sign) or over the range $0 \leqslant x < \infty$.

We observe that the error test, equation (2), would fail dismally for the Taylor series arctangent expansion near an x value of negative one. The test would again predict that on the order of 5×10^{-7} terms would be required for 10^{-8} accuracy, while in actuality the series diverged. In this case the Taylor series expansion is not rapidly converging, thus leading to a failure in equation (2). In general, equation (2) is practical only if:

- The series has decreasing alternating sign terms.
- The series has decreasing terms in which the absolute value of the ratio of neighboring terms is greater than 10 for all pairs of terms past the termination point. That is, if:

$$f(x) = \sum_{n=0}^{\infty} c_n x^n,$$

then beyond the termination point it is required that:

$$\left| \frac{c_n x^n}{c_{n+1} x^{n+1}} \right| = \left| \frac{c_n}{c_{n+1} x} \right| > 10.$$

Observe that this latter restriction is valid only for MacLaurin series expansions. For Taylor series expansions, replace x with $(x-x_o)$. Observe that small values of $(x-x_o)$ are very conducive to rapid convergence, but are not sufficient.

Approximate Series Expansions

Taylor series expansions have many nice properties. However, one of their less desirable properties is that they are not the optimal expansions for a given argument interval when a truncated series is to be used. This may be easily seen by a simple example.

Consider the single term approximation to sin(x) over the interval $0 \leqslant x \leqslant \pi/2$ (see figures 1 and 2). Although the accuracy of approximating the value of sin(x) with x is good for small values of x, it leads to an error of 0.57 at $\pi/2$. If instead we wish to minimize the maximum absolute error over

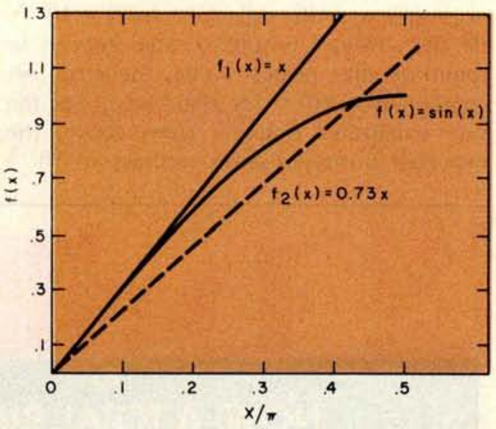


Figure 1: Single term approximations to sin(x) over the range $0 \leqslant x \leqslant \pi/2$. $f_1(x)$ is the truncated MacLaurin series expansion; $f_2(x)$ is the single term fit minimizing the maximum error.

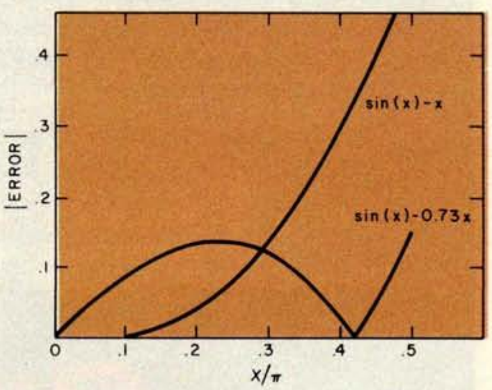


Figure 2: Graph of the errors due to single term approximations for the sine function.

Software for the Percom LFD-400

Disk Operating and File Management Systems

INDEX™: The most advanced Disk Operating and File Management System available for the 6800. INterrupt Driven EXecutive operating system features file-and-device-independent, queue-buffered character stream I/O. Linked-File disk architecture, with automatic file creation and allocation for ASCII and binary files, supports sequential and semi-random access disk files. Multilevel file name directory includes name, extension, version, protection, and date. Requires 8K RAM at $AOOO. Diskette includes numerous utilities. $99.95

MINIDOS-PLUSX: An easy to use DOS for the small computing system. Supports up to 31 named files. Available on ROM or diskette complete with source listing. $39.95

Basic Interpreters and Compilers

SUPER BASIC: a 10K extended disk BASIC interpreter for the 6800. Faster than SWTP BASIC, 9-digit accuracy, program CHAINing, BASE O subscripting, improved error reporting, disk data files, and print column alignment. Program and data files may be prepared using one of the Text Editors described below $49.95

BASIC BANDAID™: Turn SWTP 8K BASIC into a respectable random access data file disk BASIC. Includes many speed improvements and program CHAINing. When ordering, specify version (2.0, 2.2, or 2.3). Complete with listing. $17.95

STRUBAL™: A STRUctured BAsic Language Compiler for the serious professional programmer. Includes elements of BASIC, PL/M, and assembly language. Features 10-digit floating point, strings, scientific functions, and 2-dimensional arrays. Requires 16K RAM memory and LINKING LOADER (see below). Complete with RUN-TIME and FLOATING POINT packages $99.95

Text Editors

EDIT68: Hemenway Associates powerful disk-based text editor. May be used to create programs and data files. In addition to SEARCH, CHANGE, DELETE, and MOVE functions, EDIT68 supports MACROs which perform complex repetitive editing functions efficiently. Runs in only 6K RAM. "Spooling" permits text files much larger than the available RAM memory to be created and edited. $29.95

TOUCHUP™: If you already have the TSC Text Editor, TOUCHUP will convert it into a disk-based text editor. ROLL function permits text files much larger than the available RAM memory to be created and edited. TOUCHUP is supplied on diskette complete with source listing. $17.95

ASSEMBLERS

PERCOM 6800 SYMBOLIC ASSEMBLER: Requires only 8K of RAM memory to assemble programs with 200 labels. More convenient to use than most assemblers; assembly options may be specified at time of assembly. $29.95

Complete listing of above assembler $29.95

MACRO-RELOCATING ASSEMBLER: This Hemenway Associates assembler is for the programming professional. Generates relocatable and linkable object code. MACRO facility permits nested macro calls. Permits conditional assembly. Requires 16K RAM $49.95

LINKING LOADER for STRUBAL and the above assembler $19.95

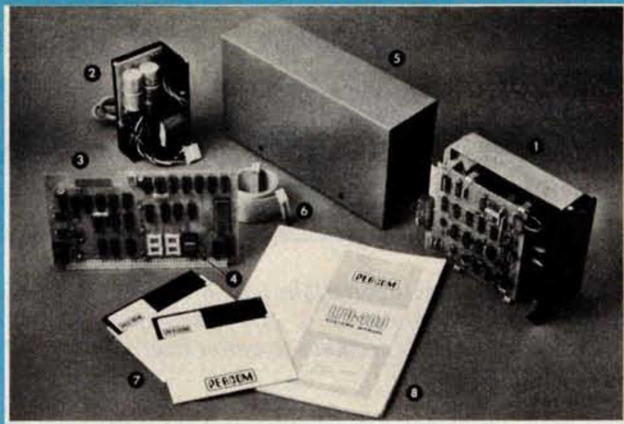
Business Applications

GENERAL LEDGER SYSTEM $199.95
FULL-FUNCTION MAILING LIST .... $ 99.95

Write or call for our complete software catalog.

PERCOM DATA COMPANY, INC.
Dept BP • 318 Barnes • Garland TX 75042
(214) 272-3421

For Your SWTP 6800 Computer . . .



PERCOM's™ Assembled and Tested FLOPPY DISK SYSTEM

the LFD-400

only 599^{95} shipping paid

The LFD-400 is ready to plug in and run the moment you receive it. Nothing else to buy! Not even extra memory!

YOU GET:

1. The popular Shugart SA 400 minifloppy™ drive. Drive alignment is double checked by PerCom before shipment.
2. The drive power supply—fully assembled and tested.
3. LFD-400 Controller/Interface—plugs into the SS-50 bus • accommodates three 2708 EPROMs • fully assembled and tested.
4. MINIDOS™—the remarkable LFD-400 disk operating system on a 2708 EPROM • plugs into the LFD-400 Controller card • no extra memory required • no "booting" needed.
5. Attractive metal enclosure.
6. Interconnecting cable—fully assembled and tested.
7. Two diskettes—one blank, the other containing numerous software routines including patches for SWTP 8K BASIC and the TSC Editor/Assembler.
8. 70-page instruction manual—includes operating instructions, schematics, service procedures, and the complete listing of MINIDOS™.
9. Technical Memo updates—helpful hints which supplement the manual instructions.
10. 90-day limited warranty.

Minifloppy is a trademark of Shugart Associates.
MINIDOS is a trademark of PERCOM Data Company, Inc.

The LFD-400 is readily expanded to either two or three drives. Write for details. Send for our free brochure for more information about the LFD-400 Floppy Disk System and LFD-400 software.

To save you money, the LFD-400 Floppy Disk System is available only from PerCom. Because of the special pricing, group and dealer discounts are not available.

MC and VISA welcome. COD orders require 30% deposit plus 5% handling charge. Allow three extra weeks if payment is by personal check. The LFD-400 Floppy Disk System is available immediately. Allow three weeks for testing and transportation. Texas residents add 5% sales tax.

PERCOM™

PERCOM DATA COMPANY, INC.
Dept B 318 BARNES • GARLAND, TX. 75042
(214) 272-3421
PERCOM™ *'peripherals for personal computing'*

Circle 301 on inquiry card.

Term	MacLaurin Coefficients	Optimal Coefficients
x	+ 1.00000000	+ 1.00000000
x^3	− 0.16666667	− 0.41123328
x^5	+ 0.00083333333	+ 0.050732026
x^7	− 0.00019841270	− 0.0029754116
x^9	+ 2.755732×10^{-6}	+ 0.000096437832
x^{11}	− 2.5052109×10^{-8}	
x^{13}	+ $1.6059045 \times 10^{-10}$	

Table 1: Coefficients for the MacLaurin and optimal series expansions for sin(x). The optimized interval is $-\pi/2 \leqslant x \leqslant \pi/2$. In this interval, the optimal (relative error) series is accurate to better than 5×10^{-9} when more accurate coefficients are used.

that interval, then approximating sin(x) by 0.73x is better. The maximum absolute error is in this case less than 0.15, with two error maxima.

If our criterion is to minimize the relative error (percent deviation), the MacLaurin series single term truncation leads to 57 percent error at $\pi/2$, while the previous minimized (relative to maximum absolute error) approximation is off by less than 27 percent, with the maximum relative error occurring at zero. The maximum relative error can be further reduced to about 22 percent by using 0.78x as the approximation for sin(x). The maximum relative errors in this case occur at x values of 0 and $\pi/2$. This series is considered to be the *minimax* or optimal series expansion for sin(x) over the range $-\pi/2 \leqslant x \leqslant \pi/2$, given that one term is allowed.

The above example illustrates the fact that either by the criterion of absolute error or the criterion of relative error, the truncated MacLaurin series for the sine function is not optimal for approximation. Also observe that the coefficient (0.73 for absolute error or 0.78 for relative error) in the optimal approximation is dependent on the interval chosen. The more the interval is restricted to that region surrounding zero, the closer the coefficient is to unity. In general, the more the interval is restricted to the region immediately surrounding the Taylor series expansion point, the more the coefficients in the optimal series approximate those of the Taylor series expansion. This is because the highest convergence rate occurs near the expansion point, and it is hard to do better than that.

Continuing with the sine example, table 1 shows the series coefficients that would be used in the MacLaurin and optimal series expansions to give a desired accuracy (excluding roundoff) of approximately 10^{-8}. Observe that because the *relative* error is the criterion, the first coefficient in the optimal series is the same (to eight places) as that in the MacLaurin series. This is required because for very small x, sin(x) is approximately equal to x. Note, however, that all the other coefficients are drastically different, though the signs of the coefficients are preserved. The sign preservation is expected from a consideration of the derivatives.

A corresponding series expansion comparison for the arctangent is shown in table 2. Observe that about 10^7 terms are required in the MacLaurin series for x equal to 1 to give the same order of accuracy as the optimal series evaluated at that point. The situation for the MacLaurin series is even worse in the vicinity of negative one.

It is interesting to note that in the series expansion comparison shown in table 2 the signs of the coefficients are conserved, and the coefficients themselves bear some semblance to one another up to the fifth term. After that the coefficients look quite different.

Variations

The optimal series coefficients given in table 2 for the arctangent approximation were stipulated to be optimal over the interval $-1 \leqslant x \leqslant 1$. This restricts us to the angular range of ±45° about 0° ($\pm\pi/4$ radians around 0 radians). A trigonometric identity was given earlier which allowed continuation through 90° ($\pi/2$ radians). However, this extra range changing step is not necessary, since another series expansion exists which is optimized over the interval $0 \leqslant x \leqslant \infty$. However, it does not have a simple x^n dependence:

$$\arctan(x) = \pi/4 + \sum_{n=0}^{7} C_n \left(\frac{x-1}{x+1}\right)^{(2n+1)} . \quad (6)$$

This series is interesting in that it uses the same coefficients as given in table 2, and also gives the same order of relative accuracy. To execute this series on a computer one need only define x to be $(x-1)/(x+1)$ and proceed as before.

Term	MacLaurin Coefficients	Optimal Coefficients
x	+ 1.00000000	+ 0.99999933
x^3	− 0.33333333	− 0.33329856
x^5	+ 0.20000000	+ 0.19946536
x^7	− 0.1428571	− 0.13908534
x^9	+ 0.11111111	+ 0.096420044
x^{11}	− 0.090909090	− 0.055909886
x^{13}	+ 0.07692308	+ 0.021861229
x^{15}	− 0.066666667	− 0.0040540580

Table 2: Coefficients for the MacLaurin and optimal series expansions for the arctangent. Accuracy (relative error) is better than 4×10^{-8}. The interval of convergence for the MacLaurin series is $-1 < x \leqslant 1$. The interval of convergence for the optimal series is $-1 \leqslant x \leqslant 1$.

Powerful Software

with Extended Documentation offered by

PRS

PRS–
THE PROGRAM OF THE MONTH CORPORATION
UNVEILS COMPREHENSIVE CONCEPT
IN SOFTWARE PRESENTATION

UNIQUE DOCUMENTATION

Unique documentation gives you clear, complete and instructive text in a graphically appealing manual. This documentation doesn't only provide step by step "how to" but also is geared to help you to understand hardware-software interactions, and to promote further applications.

POWERFUL CODE

Whether it be for games, applications (home and business), or sophisticated programming tools and monitors, PRS permeates the design of its unique code with "human engineering." And you'll appreciate the delicate balance between machine-human dialogue and resident program-space in memory.

BEAUTIFUL CASSETTE-FOLDER

PRS packages your manual and cassette in an elegant and durable gold-stamped folder.

AN ARRAY OF PROGRAMS ARE AT YOUR FINGERTIPS

PRS presently offers two S-100 compatible program packages: "MICROFILE," a DATA FILE MANAGEMENT program, and "DDS II," the famous DYNAMIC DEBUGGING SYSTEM. Also, an Extensive Function Plotter (A2FP) is available for the Apple II. New programs will be released at regular intervals.

YOU'LL DISCOVER EXTENSIVE PROGRAM CAPABILITIES

With "MICROFILE," a new DATA FILE MANAGEMENT program, you can:

- Manage any list of items
- Edit information already on file
- Sort items alphanumerically upwards or downwards
- Search by primary and/or secondary keys (labels)
- List and print selected or all items
- Total sums of columns
- Justify right or left margins of each column
- Save and retrieve data with mass storage media

PROGRAM APPLICATION EXPANDS YOUR COMPUTER USAGE

For example, "MICROFILE" manages: Home inventory • Listing of musical tapes • Calendar of activities • Vacation itineraries • Checks by categories • Home-upkeep tickler files • Family historical dates • Health records.

PRS SUPPORTS YOUR MAJOR BRANDS

APPLE, SOL, SORCERER, TRS-80, and other systems with Z-80, 8080 and 6502 processors.

VISIT YOUR NEAREST DEALER TODAY

Let him demonstrate the excellence of PRS programs.

Or write for the name of a dealer in your area to:
PRS–THE PROGRAM OF THE MONTH CORPORATION
257 Central Park West, New York, N.Y. 10024

New Product Releases From

PRS THE PROGRAM OF THE MONTH CORPORATION
257 CENTRAL PARK WEST
NEW YORK, N.Y. 10024

Gentlemen:

☐ Please place my name on your priority mailing list to receive your descriptive advance releases off all new programs developed by -PRS- at regular intervals.

☐ Please send me a list of dealers in my area.

Name ____________________

Corporation ____________________

Address ____________________

City ________ State ________ Zip ________

REFERENCE: 9B8

(a)

Form:

$$\text{Log}_{10}\ X = \frac{1}{2} + \sum_{n=0}^{4} C_n\ \gamma^{(2n+1)}$$

where $\gamma = (X-\sqrt{10})/(X+\sqrt{10})$

Coefficients:

C_0	=	0.86859172
C_1	=	0.28933552
C_2	=	0.17752207
C_3	=	0.094376476
C_4	=	0.19133771

(b)

Form:

$$10^x = 1 + \sum_{n=1}^{6} C_n\ x^n$$

Coefficients:

C_1	=	1.15128759
C_2	=	0.66284315
C_3	=	0.25360332
C_4	=	0.075467547
C_5	=	0.013420940
C_6	=	0.005654902

(c)

Form:

$$e^{-x} \doteq \left[\sum_{n=0}^{6} C_n\ x^n\right]^{(-4)}$$

Coefficients:

C_0	=	1.00000000
C_1	=	0.24999868
C_2	=	0.031257583
C_3	=	0.0025913712
C_4	=	0.0001715620
C_5	=	0.0000054302
C_6	=	0.0000006906

Table 3: Several minimax polynomial expansions. Table 3a is for the case of the base 10 logarithm function. This expansion is applicable over the range $1 \leqslant x \leqslant 10$ with a relative error of better than 1.5×10^{-7}. Table 3b is the inverse of the base 10 logarithm function. The applicable range is $0 \leqslant x \leqslant 1$ with a relative error of better than 1.5×10^{-7}. Table 3c is the expansion for the negative powers of e. This expansion is applicable over the range $0 \leqslant x < \infty$ with a relative accuracy of better than 3×10^{-7}.

Form:

$$\cos(x) \doteq \frac{1 + A_1\ x^2 + A_2\ x^4 + A_3\ x^6}{1 + B_1\ x^2 + B_2\ x^4 + B_3\ x^6}$$

Coefficients:

A_1	=	− 0.47059579
A_2	=	0.027388290
A_3	=	− 0.00037234227
B_1	=	0.029404212
B_2	=	0.00042372881
B_3	=	0.000003235543

Table 4: Rational polynomial approximation for cosine which is applicable over the range $-1 \leqslant x \leqslant 1$. The maximum relative error over this range is better than 10^{-8}.

$$\text{erf}(x) = 1 - \text{erfc}(x) = 1 - \frac{2}{\sqrt{\pi}}\ e^{-x^2} \times \frac{1}{2x} \times$$

$$\left\{1 - \frac{1}{(2x^2)} + \frac{1 \times 3}{(2x^2)^2} - \frac{1 \times 3 \times 5}{(2x^2)^3} + \frac{1 \times 3 \times 5 \times 7}{(2x^2)^4} - \cdots\right\}$$

Table 5: An asymptotic series approximation to the error function. Range and accuracy are dependent on the smallest term in the series (before it diverges). The series is truncated at the smallest term, which represents the error. Including more terms increases the error.

Hasting's book *Approximations for Digital Computers* (see references) presents several such optimal expansions, three of which are presented in table 3. Some alterations have been made to put this information into a form readily usable with typical microcomputer software.

Polynomial expansions are not the only forms useful for function approximations. Another important type is the *rational* polynomial. Table 4 shows a rational polynomial approximation for cos(x). The rational polynomial approximation to the cosine is slightly more accurate than the minimax approximation given earlier for the sine. However, the argument range of this cosine approximation is inconveniently restricted. In cases such as this it is the user's choice as to which approximation to apply.

There is a class of series expansions in which an improved approximation is not insured by including more terms. This class of series approximations is called *asymptotic*. An example of such a series is shown in table 5 for the error function. The reason this series approximation has problems when too many terms are included may be seen by examining the ratio of a general pair of terms:

$$R = \left|\frac{f_{n+1} - f_n}{f_n - f_{n-1}}\right| = \frac{2n-1}{2x^2}. \quad (7)$$

It is apparent that one can always go out far enough in the series expansion to find a value of n such that R is greater than 1. It is also possible that if we define a suffi-

$$y_{n+1} = y_n - \frac{1}{2}\left(y_n - x/y_n\right) \qquad n = 0, 1, 2 \ldots$$

$$y_0 = \frac{0.154116 + 1.893872\,X}{1 + 1.047988\,X}$$

Table 6: Iterative technique for determining the square root of x. The accepted range is from zero to infinity. The y_0 term is an approximation for the starting value over the interval $1/16 \leq x \leq 1$. The y_0 relative error is better than 0.025.

$$y_{n+1} = y_n - \left\{\frac{y_n^2 - x/y_n}{2y_n + x/y_n^2}\right\} \qquad n = 0, 1, 2 \ldots$$

Table 7: Iteration technique for determining the cubic root of x over the range zero to infinity. The y_0 term is the same as that used for the square root.

ciently small error limit, ϵ, for use in the accuracy test, equation (2), the computer may never find a condition satisfying that relation. In such a situation the computer may continue the calculation of terms until an overflow occurs. A way to avoid this problem is to also test whether or not the minimum term (smallest in absolute value) has been passed, and if so, stop the calculation on that term. The accuracy of the thus truncated series is approximated by the value of that last term.

Quite often the series or rational polynomial approximation approach may be surpassed in accuracy and possibly speed by an iterative technique. In fact, such iterative techniques can often also be put in a form in which roundoff error has a minimal impact on final accuracy.

Tables 6 and 7 show iterative approximations to $f(x)=\sqrt{x}$ and $f(x) = \sqrt[3]{x}$. These forms were derived from the Newton method for finding the zeroes of functions (see reference 2). The relations have been algebraically written such that the previous approximation in the iteration is improved upon by the subtraction of a correction term (which may be either positive or negative). Convergence is generally very good and can be significantly improved upon by using a *predictor* for the correction, though the price paid is a more complicated routine. The reader is referred to standard texts on numerical methods for descriptions of predictor-corrector techniques.

Table 8 gives a general iterative formula for determining the m^{th} root of any nonnegative real number. Although it is potentially more accurate than using the logarithm-antilogarithm approximation routines, it suffers from speed limitations for large values of m, since many multiplications may be required (what would one do with $x^{1/137}$?).

As a final but very important example of an iterative approximation to a common function, arctangent, see table 9. This technique is due to Gauss and is a sure way to obtain accuracy limited only by the number of digits carried by your software; there is no cumulative roundoff error. Note that implementation of this method requires the use of a square root function, which we can have as another subroutine. Although the author has no literature information regarding the convergence of this iteration, the form looks fast in convergence but slow in calculation.

For example, listing 1 shows sample runs for x = 0.1, 1, 100 and 10^{30}. It seems that only 12 iterations maximum are required to attain a relative accuracy of 10^{-7}. However, the execution time for the BASIC program shown runs between 3 (x = 0.1) and 4 ($x = 10^{30}$) seconds, which is not very fast. Use of the optimal series, shown in table 2, cuts the execution time down to 90 ms with the same level of accuracy. However, a numerical overflow occurs when $x = 10^{30}$. The conclusion is that for 10^{-7} relative accuracy, the optimal series approximation is preferred over the iteration technique since it is about 40 times faster in execution. There is probably some hidden physical law which states that the elegance versus execution time product is a constant.

Some of the approximations we have discussed are not applicable over the entire possible range of arguments. The approx-

$$y_{n+1} = y_n - \left\{\frac{y_n^m - x}{m\, y_n^{m-1}}\right\} \qquad \begin{matrix} m = 2, 3, 4 \ldots \\ n = 0, 1, 2, 3 \ldots \end{matrix}$$

Table 8: Iteration technique for determining the mth *root of x. In this case, x is allowed to vary from zero to infinity. The convergence of this formula is quadratic.*

Define: $a_0 = (1+x^2)^{-1/2}$

$b_0 = 1$

Iterate: $a_{n+1} = \frac{1}{2}(a_n + b_n)$

$b_{n+1} = \sqrt{a_{n+1} \cdot b_n}$

Convergence Test: compare a_n and b_n (they approach one another)

Final Calculation: $\arctan(x) \doteq \dfrac{x}{\sqrt{a_n b_n (1+x^2)}}$

Table 9: Gauss iteration technique for approximating the arctangent function. This formula has been modified from a formula in the book Numerical Methods That Work *(see references) to improve the accuracy in the final calculation. The variable a_n has been replaced by the expression $\sqrt{a_n b_n}$.*

First,
you could play
SCELBI's
GALAXY
on your 8008 . . .

Then,
you could play
SCELBI's
GALAXY
on your 6800 . . .

Now, you can play SCELBI's new and improved GALAXY on your 8080

Captain your own starship on intergallactic journeys. It's an ever-changing adventure with battles, refueling problems, weaponry, warp factors — and more!

You can elect a display format that permits all data to be displayed on a 32-character per line device. It's great for TV typewriters too. There's even a display option for TTY and full width CRTs.

Here's your chance to match wits against the logic of your 8080. Search and destroy a random number of alien ships. But don't run out of time, out of fuel, out of ammunition or out of the galaxy. Plan your missions through 64 quadrants subdivided into 64 sectors. It's great fun that goes on for months and months. A multidimensional computer game. Listing uses manufacturer's recommended mnemonics. Manual includes assembled source listing in hexadecimal notation. Octal dump also provided. Resides in just 4K of memory. Manual includes flow charts, illustrations and lots more. It's easy to use and it makes sense!

Get your copy today! SCELBI's 8080 GALAXY Game. Only $9.95 for manual of source and object code listings. See it at your favorite computer/electronics dealer. Or order direct from SCELBI.

Scelbi Publications
P.O. Box 133 PP STN
Milford, CT 06460
(203) 874-1573
a division of Scelbi Computer Consulting, Inc.

Mail order: Optional punched tape of object code in hexadecimal format, $5 additional. All prices shown for North American customers. Master Charge, VISA, Postal and Bank Money Orders preferred. Personal checks delay shipping up to 4 weeks. Pricing and availability subject to change without notice.

IMPORTANT: Include 75¢ postage/handling for each book delivered by U.S. Mail Book Rate or $2 for each book shipped First Class or via UPS.

imations requiring range reduction are:

Function	Range
sin(x)	$(-\pi/2 \leq x \leq \pi/2)$
$\log_{10}(x)$	$(1 \leq x \leq 10)$
10^x	$(0 \leq x \leq 1)$

Table 10 outlines techniques that might be used in reducing the range to that required by the approximation.

Conclusions

Several approximation methods have been presented for use in software systems lacking particular function subroutines. These methods may be implemented in either machine language or in a higher level language. It is likely that the majority of readers will use the resulting subroutines in a BASIC interpreter unless they have an assembler with macroinstructions for multiplication and division, or something equivalent (such as floating point multiplication and division hardware).

The particular approximations given above can be used to evaluate many other functions by using identity or recursion relations. In cases where such relations do not exist, it may be possible to use statistical software packages on larger computers to find adequate functional approximations for use in microcomputers. For example, one might take a polynomial having coefficients to be determined and regress the coefficients against handbook tables representing the desired function. With some ingenuity in transforming variables, such as:

$$f(x) = \sum_{n=0}^{N} a_n \, y^n(x)$$

$$y(x) = \frac{x-1}{x+1}$$

It may be possible to obtain very accurate expansions having only a few terms. For example, if the function to be approximated goes to infinity at x = 0 and to zero at x = a, a transformation one might try is $y(x) = (x - a)/x$. Those electrical engineers familiar with pole analysis would probably be somewhat at ease with forming such expressions. However, these same people would probably have little experience with the available statistics routines.

The general conclusion is that relatively fast executing expansions exist for functional approximations. They tend not to be truncated Taylor series, but rather special (and sometimes ingenious) forms whose coefficients depend on the number of terms to be used as well as the argument interval desired. The minimax polynomials and rational polynomials are powerful examples of such approximations, and should be considered for use on small systems. Elegant iteration routines may also be very powerful, but there are situations, such as the n^{th} root iterative technique (see table 8), and the

```
10   REM ***ARCTAN(X) VIA GAUSS***
20   DIM A(20),B(20)
30   INPUT X
40   A(1)=1/SQRT(1+X*X)
50   B(1)=1
60   N=0
70   E=.0000001
80   N=N+1
90   A(N+1)=(A(N)+B(N))/2
100  B(N+1)=SQRT(A(N+1)*B(N))
110  T=X*A(1)/SQRT(A(N+1)*B(N+1))
120  PRINT N,"  ",T
130  D=(A(N+1)-B(N+1))/B(N+1)
140  IF ABS(D)>E THEN GOTO 80

?.1
1    9.9689307E-02
2    9.9673813E-02
3    9.9669946E-02
4    9.9668978E-02
5    9.9668736E-02
6    9.9668677E-02
7    9.9668662E-02
8    9.9668659E-02

?1
1    .79627295
2    .78796829
3    .78603187
4    .78555605
5    .7854376
6    .78540801
7    .78540062
8    .78539879
9    .78539831
10   .78539817
11   .78539813

?100
1    1.6692059
2    1.582113
3    1.5658381
4    1.5620399
5    1.5611064
6    1.560874
7    1.560816
8    1.5608014
9    1.5607978
10   1.5607969
11   1.5607967
12   1.5607966

?1E30
1    1.6817928
2    1.592546
3    1.5759366
4    1.5720637
5    1.5711121
6    1.5708751
7    1.5708159
8    1.5708012
9    1.5707975
10   1.5707966
11   1.5707963
12   1.5707962
```

Listing 1: The arctangent approximation shown in table 9 implemented in BASIC. Four sample runs are shown.

$\sin(x) \qquad (-\pi/2 < x < \pi/2)$

- Reduce range to first four quadrants:
  $y = x - (\text{INT}(x/2\pi) \times 2\pi)$
  (Note: It is assumed that the integer function returns next lower integer for positive and negative numbers.)
- If $|x| \geq \pi/2$, then $y = y - \pi$ and desired result is $(-\sin(y))$: otherwise the desired result is $\sin(y)$.

$\log_{10}(x) \qquad (1 \leq x \leq 10)$

- Find range reduction factor (characteristic of logarithm): Divide (or multiply) by 10, I times until result is in range.
- Reduce range:
  $y = x/10^I$
- Desired result is:
  $\log_{10}(x) = I + \log_{10}(y)$
  Note: I may be negative.

$10^x \qquad (0 \leq x \leq 1)$

- Determine power of 10:
  $I = \text{INT}(x)$
- Reduce range:
  $y = x - I$
- Result
  $10^x = 10^y \times 10^I$

Table 10: Range reduction methods for the approximations we have considered so far.

Gauss technique (see table 9), in which the approach has intrinsically greater accuracy, but the execution time of the subroutine is prohibitive.

Function approximation is a creative art; I hope readers will try some of the techniques described herein.■

REFERENCES

1. Hastings, C, *Approximations for Digital Computers,* Princeton University Press, Princeton NJ, 1955.
2. Fike, C T, *Computer Evaluation of Mathematical Functions,* Prentice-Hall Inc, Englewood Cliffs NJ, 1968.
3. Weast, R C, *Handbook of Tables for Mathematics,* Chemical Rubber Co, Cleveland OH, 1967.
4. Smith, J M, *Scientific Analysis on the Pocket Calculator,* John Wiley and Sons, New York, 1975.
5. Acton, F S, *Numerical Methods That Work,* Harper and Row, New York, 1970.
6. Abramowitz, M, and Stegun, A, *Handbook of Mathematical Functions,* Dover, New York, 1964.

Circle 359 on inquiry card.

COMPUTER SOFTWARE

For Homeowners, Businessmen, Engineers, Hobbyists, Doctors, Lawyers, Men and Women

We have been in business for over nine years building a reputation for providing a quality product at nominal prices – NOT what the traffic will bear. Our software is:

- **Versatile** – as most programs allow for multiple modes of operation.
- **Tutorial** – as each program is self prompting and leads you through the program (most have very detailed instructions contained right in their source code).
- **Comprehensive** – as an example our PSD program not only computes Power Spectral Densities but also includes FFT's, Inverse-transforms, Windowing, Sliding Windows, simultaneous FFT's variable data sizes, etc. and as a last word our software is:
- **Readable** – as all of our programs are reproduced full size for ease in reading.
- **Virtually Machine Independent** – these programs are written in a subset of Dartmouth Basic but are not oriented for any one particular system. Just in case your Basic might not use one of our functions we have included an appendix in Volume V which gives conversion algorithms for 19 different Basic's; that's right, just look it up and make the substitution for your particular version. If you would like to convert your favorite program into Fortran or APL or any other language, the appendix in Volume II will define the statements and their parameters as used in our programs.

Over 85% of our programs in the first five volumes will execute in most 8K Basic's with 16K of free user RAM. If you only have 4K Basic, because of its lack of string functions only about 60% of our programs in Volumes I through V would be useable, however they should execute in only 8K of user RAM.

For those that have specific needs, we can tailor any of our programs for you or we can write one to fit your specific needs.

Vol. I

Business & Personal Bookkeeping Programs

Bond, Building, Compound, Cyclic, Decision 1, Decision 2, Depreciation, Efficient, Flow, Installment, Interest, Investments, Mortgage, Optimize, Order, Pert Tree, Rate, Return 1, Return 2, Schedule 1

Games & Pictures

Animals Four, Astronaut, Bagel, Bio Cycle, Cannons, Checkers, Craps, Dogfight, Golf, Judy, Line Up, Pony, Roulette, Sky Diver, Tank, Teach Me

Pictures

A. Newman, J.F.K., Linus, Ms. Santa, Nixon, Noel Noel, Nude, Peace, Policeman, Santa's Sleigh, Snoopy, Virgin

Vol. II

Binomial, Chi-Sq., Coeff, Confidence 1, Confidence 2, Correlations, Curve, Differences, Dual Plot, Exp-Distri, Least Squares, Paired, Plot, Plotpts, Polynomial Fit, Regression, Stat 1, Stat 2, T-Distribution, Unpaired, Variance 1, Variance 2, XY

APPENDIX A

Beam, Conv, Filter, Fit, Integration 1, Integration 2, Intensity, Lola, Macro, Max. Min., Navaid, Optical, Planet, PSD, Rand 1, Rand 2, Solve, Sphere Trian, Stars, Track, Triangle, Variable, Vector

Vol. III

Billing, Inventory, Payroll, Risk, Schedule 2, Shipping, Stocks, Switch

Vol. IV

Bingo, Bonds, Bull, Enterprise, Football, Funds 1, Funds 2, Go-Moku, Jack, Life, Loans, Mazes, Poker, Popul, Profits, Qubic, Rates, Retire, Savings, SBA, Tic-Tac-Toe

Vol. V

Andy Cap, Baseball, Compare, Confid 10, Descrip, Differ, Engine, Fourier, Horse, Integers, Logic, Playboy, Primes, Probal, Quadrac, Red Baron, Regression 2, Road Runner, Roulette, Santa, Stat 10, Stat 11, Steel, Top, Vary, Xmas

APPENDIX B

Vol. VI

Ledger	Maintains Company accounts and generates financial reports. Includes routines for: Pyrl, Inv, Depr, A/R, A/P.

Vol. VII

Chess	Designed to challenge the average player, fairly comprehensive. Great fun for all, offers a unique opportunity for beginners in need of an opponent.
Medbil	For Doctors and Dentists alike, a complete patient billing system which also permits the maintaining of a patient history record.
Wdproc	Wordprocessing for lawyers, publishers, writers, etc. Write, store, and change from rough draft to final copy in a variety of formats.
Utility	Disk utility program with memory testing.

Vol. VIII

1040-Tax	Taxpayers return, itemized deductions or standard
Balance	Reconciles bank statements
Checkbook	Balances your checkbook
Inst1 o 78	Computes real cost on bank financed items; cars, boats, etc.
Deprec 2	Computes depreciation, 4 methods, any time period

APPENDIX C – FAVORITE PROGRAM CONVERSIONS

Vol. I – $24.95 Bookkeeping Games Pictures	Vol. II – $24.95 Math/Engineering Plotting/Statistics Basic Statement Def.	Vol. III – $39.95 Advanced Business Billing, Inventory Investments Payroll	Vol. IV – $9.95 General Purpose	Vol. V – $9.95 Experimenter's Program	Vol. VI – $49.95 Mini-Ledger	Vol. VII – $39.95 Professional Programs	Vol. VIII – $19.95 Homeowner's Programs

AVAILABLE AT MOST COMPUTER STORES
Master Charge and Bank Americard accepted.

Our Software is copyrighted and may not be reproduced or sold.

Add $1.50 per volume handling, all domestic shipments sent U.P.S. except APO and P.O. Box which go parcel post. Foreign orders add $8.00/volume for air shipment and make payable in U.S. dollars only.

SRI SCIENTIFIC RESEARCH

P.O. Box 490099-B Key Biscayne, FL 33149

Phone orders call 800-327-6543 Information – (305) 361-1153

Original painting by Mike Smithwick

The Sky's the Limit

Use Ham Radio Bands for Intercomputer Communication

Joe Kasser
11532 Stewart Ln
Silver Spring MD 20904

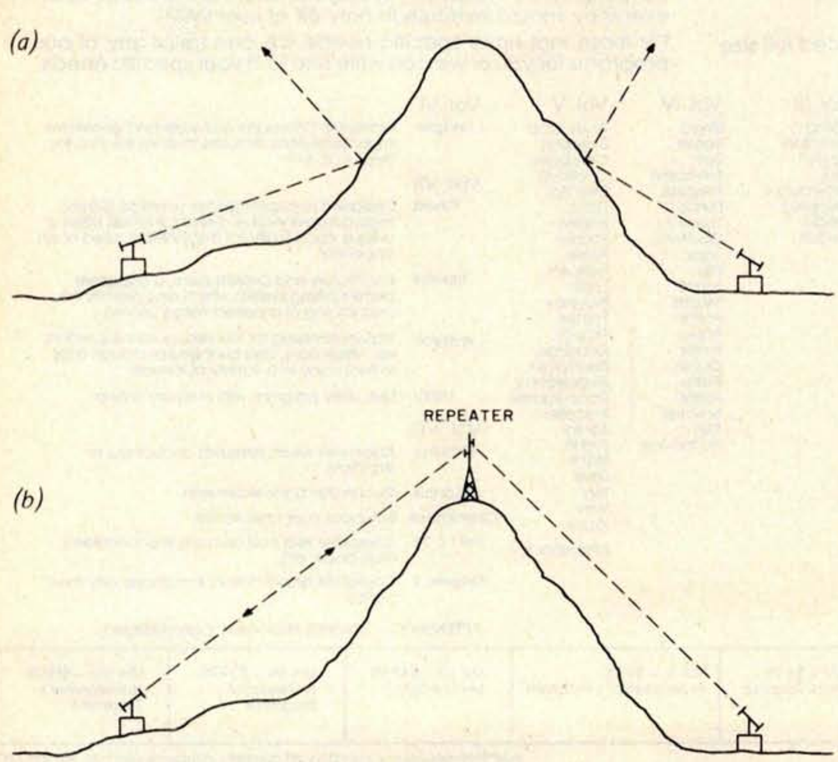


Figure 1: Difficulties in radio communications. Figure 1a shows that if there is a hill between two local stations, it is not possible for them to communicate by means of VHF. If a repeater station is positioned at the top of the hill as in figure 1b, communication becomes possible.

Introduction

This article outlines how intercomputer communication networks can be set up using existing amateur radioteletypewriter (RTTY) networks, without requiring the computer user to have an amateur radio license. Using modified radio message handling codes, a communications language is developed that allows message forwarding, file transfers for batch processing, and interactive use of a central computer. In all cases compatibility has been established between simple systems and complex ones such that the network can be accessed by terminals as well as computers. Finally, the network concept is merged with space age technology to outline how a worldwide personal computing communications network can be established with minimal message routing information requirements, using the AMSAT Phase II and Phase III amateur radio communications satellites.

While discussing intercomputer communications, let us consider that which exists now and that which is soon going to exist. The communications system should be set up so that one can use the network with a minimum investment in equipment. As the system becomes more sophisticated, so will the system capability.

Amateur Communications Network

The closest thing to computer communications in the *hobby* area is amateur radioteletypewriter communications. Hams have been using radioteletypewriters for many years and have developed equipment and techniques that are of interest to the computer experimenter. Radioteletypewriters are used by amateurs in two different ways:

- Long distance contacts via shortwave.
- Local area contacts on very high frequencies (VHF).

Shortwave communication paths are affected by many things including propagation conditions and interference from atmospheric or man-made signals (called noise). VHF communications paths are, for all practical purposes, unaffected by propagation conditions and can be made immune to noise.

In the United States much of the VHF radioteletypewriter operation is by means of *repeaters*. VHF communication paths are relatively short range because the signals are usually limited to line of sight paths. Thus, two radio amateurs would find it difficult to communicate if there were a hill between them, as shown in figure 1a. If a relay or repeater were placed at the top of the hill as shown in figure 1b, the two stations could communicate. Users of the VHF portion of the radio spectrum have long recognized the advantages of such repeaters. Most cities now abound with radio towers relaying police, fire, taxicab and amateur signals.

A typical citywide amateur RTTY repeater communications system is depicted in figure 2. Notice the similarity to a centralized computer network. Many amateur stations operate with *autostart*. They leave their equipment operating all day to print any messages that turn up on the frequency. Others have *stunt boxes* or selected signal decoders that allow only messages addressed to a particular station to be printed. Still others are interfacing microcomputers to their radio stations and using the computer to perform selective call decoding and storage of commonly transmitted messages or descriptions of their stations (colloquially known as brag tapes).

The VHF repeater system can thus be considered a communications network wherein any amateur station can originate or receive messages with only a minimum amount of equipment. However, the amateurs can only communicate if both are operating (on line) at the same time. If one is not present, messages can be received only if the equipment is in the autostart mode. Even then, the originator cannot know that the message was received because the receiving station might have been switched off

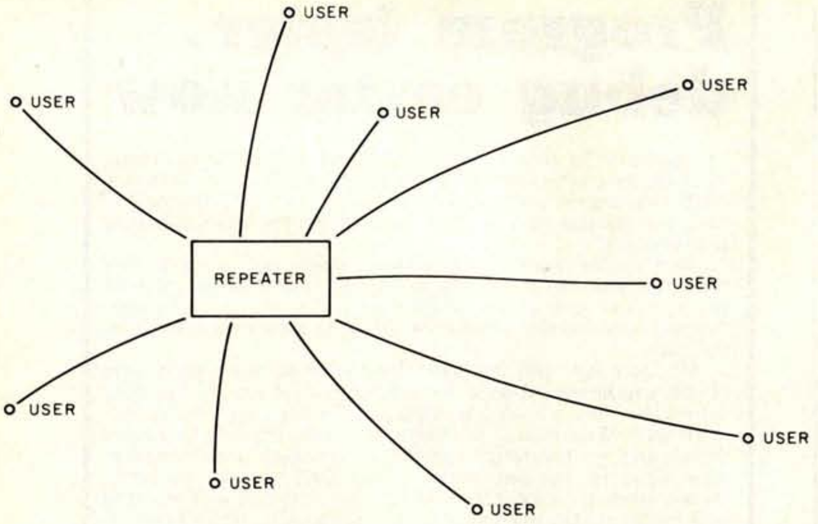


Figure 2: A city-wide radioteletypewriter VHF repeater link. Note that each of the users is an amateur radio station.

at the time the message was sent. The occupancy of these repeaters at the present time is low, so all users can timeshare them easily. When two stations are in communication, the others may either stand by or join in and have a multistation contact.

As radio amateurs recognize the capabilities of the microcomputer, they are installing them in their stations to take care of autostart and answer back so that messages can be received and even acknowledged in real time. However, this is only valid for those equipped with a computer. What about those who only have a simple terminal? How can they obtain the same advantages of communications capability?

The computer experimenters among the amateurs have come up with a solution. They are connecting one computer as a message store and forward relay. This computer continuously monitors the repeater, and may or may not be located at the repeater site. When accessed by a user it may be used to store a message addressed to any other user for later retrieval. Anyone can come on the air (on line) and talk to the computer. Thus anybody equipped with a terminal compatible with the system can use the central computer to send and retrieve messages. One can even use the computer in an interactive mode if the network is set up to allow it. Those who own their own computers can use the network to exchange software in machine readable form as well as a message relay. A typical sequence of operation in the message mode would consist of one station coming on the air and calling his or her party. If no reply is received the message could be stored in the central computer. Later when the party comes on line, he or she can ask the computer if there are messages waiting, and if so, to retrieve them.

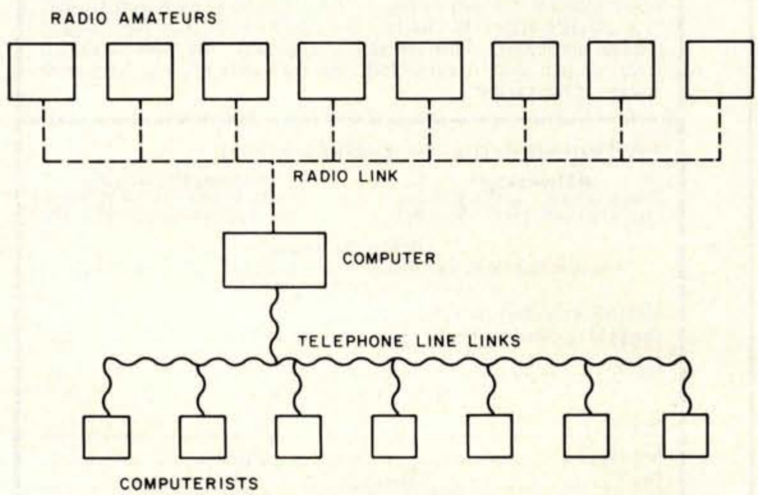


Figure 3: Radio amateur and computer experimenter city-wide communication network. This type of network allows the computer user to have access to the radio network without having to obtain a radio license.

Computer experimenters in general do not have amateur radio licenses, probably don't want to make the effort to obtain them, and don't think that they need them. What good is a VHF repeater to them?

Computerists' Access to Repeaters

In the United States, the regulations governing amateur radio allow third party traffic. This means that amateurs are allowed to relay certain kinds of messages for nonamateurs. Amateurs have connected their radio sets to the telephone line and are able to relay telephone calls across the continent. They have also connected the VHF repeater to the telephone network, and are able to make telephone calls from cars by the use of tone signaling. Some repeaters have reverse call-in capability by which the nonamateur can dial the repeater and the telephone will ring over the air. An amateur can then answer the call and control the conversation.

If the central computer in the VHF repeater network is also connected to the telephone line, any experimenter could call the computer and either store or retrieve a message. The whole network is suddenly expanded to include the computer experimenters, as shown in figure 3. They can now access the central computer over the telephone line and operate it in much the same manner as the amateur would over the air. The nonamateur would not, however, be able to initiate transmissions over the air. Messages would have to be stored for later pick up over the air by radio amateurs or pick up over the telephone line by other nonamateurs. In this way, with no petitions

LOOK WHAT'S NEW FROM LEAR-SIEGLER ...and SYNCHRO SOUND'S GOT 'EM!

ADM-31 VIDEO TERMINAL

- Two full pages of display
- Protected fields/dual intensity
- RS232 or current loop
- Block mode

$48.05 per month* | **$1450.00** complete

ADM-42 SMART TERMINAL

- Up to 8 pages of memory
- Blanking, blinking and reverse fields
- 3 tab modes/protected fields
- 32 special functions
- Detachable keyboard

$57.50 per month* | **$1795.00** complete

300 SERIES Model 301 BALLISTIC PRINTER

- 96 ASCII character set
- 7 X 9 dot matrix
- 180 characters per second
- Selectable top of form
- Foreign alphabets available

$65.27 per month* | **$1995.00** complete

VDP-1000 DATA SYSTEM

1.24/MB

- Choice of 2 disk systems, 3 languages
- 16 bit CPU
- 32 K RAM
- Includes ADM-3A and desk

$254.88 per month*
$7995.00 complete

10/MB **$585.55** per month* | **$17,999.00** complete

ADM-3A VIDEO TERMINAL

- Full addressable cursor
- Display format-24 lines of 80 characters per line
- Communications rates —75 to 19,200 Baud
- Computer interfaces— EIA standard

$29.44 per month* | **$854.95** complete

SYNCHRO-SOUND ENTERPRISES, INC.
The Computer People
193-25 Jamaica Avenue, Jamaica, New York 11423 Dept. HB
212/468-7067 TWX 710-582-5886
West Coast: 5810 Commerce Blvd., Rohnert Park, CA 94928
707/544-2865

***36 month lease**
Maintenance additional
OEM pricing available

Hours 9-4 Daily and Saturday | Visit our new showroom Working units on display

BankAmericard ■ Master Charge

Circle 355 on inquiry card.

to the FCC, both computer enthusiasts and radio amateurs would have access to the computer. Amateurs would be identified by their call signs while computer experimenters would be identified by their computer club initials and membership numbers. Existing radio amateur VHF radioteletypewriter repeater networks could easily become computer communication networks.

Communication Language

Having more than one computer on the network introduces protocol problems. How does the network ensure that only the desired computer is on line? How do messages get sent and stored? How does one user gain access to another's computer? These problems require solutions before a usable system is put on the air. A communications language must be used to perform these functions. Luckily, such a language does not have to be invented because one exists in rudimentary form.

Years ago, when radio communications were by means of Morse code, a communications code was developed to allow commonly used conventions in station management and message relaying to be sent speedily and efficiently. This communications code is known as the **Q** code. It comprises a large number of three letter groups each beginning with the letter **Q**. By the use of this code, radio telegraph operators speaking different languages can communicate and pass traffic. Radio amateurs adopted some of the language and modified it to suit their needs. There is no reason why computer experimenters could not do the same.

Extracts from the code and the proposed modifications for intercomputer network use are presented in table 1. By the use of such a code, amateurs with Baudot terminals can use the code by typing the 3 letter group. Others with ASCII terminals and computers can type ASCII control characters, have the computer translate them to the 3 letter group, and then transmit them. In this way the network can use Baudot even though ASCII may eventually be licensed.

Examples of the use of the **Q** code in the network follow. When sending a message to be stored in the central computer relay:

WR3ABU :QTC: G8BTB :QRV: DE G3ZCZ

which means:

Attention repeater whose call sign is WR3ABU, I have a message for G8BTB. Are you ready? From this station G3ZCZ.

The computer would reply:

G3ZCZ :QRV: DE WR3ABU

which means:

Yes, G3ZCZ, I am ready, from the repeater WR3ABU.

The computer is now ready to receive and store the message. G3ZCZ would then send the message in the following format:

WR3ABU :QSP: G8BTB :QSO: *[text of message]* :QSL: DE G3ZCZ

which means:

WR3ABU please relay the following message to G8BTB. The message follows *[text of message]*, please confirm reception from G3ZCZ.

G3ZCZ would then confirm and the message would be stored in the computer.

Table 1: Extracts from the proposed **Q** *code modifications for a network communications language. The original* **Q** *code reference was the* FCC Study Guide and Reference Material for Commercial Radio Operator Examinations, Revision May 15 1955.

Code	Question	Answer or Advice
QRA	What is your identification or call sign?	My call sign is . . . .
QRL	Are you busy?	Yes, I am in use by . . . .
QRM	—	Your transmissions are being interfered with.
QRQ	Shall I speed up to . . . . bauds?	Yes.
QRR	Are you equipped for automatic operation?	Yes.
QRS	Shall I slow down to . . . . bauds?	Yes.
QRT	—	Signing off (log off).
QRU	Have you any messages for me?	Yes, messages are from . . . .
QRV	Are you ready?	Yes.
QRX	Will you wait?	Yes.
QRY	What is my turn?	Your turn is number . . . .
QSG	—	Send . . . . messages.
QSK	Can you operate full duplex?	Yes.
QSL	Will you confirm?	Confirmed.
QSM	—	Repeat last message.
QSO	—	Message for . . . .
QSP	—	Relay via . . . .
QTA	Cancel message to	Cancelled
QTC	—	The message is . . . .
QTH	What is your address?	My address is . . . .
QTR	What is the correct time?	It is . . . . UTC.
QTX	—	Log on.
QUA	Send me all new messages.	—
QUC	Who did the last message I sent go to?	It went to . . . .
QBM	Send me the message from	—
QDB	—	The message to . . . . is forwarded.
QIC	May I call . . . direct?	Yes.
QJG	—	Revert to message mode (log off interactive mode).
QNO	—	Negative response or action.

When signing on the network, anyone can ask if messages are awaiting them in the following way:

WR3ABU :QRU: DE G8BTB

which means:

WR3ABU, do you have any messages for me, this is G8BTB asking.

The computer would then reply:

G8BTB :QRU: DE G3ZCZ, FOWN, ON8IK DE WR3ABU

which means:

G8BTB, there are messages from G3ZCZ, FOWN and ON8IK for you.

Messages could be retrieved from the computer in the following format:

WR3ABU :QBM: DE G3ZCZ :QRV: DE G8BTB

which means:

OK WR3ABU, send me the message from G3ZCZ, I am ready.

The computer would then respond:

G3BTB :QSO: DE G3ZCZ *[text of message]* :QSL: DE WR3ABU

which means:

G8BTB, the message from G3ZCZ is *[text of message]*, please confirm reception, from WR3ABU.

G8BTB would then confirm reception of the message with the following transmission:

WR3ABU :QSL: DE G8BTB

The computer could then delete the message from the storage area.

Almost any radio amateur could read the exchange and understand what is taking place. This language can be used by anyone, however simple the system being used is.

Another advantage of the Q code is that most amateurs will have to learn relatively few of the words. As long as the computer experimenter is learning a language, it does not matter which one is learned. Using words of fixed length, bounded by colons, with the letter Q as the first character will simplify the software required to process the

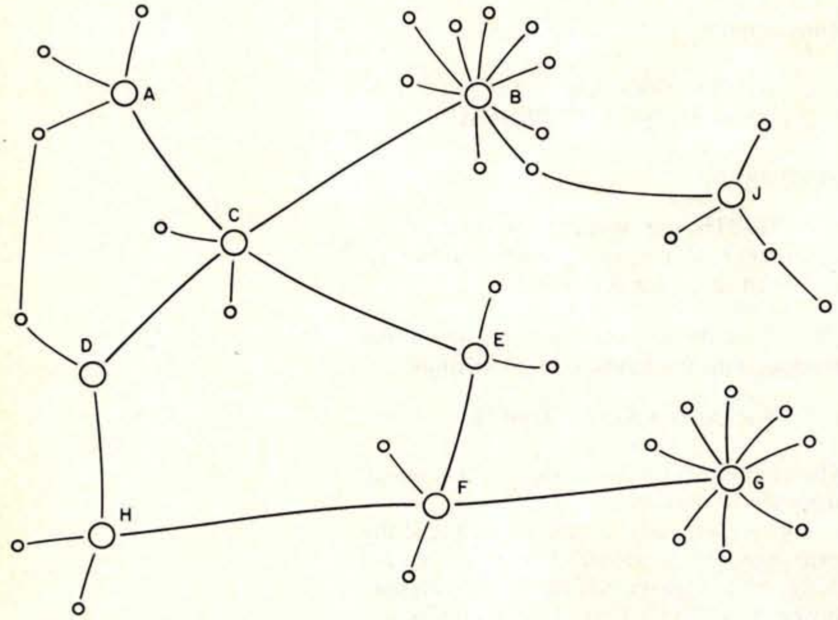


Figure 4: A distributed wide area network. This type of network would have a specified sequence of links. For a message to go from a station in network A to a station in network G, there would have to be a path for the message to follow through other stations in the network.

instructions. Since the choice of signals to be exchanged between the computer and the stations using the network is arbitrary, it may as well be in humanly readable form to allow people using terminals without computers to use the network.

If it becomes legal to use ASCII over the air, it will be up to the repeater groups to decide if they want to change from Baudot to ASCII. The decision may depend on how many users have Baudot machines. Both modes could be used with conventional tones for Baudot and Kansas City or Bell 103 standard tones for ASCII.

So far we have seen that local level networks can be set up using a computer linked to a VHF repeater and the telephone network, to give a wide area coverage. Since the range of individual interests is wide and different, it should be possible to set up such systems using the talent available in the group. Thus, at the local level, radio amateurs and computer people can merge their talents and put together a system useful to both.

The actual modulation techniques used in the links has not been addressed as yet. Bell 103 compatible tones can be used for the telephone lines. Any legal method can be used for the radio link. The network can make its own decision about which technique to use, depending on how much hardware and software they want each user to assemble when using the network. Since there are many different types of microcomputers in the real world, different stations will need different software at the machine language level. A high level communications language that can be understood by humans as well as computers will make the link comprehensible to any user no matter how simple or sophisticated the available equipment.

Network Links

Once networks are established there will be a desire to expand the coverage and services of each by linking them together. Conventional techniques can be used to link networks in which each local network establishes a gateway to another network. If enough locals link together they can form a large network of the type shown in figure 4. It would then be technically possible to send a message from a station in network A to a station in network G. Note that the figure does not differentiate between radio and telephone links.

Operationally, the routing of the message poses many problems. How is the message to be routed? If a fixed route is always used (A to C to E to F to G) what happens if the E node is inoperable? Should the message be bounced around at random, eventually getting to G, which would then send a cancel command to the network to delete that message because it has arrived at its destination node? What other techniques would be suitable for use in a personal network which may or may not be 100% reliable in terms of long distance links? The professionals have been dealing with similar problems for years.

Here again amateur radio has the potential to provide a simple solution to the problems of internetwork communications by making use of modern technology. Just as the developing nations are bypassing complex terrestrial microwave links and employing satellites in long-range communication links, amateur radio can provide the capability for computer enthusiasts to do the same thing. Currently under construction and planned for launch in late 1979 is the first AMSAT Phase III radio amateur communication satellite, shown in photo 1. This satellite will be put into a high altitude elliptical orbit and will provide communication capability over the entire northern hemisphere for up to 16 hours per day with the same degree of reliability as a conventional VHF radioteletypewriter repeater. A message from Washington DC could be sent to London England by means of the following Q code:

> WR3ABU :QSP: G8BTB :QSP:
> GB3LO :QSO: /*text of message*/
> :QSL: DE G3ZCZ/W3

The only difference between this message and the transmission given earlier is the addition of QSP GB3LO which means, "Re-

An advanced desktop data system for $1,995? Quitcherkiddin,* TANO.

We're not kiddin'.
Outpost 11 is exactly that. You get:

- **A ruggedly designed unit, intended for heavy use**
- **Full ASCII keyboard**
- **24 x 80 character CRT**
- **32K bytes RAM**
- **M6800 CPU**
- **Mini-floppy disk drive and controller**
- **BASIC software package**
- **Over 20 applications packages available**
- **User documentation**

Just like this, for only

OUTPOST 11

$1995

COMPLETE

With Outpost 11 and the BASIC software package, you can:
Do full floating point BCD math ● Program SAVE and LOAD with file names ● Program MERGE and CHAIN ● ON ERROR statement ● String variables and functions ● Trig and math functions ● GET, PUT and FIELD statements ● Floppy file processor.

You've been waiting a long time for a Data System with all these features at a price like this. Now it's yours. No kiddin'.

**Quit your kidding*

Order Form

Quitcherkiddin', TANO. I'm calling your hand. Send me ______ Outpost 11's as described in your ad. Here's my deposit of $200 each, plus $35 freight and insurance. I'll pay the balance on delivery.

Name ______________________

Company/Title ______________________

Shipping Address ______________________

Telephone ______________ Signature ______________

Mail to:

TANO Corporation
4521 W. Napoleon Avenue
Metairie, La. 70001
(504) 888-4884
TWX 810-591-5229

Sales Offices: Los Angeles, CA—213/426-7375 • Montreal, Canada—514/934-0000 • Boston, MA—617/969-4650 • Dallas, TX—214/358-1307 • Houston, TX—713/461-0038 • Denver, Col.—303/841-2788 • San Francisco, CA—408/377-7001 • Chicago, IL—312/830-0060 • Atlanta, GA—404/252-6609 • Washington, DC—301/593-1559 •

Circle 358 on inquiry card.

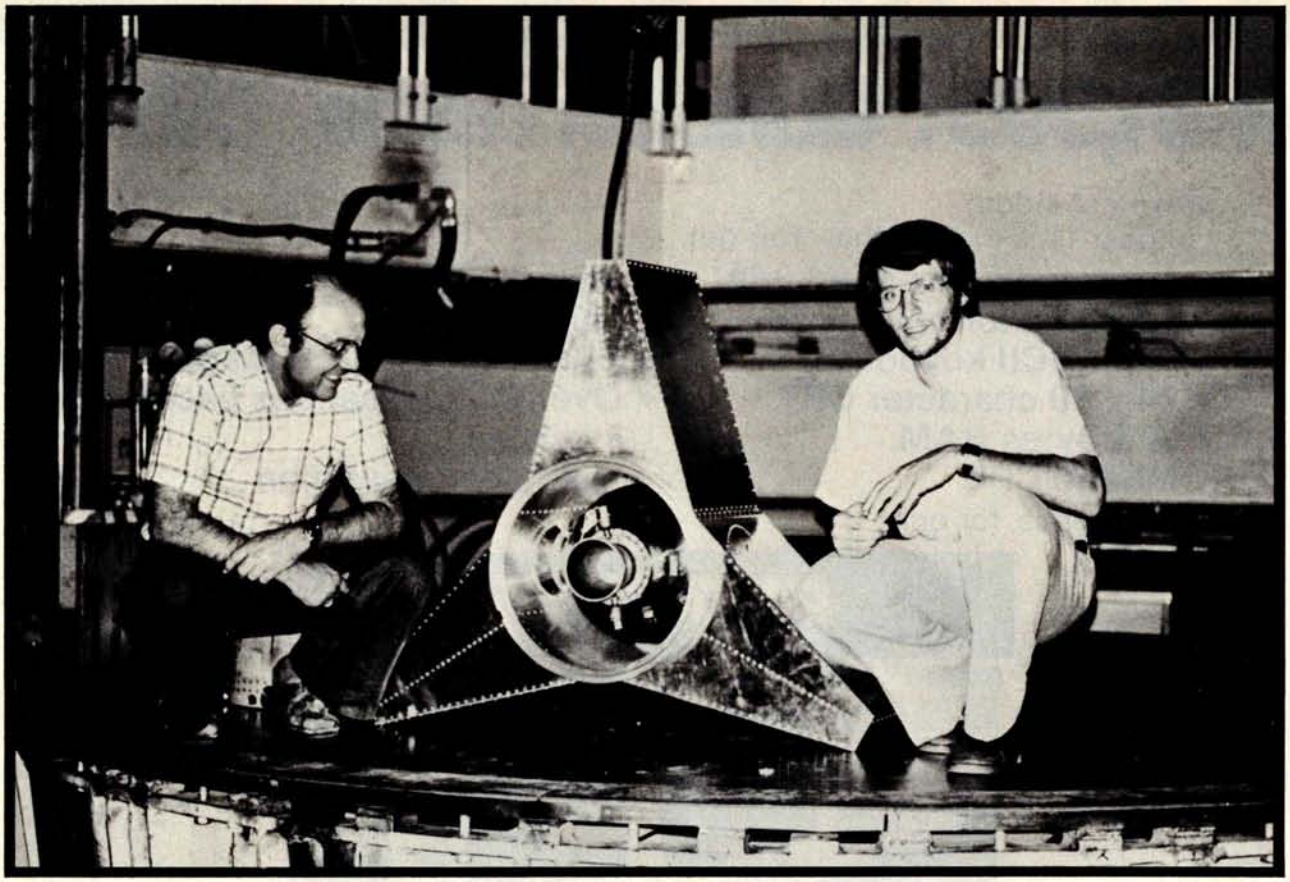


Photo by Dick Daniels, courtesy of AMSAT.

Photo 1: Jan King (right) and Karl Meinzer (left) with model of the AMSAT Phase III satellite. The satellite has a mass model kick motor installed in the structure. Jan is the project manager, and Karl designed the satellite, which is controlled by an RCA COSMAC microprocessor. He also designed IPS, a high level machine-independent language for use with the satellite.

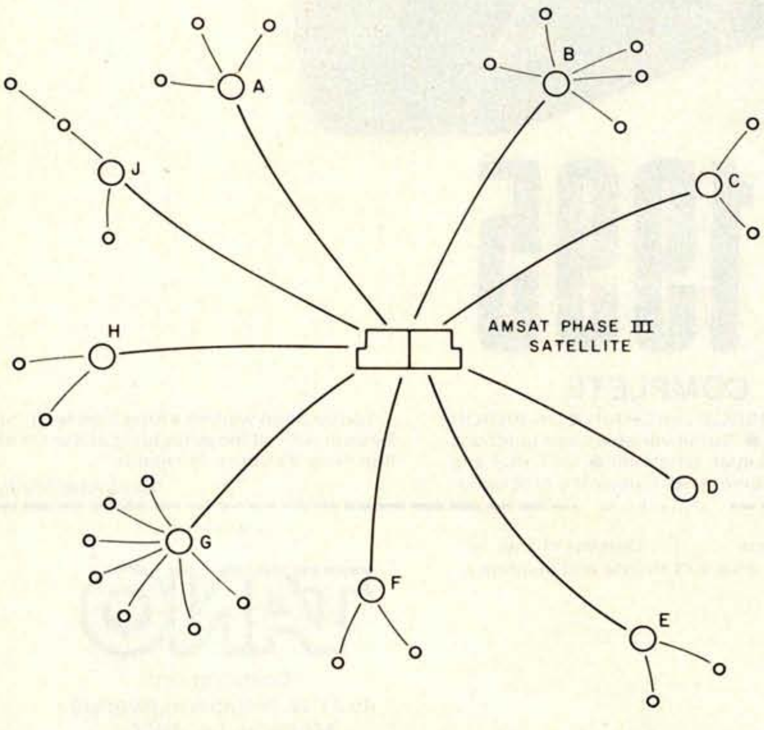


Figure 5: A distributed wide area network using a satellite. This approach would solve many of the message transmission problems. For a message to go from a station in network A to a station in network G it would have to travel only through the satellite and then be relayed to the other station.

lay the message via gateway station in London." The call sign of the gateway station is GB3LO.

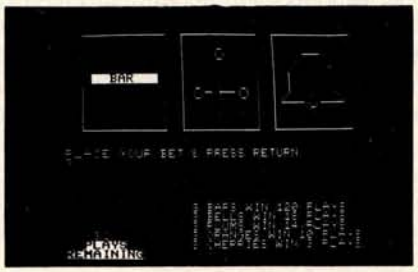
Just as a long-distance telephone user must know the area code when placing a call, the network user must know the call sign for the gateway station of the person being called. With a satellite in the link, the network simplifies to the type shown in figure 5. This configuration has a striking resemblance to a local network. Each local network can be accessed by any other local network, just as any user can be accessed by any other user in the local network. If a gateway is off line, due either to equipment problems or being out of range, the originating gateway can store the message for later transmission when communications are reestablished. The satellite network would be used similarly to the local network.

Since the satellite network will be lightly

NEW FREE CATALOG OF Ready to load Cassettes for PET 2001 Computer*

Available *now* from NAIL**
Ready for immediate shipment.
Dozens and dozens of interesting programs are now available from NAIL (National Artificial Intelligence Laboratory). The first programs are games and graphics. Unlike most other sources of game programs, NAIL has created game programs that don't just tell you what's happening; they *SHOW* you right on the screen with vivid graphics in motion.

For example, NAIL program G-1, SLOT MACHINE; randomly selects cherries, lemons, plums, oranges, etc. from program memory and then *displays three* on the screen. You select the number of free plays, and on each turn you can bet from one to nine plays. The machine displays at all times what you have bet; if you have won, how much, and how many free plays you have



left. To "pull the lever" on the machine, you need only hit return, and the old characters are erased from the screen, and three new ones appear, one by one, "as the wheels stop spinning".

Whether you have a 4K or an 8K machine, this program will run on it. Many NAIL games will run in 4K machines. The new free catalog lists dozens of games just like this one with their descriptions, and price. Be sure to send in your name and address on a postcard or use the coupon at the bottom of the page for a free catalog by return mail — no obligation. If you want to order the SLOT MACHINE program, send $2.00 with your name and adress and NAIL will send you by return mail the SLOT MACHINE game cassette and the new free catalog. You do not have to buy anything to receive the catalog. If you want to order the SLOT MACHINE program at the special introductory price of $2. you may do so.

NAIL programs are reasonably priced. They will enable you to get so much more enjoyment from your PET, whether in machine language or BASIC, you may discover some things in the NAIL programs you didn't know before. If you're really good, we may be interested in purchasing programs from you for distribution to NAIL customers. (see below — Do you write programs?).

What is your time worth?
How long would it take you to enter a program with 300 lines of code, and then debug it? Even if you figure your time is worth only the minimum wage, you can order most NAIL programs for less than your time would be worth to enter the same program from a printed listing. NAIL cassettes each contain just one program. No searching through a half dozen others to find what you want. Just slip it in your PET and within seconds your program is loaded and ready to run.

Know what you are buying.
The NAIL catalog gives lengthy and complete descriptions of programs offered, including actual unretouched off-the-screen photographs of the program in action. You will know just what you are buying.

You will also know just what the program does, how it works, and thus be able to decide if you want to order it. No more having to order four or five programs on a single cassette in order to get just the *one* you want.

Do you write programs?
NAIL is interested in purchasing sophisticated, well structured, and completely debugged programs.

Although there are already CHESS, CHECKERS, BACKGAMMON and other similar program in the NAIL inventory, NAIL will buy programs of the same games provided they are substantial improvements over the ones already in inventory. Adversary programs are played against each other. Programs may be in either BASIC or machine language.

Machine Language programming?
Then you might consider the NAIL SYS 7171 machine language monitor for PET. It is written in machine language and comes ready to use with complete written instructions. Written in machine language, it uses only 5 pages of RAM (1,283 Bytes) leaving 5884 bytes free for programming. Enables you to load and save in both machine language and in BASIC. Also allows you to *append* one BASIC program to another.

Once loaded, the SYS 7171 machine language monitor stays in the machine in protected high memory until you turn off the power. You can skip back and forth to and from BASIC to the monitor.

Displays contents of *all* memory, including ROMS. Do not confuse this monitor with destructable non-versatile types offered by others.

This is the *only* monitor currently available which provides for coresidence of BASIC and machine language programs and the capability of saving either on cassette or a combination of both. The ultimate in versatility!

Cassette of monitor, instructions, and license for your personal use on your machine for 50 years. $29.71.

Send coupon below now for your copy of the new free NAIL catalog of cassette programs for PET. If coupon is missing, send name and address on postcard to NAIL, Box F, Mobile, Al. 36601 and request catalog #8B.

The descriptions of the programs are extensive and usually have sample pictures taken right off the PET screen. This is important so that you know just what you are getting when you order.

**NAIL is a trademark of National Artificial Intelligence Laboratory
Copyright 1978 NAIL

*PET is a trademark of Commodore Business Machines

RT 164

NAIL, Box F, Mobile, Al. 36601
Please send me the items checked:

____New free NAIL catalog, without obligation.

____Sample Slot Machine cassette — I enclose $2.
(limit 1 per order)

____I write programs in ___BASIC, ___machine language, and may be interested in your distribution of them.

____SYS 7171 Machine Language Monitor (for 8K PET) — I enclose $29.71

Name ______________________________
(please print)

Address ______________________________

City ____________ State ________ Zip ____________

Circle 268 on inquiry card.

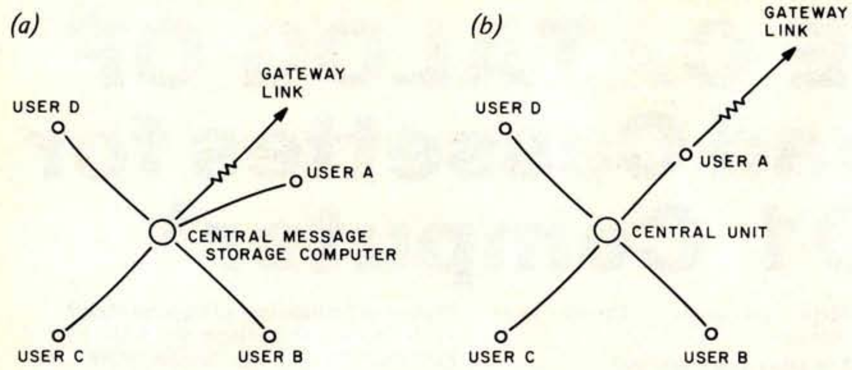


Figure 6: Two types of gateways that could be used. Figure 6a shows a generalized gateway that contains a central message storage computer that distributes messages to all of the users in the local group. A distributed gateway, figure 6b, would have the communication link enter through one of the users in the network.

loaded at first, it can be started using conventional radioteletypewriter techniques. Additional channels can be added using different modulation techniques for the advanced gateways as use increases. This will allow people with both types of equipment to have access to the system.

Gateway Requirements

There is no need for the central message storage computer in the local network to also act as the gateway to the satellite network. Figure 6 shows two network configurations. In figure 6a the central relay computer is the gateway, and in figure 6b user A is the gateway. Figure 6b can be considered an example of distributed processing within the network. In this case, the central relay forwards all messages with destinations outside the local network to user A when user A comes on line. Thus, if user A does not own a computer but has the equipment to communicate via the satellite, messages can still be sent to the satellite. Similarly, if user A does have a computer, it can come on line, get the messages for forwarding, then go off line and communicate with the satellite. When the satellite moves out of range, the computer can come back on line with replies. User A can define this mode because he or she may not be willing to donate the computer to the network full time.

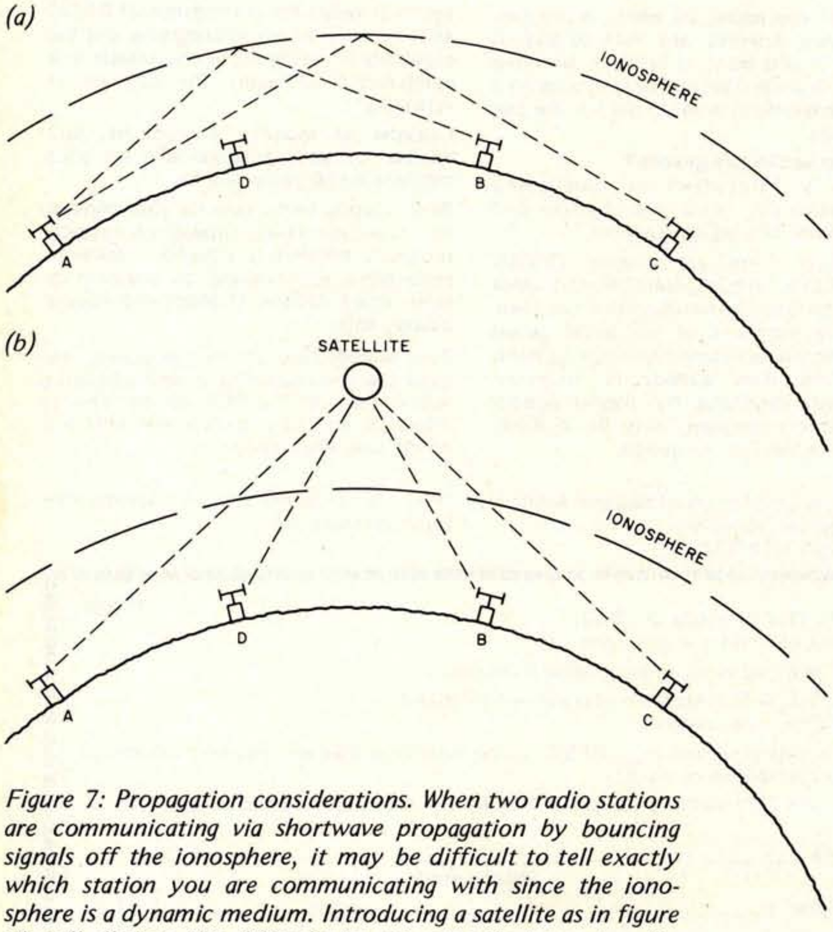


Figure 7: Propagation considerations. When two radio stations are communicating via shortwave propagation by bouncing signals off the ionosphere, it may be difficult to tell exactly which station you are communicating with since the ionosphere is a dynamic medium. Introducing a satellite as in figure 7b will alleviate this difficulty by not using the ionosphere for propagation.

Why the Satellite?

It was stated earlier that radio amateurs can communicate worldwide using radioteletypewriters by means of shortwave. Why not choose a shortwave frequency that is standard for all gateways and do without the satellite completely? Radio amateurs do operate in autostart mode on shortwave but these conventional shortwave communication paths possess many undesirable characteristics when compared with a satellite link. The professionals have long recognized this and are switching over from conventional shortwave communications to satellite communications as quickly as possible.

Let us consider some of these problems. The ionosphere that reflects shortwave signals back to earth is a dynamic reflecting medium. Its properties change from minute to minute, are different during the day and night, and are affected by solar activity. Solar flares on the sun may enhance or detract from the reflecting properties of the ionosphere at any particular frequency, sometimes eliminating them entirely. Thus the situation shown in figure 7a is typical of the conditions under which radio amateurs operate. Stations A and B are in contact with each other. Station C can also hear A but cannot hear B because those signals are skipping over station C.

During the contact between A and B conditions may change so that B becomes able to hear C but unable to hear A. Station D, who cannot hear A, B or C, may begin to transmit and (as conditions change) interfere with A, B or C. Alternately, C may not know that B is transmitting and may call A, thus causing interference to B's signals as received by A. Since one can't receive on one's own frequency during transmissions, one can't determine if any interference is present on the frequency. One could not determine its effect at the other station's site even if one could monitor the transmission.

Users of shortwave radioteletypewriter autostart recognize the problems associated with that mode of operation. The effects of propagation can be compensated to some extent by using relatively complex terminal units. At present there is no option but to put up with the idiosyncrasies of the ionosphere. However, the use of satellites for communications links can change all that. The satellite relay is depicted in figure 7b. Using frequencies that pass through the ionosphere with little or no reflection, all stations can transmit signals up to and receive signals from the satellite. A satellite *transponder* works in a similar manner to a terrestrial VHF repeater. Signals received by the satellite on one frequency (uplink) are amplified on board and retransmitted to the ground on a second frequency (downlink). Since the uplink and downlink are at different frequencies, it is possible to listen to your own signals coming back from the satellite. Thus, if any one channel is busy, all users can hear the signals and will know that they should wait until the frequency is clear before initiating a transmission. Since everybody can hear everybody else, the communication satellite provides wide area coverage nearly equal in performance to the local VHF repeater as far as the user is concerned.

Viability of Satellite Link

Implementing an amateur computer communications network by means of a satellite poses far fewer operational problems than implementing the network by means of conventional terrestrial relays. Thus it may be desirable – but is it viable?

The Radio Amateur Satellite Corporation (AMSAT) is currently building two satellites capable of meeting the space segment requirements for the network. AMSAT has a proven record in the construction and day-to-day management of Orbiting Satellites Carrying Amateur Radio (OSCAR). The first amateur built satellite was constructed by Project OSCAR in California and launched in 1961. OSCAR II, III and IV followed over the years. These satellites were short-lived experimental spacecraft that proved that amateurs had the capability to build and communicate through space satellites.

AMSAT, founded in 1969, arranged the launch of OSCAR V, built by a group of amateurs at Melbourne University in

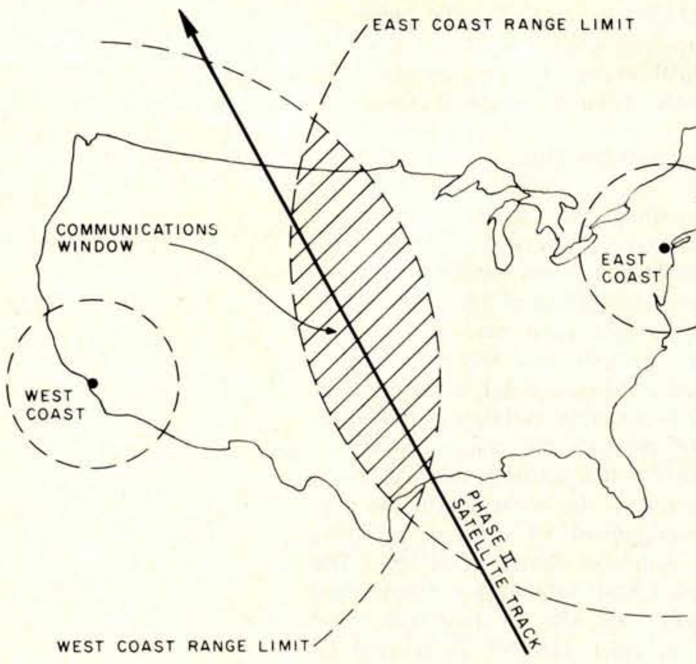


Figure 8: Coast to coast communication. As the Phase II satellite orbits, it is in a position to relay messages between the East Coast and the West Coast. Communication is possible only when the satellite is passing through the window (overlap occurs two or three times a day).

Australia. This was the first amateur built satellite that could be commanded from the ground. AMSAT used that capability to turn a beacon off during the week and on during weekends so that the majority of radio amateurs who were at home on weekends would be able to receive its signals. AMSAT subsequently constructed and obtained launches for the AMSAT OSCAR VI, VII and VIII spacecraft. These satellites were long life communications relay spacecraft, and contained equipment built by amateurs in Germany, Canada, Australia, Japan and the US. AMSAT OSCAR VI, launched in 1972, had a design lifetime of one year but operated successfully for almost five years until its batteries failed. AMSAT OSCAR VII, launched in 1974 with 3 year design life, is still going strong. AMSAT OSCAR VIII, launched as a replacement for the AMSAT OSCAR VI spacecraft, was put into orbit in March 1978.

These satellites, although relatively difficult to use, provide reliable communications links to radio amateurs separated by ground distances of up to 2400 km (1500 miles). The satellites travel in low orbits and hence are only within range of any one particular ground station for at most 25 minutes at one time, up to six times a day. Being in low orbits, they travel relatively fast; consequently signals relayed through them are subject to Doppler shifts. However, these satellites have shown the prowess of the radio amateur.

AMSAT is a worldwide organization with members in over 80 countries. Their satellites are used daily for educational demonstrations of the space sciences in schools all over the world. Free applications literature and ideas are available for the asking from the American Radio Relay League. These educational aspects were the prime reasons for NASA agreeing to provide launch places for them as secondary payloads. The European Space Agency (ESA) has agreed to provide a launch for the first AMSAT PHASE III spacecraft in late 1979. Thus, since AMSAT has a proven record and a launch date for the first PHASE III satellite, a satellite based amateur computer communications network is viable.

Using the Satellites

Although the first Phase III spacecraft will not be launched until December 1979, AMSAT OSCAR VII and VIII are up there and usable. It was these satellites that drew, and are still drawing, radio amateurs into personal computing. The orbits of these satellites, unlike that designed for the PHASE III satellite, are such that the spacecraft pass from horizon to horizon in less than 25 minutes. Amateurs soon found that communicating via these Phase II satellites required one hand to point the antennas, one to tune the receiver, one to tune the transmitter, another to hold the Morse code key or microphone and yet another to log the contact. Unlike the octopus, radio amateurs have only one pair of appendages and have therefore turned to computers for help. Computer programs have been developed that track the satellites, point antennas and calculate and compensate for Doppler shifts. AMSAT ground command stations have been using microprocessors ever since the days of the 8008. The Phase III spacecraft is itself controlled by a COSMAC microprocessor.

AMSAT now has an ongoing project for its members to build a low cost S-100 bus 8080 or Z-80 computer. AMSAT members have developed S-100 based hardware to interface antenna controllers to the computer, high power debugging software, and even a high level language called IPS. Thus any satellite user who is also interested in personal computers could set up a gateway for the local VHF radioteletypewriter repeater to use the Phase II satellites for relaying messages. The gateway would store messages until the satellite came into range and then forward them to the destination gateways. Figure 8 depicts how such a message might be transferred from the East Coast to the West Coast of the United States as a Phase II satellite passes over. However this technique is only practical for experimentation and relaying short messages.

Using Phase III Satellites

The AMSAT Phase III satellite will be placed in an elliptical orbit with its apogee (highest point) in the northern hemisphere. Studies by AMSAT have determined that this orbit would provide optimal service to the world's radio amateurs, most of whom live in the northern hemisphere. The satellite, whose orbit is sketched in figure 9, will be available for periods of between eight and 11 hours twice daily. Once the satellite gets reasonably above the horizon, there will be minimal Doppler effects, and pointing antennas will not be a problem because the apparent rate of travel will be slow. AMSAT will provide simple tracking software that will enable anyone to point an antenna at the satellite. This information may even be provided over the satellite. The satellite will be usable by any one station for periods of between 16 and 22 hours each day. This will allow communication links to be set up between any gateway in the northern hemisphere and any other one.

Summary

This article has discussed several new concepts in personal computing. It has outlined how amateur radio can easily provide area wide communications networks by connecting message storage computers to both the telephone line and the amateur radio VHF repeater. This allows computer experimenters access to the network without an amateur radio license. The AMSAT OSCAR amateur radio communication satellites can provide the computer enthusiast with an operationally simple worldwide personal computing network.■

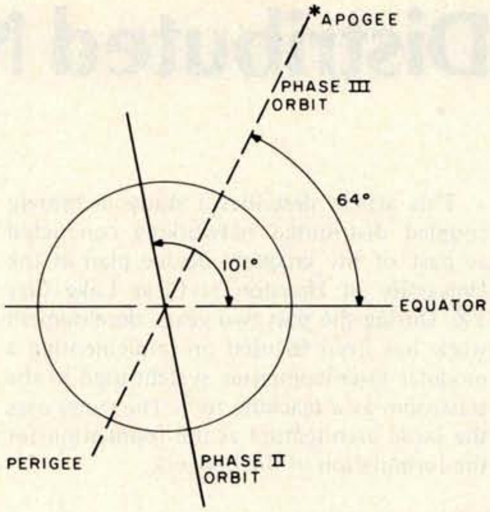


Figure 9: Typical orbit of the Phase III satellite. The greater the altitude of the spacecraft, the more ground it sees and thus the greater the communications range. The Phase II orbit at 1440 km allows communication ranges of up to 2400 km. The Phase III orbit, with a much higher altitude, will provide communications over the entire northern hemisphere and much of the southern hemisphere.

ORGANIZATIONS

AMSAT, the Radio Amateur Satellite Corp, POB 27, Washington DC 20044.

The American Radio Relay League, OSCAR Dept, 225 Main St, Newington CT 06111.

REFERENCE

DuBois, John L, "A Real Time Tracking System for Amateur Radio Satellite Communication Antennas," *AMSAT Newsletter,* March 1978.

Distributed Network

Glen Horton
Hickok Teaching Systems
2 Wheeling Av
Woburn MA 01801

This article describes a study in loosely coupled distributed networking conducted as part of my graduate degree plan at the University of Houston at Clear Lake City TX. During the past two years, development work has been focused on implementing a modular microcomputer system used in the classroom as a teaching tool. The study uses the basic architecture as the foundation for the formulation of the network.

Primary Modules

There are basically two modules needed to form a network of this type. First, a module is needed to effect communication between the processors which are part of the local network. This module is termed a "protocol module." A protocol module is essentially a block of 16 memory locations regarded as programmable memory by one processor and as read only memory by the rest of the processors in the local network. The protocol modules for each of the other processors in the local network function independently of the other protocol modules. Therefore, concurrent reads of system information are possible.

Second, a module for sharing resources is needed. This module is termed the "shared slot arrangement," an interface to the basic microcomputers in the local network. The shared slot arrangement is essentially a bus switch capable of logging and returning status and control information. The hardware of the shared slot arrangement, based on current control information, either "makes" or "breaks" a digital signal path between one of the local processing elements and a shared slot with its attached interface. Each shared slot arrangement is capable of controlling four shared slots that are independent of one another. Software routines which form part of an integrated operating system control the shared slots through the shared slot arrangements.

Example

As an example, figure 1 shows three processors loosely coupled by using shared slot arrangements and protocol modules. The user at processor 2 would like to read a file from the disk unit which is dedicated to processor 1. The user at processor 1 would like to print the contents of a file to the line printer on processor 3. The shared 8 K byte memory block is currently controlled by processor 2. The following operations describe a sequence to effect such transfers of information.

The executive software communicating with user 2 generates a disk access protocol. The disk access protocol contains all of the information necessary to retrieve the designated file. Processor 1 is then interrupted by processor 2. Processor 1 recognizes the interrupt request and protocol communication as a disk access protocol. Processor 2 then passes the shared slot with the 8 K memory block to processor 1. Processor 1 reads the designated file from the disk unit into the 8 K memory block. Then processor 1 passes the shared slot with the 8 K memory block back to processor 2. Processor 2 may then use the file's contents as desired by user 2.

User 1 has requested a shared slot with an 8 K memory block which is to be used to transfer a file to the line printer. Since processor 2 has finished with the shared slot and its attached 8 K memory block, it passes the slot over to processor 1. User 1 then reads the requested file into the 8 K memory block. The protocol for printing the file is initiated (processor 1 interrupts processor 3). Processor 3 recognizes the interrupt request and the line printer access protocol. Processor 1 then passes the shared slot and the 8 K memory block over to processor 3. Processor 3 reads the contents of the memory and prints them to the line printer.

Even though the description is tedious and long, the actual transfers of information are quick and relatively easy. It is important to note that the software is distributed among the three processors in the local network. The software is therefore broken into distinct modules best suited to their host processors and current configurations. This keeps the software overhead per processor small in relation to the user buffer areas and user support programs.

Shared Slot Arrangement Hardware

The shared slot arrangement is used to extend the bus of an existing microcomputer. The actual switching mechanism is under the control of small software subroutines dis-

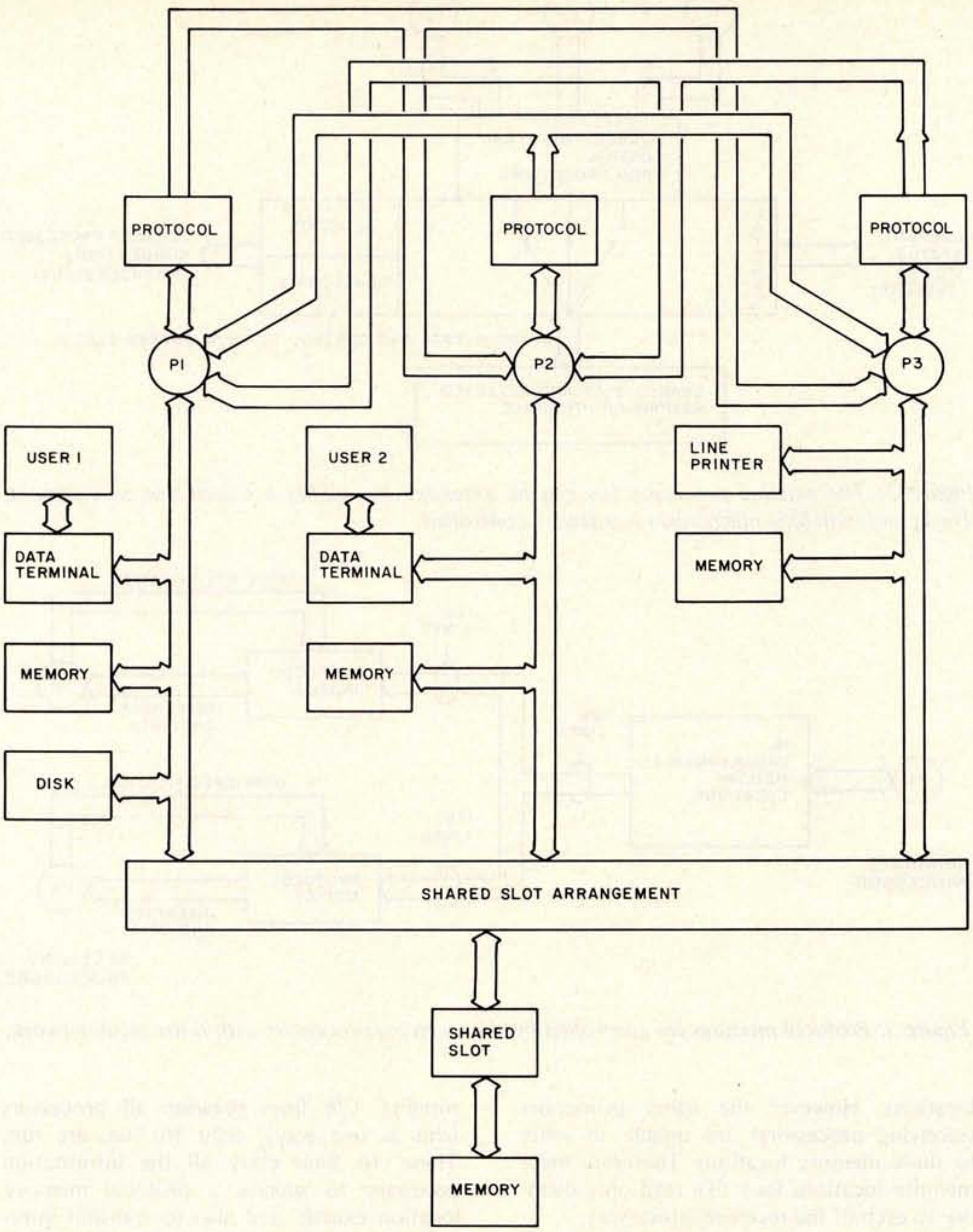


Figure 1: Loosely coupled network using protocol and shared slot modules.

tributed throughout the local network. An 8 line to 1 line multiplexer is used to construct the actual switching mechanism. Three lines, which describe the current processor number, determine which of the switches is to be made or broken. When a connection is broken, the shared slot is said to be deallocated. The status lines then reflect this condition. When a shared slot connection is made, it is allocated and the status will again reflect the change in the condition of the shared slots. The last available status can be read by any processor in the local network at any time. This also holds true for the current processor number (see figure 2).

Protocol Module Hardware

Figure 3 shows how the protocol modules are controlled by the receiving processors in the local network. Processor 1 is the originating processor in this example. It has 16 programmable memory locations which it uses to communicate with the other processors of the system. The other processors in the system each have a protocol module capable of reading one of those 16 memory

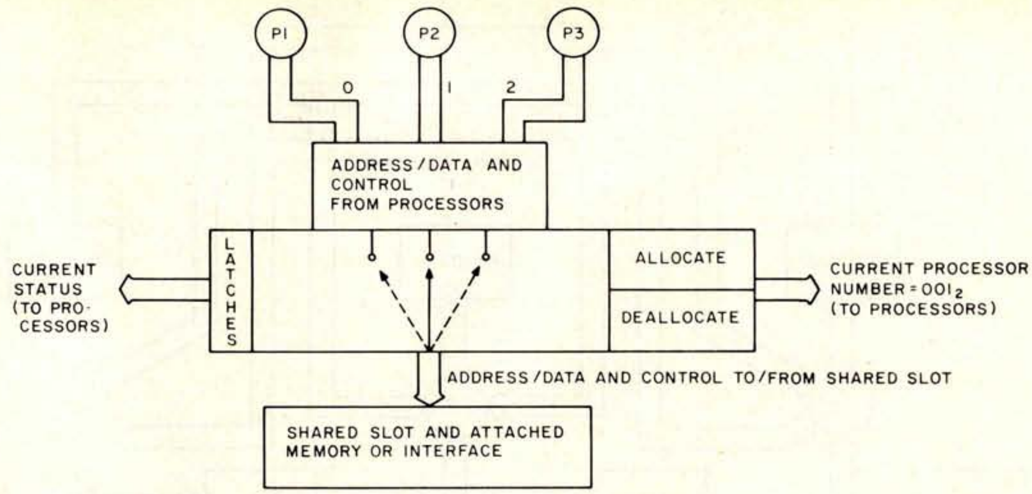


Figure 2: The existing processor bus can be extended by adding a shared slot arrangement. The actual switching mechanism is software controlled.

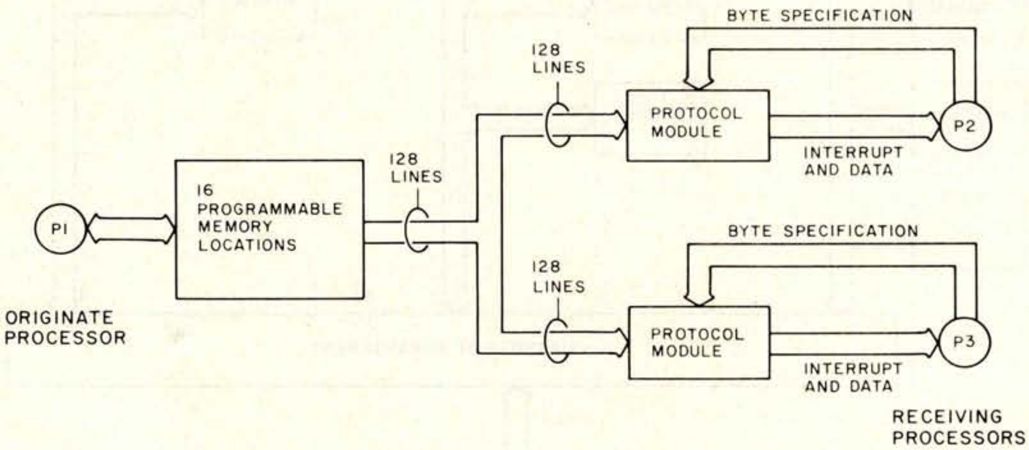


Figure 3: Protocol modules are controlled by the receiving processors within the local network.

locations. However the other processors (receiving processors) are unable to write to these memory locations. Therefore these memory locations look like read only memory to each of the receiving processors.

Since each receiving processor has a protocol module unique to each of the other processors, it may read another processor's protocol independent of the other processor's reads. The receiving processor has a multilevel interrupt interface that determines the number of the processor generating the interrupt. It is from this interpretation that the receiving processor determines which protocol to read. A protocol module then decodes the byte specification (from the receiving processor) to determine which memory location (from the originating processor's communication area) is to be transferred to the receiving processor.

By multiplexing the memory in this fashion, a considerable saving in actual wire connections is possible. Instead of running 128 lines between all processors (this is one way), only 16 lines are run. These 16 lines carry all the information necessary to specify a protocol memory location exactly and also to transmit interrupt request signals from the originating processor to the receiving processor.

Summary

A loosely coupled distributed network of microcomputers was formed using protocol and shared slot hardware modules. System information is passed in bursts of read operations by the receiving processors. Software at the receiving processor determines the appropriate action to take in each instance. The software routines also update the shared slot arrangement status and control signals and thus allow any of the attached resources to be shared. By joining multiple copies of the basic modules, expansion of this framework is easily accomplished.■

Programming Quickies

Checkbook Balancer

Rod Hallen
Road Runner Ranch
POB 73
Tombstone AZ 85638

```
100 PRINT "CHECKBOOK BALANCER"
105 REM "BY ROD HALLEN – TOMBSTONE, ARIZONA-17 DEC 1977"
110 PRINT
120 INPUT "STATEMENT BALANCE $", A
130 INPUT "CHECKBOOK BALANCE $", B
140 PRINT "CHECKS NOT CLEARED $";
150 INPUT C,
160 LET T=T+C
170 IF C>0 THEN GOTO 150
180 PRINT "DEPOSITS NOT RECORDED $";
190 INPUT D,
200 LET U=U+D
210 IF D>0 THEN GOTO 190
220 LET E=(A–T)+U
230 LET F=B–E
240 PRINT
250 PRINT
260 PRINT"————————————————————————————————————"
270 PRINT "CHECKS NOT CLEARED – TOTAL $"; T
280 PRINT
290 PRINT "DEPOSITS NOT RECORDED – TOTAL $"; U
300 PRINT
310 PRINT "ACTUAL BALANCE $"; E
320 PRINT
330 IF F>0 THEN PRINT "CHECKBOOK OVER BY $"; F
340 IF F=0 THEN PRINT "CHECKBOOK BALANCES"
350 IF F <0 THEN PRINT "CHECKBOOK UNDER BY $"; ABS (F)
360 PRINT" ————————————————————————————————————"
370 END
```

Listing 1: Checkbook balancing program written in Processor Technology BASIC 5.

I balance my checkbook each month by adding together the statement balance and any deposits which are made after the statement date. I then subtract from this the total of all checks which have been written but not cleared by the statement date. The answer should agree with the balance shown in my checkbook.

The program in listing 1 is my checkbook balancer written in Processor Technology's BASIC 5 for my SOL system. It follows the formula:

Statement Balance + Deposits Not Recorded – Checks Not Cleared = Actual Balance.

Lines 120 to 210 input the necessary information, lines 220 and 230 compute the answers, and lines 240 to 360 print the results. Lines 150 to 170 ask for "checks not cleared" one at a time and keep a running total. When you have entered all of the checks, type 0 and program control will move to line 180. Lines 190 to 210 work in the same way. Entering a 0 after all deposits not recorded have been entered causes program execution to move to line 220.

A quick scan through listing 1 will reveal that the printed results are the total amount of checks not cleared, the total amount of deposits not recorded, the current balance and the status of your checkbook balance. This last value is indicated as either over or under balanced and by how much. If the answer is over or under, you have the task of finding out why. One thing to remember; if you haven't been balancing your checkbook every month, you may have been carrying a bad balance for some time without knowing it. If the difference is not too large, the best method may be to enter the actual balance in your checkbook and run the program each month from now on.

The task that the checkbook balancing program performs used to take me ten minutes with a pencil and pocket calculator. Considering that it now takes about three minutes and is only required once a month, not much time has been saved. However, as more and more of my chores are computerized, the amount of time freed for other uses increases rapidly.

[A variation of this program will be required for those who use "negotiable order of withdrawal" checking accounts which pay about 5 percent per annum interest. . . .CH]■

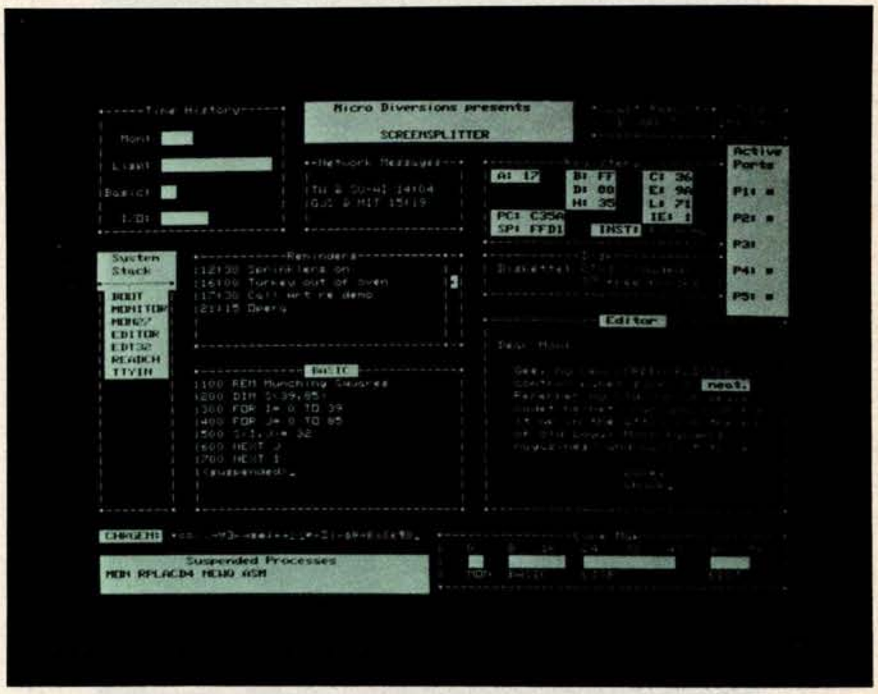
Circle 95 on inquiry card.

See It All at Once!

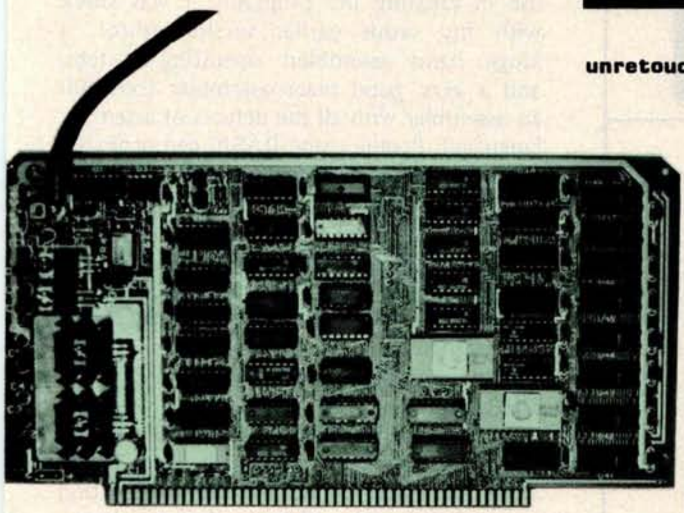
with a **Screensplitter**

Text Editing and Multiple Process Display System

from **Micro Diversions**



unretouched photo **13" or larger slow phosphor monitor recommended**



S-100 Buss

- **40 lines, 86 characters/line UPPER/lower case full ASCII with special symbols**
- **Onboard intelligence: 1K byte fast subscreen control, positioning and text formatting software**
- **Custom reprogrammable character generator for unusual applications**
- **140 page Owner's Guide covering hardware, software, system integration**
- **Available from stock**

$295 Kit
$395 Assembled
$15 Owner's Guide only

VISA
master charge

Micro Diversions, Inc.
8455-D Tyco Road
Vienna, Virginia 22180
(703) 827-0888

Circle 225 on inquiry card.

Circle 293 on inquiry card.

Continued from page 6

number of vendors, at prices in the $2000 to $5000 range. This hardware basis for software tools includes a processor, as much memory above about 16 K bytes as possible, a terminal device, and electronically addressable mass storage on floppy disk or its equivalent. (Two drives are much much preferable to one, so copies can be made; I have only one drive and feel the lack every time I have to transfer to a new disk.) A printer, such as my obsolete Model 33 Teletype, is useful whenever final listings or working maps of the source code are needed. In my case, the processor and peripherals were designed and integrated personally, but numerous commercial hardware products are essentially identical in function and probably less expensive if you value time spent.

Readers interested in doing similar projects with nonhomebrew equipment will undoubtedly have an easier time of it, since chances are that a good operating system and assembler, BASIC interpreter or Pascal compiler would be available for use in creating the programs. I was stuck with my crude earlier version editor, a kluge hand assembled operating system, and a very good macroassembler (but still an assembler with all the defects of assembly language). People using BASIC can probably create assemblers, text editors, or interpreters far easier than I could with my system such as it was: witness the Tiny Pascal project of Herbert Yuen and Kin-Man Chung in this BYTE, several text editors and assemblers I know about, and my friend Dan Fylstra's 6502 assembler in BASIC which is marketed for the PET by Personal Software.

But never mind the lack of the ultimate in tools. Part of the reason for deciding to work on a text editor was that in addition to exploring the software design aspects, I would in effect be using the previous tools to bootstrap a better set of tools. (Ah, you may note, there was a pragmatic and short-term result of the project; like many activities with long-term good effects, there are short-term good effects here, too.)

In my earlier editor, written in 1976, the emphasis was on simplicity and getting an editor written in minimal time subject to the constraints of not having an editor. As a result, I took coding shortcuts which are functionally correct but not particularly fast. Thus all operations were on an incremental character by character basis. For example, insertions in very long files take incredibly long amounts of time per character since the entire file must be shifted.

Circle 41 on inquiry card.

The new editor program was the last software project which I would implement while having to put up with the older editor's faults.

Designing and Implementing the Program

I have a number of biases in the design of editor programs, biases which reflect the hardware I have and my philosophies of interaction. This is one of the reasons for engaging in a do-it-yourself design.

For example, I am personally devoted to that school of text editing which uses a minimum of keystroking, where single keys for the most part are used to invoke all commands. I am also committed to avoiding my own stupid errors which can be disastrous if the wrong single key is pressed. So, I oriented the editor design to single keystroke inputs which have three general classes of potential disaster if inadvertently invoked.

There are benign functions which can always be executed by mistake without hurting the file. These are commands like *print* or *go to the next line* or *back up the character pointer* or *go to the beginning of the file.* In each case, the command merely manipulates file pointers without many side effects. These commands are executed by single keystrokes.

A second class of functions are what might be called really serious because of the effects they may have on the file. In this class are found things like *insert text at the current character location, replace text at the current character location* or *execute the macrostring.* To protect the file from inadvertent use, these commands require use of the control or shift key on my terminal to generate various ASCII control characters. The control key must be pressed at the same time as some other key, so a 2 key and 2 finger combination is required, typically with the left hand on the control or shift key and the right hand picking the particular command key.

And finally, there is a whole class of potentially disastrous commands like *write the file to disk, read the file from disk, assemble a program, load a program, jump to an arbitrary address* and so on. These commands are typically invoked by a shift or control character so that two keys must be pressed as in the previous class. Then, to provide further protection, a message is displayed on the terminal requesting confirmation with an arbitrary character. An escape character can be input if the operation is to be aborted before the text buffer is actually or potentially zapped.

Another example of a personal bias is that the editor would have to have provisions for search and change operations, execution of macro strings of editor commands, and the ability to move blocks of text around from one place to another in the file. Not all editors (especially not the typical BASIC interpreter's editor) have this sort of capability. Most separately available programs which go under the name of text editor have these capabilities.

There is a myriad of much more detailed choices involved in the design of a program for text editing, for example: how to allocate memory, how to decode commands, internal coding conventions, provision for expansion into a full operating system monitor within the framework of an editor. The educational results of the editor writing process are: a specific personal involvement with complex questions, the necessity of making the choices involved in the design within the framework of these questions, and a broader understanding of computing. So, if you want an answer to the question: "What educational and pragmatic project can I do with my computer?", my answer would be, "write an editor."■

SOLID STATE MUSIC IS MISLEADING!

We won't mislead you any longer. Solid State Music—the name is misleading because we're so much more than just music synthesizer boards. We offer the widest line of S-100 boards in the industry. Memory boards for RAM, ROM, and EPROM. Video interface and I/O boards, extender and mother boards, prototyping and vector jump boards, a new CPU board, and, of course, music boards. So we're changing our name to SSM.*

You've been enjoying the pleasures of our company for over four years now. Few competitors can say as much. In fact, Solid State Music was building quality boards when the first S-100 bus left the Altair® station.

As SSM,* we will continue to offer boards that provide quality, flexibility, and good design. Boards that represent value. Not the least expensive. Not the most expensive. Simply the best combination of price and quality available. That's SSM.*

Solid State Music has been known for service. Fast delivery, ready customer support, and a strong product warranty. Our name is changing. Our tradition of service is not. Orders still shipped from stock. A one year warranty on assembled and tested boards, 90 days on kits. And our people are still on hand, still glad to answer your questions.

Check out the Blue Boards of Happiness when you're after Solid State Music quality, reliability, and flexibility, backed by the friendly support of the oldest S-100 board manufacturer in the marketplace. If you liked us as Solid State Music, you'll love us as SSM.*

SSM*

2116 Walsh Avenue
Santa Clara, CA 95050
(408) 246-2707

***We used to be Solid State Music. We still make the blue boards.**

Circle 335 on inquiry card.

SCELBI MAKES

**You can rely on SCELBI publications. Because
Computer info that is fun and exciting.
Written and illustrated with YOU, the compute**

NEW!

***"Learn Micro-Computers."* Quickly, easily and enjoyably. For the beginner just getting started in microcomputers.**

Here are all the fundamentals behind the operation of virtually all microcomputers. Everything the beginner needs in one handy package. Contains the popular book **Understanding Microcomputers.** Written in simple English. Clear. Concise. Accepted as the standard for the neophyte, you must own this 300-page no-nonsense, easy-reading text. Includes simple-to-use glossary of key microcomputer words. Companion cassette helps you get right into it. Chapter-by-chapter talking synopsis. Key review questions. References to page numbers for further review. You'll gain extra knowledge needed for reading and understanding computer magazines, manufacturer's literature and to feel "at home" around computers. A great new idea developed by the computer experts at SCELBI. Get Yours Now! At your favorite computer/electronics dealer. Or order direct. **Learn Micro-Computers,** only **$14.95.** Paperback edition of book alone: **$9.95.** Hardcover edition alone: **$14.95.** Shipping/handling extra.* (Available after December 1)

SENSE

they make sense! They're easy to read and understand. For the neophyte. The pro. The advanced wizard. Everyone. hobbyist in mind.

SCELBI's Software Gourmet Guides and Cookbooks for "8080" and "6800"

Now **you** can cook up mouth-watering programs. Delectable "how to" facts include "8080" or "6800" instruction sets. How to manipulate stacks. Flow charts. Source listings. General purpose routines for multiple precision operation. Programming time delays for real time. And much, much more. Includes floating point arithmetic routines. Specify "8080" or "6800". Get your copy today! They're at your favorite computer/electronics dealer. Or, order direct. 8080 Paperback: **$10.95.** Vinyl: **$12.95.** Hardcover: **$14.95.** 6800, Paperback only: **$10.95.** Shipping/handling extra.*

NEW!

Steve Ciarcia's hilarious new computer novel, "Take my computer . . . please!"

An uproariously funny full length book about Steve's true-to-life misadventures and his computer's inability to cooperate. Read his off-the-wall attempt to beat the Jai-Alai system. Or how he attempts a stock market killing but logs wrong info into his broker's computer. What happens when Steve sets up a computerized speed trap? And page after belly laughing page of nonsense, jollies and even illustrations to bring the stories to life. 128 pages. Hard cover. You've got to get yours now! See it at your favorite computer/electronics dealer. Or, order direct. **$5.95.** Shipping/handling extra.*

The *Z80 Instruction Handbook* now available!

Your complete guide to the powerful Z80 instruction set. This handy, compact reference provides a clear explanation. It's an ever ready instant reference that can be carried in your pocket. Explains the instruction set in meticulous detail. Industry standard mnemonics are used throughout. Machine codes are presented in both octal and hexadecimal format. A convenient index lists all instructions alphabetically along with machine codes and timing information. Don't wait. It's available at computer and electronics retail outlets or direct from SCELBI. Only **$4.95.** Shipping/handling extra.*

Looking for holiday gift ideas? Give someone the world of computers! Make it a SCELBI publication. They make excellent Christmas gifts. There's one just right for that special someone on your list.

Scelbi Publications
P.O. Box 133 PP STN, Milford, CT 06460
(a division of Scelbi Computer Consulting, Inc.)
(203) 874-1573

*Prices shown for North American customers. Master Charge, VISA, Postal and Bank Money Orders preferred. Personal checks delay shipping up to 4 weeks. Pricing and availability subject to change without notice. **IMPORTANT!** Include 75¢ postage/handling for each book delivered by U.S. Mail Book Rate or $2 for each book shipped First Class or via UPS.

Machine Language Puzzler

TIMOUT

Christopher Strangio
CAMI Research
43 Bailey Rd
Watertown MA 02172

Subroutine "TIMOUTN" is used to introduce a time delay in the execution of a main program. The program, written in 8080 assembly language, is shown as follows:

```
TIMOUTN:  PUSH B         10
          MVI B, #X       5
          DCR A           5
LOOP:     CPI 0           5
          CNZ TIMOUTN    10*
          DCR B           5
          JNZ LOOP       10
          INR A           5
          POP B          10
          RET            10
```

Assume that this many total machine states are required for each instruction. Note that the call instruction marked * has been arbitrarily assigned a *fixed* time of 10 to simplify the calculation.

Two numerical values are needed by TIMOUTN to define the delay period. The first value, X, is loaded with an MVI (move immediate) instruction inside the subroutine; the second value, Y, is passed to the subroutine in the accumulator.

To call TIMOUTN, the following general procedure is used:

```
PUSH A
MVI A, #Y
CALL TIMOUTN
POP A
```

Questions

(a) What is the *total* time delay (calculated in machine states) obtained from TIMOUTN if:

1) X = 1 and Y = 1
2) X = 2 and Y = 1
3) X = 1 and Y = 2
4) X = 2 and Y = 2

(b) Which instructions determine the basic period of delay in TIMOUTN? What is the period of the basic timing loop?

(c) It is interesting to note that if X = Y = hexadecimal 0F, the time delay provided by TIMOUTN would be approximately *416,000 years* (assuming the processor clock period is 1 μs per state)! Under these conditions, what is the maximum number of bytes by which the stack would grow during the execution of TIMOUTN? (If X = Y = hexadecimal FF, the total time delay would be far longer than the known age of the universe, about 4 billion years!)

(d) The period of the basic loop, found in part (c), can be labeled Q. In terms of X, Y, and Q, write a simple algebraic equation that approximately describes the total delay of TIMOUTN. The answers to this puzzle are found on page 138. But don't go there just yet; try puzzling out the answers first.■

Technical Systems Consultants, Inc.

The Software Company

TSC, Technical Systems Consultants, is **the software company** for all the newest, most innovative ideas in computer software. TSC builds a variety of programs, packages and games so you can get down to business or just some fun.

Text Editing System

The most complete and versatile editor available for the 6800 and 8080 micro. The system is line and content oriented for speed and efficiency and features such commands as block move and copy, append and overlay, as well as string manipulators. The 6800 version requires 5K beginning at 0 hex, the 8080 needs 6K starting at 1000 hex. Both should have additional file space as required.

SL68-24	**6800 Text Editing System**	**$23.50**
SL68-24C	**w/cassete**	**$30.45**
SL68-24P	**w/paper tape**	**$31.50**
SL68-24D	**w/mini flex disc**	**$31.50**
SL68-24F	**w/flex disc**	**$50.00**
SL80-10	**8080 Text Editing System**	**$28.50**
SL80-10P	**w/paper tape**	**$37.50**
SL80-10F	**w/ CP/M disc**	**$40.00**

Space Voyage

SL68-5C	**w/cassette**	**$18.95**
SL68-5P	**w/paper tape**	**$19.00**
SL80-9P	**w/paper tape**	**$19.00**

6800 Disassembler

SL68-27C	**w/cassette**	**$15.95**
SL68-27P	**w/paper tape**	**$13.00**

Micro BASIC Plus for 6800

SL68-19C	**w/cassette**	**$22.90**
SL68-19P	**w/paper tape**	**$21.95**

6800 Floating Point Package

SL-68-4	**Floating Point Package**	**$6.50**
SL68-4P	**w/paper tape**	**$9.50**

Text Processing System

As a complement to the Editor, the Processor supports over 50 commands for left, right or center justification, titling, paging and general text output formatting. A loop command is available for repeated formatting jobs such as form letters. Also included are capabilities for macro definition to build special formatting commands. The program requires about 8K of RAM and previously edited text.

SL68-29	**6800 Text Processor**	**$32.00**
SL68-29C	**w/cassette**	**$38.95**
SL68-29P	**w/paper tape**	**$40.00**
SL68-29D	**w/mini flex disc**	**$40.00**
SL68-29F	**w/flex disc**	**$75.00**
SL80-11	**8080 Text Processor**	**$32.00**
SL80-11P	**w/paper tape**	**$41.00**
SL80-11F	**w/ CP/M disc**	**$50.00**

Relocator

This self-prompting, easy to use program relocates object code in RAM or from tape. Complete instructions included for making the TSC Editor and Assembler or Editor and Text Processor co-resident. (As sold they reside in the same area.) Just over 1K in length.

SL68-28	**6800 Relocator**	**$ 8.00**
SL68-28C	**w/cassette**	**$14.95**
SL80-13	**8080 Relocator**	**$ 8.00**
SL80-13P	**w/paper tape**	**$13.00**

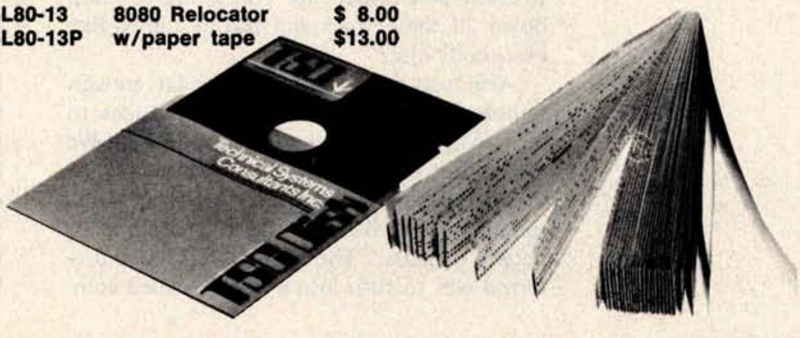
The Mnemonic Assembler

The ideal addition to the Text Editing System, together they form a complete program development center. The Assembler is one of the most versatile available and allows for easy adaptation to most systems. The Assembler is many times faster than other resident assemblers. Requires aproximately 5.5K plus file and symbol table space.

SL68-26	**6800 Mnemonic Assembler**	**$23.50**
SL68-26C	**w/cassette**	**$30.45**
SL68-26P	**w/paper tape**	**$31.50**
SL68-26D	**w/mini flex disc**	**$31.50**
SL68-26F	**w/flex disc**	**$50.00**
SL80-12	**8080 Mnemonic Assembler**	**$25.00**
SL80-12P	**w/paper tape**	**$34.00**
SL80-12F	**w/ CP/M disc**	**$40.00**

All programs include complete source listing. Cassettes are in the Kansas City standard format.

All orders should include check or money order. Add 3% for postage and for orders under $10, please add $1 for handling. Send 25¢ for a complete software catalog.



TSC™ Technical Systems Consultants, Inc.
Box 2574 W. Lafayette, IN 47906

Specialists in Software & Hardware for Industry & the Hobbyist

Circle 370 on inquiry card.

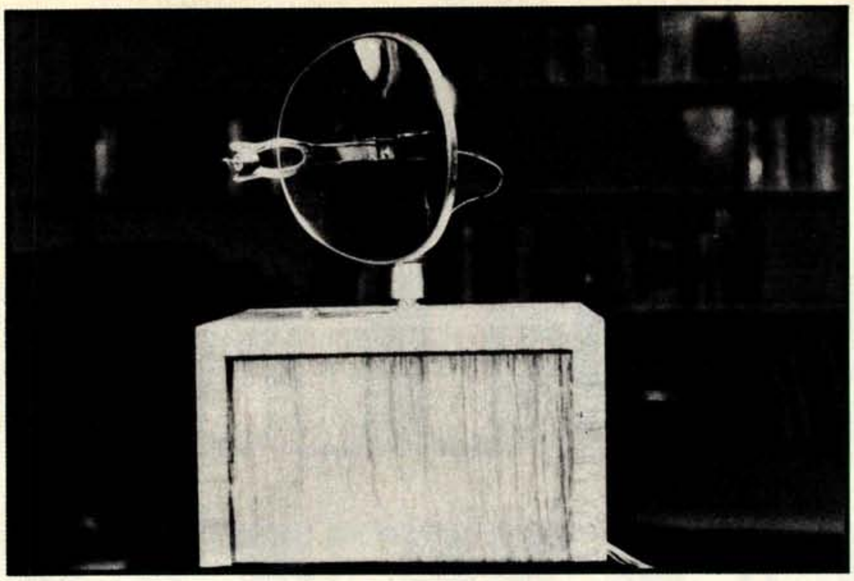


Photo 1: A stepper motor controlled scanning sensor capable of detecting both infrared and visible light. A photo detector is mounted at the focal point of an inexpensive parabolic solar cigarette lighter. The computer controlled unit is capable of following a moving flashlight, detecting headlights in a driveway, and many other applications.

I've Got You in My Scanner!

A Computer Controlled Stepper Motor Light Scanner

Ciarcia's Circuit Cellar

Steve Ciarcia
POB 582
Glastonbury CT 06033

"Boy, sitting here is really relaxing, isn't it, Lloyd?" I leaned back in the recliner and looked out of my living room window at the dense forest no further than 30 feet from where we sat. The sliding glass doors were open and occasionally some furry little animals could be seen darting in and out through the underbrush feverishly searching for dinner. The setting midsummer sun created an orange and yellow background for the beautiful scene. I wondered why I had ever waited for five years before moving out of the city. All that noise and congestion. This was so peaceful.

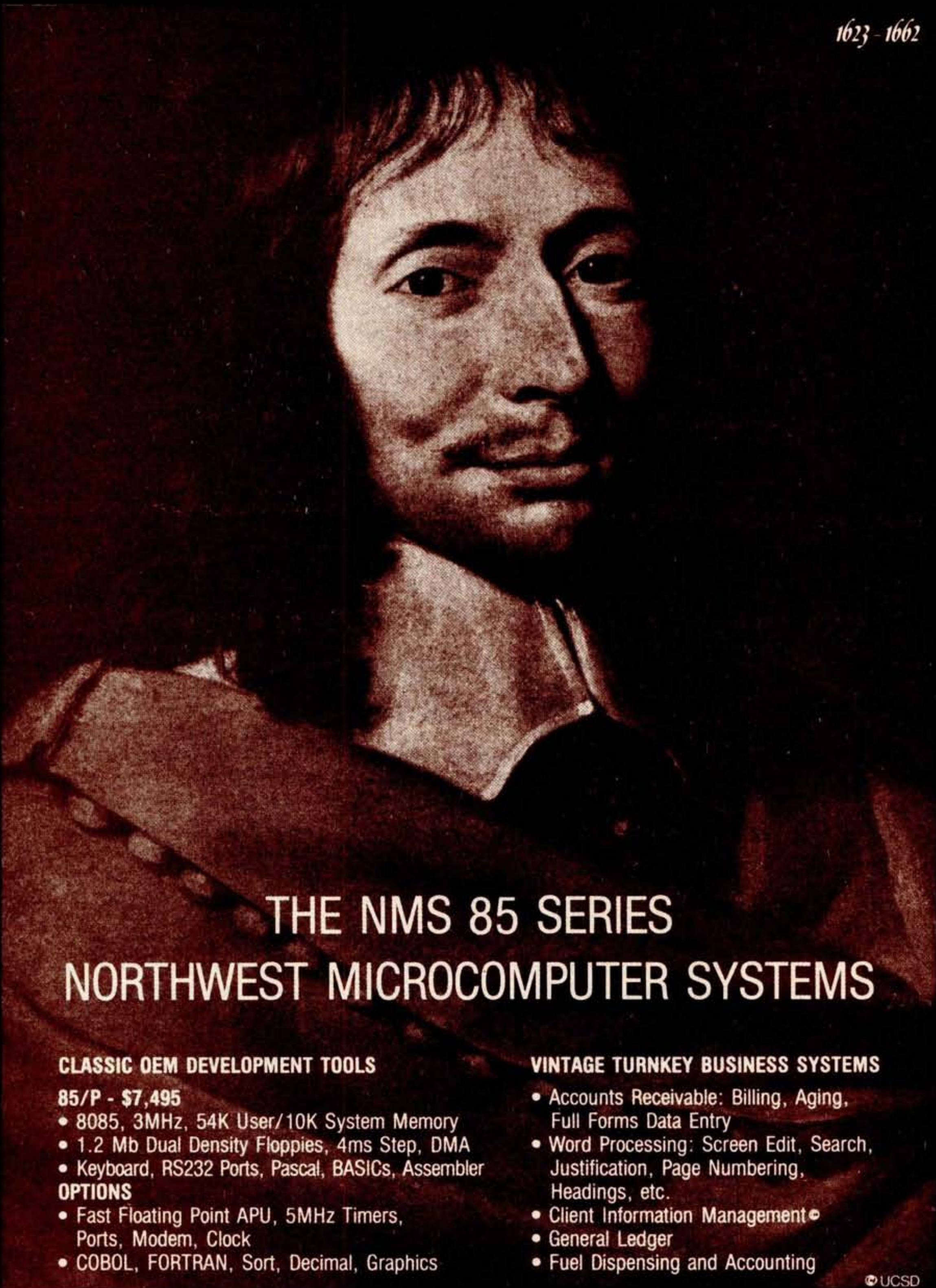
"It's very nice up here, Steve. I especially like the big driveway. You have lots of room to park cars. When are you going to chop down all these trees and put in a lawn like everybody else?"

Although I was not actually far enough removed from such suburban beatitudes to scoff with impunity, I piped back, "Bah! We moved out here to get away from the rest of the world. The last thing I want to do is be reminded of civilization, whether that means people or grass." The alternative I much preferred was to turn into a leaf shrouded computer hermit. "Lloyd, if I could figure a way to put in a moat with alligators, I'd do it," I said, tongue in cheek.

Such was the tone of our conversation for the next few hours. Whenever Lloyd ventured into the Connecticut wilderness (as he called it) he would stop by and visit me. Because of Lloyd's practical knowledge of computer related subjects I often used him as a sounding board for article ideas. His diplomatic responses sometimes disguised his opinions so tactfully, though, that I wasn't always sure what he really thought.

Before we knew it the sun had set and we were enveloped in darkness. The moonlight cast a silvery glow across the tops of the trees but hardly penetrated to the underbrush. But the moonlight was of small consequence to us. Even with the additional dim light escaping from the next room there was barely enough illumination to discriminate facial expressions, but there was sufficient backlighting for the little night creatures to observe us. Having lived in my new house for two months I had finally become accustomed to the nocturnal sounds and no longer experienced heart failure whenever I

Blaise Pascal



121 EAST ELEVENTH, EUGENE, OREGON 97401 (503) 485-0626

Circle 286 on inquiry card.

detected a pair of eyeballs peering at me from between the tree limbs. It was, after all, the domain of the owl and deer and I was the intruder.

I was less sure of the effect on Lloyd. Far away from the accustomed roar of jets at JFK and the traffic jams on the Long Island Expressway, he was suddenly very quiet, almost subdued, as he stared out the window into the darkness. Suddenly his eyes became focused on something in the distance and, gripping the arms of the chair tightly enough to leave an impression, he craned his neck to get a closer look. Something had obviously attracted his interest.

"I saw something!" he said.

"It's probably some possum checking us out or some other small animal after the dinner scraps I put outside."

"No, it's no animal. At least no small one. I thought I saw a light too. How many possums glow in the dark?"

"Don't be an alarmist. There's nothing to worry about."

"Look, there it is again, Steve! I think someone's out there."

I saw a form way off in the darkness. It was definitely an erect biped moving through the trees. . . .

I, too, saw a form way off in the darkness. It was definitely an erect biped moving between the trees and making considerable noise as it went.

We jumped from our chairs and crouched together looking out through the screen. The same thought came to both of us: "Is it Bigfoot?!"

"Wait a minute," I said in a hushed tone. "This is Connecticut. That's absurd! How can it be Bigfoot? Besides, since when does Bigfoot carry a flashlight?"

The bright beam of a flashlight shot from the stranger's hand. The dim light revealed a large man in coveralls dragging a heavy sack and carrying something over his shoulder.

"He must have a gun!" Lloyd gulped, and we both dropped to floor level. "Quick! Call the police or something! Better yet turn on the outside flood lights. Maybe it will scare him away."

"Look, Lloyd, if you want to become a moving target walking across the room to the light switch go right ahead."

"How come your burglar alarm hasn't turned the lights on?"

I thought about the alarm system for a moment and then answered, "I've got sensors all over the driveway and the road leading to the house. I didn't put them out in the woods because it's more likely that someone would come down the road rather than hike through the woods."

"How come nobody told *him* that?"

"Look, it'll pick him up anyway if he comes within 50 feet of the house."

Before Lloyd could reply, the man in the woods stopped in a clear area. The object slung over his shoulder wasn't a gun, but a shovel. He started to dig.

"Steve, do you think he's burying a body?"

I gathered up what courage I could and decided to go out and confront the perpetrator before my front yard looked like the aftermath of Dunkirk. "Come on, let's find out what he's doing."

As we approached, the man ignored our presence and kept digging. Occasionally he pointed his flashlight into the hole, then dumped the contents of the shovel into the sack. Was there buried treasure on my property?

"Excuse me, sir? Excuse me?" I said softly but with resolve. When I did not receive a response I stepped closer and repeated a little louder, "Excuse me, sir?!"

"Shhhh, Sonny! Da ya wanna scare all these critters away? It's hard enough making a living these days without everyone getting into the act. This here is my mound, Sonny!"

Mound? Sonny? I listened to his voice closely now and examined his features as best as I could in the moonlight. His accent was definitely Maine — deep woods Maine, and I put his age conservatively at 70. He seemed harmless enough, but I still had some unanswered questions.

"Sir, do you mind telling me what you're digging?"

He swung the shovel up over his shoulder and turned toward me. His face was weatherbeaten and aged, yet there was a youthful glint in his eyes. The gravity of the situation evaporated as he answered, "Worms."

"You're digging *what?*" Lloyd chimed in.

"Worms," he answered again. "This hea mound," he pointed at the area where he was digging," is one of the best night crawler mounds in the county. Youst to be a farm around here, few yease back. This was the compost heap. Worms love it, ya know." He chuckled as he explained the worm breeding business to us city slickers. "I been diggin' around here off an on for 30 yease. Then someone came along and put a house on it." He pointed a boney finger at my place.

"I had no idea . . . ," I said, somewhat embarrassed.

"No matter . . . them's still my worms! I got ten spots just like this one and I'm transplantin' my worms. You know what a night crawler like this is worth around here, Sonny?" He reached into the hole and suddenly I had a handful of worms held in my face. I took great care to take shallow breaths lest I accidentally gasp and inhale one of the squirmers.

"No sir. I don't fish."

"Well, they're worth plenty. And I got to dig a lot of them durin summa cause there ain't no worms in winta, Sonny."

His logic was irrefutable. He obviously earned his living at this. I felt a bit sad for the old codger. His digging really wasn't an inconvenience as long as he only took the worms and left the dirt. I didn't know enough about night crawlers to know the best time of night for harvest but I was sure we could work something out. I held my hand out to his and said, "You don't have to transplant your worms. What's a few worms between friends?"

A Modification to the Alarm System

At the conclusion of this episode I couldn't help but be concerned about the detection logic of the sophisticated alarm I had installed. There were sensors across critical points in the driveway and the road leading to the house that could detect the presence of a car or person. But, because of the likelihood of false triggering by wild animals, I hesitated to place similar detectors in the woods surrounding the house. I had thought the woods were impassable, but I guess I was wrong. The common denominator for anyone trying to make it through those woods at night is the necessity of a light. It should seem easy in principle to just place a light activated switch out there and activate the sequence when it detects some light source. Unfortunately, since the sensitivity would have to be relatively high, it would no doubt be accidentally triggered from lightning bolts and wayward fireflies. Complex integration and delay logic could be incorporated which would eliminate many false alarms but light level is still the only detection quantity.

Since a light source such as a flashlight or motor vehicle would have to move to approach the house, motion is another necessary parameter to consider. Most motion detection systems are passive beams whose sequential interruption triggers an appropriate response to a time and distance algorithm. Such a system of infrared or visible light beams, sufficient to protect four or five acres of property, would be prohibitively expensive if it incorporated a laser light source, and probably couldn't work reliably without one.

If we accept the premise that anyone coming through the woods on my property would need a flashlight or lantern, etc, then to detect the presence of an intruder requires a combination of light level and motion.

There are a number of methods that achieve the desired result. The most straightforward is to use a television camera, digitize

Circuit Cellar experimental setup showing the parabolic scanner as it detects a candle.

the image, and after adjusting for ambient light changes, compare it to a previous digitized image. Many of the most sophisticated alarm systems incorporate this feature. While it is not beyond the capability of the more than modest home computer, it would be expensive in this application.

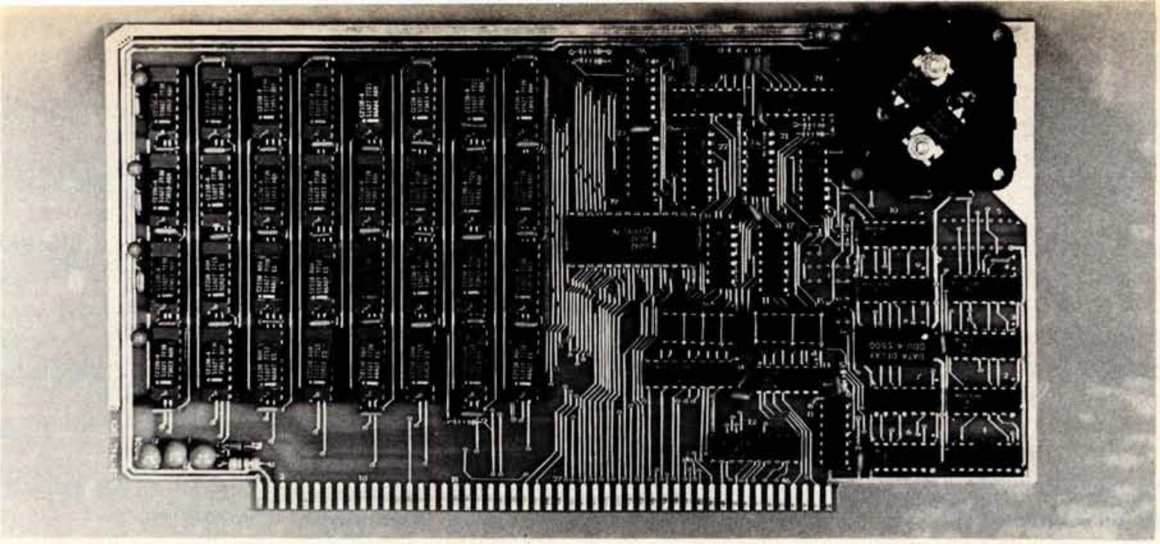
If You're Trying to Detect Motion — Move the Detector

Detecting motion with a light level sensor requires that a quantity of them be placed throughout the detection area. As the source moves, the relative light levels reaching all the sensors can be plugged into an equation and the location of the source computed. Tracking an object is simply a case of repeating this snapshot technique a number of times. Unfortunately, the concept is about the only part that's simple.

An alternative approach is to point a light sensor at a source and then move the sensor to a new location. If eventually the source is again detected in this new position the source must have motion! This of course presumes that there aren't so many sources that placement of the sensor inadvertently coincides with a stationary source. Small but discrete steps of sensor displacement will increase the resolution of this method.

Build a Light Sensitive Scanning System

I wish to back off a bit at this point and explain that this design is not merely a motion detection system. That is one of its numerous applications and, as previously stated, it is the idea that prompted its development. The design is a simple, yet effective,



We Start With A Price That's Hard To Beat.

But We Don't Stop There.

Central Data Corporation has combined the benefits of new technology and high-volume company sales to bring you a RAM board with more features and product options for less money.

Lower Prices
More Memory Capability

To begin with, we've reduced the price of our 16K RAM board by $40 to $249. At $425—a price reduction of $50—our 32K board costs less, too. Plus, we now offer a full 48K memory board for $599. These boards are expandable to 64K at a price of $185 per 16K package. Or you can start right out with a full 64K board for $775.

Improved Board Design

We've also added improvements to the board design at no extra cost to you.

- **Deselectable in 2K increments.** Our deselect feature enables you to switch off any 2K to avoid overlap with your existing memory.
- **Fully socketed memory.** This feature enables you to expand the memory board yourself.
- **Plug selectable addressing.** Now you can re-address without soldering

Other Standard Features

- Power-saving dynamic board with on-board invisible refresh
- One-year guarantee on parts and labor
- S-100 and Z-80 compatible

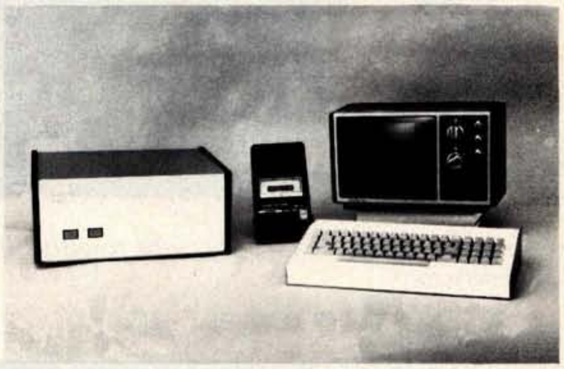
Specifications

Storage Capacity	16K, 32K, 48K, or 64K
Addressing	16K boundaries
Max. Input Load	One LS TTL load
Output Buffering	On all data lines
Access Time	450ns
Cycle Time	480ns
Wait States Generated	None
Maximum DMA Rate	1 Mhz

Power Consumption

+16	150ma
+ 8	300ma
−16	20ma

Other Products From Central Data



Central Data also offers, fully-assembled and in kits, a range of other computer products including (pictured above) our 2650 microprocessor, software packages, TV/Monitor, ASCII keyboard, and (not pictured) floppy disk system, to name just a few.

For More Information
To Place an Order

We welcome any questions you have about our RAM boards or other products. To place an order, or for more information contact:

Central Data Corporation
P.O. Box 2484, Station A
Champaign, IL 61820
Ph. (217) 359-8010

Place orders prepaid or COD. Delivery is stock to 30 days with shipping and handling prepaid in Continental United States. Please include phone as well as name and address.

Circle 45 on inquiry card.

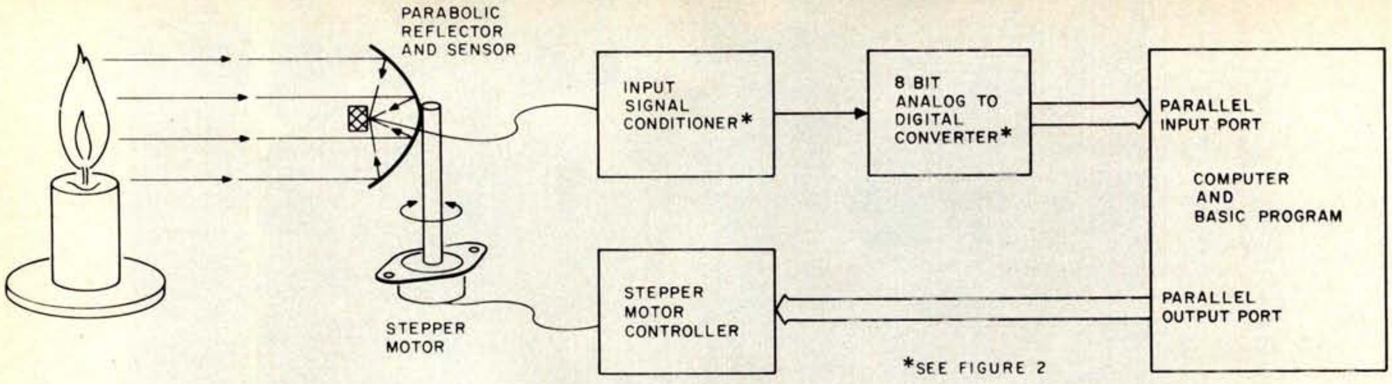


Figure 1: Block diagram of closed loop optical scanning system. The three main sections are the optics, consisting of a parabolic reflector and visible and infrared detector, the signal conditioner and input interface, and the stepper motor.

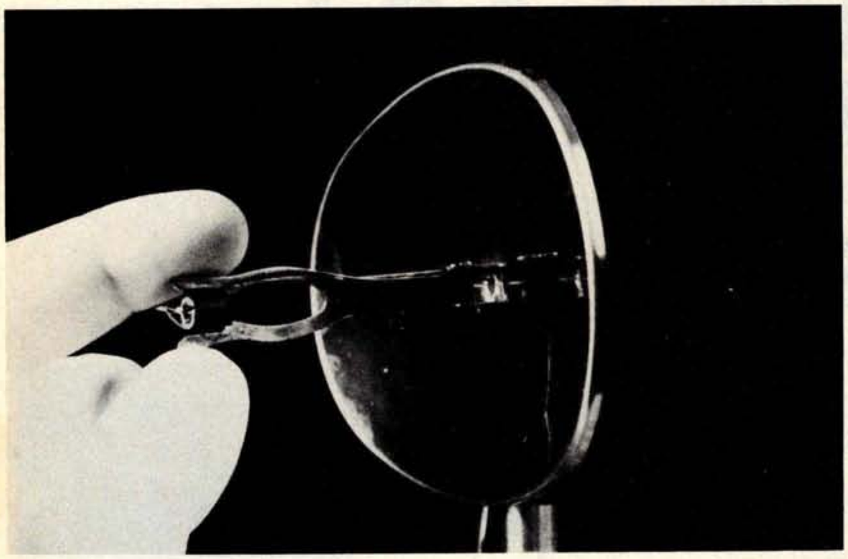


Photo 2: The parabolic reflector, used to gather light for detection by the photo detector.

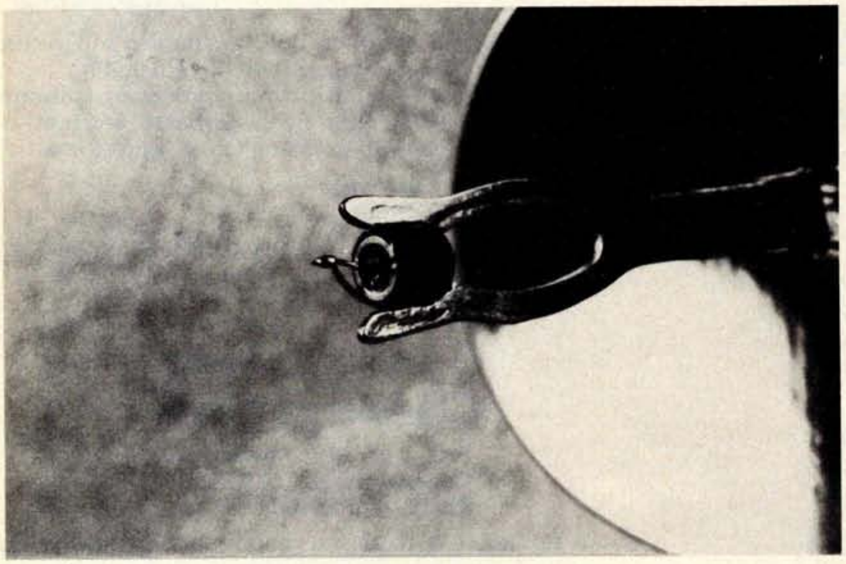


Photo 3: Closeup of the photo detector glued to the pronged holder.

light sensitive scanning system. A sort of passive radar (radio detecting and ranging system) if you will. It incorporates a sensitive visible and infrared light detector that is highly directional. In addition, it has the ability to accurately position itself on a rotational axis and sweep a wide area, much like a radar.

Figure 1 is a block diagram of the device. The scanning system consists of three prime components: optics, including sensor and reflector; signal conditioner and input interface; and finally, for closed loop control, a rotational positioning mechanism consisting of a 7.5° resolution stepper motor. The completed unit allows the computer to position its sensor in a known direction, read an analog value of the light level in that direction, and move to another point or track a moving source (more on this later).

The prime consideration in any light detection system is the optics. To take full advantage of any positioning mechanics, the light sensor must be highly directional. This is usually done with a series of lenses, the whole affair resembling a telescope. This technique is quite expensive and heavy. Instead of lenses, a highly polished parabolic reflector can be used to concentrate the light. One such device ideal for this application is an inexpensive parabolic mirror sold by Radio Shack for under $2 as a solar cigarette lighter. The unit, shown in photo 1, has a fork tipped hinged prong which extends from the center to hold the cigarette. Already designed to be at the focal point of the mirror, it serves as the perfect mounting bracket for the photo sensor. A GE L14F2 infrared photo Darlington inserted into a phenolic sleeve is glued to the cigarette holder at the focal point as in photos 2 and 3. The lens of the photo sensor should face the reflector.

While the photo detector is infrared by design, it is highly sensitive to visible light as well. By choosing the infrared unit, a de-

Jim-pak® electronic components KITS

Regulated Power Supply

- ★ Uses LM 309K
- ★ Heat sink provided
- ★ P.C. board construction
- ★ Provides a solid 1 amp @ 5V
- ★ Includes components, hardware and instructions
- ★ Size: 3-1/2" x 5" x 2" high

JE200 5v 1amp $14.95

Function Generator Kit

- ★ Provides 3 basic waveforms: sine, triangle & square wave
- ★ Frequency range from 1 Hz to 100K Hz
- ★ Output amplitude from 0-volts to over 6 volts (peak to peak)
- ★ Uses a 12V supply or a ± 6V split supply
- ★ Includes chip, P.C. board, components and instructions.

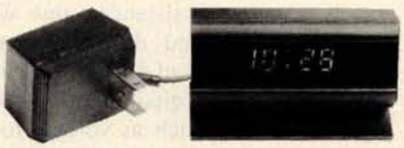
JE2206B $19.95

Digital Stopwatch Kit

- ★ Use Intersil 7205 Chip
- ★ Plated thru double-sided P.C. Board
- ★ LED display (red)
- ★ Times to 59 min. 59.59 sec. with auto reset
- ★ Quartz crystal controlled
- ★ Three stopwatches in one: single event, split (cummulative) and taylor (sequential timing)
- ★ Uses 3 penlite batteries
- ★ Size: 4.5" x 2.15" x .90"

JE900 $39.95

4-Digit Clock Kit



- ★ Bright .357" ht. red display
- ★ Sequential flashing colon
- ★ 12 or 24 hour operation
- ★ Extruded aluminum case (black)
- ★ Pressure switches for hours, minutes and hold functions
- ★ Includes all components, case and wall transformer
- ★ Size: 3-1/4" x 1-3/4" x 1-1/4"

JE730 $14.95

6-Digit Clock Kit

- ★ Bright .300 ht. common cathode display
- ★ Uses MM5314 clock chip
- ★ Switches for hours, minutes and hold functions
- ★ Hours easily viewable to 20 feet
- ★ Simulated walnut case
- ★ 115 VAC operation
- ★ 12 or 24 hour operation
- ★ Includes all components, case and wall transformer
- ★ Size: 6-3/4" x 3-1/8" x 1-3/4"

JE701 $19.95

Jumbo 6-Digit Clock Kit

- ★ Four .630" ht. and two .300" ht. common anode displays
- ★ Uses MM5314 clock chip
- ★ Switches for hours, minutes and hold functions
- ★ Hours easily viewable to 30 feet
- ★ Simulated walnut case
- ★ 115 VAC operation
- ★ 12 or 24 hour operation
- ★ Includes all components, case and wall transformer
- ★ Size: 6-3/4" x 3-1/8" x 1-3/4"

JE747 $29.95

AVAILABLE AT YOUR LOCAL Jim-pak electronic components DEALER TODAY !!

For Dealer Information, write or phone JIM-PAK,® 1021 Howard Avenue, San Carlos, California 94070 Ph. (415) 592-8097

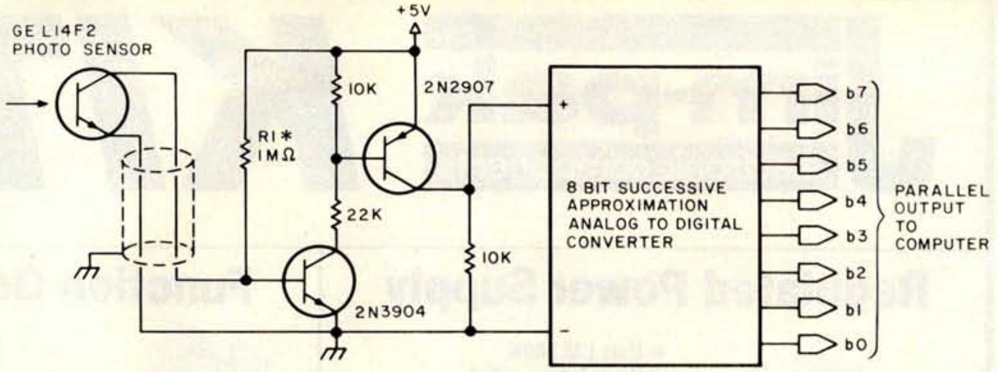


Figure 2: Optical signal conditioner and analog to digital converter that convert light input into a proportional digital output. The analog to digital converter shown was described in "Talk to Me: Add a Voice to Your Computer for $35," by Steve Ciarcia, June 1978 BYTE, page 142.

***Raising the value of this resistor will increase sensitivity. Lowering it will reduce sensitivity (range of resistor values is 4.7 k to 3.3 MΩ).**

tection system can be designed that utilizes the best of both spectra.

The output of the photo sensor is essentially a current proportional to the light hitting it. The signal conditioner section of figure 2 converts this to an analog voltage level. The sensitivity of the photo detector is governed by resistor R1; changing this resistance value will affect both sensitivity and dynamic range. For the computer to read this voltage it must be converted to a digital quantity. While in theory any method, such as voltage to frequency, or voltage to pulse width, etc, could have been used, I'm a purist. The output of the signal conditioner is fed to an 8 bit successive approximation analog to digital converter. The details of this design were outlined in the June 1978 Ciarcia's Circuit Cellar in an article entitled "Talk to Me!" (page 142). Two slight modifications were made to the circuit for this application. The sample rate was reduced by placing a 0.01 μF capacitor in parallel with the 150 pF component already between pins 1 and 6 of IC1, and the offset potentiometer was readjusted to allow full scale unipolar operation (ie: 0 V input would give hexadecimal 00 output and +5 V input would give hexadecimal FF output).

The parallel output of the analog to digital converter is attached to an 8 bit parallel input port. Either an assembly language or a BASIC program can be used to read and display this quantity by querying the input port (input port 16 in my examples).

Exercising the device with a BASIC program is relatively straightforward. Listing 1 is a program written in Micro Com 8 K Zapple BASIC which reads the light level from the analog to digital converter and converts it to a proportional pulse width on output port 16. If a Sonalert or the circuit of figure 3 is attached to the least significant bit of port 16, it will beep. The beep rate will change as the reflector is pointed toward various light intensities. Printing out the analog to digital conversion value will give an accurate account of the sensitivity and dynamic range.

Add a Stepper Motor for Positioning

Now that we have an effective light sensor, we must add rotational mechanics to provide *sweep*. The simplest method for rotating this relatively lightweight reflector is to mount it directly on the shaft of a stepper motor.

An inexpensive stepper motor is available

```
100 REM INFRARED SENSOR TEST PROGRAM
110 REM
120 REM
130 REM THIS PROGRAM CAUSES A SOUND SOURCE ATTACHED
140 REM TO LSB OF PORT 16 TO -BEEP- WITH   A PERIOD PROPORTIONAL
150 REM TO THE AMOUNT OF LIGHT SEEN BY THE LIGHT SENSOR
160 REM REV. 1.1    S.CIARCIA
170 REM
180 OUT 16,0
190 X=INP(16)
200 IF X<230 THEN GOSUB 220
210 GOTO 190
220 OUT 16,255
230 FOR T=0 TO X+5
240 NEXT T
250 OUT 16,0
260 RETURN
```

Listing 1: Program written in Micro Com 8 K Zapple BASIC that reads the light level from the analog to digital converter and converts it to a proportional pulse width on output port 16 (in my particular system configuration).

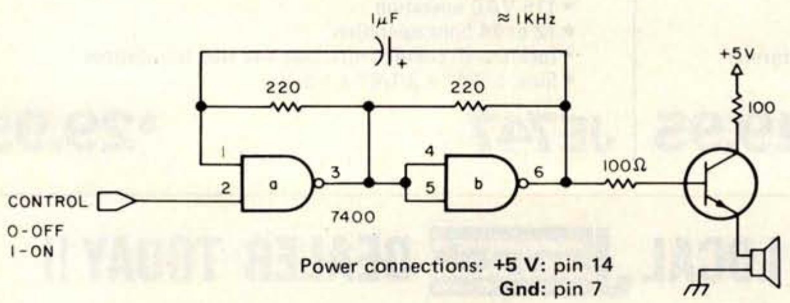


Figure 3: Sound source circuit (use with optical scanner test program above).

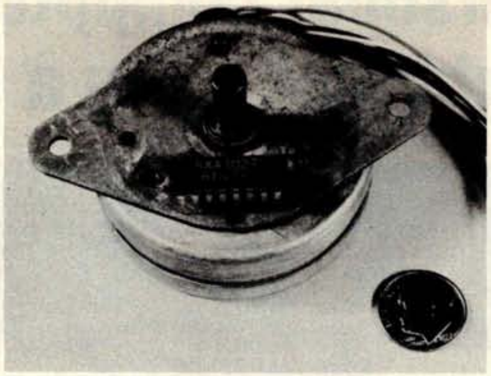


Photo 4: North American Philips stepper motor, Model K82944-M1, and the SAA1027 controller circuit used to drive it. Cost for the two pieces is less than $20.

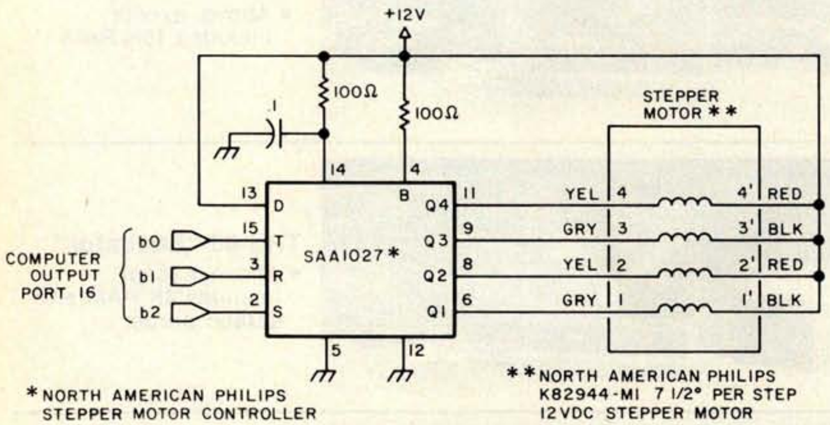


Figure 4: Connecting the stepper motor to the integrated circuit controller.

```
100 REM THIS PROGRAM DRIVES THE STEPPER MOTOR IN A BACK AND FORTH MOTION
110 OUT 16,1 :OUT 16,255 :REM PRESET STEPPER CONTROLLER
120 REM
130 REM
140 REM GO 25 STEPS CLOCKWISE
150 FOR D=0 TO 24
160 REM BIT 2 IS SET HIGH AND BIT 0 IS TOGGLED TO GO CLOCKWISE
170 OUT 16,5
180 GOSUB 390
190 OUT 16,4
200 NEXT D
210 REM
220 REM RETURN SCAN DELAY
230 FOR S1=0 TO 10
240 GOSUB 390
250 NEXT S1
260 OUT 16,1 :OUT 16,255
270 REM GO 25 STEPS COUNTERCLOCKWISE
280 FOR D=0 TO 24
290 REM BITS 1 AND 2 ARE HELD HIGH AND BIT 0 IS TOGGLED TO
300 REM GO COUNTERCLOCKWISE
310 OUT 16,7
320 GOSUB 390
330 OUT 16,6
340 NEXT D
350 GOTO 110
360 REM
370 REM IN BETWEEN STEP DELAY TIMER
380 REM DELAY TIME SET BY VALUE OF T1
390 FOR T=0 TO 5
400 NEXT T
410 RETURN
```

Listing 2: A BASIC program that drives the stepper motor and demonstrates the sweep action. It initializes the stepper motor, drives 25 steps clockwise, waits a short period, and then returns to its initial position.

from North American Philips. This unit (shown in photo 4) is relatively small, and a single integrated circuit controller is all that is needed to interface it to a computer. The particular unit in this article is a 12 VDC 7.5° stepper motor. This means that there are 48 steps per revolution, and, if one were trying to scan a 180° field of view, the stepper should oscillate between 24 clockwise and counterclockwise steps. This would give the impression of "sweep."

The electronics of the stepper are outlined in figure 4. Three bits of a parallel output port are necessary to control the direction and speed of the motor. The three signals are S, R and T, for set, rotation and trigger. When first engaging the motor it should be set to a known condition by pulsing the set input low while keeping the trigger input high. Once initialized, the direction of rotation must be chosen. This is done by setting the R input low if clockwise rotation is desired and high for counterclockwise rotation. An actual step is initiated by simultaneously making a 0 to 1 logic transition on trigger input T. By repeatedly toggling this bit, continuous motion will result. The stepper motor in this article is capable of 200 steps per second.

A BASIC program which drives the stepper motor and demonstrates the sweep action is outlined in listing 2. It initializes the stepper, drives 25 steps clockwise, waits a short period, and then returns to its initial position.

Making a Scanning System

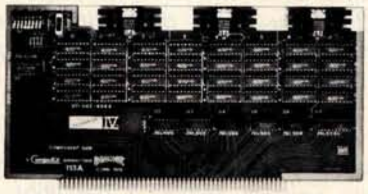
To produce a closed loop controlled scanning system, the reflector and photo sensor are attached to the stepper motor shaft by any convenient means. I glued the reflector to a sleeve which attached to the shaft of the motor. The concept of closed loop control comes from the ability of this unit to position itself, take a light reading, and perform some further action as a result. This could be to step to a new location or to stop and remain stationary on any source above a certain light level.

Listing 3 is the BASIC program of such an exerciser which seeks out and points at a light source. As the parabolic reflector steps through its sweep, it checks the reading of the analog to digital converter and compares it to a set point. If the set point is exceeded, the program will stop stepping and point at this source. Should the light be extinguished or obstructed, the sweep resumes until it finds another source of sufficient intensity.

This is a rather rudimentary program but it incorporates all the basic structure to which enhancements such as motion detec-

$279 = 16K

of **static** S-100 memory



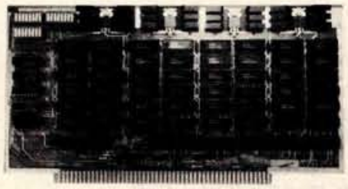
Our Econoram IV™ "unkit" comes with sockets and bypass caps pre-soldered into place for easy assembly. Plugs into ***any*** S-100 buss machine. Features under 2000 mA current, complete buffering, fully static operation, and our enviable reputation for reliability under all operating conditions.

Assembled and tested, **$314**; qualified under the CSC program*, **$414**.

$445 = 24K

of **static** S-100 memory

Econoram VII™ is an "unkit" that has sockets and bypass caps pre-soldered into place. This full feature memory is configured as two 4K and two 8K blocks for maximum flexibility, boasts current consumption under 2000 mA, and follows our design policy of fully static operation with complete buffering.

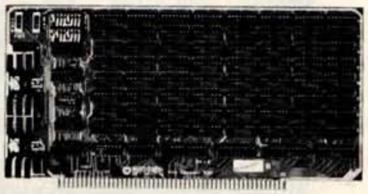


Assembled and tested, **$485**; qualified under the CSC program*, **$605**.

*(Certified Systems Components boards are assembled, tested, serial numbered, burned in for 200 hours, and guaranteed to run at 4 MHz over the full temp range. *We exchange board if failure occurs within 1 year of invoice date.*)

$135 = 8K

of **static** S-100 memory

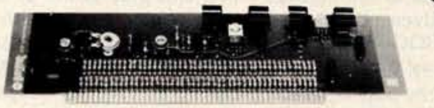


Econoram II™ is our biggest selling kit to date, thanks to the cost-effectiveness it brings to the S-100 buss. Features low current consumption, high speed, full buffering, and the same performance that has made Econoram one of the best selling peripherals in the business.

Assembled: **$155**. Quantity kit price: **3/$375**.

ALL OF THE ABOVE ARE AVAILABLE FROM STOCK. Coming soon: 32K boards for the S-100 buss, the Intel/National 80/10 and 80/20, and the Digital Group . . . plus a dual UART S-100 I/O board and our new family of S-100 Econorams. Watch this space, or inquire about price and availability.

ACTIVE TERMINATOR



$29.50 (kit)

Active termination draws much less power than passive termination, and minimizes the glitches, noise problems, crosstalk, ringing, and other gremlins that can occur with unterminated busses. Plugs into any S-100 computer. Many have copied our technique; few can match our low price.

ATTENTION—

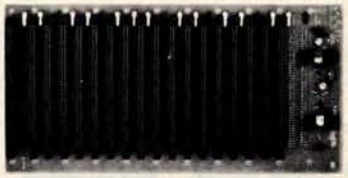
RADIO SHACK TRS-80 AND HEATH H8 OWNERS:

MEMORY PRICES ARE COMING DOWN!!

Our TRS-80 Conversion Kit chip set is now only $159 (3 kits for $450). It not only upgrades your mainframe from 4K to 16K, but can also populate the memory expansion module with extra memory. Works with APPLE computers too. We include full instructions for the TRS-80 conversion, and back up our parts with a **1 year warranty.**

Heath H8 owners are recognizing that our **12K Econoram VI™**, designed specifically for the H8 buss, offers a superior memory value at **$235. Now you can buy two of our 12K boards on special for $399** (price good through cover month of this magazine). These are "unkits" with sockets and bypass caps pre-soldered in place. Need we say more?

MOTHERBOARDS



11 slots, with active termination circuitry, heavy power/ground traces, and all edge connectors: **$90** in "unkit" form (edge connectors are pre-soldered in place, making assembly a cinch). Need more slots? Our 18 slot motherboard "unkit" includes all the above along with 18 edge connectors for **$124**.

MULLEN RELAY/ OPTO-ISOLATOR CONTROL BOARD $88 (kit)

8 reed relays respond to an 8 bit word (relays can also drive Mullen 500 Watt AC control modules for power applications); 8 opto-isolators receive outside world signals for handshaking or further control purposes. Applications include audio, time lapse photography, robots; one user even devised an automatic cat feeder. At our new low price, you can afford to let your imagination run wild.

MULLEN EXTENDER BOARD $35 (kit)

From Mullen Computer Products. We now carry the improved version that includes a built-in "kluge board" area in addition to the integral logic probe and ingenious edge connector. Greatly simplifies servicing and testing . . . when it comes to extender boards, it would be hard to improve on this unit in terms of either price or ease of use.

CompuKit™

TERMS: Cal res. add tax. Add 5% shipping, excess refunded. COD orders OK with street address for UPS. Visa® /Mastercharge® orders call our 24-hr. order desk at **(415) 562-0636**. Thank you for your business.

GODBOUT

BILL GODBOUT ELECTRONICS
BOX 2355, OAKLAND AIRPORT, CA 94614

```
100 REM THIS PROGRAM SIMULATES A CLOSED LOOP -RADAR-
110 REM IT SCANS BACK AND FORTH 25 STEPS IN EACH DIRECTION
120 REM LOOKING FOR A PRESET LIGHT LEVEL EITHER PRODUCED BY
130 REM OR REFLECTED FROM SOME OBJECT IN ITS SCAN PATH
140 REM IT WILL STOP SCANNING AND REMAING POINTING AT ANY SUCH
150 REM OBJECT IT FINDS. IF THE OBJECT MOVES, THE SCANNER WILL FOLLOW
160 REM
170 REM
180 OUT 16,1 :OUT 16,255 :REM PRESET STEPPER CONTROLLER
190 FOR D=0 TO 24
200 OUT 16,5
210 REM TAKE ONE CLOCKWISE STEP
220 GOSUB 550
230 OUT 16,4
240 GOSUB 480 :REM READ SENSOR
250 NEXT D
260 REM
270 REM RETURN SCAN DELAY
280 FOR S1=0 TO 10
290 GOSUB 550
300 NEXT S1
310 REM
320 REM
330 OUT 16,1 :OUT 16,255 :REM PRESET STEPPER CONTROLLER
340 FOR D=0 TO 24
350 OUT 16,7
360 REM TAKE ONE COUNTERCLOCKWISE STEP
370 GOSUB 550
380 OUT 16,6
390 GOSUB 480 :REM READ SENSOR
400 NEXT D
410 FOR D=0 TO 25
420 NEXT D
430 GOTO 180
440 REM -RADAR- SENSOR READ ROUTINE
450 REM A/D INPUT IS ATTACHED TO INPUT PORT 16
460 REM LOW LIGHT LEVEL IS A VALUE OF 255 AND HIGH INTENSITY
470 REM IS AN INPUT VALUE OF 0
480 X=INP(16) :REM READ A/D CONVERTER
490 L=10 :REM PRESET LEVEL SET ....THIS WOULD BE A BRIGHT LIGHT
500 IF X<L THEN GOTO 480
510 RETURN
520 REM
530 REM
540 REM DELAY TIMER TO COMPLETE MECHANICAL MOTION BEFORE READING SENSOR
550 FOR T=0 TO 25
560 NEXT T
570 RETURN
```

Listing 3: A BASIC program that causes the scanner system to seek out and point to a light source. The scanner tracks the light source as it moves. If the light source is extinguished or obstructed, the sweep resumes until another source of sufficient intensity is found.

tion and tracking can be added. It will, as now written, follow a flashlight as someone walks across a room. It is left as an exercise for the reader to drop a net over the perpetrator.

There are a few other little things you can try after you've built this gadget. The sketch on page 80 shows the portion of my basement (the "Circuit Cellar") immediately adjacent to the computer system. After modifying the BASIC program of listing 3 to print out a number on a scale of 1 to 9 (a period is 0) indicating relative intensity, and turning on a light and lighting a candle, I initiated a single scan across the room. Listing 4 is a printout of that scan. The sensitivity of the device had to be set very high to pick up the candle, and the result was rather interesting. The scan allowed the computer to "see" around the room in front of it. There is an intense light source to the right and a rather low level one to the left. By incorporating gain selection (changing the 1 MΩ resistor in the signal conditioner) under program control the computer could reduce the gain selectively to determine the origin of each light.

One further experiment entailed taking numerous sweeps and combining them to form a digitized computer picture. First the program was changed back to a threshold detector again. As it scanned the 25 steps it would print out an asterisk (*) for anything that exceeded this threshold and a period (.) otherwise. A protractor was attached to the arm of the tripod so that the angle of the reflector could be adjusted by a known increment each time the computer stopped between scans. The result was as illustrated in listing 5. The ten scans form a computer's eye picture of the wall. Again, because of the dynamic range differences between the candle and the light, the incandescent bulb appears much larger than it actually is.

Conclusion

Here's a simple device that can detect and track infrared and visible light sources. See what you can do with it. I don't want to leave anyone with the impression that I'm waiting for a burglar with a million candlepower flashlight to come tripping through the woods. This is but one sensor in a larger system, and the infrared capabilities, which I neglected to discuss in detail, are its primary application. Starting in January, I'll explain computer alarm systems.

There have been numerous articles on light seeking robots. With this detector it is quite possible that the mechanics and software could be reduced considerably. I've often thought about building a robot, but my mechanical talent is nonexistent. When I can build one with a screwdriver and a soldering iron only, I'll write about it. (My thanks to Lloyd Kishinsky for graphics ideas used in this article.)

If you have any comments or ideas, please don't hesitate to write and include a stamped, self-addressed envelope. Much to my surprise, I've actually been able to keep up with correspondence. Some people have written to ask whether I sit up nights thinking of crazy introductions for my articles. You may not believe this, but everything has some basis in fact. I've met some really strange characters. Why create fiction when the truth is often so much more hilarious!

Next month: a hexadecimal-octal display.■

```
. . . . . . . . . . 1 . . . 1 6 9 7 3 1 . . . .
```

Listing 4: A single scan of the room containing the light bulb and candle. A modification of the program in listing 3 to print numbers on a scale of 1 to 9 (a period is 0) indicating relative intensity.

```
. . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . * . . . . .
. . . . . . . . . . . . . . . . . * * . . . . .
. . . . . . . . . . . . . . . . . * * * . . . .
. . . . . . . . . . . . . . . . * * * . . . . .
. . . . . . . . . . * . . . . . * . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . .
```

Listing 5: Ten sweeps of the room. The relatively large size of the light used in the experiment accounts for the large number of asterisks at the right.

Computer Assisted Instruction on a Microcomputer

Melvin Davidson
Director, Computer Center
George A Gerhold
Professor of Chemistry
Larry Kheriaty
Systems Programmer
Western Washington University
Bellingham WA 98225

Computer assisted instruction (CAI) could alter the delivery of education profoundly. This has been known for years, but not much has happened in response, partly because of expensive hardware, and partly because of lack of quality courseware. Although minicomputer based CAI systems need not be expensive on a per terminal hour basis, the initial capital investment for a small number of terminals is too high for most school systems. (Our campus system costs 30 cents per hour per terminal amortized over 24 ports for five years at 200 hours per month exclusive of terminals.) Remote terminals are not attractive because of prohibitive line costs. Thus the field is open for microprocessor based systems. They are inexpensive to operate; the initial investment is reasonable; there are no line costs; and individual systems can be tailored for particular applications. The difficulty with microprocessor based systems lies in the second area, the lack of quality courseware.

Production of quality courseware is a demanding task which requires experienced teachers as authors, extensive testing with a variety of students, and repeated editing on the basis of this testing. Potential authors are usually not computer aficionados, so programming in appropriate high level languages must be possible. Testing requires elaborate record keeping and file sorting capabilities, and the collection of results from multiple terminals. Therefore, the system requirements for the development of quality courseware imply use of a multiterminal, large memory system (one beyond the capabilities of present microprocessors). Therefore we decided to develop a microprocessor based system which could be used to deliver CAI courseware which was written, tested and edited on our campus terminal system. In this way we can combine low delivery cost with quality courseware.

Our CAI machine consists of a microprocessor, a floppy disk and a video terminal. Our developmental system uses a Southwest Technical Products 6800 computer, a Midwest Scientific Instruments FD-8 floppy disk system or a PerCom minifloppy disk system and an Ann Arbor Design III video terminal. We have completed a PASCAL version, are working on a Z-80 and multiuser 6800 version, and are contemplating a 6502 version. Also, since December 1977, we have been running test programs on the SwTPC 6800 system.

We have experience with a variety of CAI languages including IBM Coursewriter, a variation of Coursewriter called CW3-WPL, and more recently with PILOT. Our feeling is that a PILOT/BASIC composite is currently the optimal language for CAI. The PILOT language is easy to learn, flexible, and comparatively easy to implement on any computer system. A PILOT/BASIC semicompiler forms the basis of our Western Terminal System (WTS) campus timesharing service running on Interdata 32 bit minicomputers with Telefile 80 megabyte disk units.

Our initial microcomputer hardware and software design objectives are summarized in table 1. We set the first three objectives to minimize the level of expertise required both to assemble and to use the system, because this intended to be a production system, not just a hobby system. Of course, if an instructor can assemble hardware from kits, that will be an extra saving. We set the fourth through sixth objectives to allow delivery of courses of the size and sophistication of the CAI courses currently in use at Western. The price is the lowest possible consistent with criteria of quality hardware and courseware.

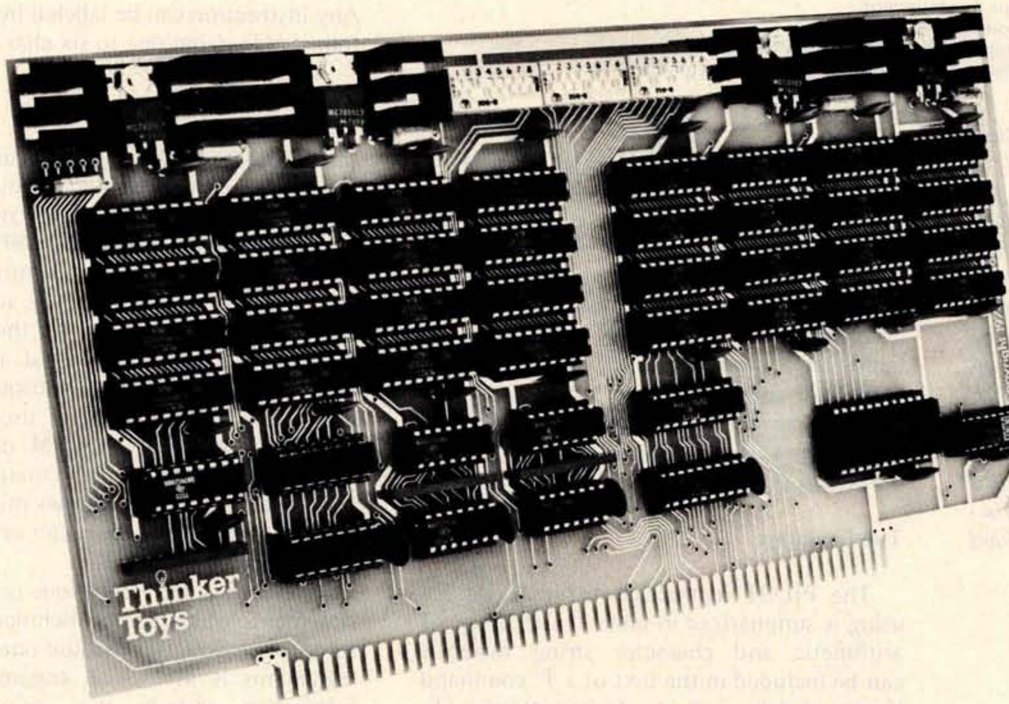
We eventually relaxed our price restriction slightly and selected a Southwest

Table 1: Original design objectives for the CAI system.

- Turnkey system with no front panel switches other than a power switch.
- Central operating system and bootstrap loader module should be in read only memory for cold starts.
- Hardware components should be available off-the-shelf, assembled.
- System should be disk oriented with only the immediately required lines of the course loaded into main memory at any one time.
- Video terminal with 24 lines containing 80 characters, the full ASCII upper and lower case character set, and cursor controls.
- Implement as large a subset of PILOT and BASIC as possible.
- Total cost of hardware exclusive of video terminal should be less than $2500.

SuperRam™

16K STATIC FOR $299



Introducing SuperRam™ 16K static memory, the one that's leaping tall price barriers at a single bound. It saves you about $100 on the usual cost of a big 16K memory for your S-100 system.

SuperRam™ 16K is the latest in cost-efficient memory designs by George Morrow, designer of the best-selling ECONORAM* memories.

SuperRam™ 16K is configured as four independent 4K blocks, each separately addressable and write-protectable. Designed to meet the proposed IEEE Standard for the S-100 bus (see IEEE Computer, 5/78), all signals are fully buffered—including address and data lines. And Morrow's design uses just 11 chips to keep the board uncrowded and trouble-free.

SuperRam™ 16K comes as an easily assembled kit, with solder mask and parts legend.

Ask for the SuperRam™ 16K memory kit at your local computer shop. Or if unavailable locally, call your BankAmericard/Visa or Master Charge order to 415-524-5317, 10-4 Pacific Time. Or send check or money order to Thinker Toys™, 1201 10th St., Berkeley, CA 94710. Add $3 for handling; Cal. res. add tax.

*ECONORAM is a trademark of Godbout Electronics

A product of Morrow's Micro-Stuff for

Thinker Toys™

1201 10th Street
Berkeley, CA 94710

Circle 255 on inquiry card.

Four Instructions that Comprise the Central Core of PILOT:

T:	Type the following text.
A:	Accept an edited answer and assign it as the value of the following string or number variable.
M:	Match the answer with the following one or more acceptable patterns.
J:	Jump to the designated instruction.

Additional PILOT Instructions:

R:	Remark or comment.
U:	Use a subroutine at the designated label.
E:	End the subroutine and return to the place of call.
C:	Calculate using the following BASIC statement.

Local Additions to PILOT:

:	Continue the previous T: statement.
PR:	A new problem or course frame.
XI:	Execute the following character string as a PILOT statement.
SJ:	Jump to a new segment, preserving symbols but clearing labels.

Table 2: The instruction set that is being used in this implementation of PILOT can be divided into three basic parts. There are four instructions that can be considered as the basic core of the language. The second part contains instructions that are not essential to the operation of the language but make it easier to use. The third section contains instructions that we have added for our own use.

Technical Products SwTPC 6800 interfaced to a Midwest Scientific Instruments FD-8 floppy disk system. The 24 line by 80 character requirement rules out home television terminal kits at present, though we are confident that this will change now that both Vamp and Pickles & Trout have home television modification kits available. Interested users should be able to acquire all the hardware, including the video system, for under $3000 off-the-shelf by the time our initial development project is complete.

The Language

The PILOT instruction set that we are using is summarized in table 2. The values of arithmetic and character string variables can be included in the text of a **T**: command if preceded by a # or a $, respectively, (T: #X3, T: $B). This feature can be used, for example, to send special characters to a terminal for graphics effects. One line of input is accepted by an A: instruction, multiple spaces are compressed to single spaces, automatic upper and lower case editing is performed as specified, and the response optionally may be saved in a string or numerical variable. The last response accepted can be compared with a specified pattern string which may include the special characters * to match one character, % to signify space or end, & to match multiple characters, and ! to match one string or another. The result of the match is an internally kept yes or no.

Any command can be made conditional by adding the labels in table 3 between the command code and the colon following it. In addition, execution of any instruction is made conditional by enclosing a BASIC logical expression, called a relational conditioner, in parentheses between the command and its following colon. The command will be executed only if the comparison is true:

T(X < 3):

Any instruction can be labeled by an asterisk followed by from one to six characters:

*LAB1 T:

The destination of the **J**: command is a label (**J**: label); a line number (**J**: number); the last accept (**J**: **@A**); the next match (**J**: **@M**); or the next problem frame (**J**: **@P**).

There are three statement modifiers **H**, **J** and **S**. An **H** modifier can be added to a **T** command (**TH**:) to suppress the automatic new line sequence (line feed and carriage return) at the end of an output line. The match jump code **MJ**: is used to automatically jump to the next **M**: instruction if no match is found, and the match code **MS**: followed by a pattern allows minor spelling errors of one wrong character or an inverted pair of characters.

There are some limitations on the BASIC statements which can be included directly in a **C**: instruction, but because one of the legal statements is the CALL statement, the **C**: instruction imbeds the entire extended BASIC language into PILOT. However, only five BASIC instructions are essential for execution of programs (**INPUT, PRINT, DIM, LET, IF . . . THEN**). The other BASIC commands make programming easier, but are not essential. We have chosen to make a major saving in the microprocessor memory and system complexity by limiting the BASIC to those five instructions. Extended BASIC programs will be written and tested on the campus terminal system and then machine translated into the five essential instructions. These five will be translated into PILOT as follows:

BASIC	*MicroPILOT*
INPUT	A:
PRINT	T:
IF . . . THEN	(relational conditioner)
LET	C: LET
DIM	C: DIM

Thus the only two BASIC instructions which

Label	**Conditions for Execution**
Y	The last match was successful.
N	The last match was not successful.
E	An error flag was turned on.
C	The last relational conditioner was true.
0 - 9	The last **A**: instruction has been repetitively executed the designated number of times.

Table 3: Conditional labels that can be added to statements to affect the order of execution.

will be legal when included directly after the C: will be

LET variable = expression and
DIM

In our first version of the system we have included floating point scalers, arrays of one and two dimensions, and varying length strings. The usual BASIC arithmetic and logical operators +,−,*,/,EQ,NE,LT,GT,LE, GE,AND, OR and NOT are provided as well as the built-in functions **ABS,INT,RND CHR,ASC,FIX** and **SGN**. In addition we support the string operations of ! ! (concatenation), **MID** (middle), **LEN** (length), **XLA** (translate) and **INS** (instring). Subroutine calls in BASIC will not be supported in the microcomputer version of the software, but the equivalent can be obtained by copying the subroutines in the appropriate places during machine translation.

In a later version we will add the exponentiation ** operator, a complete range of built-in scientific functions, string arrays and file handling with:

FI: disk address, string variable
FO: disk address, string variable

for moving the value of a string variable in from or out to a file at the given address. The segment jump will be replaced with an instruction to clear the label table, and a disk jump instruction will be included for multiple diskette courses. Programs begin with a header containing the optional automatic features in table 4. Additional headers can be imbedded within the course.

Table 4: Optional automatic features that may be added at the head of the program.

Label	Function
U	Translate all input to upper case.
L	Translate all input to lower case.
S	Remove all spaces from input.
E	Enable automatic use on any input beginning with @.
G	Enable the GOTO command.
W	Erase the label table.

Conclusion

In summary the microprocessor based CAI system will execute all instructions included in PILOT on our multiterminal campus system. Most programs written in extended BASIC will execute on the system after machine translation. Exceptions requiring author reprogramming include matrix operations, **ON GOSUB, ON ERROR, OPEN, CLOSE, DEF, READ, RESTORE, DATA, CHAIN.** The early version of the system requires 9 K bytes of memory, including a floating point arithmetic package from Technical Systems Consultants. In addition we must provide workspace. We expect to fit the early version of the complete system into 16 K bytes of memory and the later version should fit into a 20 K byte system.

Obviously it is possible to write courseware in whatever language is available and to use that courseware without extensive testing. But untested material in an inappropriate language tends to be second rate. The object of our project has been to provide access to high quality, tested, existing courseware at low cost to those who would otherwise have to pay telephone and computer costs at a network center. For that reason we have not provided for convenient course authoring or editing on the microcomputer system.

Because an effort to establish a national standard for PILOT is underway, we suggest that programmers undertaking implementations of PILOT contact the authors for the most recent documentations.■

Introducing the first user-oriented software.

Your business system is worthless until you learn to use the software. That used to take weeks.

Autoscribe™ and Bookkeeper™ software by MicroSource are different. Here, at last, are packages you can use the first day. Because they're created for you, the user.

Autoscribe is the new low-cost word processor software for North Star-based systems. A powerful business system that lets you control paper work. Less than $8000 for the complete system with hardware, software and documentation.

Bookkeeper is for accountants. For half the price of comparable packages, a firm can perform—and control—virtually all client write-up functions. On a North-Star system, under $7500.

Here's what happens before MicroSource introduces any software: We work with the author to turn *his* software into *your* software. Documentation is developed that continues the user philosophy.

Then, after field testing, we distribute to leading computer stores through MicroAge,™ the nation's largest independent microcomputer distributor.

How about support? We believe you're buying a piece of the future, a system that should become more valuable with time.

So updates are released regularly. Enhancements, enabling the software to utilize other hardware, are developed. And we maintain communications.

This isn't just an idea. It is now. Autoscribe and Bookkeeper are at your dealer now. Stop in and see for yourself. Once you see MicroSource software in action, you're convinced.

Dealer inquiries invited.

MICRO SOURCE™

1425 W. 12th Place • Tempe, Arizona 85281 • Phone: (602) 967-1421

Circle 226 on inquiry card.

Event Queue

In order to gain optimum coverage of your organization's computer conferences, seminars, workshops, courses, etc, notice should reach our office at least three months in advance of the date of the event. Entries should be sent to: Event Queue, BYTE Publications, 70 Main St, Peterborough NH 03458. Each month we publish the current contents of the queue for the month of the cover date and the two following calendar months. Thus a given event may appear as many as three times in this section if it is sent to us far enough in advance.

November 3-5, Third West Coast Computer Faire, Los Angeles Convention Center. This is a conference and exposition on personal computers for home, business and industry. For more details about this computer faire, write for a free copy of the Silicon Gulch Gazette. Contact Computer Faire, POB 1579, Palo Alto CA 94302, (415) 851-7075.

November 5-8, Computer Applications in Medical Care, Washington DC. This IEEE sponsored symposium on computer applications in medical care is designed to inform physicians and health care professionals about current and potential applications of computer technology to patient care; and to identify areas of future research and development that need to be addressed. Contact Abund O Wist, PhD, general chairman, Medical College of Virginia, (804) 770-4957.

November 6-8, Asilomar Conference on Circuits, Systems and Computers, Asilomar Hotel and Conference Grounds, Pacific Grove CA. This conference, sponsored by the IEEE Computer Society, will delve into areas such as circuit theory and design, communication and control systems, computer systems, computer aided design, etc. Contact Donald E Kirk, Electrical Engineering Dept, Naval Postgraduate School, Monterey CA 93940.

November 7-9, Mini/Micro Computer Conference and Exposition, Astrohall, Dallas TX. The conference portion of this show will feature approximately 28 sessions covering both application and design topics. The exposition will feature hundreds of product displays. Contact Mini/Micro Computer Conference and Exposition, 5528 E La Palma Av, Suite 1, Anaheim CA 92807, (714) 528-2400.

November 11-12, Colorado Computer Coral, Denver Merchandise Mart, Denver CO. Contact Denver Amateur Computer Society, 12805 W Stanford Av, Morrison CO 80465.

November 13-15, Data Processing Operations Management, Chicago IL. This three day seminar will emphasize the management skill and techniques applicable to the data processing operations function. Contact Phillip M Nowlen, director, Center for Continuing Education, The University of Chicago, 1307 E 60th St, Chicago IL 60637.

November 13-16, COMPSAC, The Palmer House, Chicago IL. The IEEE Computer Society's second international computer software and application conference. This conference will bring together computer practitioners, users and researchers to share their ideas, experiences and requirements for applications software, management techniques, and software development support, including automated techniques. Contact Wallace A Depp, executive director, Processor and Computer Software System Division, Bell Laboratories, Naperville IL 60540, (312) 690-2111.

November 13-17, Advanced Interactive Computer Graphics, UCLA. This course is designed to broaden the perspective of scientists, engineers and computer professionals who already comprehend basic interactive computer graphics concepts. The course will contain the latest concepts and properties in both current and next generation hardware and software. Contact UCLA Extension, POB 24901, Dept K, Los Angeles CA 90024.

November 15-17, Software Quality Assurance Workshop: Functional and Performance Issues, San Diego Hilton Hotel, San Diego CA. Sponsored by the ACM SIGmetrics and SIGsoft groups, this workshop is concerned with current experiences and new developments in software quality assurance. Papers will be presented on all aspects of quality assurance. Contact JoAnn Lockett, The Rand Corp, 1700 Main St, Santa Monica CA 90406.

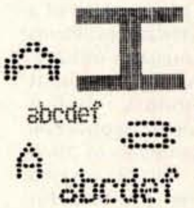
November 19-22, The 11th Annual Microprogramming Workshop, Asilomar Conference Ground, Pacific Grove CA. This workshop will provide a forum for the discussion and comparison of design techniques for firmware and for the

Pay a little bit more and get a printer that's brighter than your computer. The BrighterWriter.™

When a few dollars more buys you a first-class impact printer, why settle for a toy? The BrighterWriter gives you quality to start with. And versatility that stays even if you outgrow your present personal computer.

Built smart like the big ones.

The BrighterWriter's a smart printer. There's a microcomputer inside. It outwits even the bigger, higher-priced printers. So you get versatility to do all kinds of printing. And power to grow on.



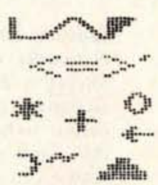
Prints fat, skinny, tall, small.*

This printer can be as creative as your imagination. Stretch out your characters. Squeeze them close. Make them high. Low. Bold. Banner. You name it.

Plugs into your computer.

No matter what personal computer you own or plan to buy, the BrighterWriter plugs in. Simply and quickly. Hundreds of BrighterWriters are working in Apple, TRS-80, Heathkit, S-100 and many other personal computer systems right now.

Pictures and fancy symbols.*



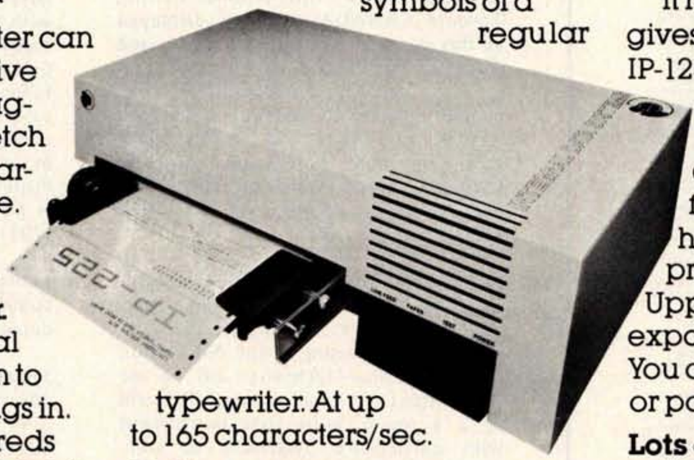
The BrighterWriter draws out your creativity. You can print drawings, graphs, diagrams, bold symbols, or just about any graphic you can imagine.

Picture your page as thousands of dots. The BrighterWriter can fill in the dots, plot them contiguously, stack them, or scatter them. And its special set of graphic characters simplifies the process.

AaBb CcDd EeFf

Prints any character a typewriter can. Faster . . .

The BrighterWriter can print plain and simple. With 7x7 dot matrix clarity. You get all the letters, numbers, and standard symbols of a regular typewriter. At up to 165 characters/sec.



Ordinary paper.

Fancy or plain, the BrighterWriter prints on ordinary paper. Better yet, it prints on many shapes of paper. Single sheets. Roll. Fanfold.

Want more copies? The BrighterWriter prints multiple copies without extra adjustments.

Four easy buttons.

Operating the BrighterWriter couldn't be simpler. Up-front controls are easy to get to. A power button to turn it on. A test button to self-test your printer. A paper feed button to advance the sheets or forms. A line feed button to advance the paper a line at a time.

$949 **Prints any-which-way.**

The BrighterWriter comes in two models. The IP-225, at $949, gives you a BrighterWriter with tractor-feed drive for precision forms control. This one can handle everything from labels to 8½" paper widths.

It has eight form lengths and gives you all the features of our IP-125.

A brighter buy. $799

Our IP-125, friction-feed, BrighterWriter has a 96 character set and prints on 8½" wide paper. Upper and lowercase. It prints expanded characters, too. You can choose a RS-232 serial or parallel interface. $799

Lots of goodies.

There's more. Choose all kinds of options for your BrighterWriter. Up to 132 characters per line, variable character densities, larger buffers, special graphics packages, interface cables, and more.

Give us a call or write. Integral Data Systems, 14 Tech Circle, Natick, MA 01760, (617) 237-7610.

Better yet, see the BrighterWriter at the store nearest you.

Integral Data Systems, Inc.

*Some of these advantages require extra-cost options.

Circle 177 on inquiry card.

supporting hardware. Informal interaction between groups working in similar research and application environments will highlight the topical session. For more information contact Dr Alice G Parker, Micro-11 program chairman, Dept Electrical Engineering, Carnegie-Mellon University, Pittsburgh PA 15213, (412) 578-2472.

November 27-29, European Communities Symposium on Computer Aided Design of Digital Electronic Circuits and Systems, Hotel Hilton, Brussels BELGIUM. The aim of the symposium is to disseminate the results of the *Computer Aided Design Electronics Study* undertaken by the European communities and further to present an assessment of the state of the art of techniques, problem areas and possibilities of further developments in the field of computer aided design of digital electronics. Contact Keness Belgium Congress SA, Rue de l'Industrie 17, 1040 Brussels BELGIUM, (02) 230 09 53.

November 27-December 1, Micro Programming Workshop, Lafayette IN. This 5 day hands-on advanced programming workshop is for individuals interested in developing skills required to plan, prepare, test and document 6800/6801 microprocessor applications software. Contact Jerilyn Williams, Wintek Corp, 902 N 9th St, Lafayette IN 47904.

November 28-30, 9th Annual Canadian Computer Show, International Centre, Toronto CANADA. Products displayed at this show will include: computer and data processing equipment, supplies and services, including minicomputers, peripheral hardware and software, keypunch services, consulting and contract programming and timesharing. Contact Industrial Trade Shows of Canada, 36 Butterick Rd, Toronto Ontario M8W 328, (416) 252-7791.

December 3-5, 9th North American Computer Chess Championship, Sheraton Park Hotel, Washington DC. The 1978 annual meeting of the Association for Computing Machinery will be the site of this chess championship. This will be a 4 round Swiss style tournament with participants restricted to computers. Two rounds will be played on December 3 (1 PM and 7:30 PM), one on Monday (7:30 PM) and the last round on Tuesday (7:30 PM). Deadline for entries is October 20. Contact Prof M M Newborn, School of Computer Science, McGill University, Montreal Quebec H3A 2K6 CANADA.

December 4-6, 1978 Annual Conference of the Association for Computing Machinery, Sheraton Park Hotel, Washington DC. Contact Dr Richard Austing, Dept of Computer Science, University of Maryland, College Park MD 20742, (301) 454-2004.

December 4-6, Minicomputers and Distributed Processing, Atlanta GA. This 3 day seminar will examine the uses, economics, programming and implementation of minicomputers. Contact Philip M Kowlen, director, Center for Continuing Education, The University of Chicago, 1307 E 60th St, Chicago IL 60637.

December 6-8, Data Processing Operations Management, Washington DC. For details, see November 13-15, Chicago.

December 12-14, Midcon/78, Dallas Convention Center, High Technology electronics show and convention. Contact Electronic Conventions Inc, El Segundo CA, (800) 421-6816 (toll free).

December 13, Computer Networking Symposium. Sponsored by the IEEE Computer Society's Technical Committee on Computer Communications and the Institute for Computer Sciences and Technology of the National Bureau of Standards. This symposium will highlight papers of practical and research experiences concerning both computer and communication networks. Contact Dr George Cowan, Computer Sciences Corp, 6565 Arlington Blvd, Falls Church VA 22046.

December 13-15, Distributed Minicomputer Networks, Executive Tower Inn, Denver CO. This seminar will address the minicomputer from the viewpoint of the distributed network user. The structure and management of a large data base and software problems with the trade-offs of languages utilized, hardware types, input and output options, device controllers, system failure and recovery, sample application case studies and the economics of minicomputer applications will be covered in depth. Contact The Institute for Professional Education, Suite 601, 1901 N Fort Myer Dr, Arlington VA 22209, (703) 527-8700.

January 15-17, Minicomputers and Distributed Processing, San Francisco. For details, see December 4-6, Atlanta.

January 17-19, Distributed Minicomputer Networks, Ramada Inn, Arlington VA. For details, see December 13-15, Denver.

January 24-27, International Microcomputers/Minicomputers Microprocessors '79/Japan, Harumi Exhibition Center, Tokyo. Contact ISCM, 222 W Adams St, Chicago IL 60606, (312) 263-4866.

January 30-February 1, Communication Networks Conference and Exposition, Sheraton Park Hotel, Washington DC. Designed to bring together communication network users, consultants, vendors and regulatory officials so that issues can be discussed and analyzed. It is particularly aimed at executives and managers who purchase communication products and services. Contact The Conference Company, 60 Austin St, Newton MA 02160.■

Can you upgrade to 16-bits on a tight budget?

Of course!

Let me introduce you to the 16-bit world of Alpha Micro. Consider *all* the advantages of upgrading your present 8-bit system with the first fully integrated concept of hardware and software. A concept that provides all the features of a 16-bit minicomputer, yet retains all the flexibility, peripheral options, and low cost of S-100 bus systems.

Now you can have much of the sophistication that was previously restricted to only the mini's... Multi-User, Multi-Tasking, Time-Sharing, Memory Management, Disk-File Management, and yes, Peripheral Independence.

Come on, Join Alpha Micro's 16-bit world... You can do it on an 8-bit budget. Five subsystems let you expand your 8-bit system capability as your needs, and your budget, allow:

AM-100 16-bit CPU. Two board S-100 bus compatible microprocessor set, complete with software — a year and a half field proven reliability.

AM-200 Floppy Disk Controller. S-100 bus compatible supports PerSci 277 and Wangco 87 disk drives.

AM-300 Six Port Serial I/O. S-100 bus compatible provides six fully programmable RS-232 ports.

AM-400 Hard Disk Subsystem. S-100 bus compatible interface and CALCOMP Trident series drives (25, 50, 80, 200 and 300 megabyte configurations).

AM-500 Hard Disk Subsystem. S-100 bus compatible formatter-controller and CDC Hawk 10 megabyte hard disk drive.

All Alpha Micro systems are fully software supported including the new completely integrated, minicomputer class accounting system which consists of several hundred programs.

For more information see your local Alpha Micro dealer or write or call.

alpha micro

17881 Sky Park North
Irvine, California 92714
(714) 957-1404

Circle 4 on inquiry card.

Languages Forum

Defining a Language: PL/B

David L Wilson
Engineering Computing Lab
B554 Engineering Bldg
University of Wisconsin
Madison WI 53706

The person who writes a software package for microprocessors is faced with a difficult choice. If the package is written in BASIC, it will use a lot of memory and execute slowly. Certain operations, such as "shift" and "exclusive or," which the microprocessor can do easily, are not available in the standard BASIC language. On the other hand, if the package is written in assembly language, it will not be portable. Four separate versions might be needed for the 8080, Z-80, 6800 and 6502.

This article gives an initial proposal for the middle level language PL/B. As in assembly language, the programmer is given access to those operations that microprocessors can do easily. PL/B programs will be compiled into machine language; they will not be interpreted as in BASIC. PL/B is not an easy language; it is almost as difficult to write a program in PL/B as it is to write it in one of the assembly languages. On the other hand, PL/B programs will be portable.

The first part of this discussion describes the PL/B language. It is written for the experienced assembly language programmer. The second part, written for the person who has previously implemented an assembler or BASIC interpreter, deals with how the PL/B language could be implemented. The PL/B language is designed to be easy to compile: a PL/B compiler should fit into an 8 K byte computer.

Registers

Table 1 gives the names of the registers in PL/B and the corresponding registers to be used on the 6800, 8080, Z-80 and 6502.

The apostrophe in front of the register name distinguishes the register from a symbol name.

On the 8080, if indexing with an offset is used, **HL** is loaded with the offset, and **DE** is added to **HL**. If indexing without an offset is used, **DE** is just transferred to **HL**. In either case, **HL** is then used for the memory reference.

On the 6502, registers **'X** and **'L** will occupy four bytes in page zero of memory. When indexing is used, the 6502 register **X** will be loaded with the offset, and the indirect indexed addressing mode will be used. The 6502 register **Y** will be permanently loaded with a zero, and will be used for indexing without an offset.

Symbol Names

Symbols start with a letter which is followed by letters and digits. The symbol may contain internal blanks (which are ignored) and may be any length. In PL/B all blank, carriage return and line feed characters are ignored.

The symbols in the PL/B language are of three distinct types:

1. name for data storage area
2. name for a procedure (subroutine)
3. name for a constant (parameter)

The name for a constant is used when some parameter will have the same constant value for any one compilation of the PL/B program, but may be changed for another compilation.

Constants

Octal numbers are preceded by the back arrow or its equivalent. The maximum octal number allowed is ←377, except in arithmetic statements using the **'L** or **'X** register when the maximum is ←177777.

Hexadecimal numbers are preceded by $. The number of digits in the hexadecimal number must be a multiple of two. In data area initialization of "declaratives," a hexadecimal number may be as long as desired; in arithmetic statements using the **'L** or **'X** register, it must have four digits; for the **'A** register it must have two digits.

Binary constants are preceded by %. The number of digits in the binary number must be a multiple of eight. In declaratives, the number may be any length; when using the

About the Author

David Wilson works for the University of Wisconsin as a systems engineer, applications programmer and technical writer. One of his current projects is rewriting FORGO, a diagnostic FORTRAN compiler in the C language (see October 1977 BYTE). He has an MS and ABD in computer sciences from the University of Wisconsin.

'L or 'X register, it must have 16 digits; for the 'A register, it must have eight digits.

Decimal numbers have no prefix. In declaratives, the allowable range is −128 to 255. Of course the numbers −1 and 255, for example, are really the same bit pattern. The only difference is whether the number is seen as being signed or unsigned. In an arithmetic statement, the allowable range (see table 2) depends on the register being used.

In an arithmetic statement using 'L, an alphanumeric constant is the two ASCII characters immediately following a ". For the 'A register, it is the first character following a ". In declaratives, the constant is the string of characters surrounded by "s. If a " is to appear within the string, it must be doubled. That is, " " within such a string would result in just one " being stored in memory. In alphanumeric constants, blanks, carriage returns and line feeds are not ignored.

One other constant is the address corresponding to a symbol. This is indicated by preceding the symbol with @. This type of constant may be used only with the registers 'L or 'X. An address constant may be offset by using the form:

@ symbol # constant

For example:

@ TABLE #10

would give the address of the byte ten bytes past TABLE.

Declaratives

The form of a declarative which will assign some memory space for a symbol and give the initial values for that byte or bytes is:

symbol , value list ;

The ; is used to end all statements in PL/B. The value list is composed of one or more constants, separated by commas. The symbol is assigned the address of the first byte filled from the value list. For example:

SYMBOL TABLE ," ",$0000;

would set up three bytes containing a blank and two binary zeroes. SYMBOL TABLE would be assigned the address of the first byte.

The EQU (equivalence) operation of many assemblers is done as follows:

symbol ,= symbol ;

or if an offset is used:

symbol ,= symbol # constant ;

or if the symbol corresponds to an absolute address:

symbol =# constant ;

or if the symbol is the name for a constant parameter:

symbol ,= constant ;

For example:

END OF TABLE, = TABLE # 10;

will assign the symbol END OF TABLE the address of the byte which is ten bytes past TABLE.

The ORG (origin) operation of many assemblers is done as follows:

constant ;

(Remember that the constant may be the address of a symbol with an offset.)

To reserve a block of memory without giving any initial values, the form is:

symbol ,+ constant ;

For example:

TABLE ,+ 100;

would set aside 100 bytes for TABLE. The bytes would all be initialized to binary zeroes.

Declaratives will come at the beginning of the program before the first procedure or

Register Type	PL/B	6800	8080	Z-80	6502
8 bit register	'A	A	A	A	A
16 bit register	'L	M	BC	BC	M
index register	'X	X	DE	IX	M
carry bit	'C	C	CY	CY	C

Table 1: Registers used by PL/B and in the more popular microprocessors.

Register	Range
'A	− 128 to 255
'L	−32768 to 65535
'X	0 to 65535
'C	0 to 1

Table 2: Summary of allowable ranges for registers used in PL/B. Register 'A contains eight bits. Registers 'L and 'X contain 16 bits. 'C is used as a flag.

Symbol	Direct Reference
symbol #constant	to get an offset
#constant	absolute address
#'X	via index register, no offset
#'X #constant	via index register with offset

Table 3: Various ways to address memory locations.

=	load	(	rotate left
.	store	)	rotate right
+	add	<<	shift left
−	subtract	>>	shift right
&	and	.@	push
\	or	=@	pull
↑	exclusive or	↑↑	complement
+&	add with carry	−−	negate
−&	subtract with carry		
.+	decimal add		
.−	decimal subtract		

Table 4: Summary of all operations performed by PL/B.

between procedures. No symbol, other than a procedure name, may be used before it is defined in a declarative. Also, a memory byte or pair of bytes is normally referenced by giving the corresponding symbol. However, there are other ways to reference a memory location as shown in table 2. The complete list is summarized in table 3.

Arithmetic Statements

The form of an arithmetic statement is:

arena-of-action operation-list ;

The "arena of action" is a register name or memory reference. All the operations in the operation list will take place in that register or memory location. The operations are summarized in table 4. Any operation in the first column must be followed by a memory reference or a constant. Operations in the second column must not be followed by a memory reference or a constant. If a constant is used, the PL/B compiler will produce an instruction with an immediate operand. A special check is made for the operations +1 and −1. For these, PL/B will produce increment and decrement instructions. For example:

```
'A +1 .WHERE ( .WHEN .@;
```

will increment the previous contents of the accumulator, store the result at WHERE, rotate the accumulator and carry bit left, store that result at WHEN, and also push it onto the stack.

Not all operations will be permitted for all possible arenas of action. For example, if the arena of action is a memory location, one may not add to it the contents of another memory location. That is:

```
PLACE +INCR;
```

would not be allowed, but:

```
'A =PLACE +INCR .PLACE;
```

would be allowed.

The shifts are logical shifts. They are equivalent to setting 'C = 0; and then rotating. On the 6502, the right rotate is done by a subroutine.

PL/B does not use labels and GO TO commands. It does require structured programming, which is a superior method for organizing a program in order to make it easier to debug and easier for others to understand. Three structures are used in PL/B: procedures, if-else-endif, and loop-endloop.

Procedures

The form of a procedure (subroutine) is:

PROCEDURE *name* [

executable statements

]

The first (main) procedure is unnamed. When the bottom of a named procedure is reached, control returns to the procedure that called it. When the bottom of the main procedure is reached, control returns to the system monitor. The final procedure of a program is followed by an extra] to mark the end of the program.

A procedure is called by:

! PROCEDURE *name* ;

or, if the procedure is at an absolute address:

! #constant ;

or, if the procedure address is in 'L:

! #'L;

This last possibility is implemented by storing the contents of 'L into an upcoming JSR instruction. If the program is to be stored in read only memory, the contents of 'L are stored in a JMP instruction in programmable memory, and then a JSR is done to that JMP.

An argument list may follow the procedure name. The list may contain only constants, but remember that the address of a symbol is a constant (therefore, call by address is possible). The form of the argument list is identical to that of an initial value list in a declarative. The constants are placed in memory directly following the JSR to the procedure. For example, a possible call could be:

!MULTIPLY, @TIMES;

If-else-endif Structure

The form of this structure is either:

logical-expression ?

==	equal
!=	not equal
>	greater than
<	less than
>=	greater than or equal
<=	less than or equal

Table 5: Relational tests used by if-else-endif statements.

```
        .
        .
    statements done
    if expression true
        .
        .  /
or:

    logical-expression ?
        .
        .
    statements done
    if expression true
        .
        .
      : .
        .
        .
    statements done
    if expression false
        .
        .  /
```

The simplest form of a logical expression is a logical relation:

arena-of-action operation-list test

The *operation list* may be empty. The *test* is a relation followed by a memory reference or constant. PL/B will generate a compare unless the constant zero is used and some previous operation would have set the condition register. The possible relations are summarized in table 5.

A logical expression may also be made up of several logical relations connected by && or \\. If a logical relation is followed by &&, and if the relation is false, control immediately passes to statements which are to be done if the logical expression is false or to the end of the structure if there are no such statements. If it is true, the next logical relation is tested. For example:

```
'A =N >0 && 'A =M +1 .M <0 ?
'A =I -1 .I; /
```

If *N* is greater than zero, then M will be incremented, and the result checked. If that result is negative, I will be decremented. If that result is zero or positive, I will not be decremented. On the other hand, if N is zero or negative, not only will I not be decremented, but M will not be incremented either.

If a logical relation is followed by \\, and if the relation is true, control immediately passes to the statements which have to be done if the logical expression is true. If it is false, the next logical relation is tested. For example:

```
'A =N >0 \\ 'A =M +1 .M <0 ?
'A =I -1 .I; /
```

If N is zero or negative, then M will be incremented and the result tested. If that result is negative, I will be decremented. If that result is zero or positive, I will not be decremented. On the other hand, if N is positive, I will be decremented, but M will not be incremented.

Loop-endloop Structure

The form of this structure is:

```
[   .
    .
    .
  statements within the loop
    .
    .  ]
```

When the computer reaches the bottom of the loop, it will start again at the top. The loop is exited by using the !; statement within an if-endif structure. For example, the following will execute a loop ten times, using location / to save the count.

```
'A =10 .I;
[   .
    .
    .
  statements in loop
    .
'A =I -1 .I == 0 ? ! ; / ]
```

The !; may also be used to exit a procedure early. Multiple exclamation points may be used, in which case that many loops (possibly including the procedure itself) will be exited. In the example above, the test to exit the loop was put at the end of the loop. This is not required. The test may be at the start of the loop or even in the middle of the loop. One loop may contain several conditional exit instructions.

All programming needs documentation so it can be understood. Comment lines start with an * and finish with a ; like any other statement.

Input/Output

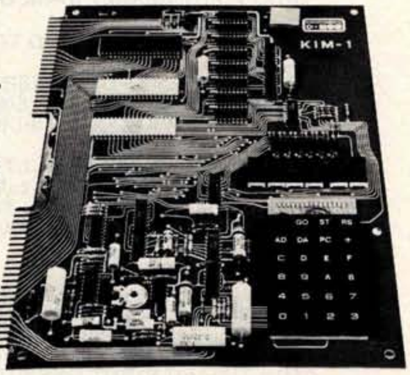
Input devices will be assigned to letters of the alphabet. They will be referred to by an * followed by the letter. The standard input device will be ***I**; the standard list output device will be ***L**; the standard device for object program output will be ***O**. A program to echo input would be:

```
[['A =*I .*L;]]]
```

The MULTIPLY procedure in listing 1 is an example of the use of the PL/B language. It leaves in **'L** the product of 'A and the byte pointed to by the argument following the MULTIPLY call.

Our competitors hated the KIM-1* at $245!

No wonder; the KIM-1 microcomputer was less money for more features than other microcomputer boards.



But KIM-1 users loved us. Up until this week, for only $245, you could own a completely assembled microcomputer board with 1K RAM, 2K ROM, cassette and serial interfaces, 15 I/O lines, 2 interval timers and a six digit LED display. With MOS Technology's comprehensive 6502 mpu instruction set. And the best documentation in the business.

And now the KIM system is the most expandable at any price. A motherboard (KIM-4), memory expansion board (KIM-3B), editor/assembler (KIM-5), prototyping board (KIM-6). All available now.

Just about the only problem potential KIM-1 users encountered was the fact that the KIM-1 was in short supply: demand kept growing faster than our expanding production. But not any more.

Will they hate KIM-1 even more at $179^95?

That's right: $179.95. For the same, superb microcomputer. And available *now.* Further evidence of the power and ability of Commodore's semiconductor and computer divisions. A synergism of MOS Technology's leadership in the design and development of mpus, RAMs and ROMs together with Commodore's know-how in mass production and mass marketing.

Here's how the line-up looks now:

Rockwell AIM 65	Synertek VIM	RCA VIP	Commodore MOS KIM-1
$375	$269	$245	**$179^95**

We wonder what our competitors are saying now!

*KIM is a registered trademark of Commodore Bus. Mach., Inc.

☐ Yes... I want ______ KIM-1s @ $179.95 now.*

☐ I'm enclosing my check in the amt. of $______

☐ Please charge this to my credit card:

☐ Master Charge ☐ BankAmericard/VISA

☐ American Express

My Card No. ______ Expires______

Name (print) ______

Address ______

City______ State______ Zip______

**Calif. residents add 6% sales tax (Santa Clara County – 6½%).*

☐ Please send me information on the KIM Expansion Modules.

☐ Please supply me with the name of your nearest dealer.

Commodore/MOS • 901 California Ave. • CA 94304

Dealer inquiries invited (415) 326-4000

```
MULTIPLICAND ,+ 2;
MULTIPLIER ,+ 1;
PRODUCT ,+ 2;

MULTIPLY [
        'A .MULTIPLICAND#1;
        * GET RETURN ADDRESS FROM STACK, COPY TO 'X;
        * ADD 2, PUT RESULT BACK ON STACK FOR RETURN;
        'L =@ .'X +2 .@;
        * LOAD ADDRESS POINTED TO BY 'X INTO 'X;
        'X =#'X;
        * GET AND TEST MULTIPLIER;
        * IF MULTIPLIER N<0, REVERSE SIGN OF BOTH;
        * MULTIPLIER AND MULTIPLICAND;
        'A =#'X .MULTIPLIER < 0 ?
            'A =0 -MULTIPLIER .MULTIPLIER;
            'A =0 -MULTIPLICAND#1 .MULTIPLICAND#1; /
        * CONVERT MULTIPLICAND TO 16 BIT NUMBER;
        'A =MULTIPLICAND#1 < 0 ? 'A =-1; : 'A =0; /
        'A .MULTIPLICAND;
        'L =0 .PRODUCT;
        * GO THROUGH LOOP FOR EACH BIT IN MULTIPLIER;
        * IF LAST BIT OF MULTIPLIER ON;
        * ADD MULTIPLICAND TO PRODUCT;
        * THEN DOUBLE MULTIPLICAND, AND SHIFT;
        * MULTIPLIER RIGHT ONE BIT;
        [ 'A =MULTIPLIER == 0 ? !; /
            'A >> .MULTIPLIER;
            'C == 1 ?
                'L =PRODUCT +MULTIPLICAND .PRODUCT; /
            'L =MULTIPLICAND +MULTIPLICAND .MULTIPLICAND;]
        'L = PRODUCT; ]
```

Listing 1: An example multiplication program in PL/B. This program leaves the product of 'A *and the contents of the address pointed to by* 'X *in* 'L.

Conditional Compilation

If all the logical relations within the logical expression of an if-else-endif structure consist of a symbolic name for a constant being compared to another constant, then the result of the logical expression will be determined by the PL/B compiler during compilation. If it turns out that some statements cannot be executed, because the logical expression will either always be true or always be false, then no code will be generated for those statements. Normally, the if-else-endif structure is permitted only within a procedure. However, conditional compilation may be used in the declaratives. The conditional compilation feature could be used, for example, to produce different versions of a program according to how much memory is available.

In order to keep the PL/B compiler small, conditional compilation and symbolic names for constants could be handled by a separate precompiler program analogous to a macroassembler's macrogeneration pass. The precompiler would eliminate the conditional compilation ifs along with any skipped code and substitute the corresponding constant for any symbolic name of a constant.

This completes the specification of the PL/B language itself. The rest of this article considers implementation problems.

Symbol Table

The symbol table will initially be:

SYMBOL TABLE," ",$0000;

NEXT will point to the next available spot:

NEXT, @SYMBOL TABLE#3;

Every character in the symbol table will be followed by an address. While looking for a symbol, the compiler first checks to see if the current character matches the character in the symbol table. If it does not, the associated address will tell the compiler where to look next for a match. If that address is zero, then this is a new symbol, and the rest of its letters must be stored starting at the place pointed to be NEXT. If it does match the character from the symbol table, then the compiler moves up three bytes, and tries to match the next character of the symbol starting at this point. The compiler will use a binary zero for end-of-symbol character. Once the end-of-symbol character is matched, then the definition of the symbol can be found by moving up three bytes. A symbol definition takes three bytes: one byte tells what kind of symbol it is; the other two give the symbol's address.

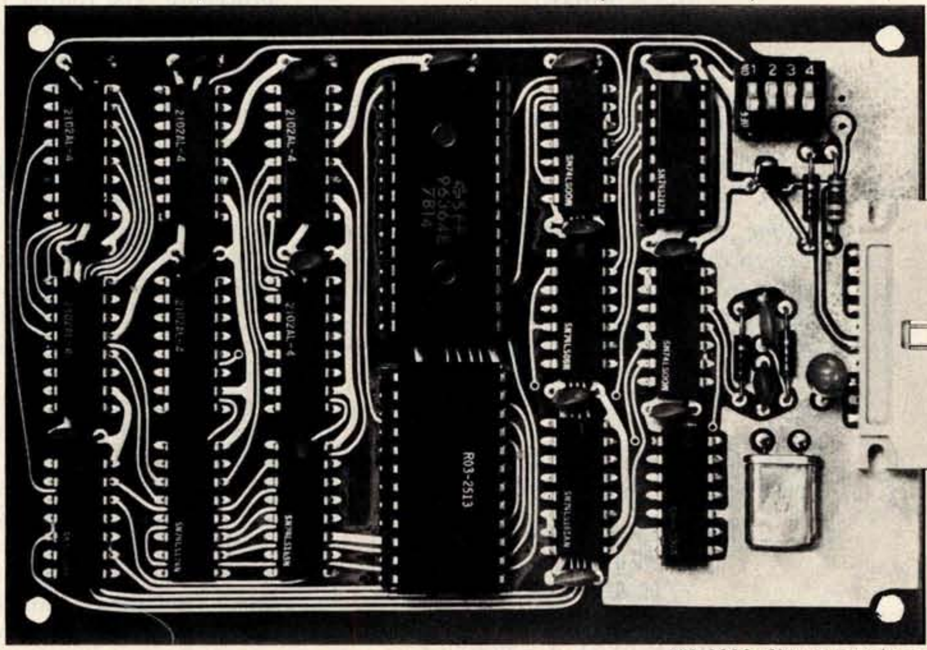
The following routine is used to get a

Everything you ever wanted in a video interface.

(Automatic scrolling, cursor, erase functions, etc.)

And less.

(Stand-alone — requires no processor.)



VIB-1000. Shown actual size.

NPC's NEW VIB SERIES. FROM $129^95

Don't let the low price fool you. These are not kits. They are fully assembled and tested boards. NPC's custom CRT Processor Chip is the secret behind the entire family. All perform character entry and cursor control functions automatically. They execute all normal cursor commands as well as more sophisticated functions such as erase page, erase to end of line, etc. Text automatically scrolls when cursor reaches bottom of page. Display format is fixed at 16 lines of 64 characters for optimum efficiency.

VIB-1000 All standard features described above. $129.95
VIB-2000 All features of the VIB-1000 plus FOREGROUND AND BACKGROUND video modes, SCREEN PROTECT and READY/BUSY handshake. $159.95
VIB-3000 All features of the VIB-2000 plus SCREEN READ capability. $169.95
VIB-2000S Serial interface (RS-232C) version of the VIB-2000. $189.95

Where Pricing is as Important as Technology

NPC Nucleonic Products Company

6660 Variel Avenue
Canoga Park, California 91303 • (213) 887-1010
Telex 69-8481 • TWX 910-494-1954
A Division of DuMont Electronics Corporation

If your local computer store does not have NPC's new VIB series yet, contact the distributor nearest you to order direct.

STEP SYSTEMS, INC.
4815 West 77th Avenue
Minneapolis, MN 55435
(612) 831-3644

NATIONAL ELECTRO SALES
12063 W. Jefferson Blvd.
Culver City, CA 90230
(213) 391-6294

DEALER INQUIRIES INVITED

Circle 287 on inquiry card.

```
SEARCH [
    'X = @SYMBOL TABLE;
    [   [   'A == #'X ? !;/
            'A .@;
            'L =#'X#1 == 0 ?
                'L =NEXT .#'X#1 .'X;
                'A =@;
                [   'A .#'X;
                    'L =0 .#'X#1;
                    #'X == 0 ? !; /
                    'X+1+1+1;
                    !GET CHARACTER;
                    'A < "1 \\ 'A > "Z \\ 'A > "9 &&
                        'A < "A ? 'A .@ =0;/ ]
                'A =0 .#'X#3 .#'X#4 .#'X#5;
                'L ='X +6 .NEXT;
                !!;/
            'X ='L;
            'A =@; ]
        'A == 0 ? !; /
        !GET CHARACTER;
        'A < "1 \\ 'A > "Z \\ 'A > "9 && 'A < "A ?
            'A .@ =0;/ ]
    'X+1+1+1;
    'A =@; ]
```

Listing 2: This is a search routine which tries to match a given symbol against those contained in the symbol table. When the routine is finished, 'X *contains the address of the symbol definition and* 'A *contains the character following the symbol.*

character from the input. It is not used when constructing an alphanumeric constant.

```
GET CHARACTER [
    [ 'A =*I != "  && 'A !=
        ←12 && 'A != ←15 ?
        !;/ ]
    'A > ←137 ? 'A – ←40; / ]
```

The routine in listing 2 searches for a match for a given symbol. When the routine is entered, the first letter of the symbol is already in 'A. When the routine exits, 'X points to the symbol definition (all zeroes for a new symbol), and 'A has the special character that follows the symbol.

Forward References

Unlike an assembler, PL/B will be a 1 pass compiler. Thus at times it will have to backtrack and insert addresses in code already generated. This is done by generating an entirely separate line for the loader in order to stuff the one or two bytes into the address area of an instruction.

Consider first a forward reference to a procedure. The type byte in the symbol definition for the procedure will not only reveal that this is a procedure, but that it still is not defined. In this case, the so-called address actually points to a list of places where the address must be inserted when its value is known. Each cell of this list will be five bytes long, of which only four are used. This type of cell will also be used to keep a status stack for the if-else-endif and loop-endloop structures currently in use. The cell will contain two addresses: the first two bytes will contain the address of the next cell on the list (or zero at the end of the list), and the next two bytes will contain the address where the procedure's address should be placed when the program is loaded.

Let LOC be the current program counter, FREE be the pointer to a list of free cells (this list is initially empty), and let 'X point to the symbol definition for the undefined procedure. The routine of listing 3 will add the forward reference to the symbol's forward reference list.

The routine in listing 4 fills in the address when it is known. Let LOC contain the address of the procedure and let 'X point to the first cell of the address insertion list. OBJECT CODE is a routine that produces object code. The byte which goes out is placed into 'A and the address where it goes when loaded is placed into 'X.

It is necessary to keep a status stack of structures in use. For this stack (last in first out, LIFO list, not the hardware stack) all five bytes of a cell will be used. The fifth cell marks the "type" of the cell. The possibilities are summarized in table 6. Thus, there will be two stack entries for every loop, and one entry for each if currently active.

When an if is compiled, the code for the ? will be a branch if true to the code following the ?, and then a JMP instruction to make the possibly big jump over the code when the condition is false. The code for a && would be set up to branch to this JMP instruction if the preceding condition is false. The code for \\ and && do set up local forward references, which can be handled in a manner similar to the method given above. Of course, these addresses are relative and only use eight bits.

Automically Indented Listing

A separate PL/B listing program should be provided which will provide an automatically indented listing. Structured programs are much easier to read if they are properly indented. In an automatically indented listing, the listing program removes any indenting the user may have done (it skips leading blanks) and indents the listing according to the structure present in the user's program. For each level the listing program should produce two characters at the start of the line: a vertical bar and a blank. Remember, if a line starts with a :, it reduces the "level" by one for that line only. If a vertical bar is not available a slash or exclamation point might be used. The result is a column of vertical bars stretching between the top and bottom of a structure. If the programmer has made some mistake in structures, (s)he may easily spot it from such a listing. All trailing comments should be started at the same column, if possible.

Each line of the listing should be preceded by four characters giving the location counter and a blank. If the line starts with a # (an ORG command), the location printed should be the one given in that line. If a ,= operation is used (an EQU operation), the resulting value for the symbol should be given.■

```
TEMP ,+ 2;
TEMP2 ,+ 2;
FREE , $0000;
LOC ,+ 2;

FORWARD [
    'X .TEMP;
    'L =#'X#1 .TEMP2;
    !GET CELL;
    'L =LOC .#'X#2 =TEMP2 .#'X;
    'X 'L =TEMP;
    'L .#'X#1; ]

GET CELL [
    'X =FREE == 0 ?
        'X = NEXT;
        'L = NEXT +5 .NEXT;
    :   'L = #'X .FREE; / ]
```

Listing 3: Routine to add forward references to a symbol's forward reference list.

Type	"Address" Field Contents
1	Address of the start of a loop.
2	Link to address insertion list for loop end. Used for !; instructions.
3	Place to put address for : or / (else-endif).
4	Place to put address for / (: already found).

Table 6: Various address field types.

```
CHAIN [
  [   'X .TEMP == 0 ? !; /
      'X =#'X#2;
      'A = LOC;
      !OBJECT CODE;
      'X+1;
      'A = LOC#1;
      !OBJECT CODE;
      'X =TEMP;
      !FREE CELL; ] ]

FREE CELL [
    'L = #'X .TEMP2 =FREE .#X;
    'X .FREE =TEMP2; ]
```

Listing 4: This routine takes the forward reference list generated by listing 3 and fills it in when the address is known.

Making a list?

Check here for new ideas

DR DOBB'S JOURNAL OF COMPUTER CALISTHENICS AND ORTHODONITA, volume I edited by Jim C Warren.
☐ This is a collection of everything from the first year of Jim Warren's outstanding journal. 359 pp. $13.00 (no photo).

YOUR HOME COMPUTER by James White.
☐ A clearly written nontechnical description of personal computers that required no prior knowledge of computers or electronics. Yours for $6.00.

GETTING INVOLVED WITH YOUR OWN COMPUTER by Leslie Solomon and Stanley Viet.
☐ Whether your interest in business applications, word processing, education, security, computerized video games, computer art and music. Robots, or amateur radio, this book will get you started and keep you on the right track. 216 pp. $5.95.

UP YOUR OWN ORGANIZATION! by Donald M Dible.
☐ A great handbook on how to start and finance a new business, this is the most comprehensive reference we've seen on the subject. 372 pp. Available for $14.95 in hardcover.

HOW TO PROFIT FROM YOUR PERSONAL COMPUTER by Ted G Lewis.
☐ If you have wanted to use your personal computer in a profitable manner, and you feel you need some help planning programming techniques for common business applications, this book is for you. 191 pp. $7.95.

BASIC AND THE PERSONAL COMPUTER by Thomas A Dwyer and Margot Critchfield.
☐ After giving an in-depth course in BASIC, the book discusses microcomputer hardware, graphics, word processing, sorting, simulation and data structures. Easy to read and informative. 438 pp. $12.95 (no photo).

TV TYPEWRITER COOKBOOK by Don Lancaster.
☐ A complete guide to low cost television display of alphanumeric data, several chapters of which were published ahead of the book in early issues of BYTE magazine. $9.95.

THE CHEAP VIDEO COOKBOOK by Don Lancaster.
☐ This latest in Don Lancaster's series of hardware books continues where the **TV Typewriter Cookbook** leaves off. New, inexpensive video display circuits (and the software to drive them) are described. 256 pp. $5.95 (no photo).

APL AN INTERACTIVE APPROACH Second Edition, Revised by Gilman and Rose.
☐ Here's an excellent way to introduce yourself to the APL language. Gilman and Rose have extensively updated their popular book to include the latest information about APL and the various forms now in use. $11.95.

SARAGON by Dan and Kate Spracklen.
☐ Written about the West Coast Computer Faire chess champ, "Saragon," which is a program in Z-80, this book is available for the first time in October 1978. $14.95 (no photo).

THE WIRED SOCIETY by James Martin.
☐ This is a fascinating and not-to distant account of what the future might be like, changed by the new telecommunications technology. Prefacing his discussions with historical as well as contemporary references, Martin lures the reader into an impending technical revolution, which will change corporations, change government, and changes the lives of everybody. $12.95 hardcover (no photo).

MICROPROCESSOR INTERFACING TECHNIQUES (Expanded 2nd edition) by Austin Lesea and Rodnay Zaks.
☐ Now available in this newly revised edition featuring: new material on "useful circuits,"an enlarged floppy disk section with new double density controllers, updates on LSI controllers—and much more. Including basic concepts and techniques, this book is a very comprehensive handbook on interfacing. 416 pp. $9.95.

MICROPROCESSORS: FROM CHIPS TO SYSTEMS by Rodnay Zaks.
☐ A complete and detailed introduction to microprocessors and microcomputer systems. $9.95.

BASIC BASIC, 2nd EDITION by James S Coan.
☐ If you're not already familiar with BASIC, this book is one of the best ways to learn about this popular computer language. BASIC (which stands for Beginner's All-Purpose Symbolic Instruction Code) is easy to learn and easy to apply to many problems. $8.95.

ADVANCED BASIC by James S Coan.
☐ This is the companion volume to James Coan's **Basic BASIC.** In this book you'll learn about some of the more advanced techniques for programming in BASIC, including string manipulation, the use of files, plotting on a terminal, simulation and games, advanced mathematical applications and more. $7.95.

CHESS AND COMPUTERS by David Levy.
☐ This 145-page paperbound book is loaded with chess games played by computers. The games are computer versus computer and computer versus human. $8.95.

CHESS SKILL IN MAN AN MACHINE edited by Peter W Frey.
☐ This is a most fascinating book, concerning itself with the when, how, and why of computer chess. $14.80 hardcover.

SCIENTIFIC AND ENGINEERING PROBLEM SOLVING WITH THE COMPUTER by William Ralph Bennett Jr.
☐ Besides teaching BASIC (which it does admirably), this lively, lucid book presents a wealth of imaginative and unusual applications programs taken from many disciplines. In its diversity and elegant style, it ranks with Donald Knuth's works as a milestone in the art of computing. Hardcover, $19.95.

GAMES, TRICKS AND PUZZLES FOR A HAND CALCULATOR by Wallace Judd.

☐ Got a friend who won't have anything to do with calculators? You just might woo him or her into the electronic age with some of the tricks in this book. Capers, games and astounding calculations are here to dazzle. They've even pulled a calculator apart to show you what the insides look like! A neat book for anyone suffering from technophobia or plain boredom. 91 pp. $4.00.

FUNDAMENTALS OF RECORDKEEPING AND FINANCE FOR THE SMALL BUSINESS by Robert C Ragan, CPA, and Jack Zwick, Ph.D.

☐ Once you have your organization or business up and running, records must be kept. 196 pp. Only $10.00 in hardcover.

THE ART OF COMPUTER PROGRAMMING by Donald E Knuth.

Praised by many critics as the best books in their field. These books are part of a projected seven volume omnibus survey of computer science now being completed by Donald E Knuth.

☐ **Volume I, Fundamental Algorithms,** 634 pp. $21.95 hardcover.

☐ **Volume II, Seminumerical Algorithms,** 624 pp. $21.95 hardcover.

☐ **Volume III, Searching and Sorting,** 722 pp; $21.95 hardcover.

FUNDAMENTALS AND APPLICATIONS OF DIGITAL LOGIC CIRCUITS, 2nd edition by Sol Libes.

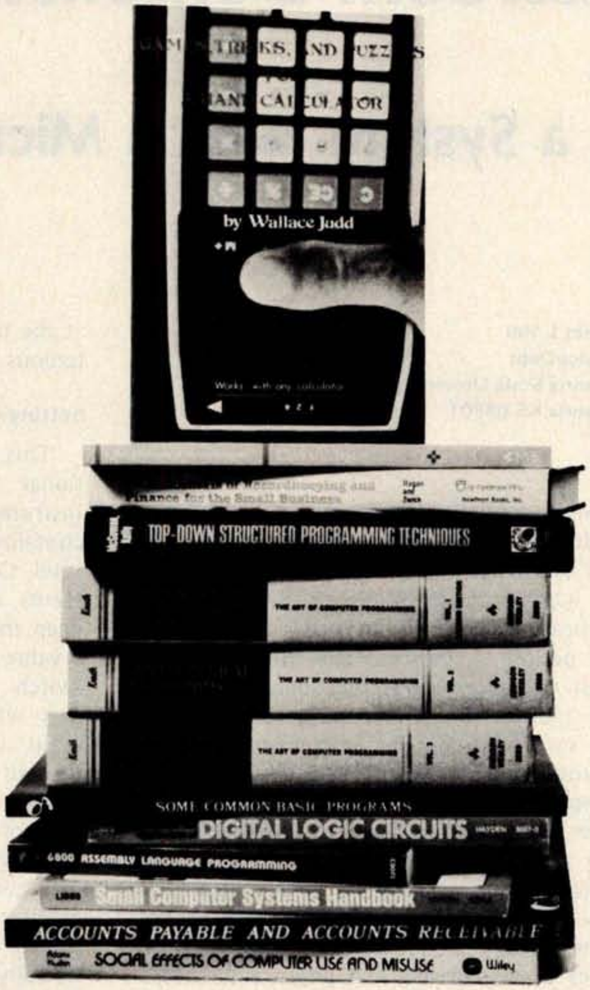
☐ You'll find everything from the basic theory, to various circuit designs, to the most recent applications, including the circuitry of calculators, digital voltmeters, frequency counters, and computers. 200 pp. $7.95.

SMALL COMPUTER SYSTEMS HANDBOOK by Sol Libes.

☐ This is an introductory text for those with a minimal knowledge of electronics who are interested in microcomputers as a hobby. The author explains microcomputer concepts from number systems and logic circuits to programming languages and applications with the emphasis on practical knowledge. 197 pp. $8.45.

THE LITTLE BOOK OF BASIC STYLE by John M Nevison.

☐ This book teaches you how to write clear, well-planned programs you can read. 151 pp. $5.95 (no photo).



SOME COMMON BASIC PROGRAMS by Lon Poole and Mary Borchers.

☐ At last, a single source for all those hard to find mathematics programs! All the programs are written in a restricted BASIC suitable for most microcomputer BASIC packages, and have been tested and debugged by the authors. $8.50.

SOCIAL EFFECTS OF COMPUTER USE AND MISUSE by J Mack Adams & Douglas H Haden.

☐ Most of us have come to accept the presents of computers in our lives. Their effect on society as a whole is worthy of carefull consideration. This book surveys the evolution of computers, their application, and the social, economic and philosophical implications of their increasing use. 326 pp. $10.00.

PROGRAMMING A MICROCOMPUTER: 6502 by Caxton C Foster.

☐ This informal, well-written book may be just what you need to enter the world of microcomputers. Foster uses the 6502 microprocessor (used in the KIM-1, the PET, and the Apple microcomputer systems) as a basis for discussing the techniques of writing programs. 231 pp. $9.95.

TOP-DOWN STRUCTURED PROGRAMMING TECHNIQUES by Clement L McGowan & John R Kelly.

☐ What is structured programming? The authors answer this question in this lively, well-written book. 288 pp. $15.95 hardcover.

6800 ASSEMBLY LANGUAGE PROGRAMMING by Lance A Leventhal.

☐ From Osborne & Associates we have another in their series of microcomputer books. Lance A Leventhal discusses assembly language programming for computers using the 6800 microprocessor. The 6800 instruction set is presented in depth with chapters on assemblers, simple programs, code conversion, tables and lists, subroutines, input/output, interrupts, program design and documentation, and sample projects. $8.50.

ACCOUNTS PAYABLE AND ACCOUNTS RECEIVABLE by Lon Poole and Mary Borchers.

☐ This is a set of 21 programs which constitute an invoice-linked accounts payable and a low-volume invoice accounts receivable system for small businesses. It is the third in Osborne & Associates' series of BASIC business programs and will be complemented by the soon to be published General Ledger. 318 pp. $15.00.

BITS™ inc
Books to erase the impossible

POB 428, 25 Route 101 West
Peterborough, NH 03458

DIAL YOUR BANK CARD ORDERS ON OUR **TOLL-FREE HOT LINE**: **1-800-258-5477.**

Name ____________

Address ____________

City ____________ State ____________ Zip ____________

Signature ____________

My check enclosed

Check # ____________

Amount ____________

My card # ____________

Expires ____________

		Total
Number of books	______	______
Postage $.75/book or $1.00 outside U.S.		______
Grand Total		______

Prices subject to change without notice. *Dealer Inquiries Invited.* *You may photocopy this page.*

A Classroom Demonstration

Controlling a System with a Microcomputer

Garnet L Hill
Physics Dept
Emporia State University
Emporia KS 66801

Microprocessors are large scale integrated circuits that consist of a central processing unit and a control section, with provisions for handling parallel binary data in communication with various types of memory and peripheral devices. The use of such a small computer is fast becoming an everyday phenomenon. The ubiquitous nature of small computers implemented with microprocessors makes it imperative that the complete course in the physical sciences cover these tools of control in some way. The purpose of this article is to point out a fairly simple application which illustrates how microprocessor control systems can be demonstrated to students in teaching laboratories of engineering and physical sciences. The example can be used as a classroom demonstration or laboratory experiment suitable for advanced high school or college levels depending upon the degree of detail provided by the teacher.

The object of this experiment is to illustrate how a computer program, running in a simple microprocessor, can be used as a control element for a system. The microprocessor and its program form a sort of complicated "black box" receiving inputs from sensors and calculating an output presented to an actuator. In the 1960s, before the existence of microprocessor technology, such black boxes were often constructed out of smaller integrated circuits and programmed by wiring the pins. Earlier than the 1960s, many of the functions now commonplace in digital electronics were simply prohibitively expensive or impossible to achieve. The advent of the programmable processor, however, brought a flexibility of function not possible with such "hardwired" logic: changing a program, the "software" of the application, can be done at the flick of a button rather than through tedious rewiring.

Setting Up the Experiment

This experiment utilizes a typical educational microprocessor system, the E&L Instruments "Dyna-Micro" computer which contains an 8080A processor made by the Intel Corporation. The experiment implements a simple control system designed to keep the rate of flow of an airstream set at a value as measured by three simple binary switch sensors. A stock solid state relay part with microprocessor compatible TTL input and 120 V at 3 A AC power ratings is used to turn on and turn off the fan motor, with a program interpreting the state of the wind sensors and deciding what the motor should be doing.

Figure 1 shows the design of the experiment at a global level. Three wind vanes are placed in front of the fan and displaced laterally to avoid blocking the wind from the fan. Due to the mixing of the fan's air stream with the "still" air of the rest of the room, the wind pressure is greatest near the fan, and least furthest away from the fan. Thus sensor A experiences the greatest displacement and sensor C experiences the least displacement for a given average speed of the wind. Each vane is a binary sensor: if the wind speed is above a certain level, the switch of the sensor is closed; if the wind is below that level, the sensor switch is open. Three bits of digital information are read by the 8080's computer program using an "IN" instruction for the appropriate port.

The design of the wind vane sensor elements is simple, as shown in figure 2. The base of the sensor is a block of wood. The dimensions are chosen so that the block can be clamped at the edge of the laboratory table with the proper spacings as shown in figure 1. The dimensions of figure 1 are based on the author's experiences, but experimentation with the relative spacing of the sensors and fan can prove fruitful in demonstrations. Each sensor is a single

Osborne & Associates, Inc.

The World Leaders in Microprocessor Books

If you want information on microprocessors, read the Osborne books.

An Introduction to Microcomputers

Volume 0 - The Beginner's Book

If you're not familiar with computers, but would like to be, then this is the book for you. Computer logic and terminology are introduced in a language the beginner can understand. Computer software, hardware and component parts are described, and simple explanations given for how they work. Text is supplemented with creative illustrations and numerous photographs. 300 pages.

Volume 0 #08-X $7.95

Volume 1 - Basic Concepts

A must for anyone in the computer field, this best selling text explains hardware and programming concepts common to all microprocessors. Its universal appeal is reflected by its having the greatest yearly sales volume of any computer text. 350 pages.

Volume 1 #02-0 $8.50

Volume 2 - Some Real Microprocessors
Volume 3 - Some Real Support Devices
and update subscriptions

These two books provide complete descriptions of virtually every microprocessor and most support devices. There are no other books like these; they provide detailed part descriptions from an independent source.

To cope with the rapid evolution of microprocessor products, Volumes 2 and 3 have been printed in loose leaf form; each volume has its own series of six bimonthly updates, allowing you to remain current with all parts as soon as they are really available. Updates sold separately.

These two books replace the 1977 edition of Volume II - Some Real Products.

Volume 2 with binder #14-4 $20.00
Volume 3 with binder #17-9 $20.00
Volume 2 update only $25.00/yr.
Volume 3 update only $25.00/yr.
Volume 2 and 3 updates $40.00/yr.

Program Books Written in BASIC

Payroll with Cost Accounting
Accounts Payable and Accounts Receivable
General Ledger

These books feature complete, quality applications software for small-to-medium sized businesses. Each book includes fully documented program listings, sample printed reports, installation instructions and user's manual. Written in an extended Wang BASIC (write to ask us about our CP/M CBASIC version and other conversions). 375 pages each.

Payroll #09-8 $15.00
AP & AR #13-6 $15.00
G. Ledger not yet available, see order form

Some Common BASIC Programs

76 short practical programs, most of which can be used on any microcomputer with any version of BASIC. Complete with program descriptions, listings, remarks and examples. 200 pages.

SCBP #06-3 $8.50

Assembly Language Programming

8080A/8085 Assembly Language Programming
6800 Assembly Language Programming

These books describe how to program a microcomputer using assembly language. They discuss classical programming techniques, and contain simplified programming examples relevant to today's microcomputer applications. 400 pages each.

8080A/8085 #10-1 $8.50
6800 #12-8 $8.50

Programming for Logic Design

8080 Programming for Logic Design
6800 Programming for Logic Design
Z80 Programming for Logic Design

These books describe the meeting ground of programmers and logic designers; written for both, they provide detailed examples to illustrate effective usage of microprocessors in traditional digital applications. 300 pages each.

8080 #04-7 $8.50
6800 #05-5 $8.50
Z80 #11-X $8.50

Osborne & Associates, Inc.
P.O. Box 2036, dept. 114 (415) 548-2805
Berkeley, CA 94702 USA TWX 910-366-7277

NAME
ADDRESS
CITY
STATE ZIP PHONE

Please send the following information:
- ☐ Becoming an O&A dealer
- ☐ School discounts
- ☐ List of foreign distributors
- ☐ More information on O&A books
- ☐ Notice when General Ledger is available

Payment in advance must be enclosed for purchases of up to $70.00. Invoicing U.S. purchases over $70.00 available upon approval of your account. All foreign orders must be prepaid in U.S. dollars drawn on a U.S. bank.

BOOK # & TITLE	PRICE	QUANTITY	AMOUNT

SHIPPING (Shipping for large orders to be arranged)
Vol. 2 and Vol. 3 update subscriptions:
- ☐ All foreign orders $4.00 per 6-issue subscription for airmail
- ☐ No charge in the U.S. - 4th class mail ONLY

Books:
- ☐ All foreign orders $3.00 per book for airmail
- ☐ $0.35 per book 4th class (allow 2-3 weeks in the U.S.)
- ☐ $0.75 per book UPS in the U.S. (allow 10 days)
- ☐ $1.50 per book special rush shipment by air in the U.S.

California residents tax
shipping
TOTAL AMOUNT ENCLOSED

Calif. residents add 6% sales tax.
S.F. BART residents add 6-1/2 % sales tax.

F11

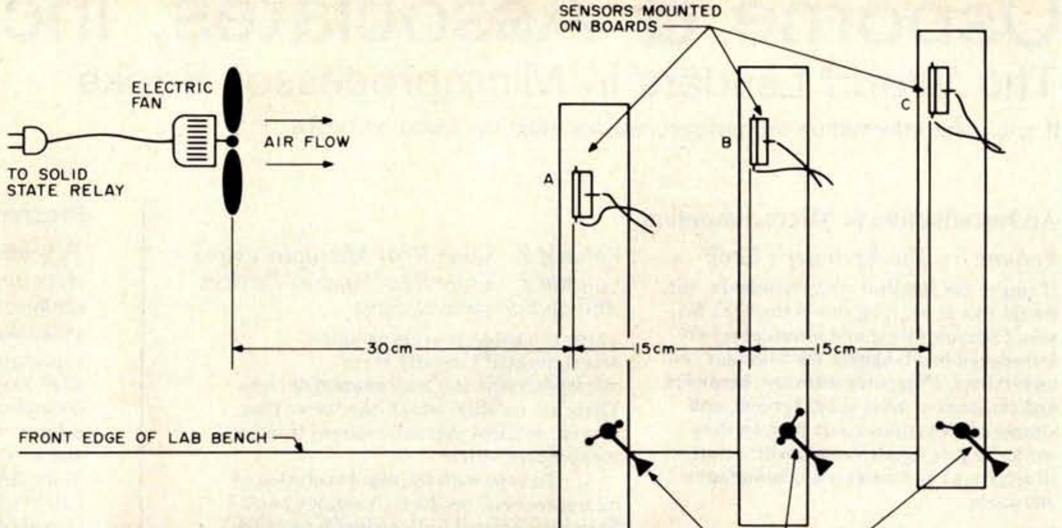


Figure 1: A microprocessor based control system. The goal of the experiment is to keep a constant rate of air flow from the fan shown at the left. Three binary switch sensors monitor the air flow. A microprocessor examines the state of the sensors and controls the fan by means of a solid state relay.

pole single throw switch which is sensitive to the air pressure felt by the movable element of the switch, an aluminum vane.

The fixed contact of the switch is a simple piece of #12 copper wire, which is sharpened to a point at one end, bent into an "L" shape, soldered to a stranded wire for connection to the computer circuit, and inserted into a hole drilled in the wooden mounting block. When the "L" is inserted in the mounting block, the point contact should face "upwind" toward the fan when the mounting block is clamped into its position in figure 1.

The wind sensitive moving contact of the switch has two parts. One part is a support frame, also made out of #12 copper wire, bent into a much larger "L" shape than was used for the fixed contact. As with the fixed contact, this support frame has an interconnection wire soldered to it before it is inserted into the wooden base of the switch. The second part of the wind sensitive contact is an aluminum vane measuring 3 cm wide by 15 cm long including a suspension hook bent in its end. The vane was fabricated from scrap sheet aluminum of about 1 mm thickness. When it hangs without any wind pressure, the equilibrium position of the vane should be about 1/2 cm away from the point contact. The mounting holes for the frame and point contact should be chosen with this in mind, although some adjustment is possible by bending the frame or point contact after assembly.

With this switch design, when the air from the fan produces a sufficient force, it displaces the bottom of the vane causing it to touch the wire point as shown by the dotted line in figure 2. This closes the switch. The three switches are used to define three binary input bits by wiring them to the circuit of figure 4 (which will be discussed in more detail later).

With this experiment's physical design, how can we obtain the desired behavior of an air flow sufficient to deflect switch A into a closed position, but insufficient to deflect switch C into a closed position, with switch B "hunting" a stable state? Basically, this is achieved by turning the motor of the fan on and off based on the sensed conditions of all three switches. Let's consider each switch in turn:

First, if switch A is open we definitely do not have enough air flow, since we want

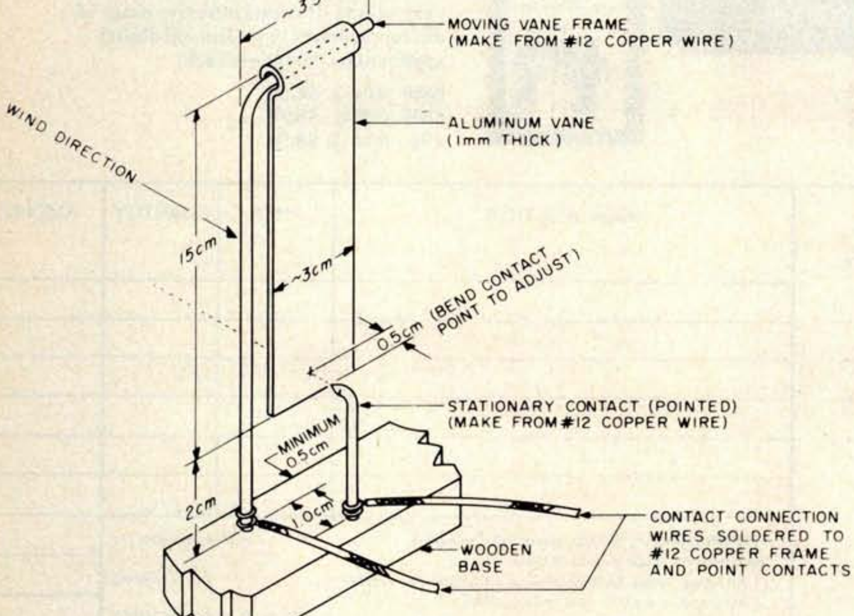


Figure 2: Detailed view of one of the three binary air flow sensors (typical for three).

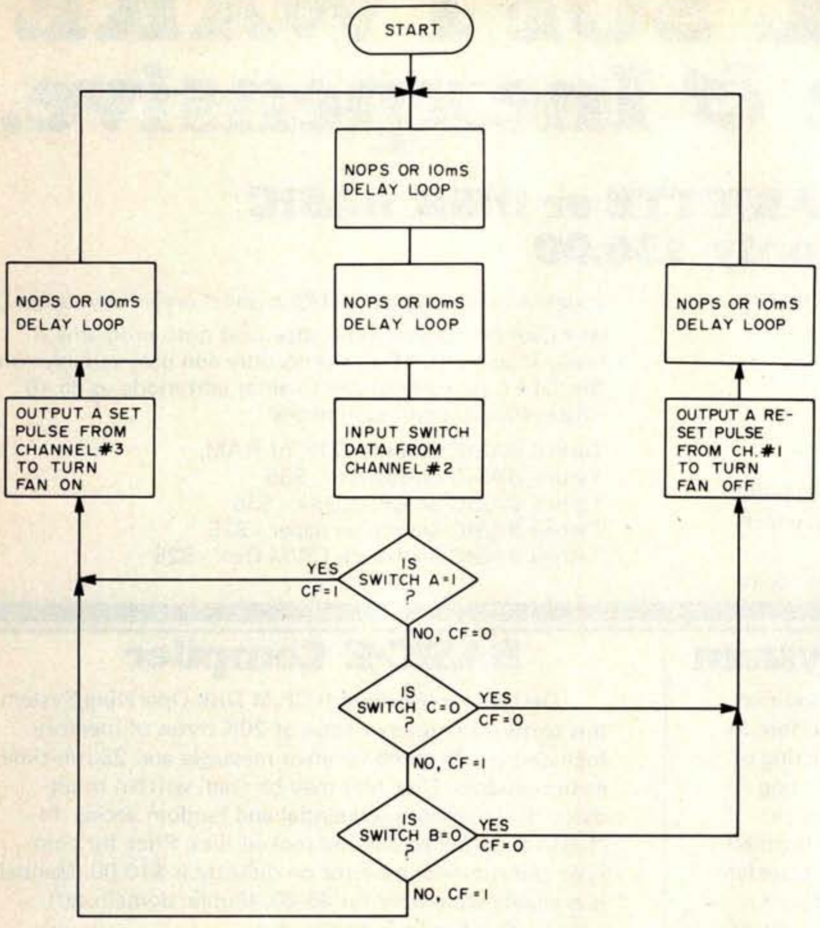


Figure 3: Flowchart of the air flow experiment.

switch A to be held closed all the time. Thus the motor must be forced on unconditionally if switch A is open.

But if switch A is closed, then we may have enough air flow already, but we might in fact have too much air flow. If switch C is then tested and found to be closed, we know that the air flow is too much, and must unconditionally turn off the fan motor.

If both switch A and switch C have normal "steady state" conditions of being closed and open respectively, then switch B is used to directly control the state of the fan motor. If switch B is open then its vane is not deflected enough, so the fan motor is turned on; if switch B is closed, then its vane is deflect enough so the motor is turned off. The inertia of the fan blades smooths out the sudden jerks of power to the motor, so that the effect of control exerted by switch B is to keep its vane just at the point of touching the contact.

This logic is shown in flowchart form as figure 3. The fineness of the control of the wind speed (and switch B's vane position) can be set by the amount of delay "wired" into the software of the flowchart. This flowchart implements a "dynamic" equilibrium situation through active control. The behavior of the system under different amounts of time delay is well worth investigating in the tutorial laboratory. How does the inertia of the fan blades (and consequent sluggish response to commands) affect the stability of the system? How does the amount of the delay in the control flowchart affect the stability of the system? Is it possible to use sensor A or sensor C alone to control the loop? Many, if not all, the experiments to be performed on this system can be implemented simply by changing the program running in the microprocessor. This ease of change (for a new program need only be typed into the entry keys of the processor) is one of the main reasons why inexpensive computers are now being widely used to control such systems.

A Look at the Details

The components used in the fan control system are shown in figure 4; they were breadboarded on the SK-10 socket of the Dyna-Micro 8080 based microcomputer. The 120 VAC fan was turned on or off by the Hamlin optically isolated solid state relay. All three sensor switches were wired with "pull up" resistors to +5 V. When the switch is closed, 0 V or a logic zero is input to a pin of IC2, the 8095 three state buffer, and when the switch is open, 5 V or a logic one is input to pin 2. The data from the 8095 buffer, IC2, is input to the accumulator via bits D_0, D_1, and D_2 on the data bus of the computer when a negative 1.333 μs device select pulse strobes pin 15 on the 8095 chip. This pulse occurs as a result of addressing IO port 2 with an IN instruction in the program. The 74154, IC1, serves as a 4 to 16 line decoder which directs device-coded control signals to the 8095 buffer or the 7476 flip flop chip, depending on which 8080 port was selected. A set pulse is applied to the 7476 flip flop, IC3, to turn the fan on. The flip flop serves as a memory to keep the fan on until the next bit control signal is passed to it. The fan is turned off by a negative-going pulse applied to the clear input of the 7476. The $\overline{\text{IN}}$ and $\overline{\text{OUT}}$ signals coming from the 8080A microprocessor are connected to the 7408 AND gate, IC6, and form a negative logic OR gate for the 74154 by DeMorgan's theorem: $\overline{A}$ & $\overline{B}$ = $\overline{A + B}$, $\overline{A \text{ or } B}$ = $\overline{A}$ & $\overline{B}$.

The 8080A microprocessor is capable

An INFO 2000 DISK SYSTEM gives you a lot less than you expected:

Less Cost

The DISCOMEM Controller board costs us less to manufacture. So your complete INFO 2000 Disk System costs you less – at least $400 less than comperable disk systems.

Less Hardware

Only three S-100 boards are needed to create a complete, high-performance disk-based microcomputer system–the DISCOMEM Controller Board, a 32K memory board, and any 8080, 8085 or Z80 CPU board. You don't need extra interface or EPROM boards since DISCOMEM contains 2 serial ports, 3 parallel ports and provision for 7K of EPROM and 1K of RAM.

LessTime

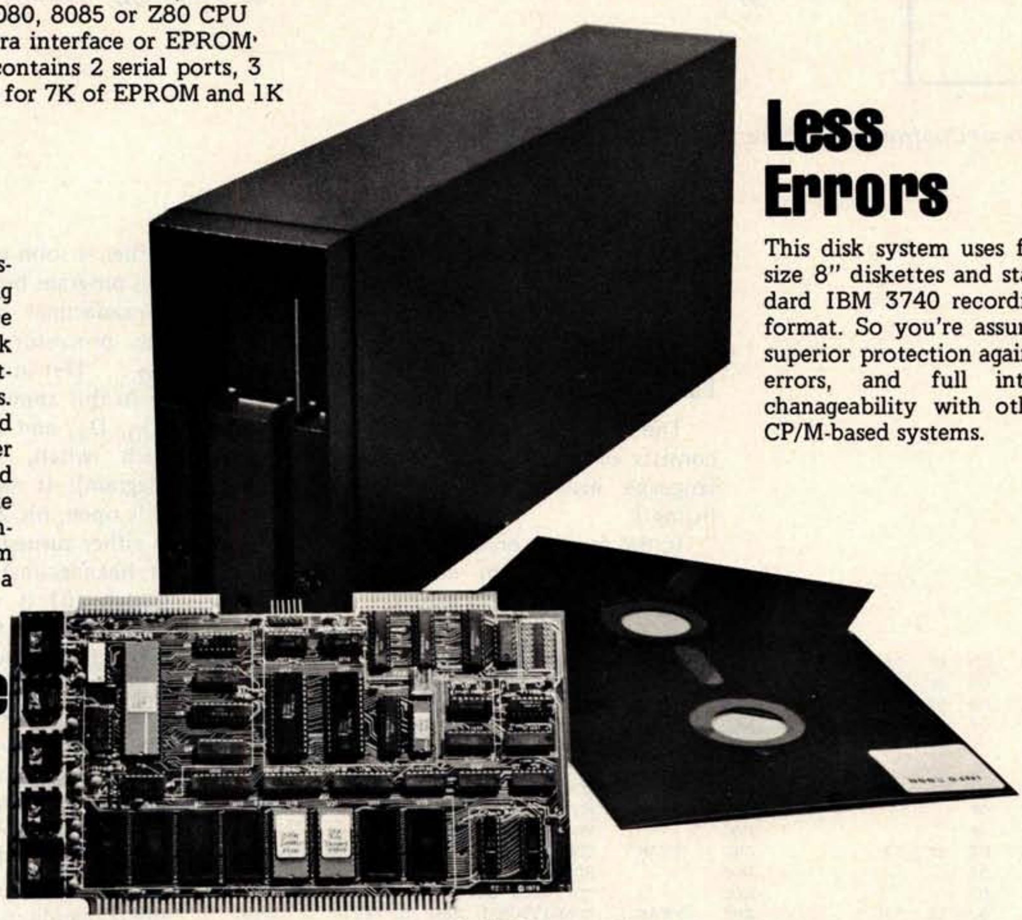
The INFO 2000 Disk System is incredibly fast! Using the PerSci Drives with voice coil positioning, disk seek times are up to 8 times faster than with other drives. A full disk-to-disk copy and verification takes well under a minute. Formatting and verifying a new diskette takes less than half a minute. Reloading CP/M* from diskette takes a fraction of a second.

Less Space

The system is remarkably compact, requiring only 1/2 to 1/3 the space taken by other 2-drive disk systems.

Less Hassle

The INFO 2000 Disk System eliminates the "I/O configuration blues" by incorporating all necessary interface ports. A CP/M Loader and all I/O drivers are contained in EPROM so there is no need for special software customization. Just plug the system into your S-100 microcomputer and begin immediate operation using the CP/M disk operating system. The INFO 2000 Disk System is supported by the most extensive library of software available, including 3 different BASICs, 2 ANSI FORTRAN IVs, several assemblers, text editors, debugging tools, utilities and numerous applications packages.



Less Errors

This disk system uses full size 8" diskettes and standard IBM 3740 recording format. So you're assured superior protection against errors, and full interchangeability with other CP/M-based systems.

It all adds up to more capability for your money.

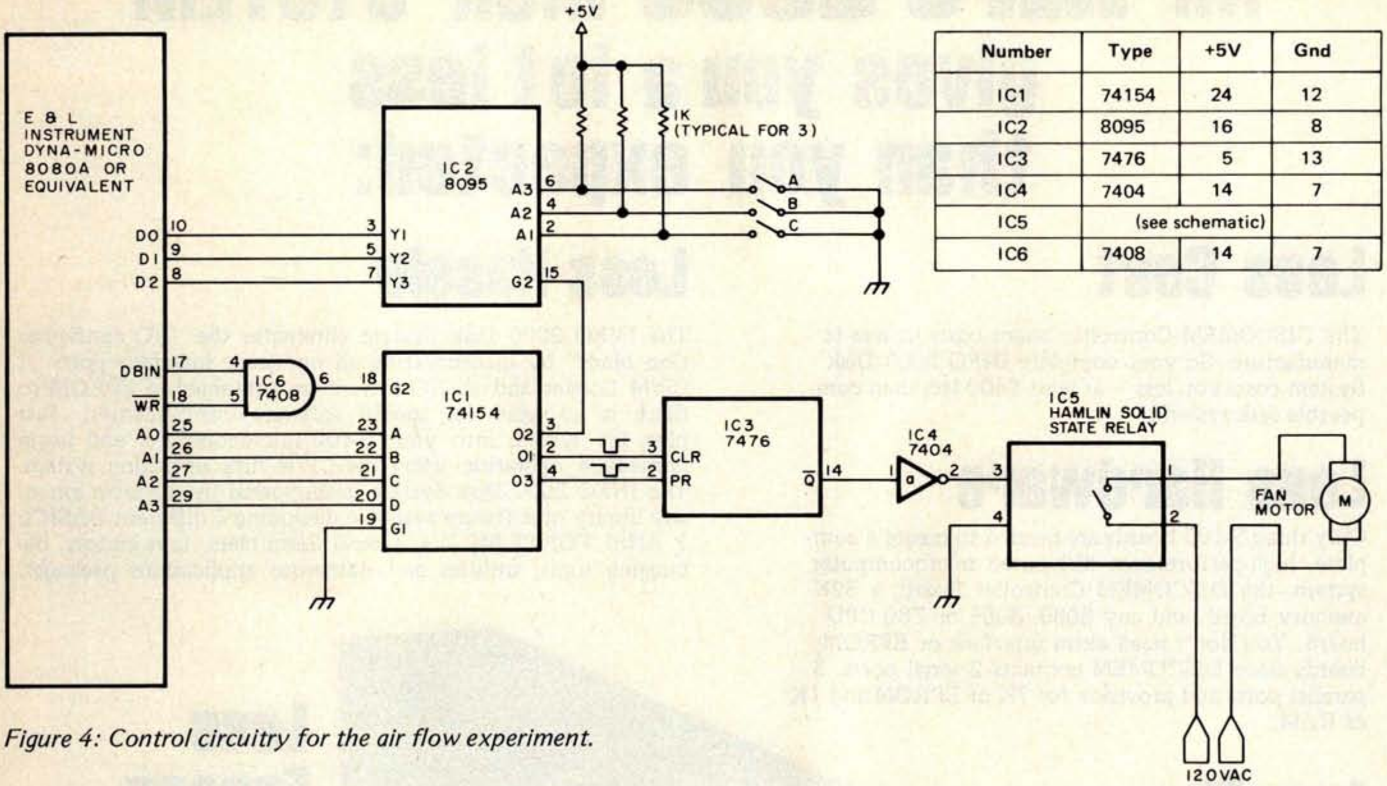
The complete INFO 2000 Disk System comes completely assembled and tested. It includes dual diskette drives, the DISCOMEM Controller, power supply, cabinet, cables and the CP/M disk operating system––everything you need for immediate plug-in-and-go operation with your microcomputer. This means less time, hassle, hardware, space, errors and less money than for comparable equipment. Now, isn't that a lot less than you'd expected in a dual disk system?

INFO 2000 Disk Systems are also available for Digital Group and Heath H8 microcomputers. Dealer inquiries welcomed.

*CP/M is a registered trademark of Digital Research.

INFO 2000
CORPORATION
20620 South Leapwood Avenue
Carson, California 90746
(213) 532-1702

Circle 176 on inquiry card.



Number	Type	+5V	Gnd
IC1	74154	24	12
IC2	8095	16	8
IC3	7476	5	13
IC4	7404	14	7
IC5	(see schematic)		
IC6	7408	14	7

Figure 4: Control circuitry for the air flow experiment.

of handling much more complex control systems than the example given here. In this control experiment, as in many simple laboratory tasks, control of a single bit (the fan motor state) is all that is necessary.

Fan Control Program

The program for controlling the fan consists of only the 38 bytes of machine language instruction which are given in listing 1.

If the fan can produce enough wind, the dynamic equilibrium condition described earlier is soon established after execution of the program begins. When the IN instruction (hexadecimal address 0308) is executed, the processor reads eight bits (D_0, D_1, D_2 . . . D_7) into the accumulator from port 2. In this application, only three bits of data (D_0, D_1, and D_2) are input (one bit from each switch, as shown in the schematic diagram). If switch A is closed and switch C is open, bit 2 (D_1) is tested; then the fan is either turned on with the OUT instruction at hexadecimal address 0316 if the middle switch (B) is open, or the fan is turned off with the OUT instruction at address 031E if this switch (B). is closed. A jump (JMP) back to the beginning of the program after each of these OUT instructions completes the loop. Any or all of the four 10 ms delay loops may be removed if desired by placing three NOP instructions (hexadecimal 00) in place of the CALL (hexadecimal CD BF 00) instructions. (The 10 ms delay loop is a subroutine built into the Dyna-Micro Designer.)

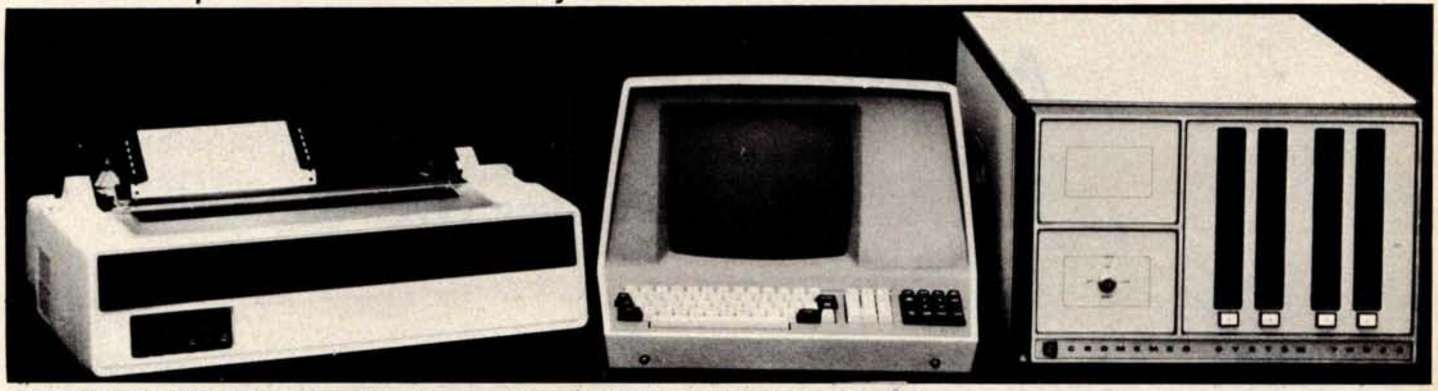
Much can be done with similar simple demonstrations of active control using microprocessor based computers and suitable inexpensive sensors like those used in this experiment. I hope that this simple control system experiment will serve to inspire teachers of science and engineering subjects, providing them with some practical examples to illustrate how computers can be used as tools in the laboratory.■

```
0300  CD  BF  00   NEXTI   CALL  DELAY   CALL SUBROUTINE DELAY IN KEX
0303  CD  BF  00           CALL  DELAY   FOR 20 MS DELAY.
0306  DB  02               IN    2       INPUT SWITCH DATA ON CHANNEL 2.
0308  0F                   RRC           ROTATE ACCUMULATOR RIGHT ONE BIT;
                   *                     TEST FOR A SWITCH DATA, LSB - MSB
                   *                     AND CF.
0309  DA  16  03           JC    ONFAN   CONDITIONAL JUMP TO ONFAN IF CF=1.
030C  0F                   RRC           ROTATE ACCUMULATOR RIGHT TWO BITS
030D  0F                   RRC           TO TEST SWITCH B DATA.
030E  D2  1E  03           JNC   OFFAN   CONDITIONAL JUMP TO OFFAN IF CF=0.
0311  07                   RLC           ROTATE ACCUMULATOR LEFT TWO BITS
0312  07                   RLC           TO CHECK SWITCH B DATA.
0313  D2  1E  03           JNC   OFFAN   CONDITIONAL JUMP TO OFFAN IF CF=0.
0316  D3  03       ONFAN   OUT   3       INPUT TO TURN FAN ON.
0318  CD  BF  00           CALL  DELAY   CALL SUBROUTINE DELAY IN KEX
                   *                     FOR 10 MS DELAY.
031B  C3  00  03           JMP   NEXTI   UNCONDITIONAL JUMP TO INPUT NEXT
                   *                     SWITCH DATA.
031E  D3  01       OFFAN   OUT   1       OUTPUT TO RESET OR TURN FAN OFF.
0320  CD  BF  00           CALL  DELAY   CALL SUBROUTINE DELAY IN KEX
                   *                     FOR 10 MS DELAY
0323  C3  00  03           JMP   NEXTI   UNCONDITIONAL JUMP TO INPUT NEXT
                   *                     SWITCH DATA.
```

Listing 1: An 8080A assembly language program for controlling the fan circuit in figure 4 (see also figure 1).

Boards . . . or Complete Systems

The best price and delivery is from MiniMicroMart!!!



CROMEMCO MODEL 3703 LINE PRINTER

180 cps bi-directional

LIST $2995 OUR CASH PRICE **$2545**

CROMEMCO MODEL 3100 CRT TERMINAL

List $1595 OUR CASH PRICE **$1350**

CROMEMCO SYSTEM 3

List $5990 OUR CASH PRICE **$4999**

Features 4 MHz CPU, 32K of RAM, dual PerSci floppy disk drive (and provision for installing two additional drives), RS232C Interface, Printer Interface; assembled and tested, ready to use. (Order as 02-5503-0)

CROMEMCO SYSTEM 2 (order as 02-5502-0) List $3990 . **$3390**

CROMEMCO DISK SOFTWARE — BASIC, Fortran, Assembler, Cobol, Word Processing System, Data Base Management, all complete with CMOS disk operating system, List $95 each . . . . **$80** ea.

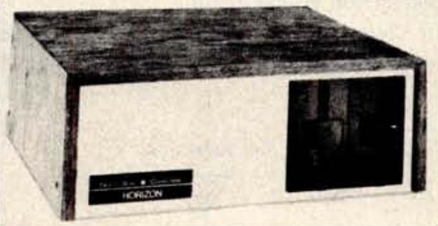
	CASH PRICE
Z-2 Computer System, Kit for rack mounting, Z-80 processor, 21 slots, power supply, front cover panel List $595 (order as 02-5301-0)	$499
Assembled and Tested, incl. fan and all edge connectors List $995 (order as 02-5401-0) List $995	845
Z-2D Disk Computer System, Kit Similar to Z-2, but comes with floppy disk controller, DOS, and minifloppy disk drive. A complete system with the addition of a RAM Board (02-5302-0) List $1495	1270
Assembled/Tested (02-5402-0) List $2095	1780

CROMEMCO BOARDS

4 MHz Single Card Computer, Kit (order 02-3511-0) List $395	335
Assembled/Tested (02-4511-0) $450 .	382
16K RAM Card with Bank Select, Kit (order 02-3216-0) List $495	399
Assembled/Tested (02-4216-0) $795 . .	499
Bytesaver PROM Board and PROM Programmer, Kit without PROM (order as 02-3308-0) List $145	123
Assembled/Tested (02-4308-0) List $245	208
TU-ART Digital Interface, Kit (order as 02-3440-0) List $195	165
Assembled/Tested (02-4440-0) $295 . .	250
Disk Controller Card, Kit (order as 02-3701-0 List $395	335
Assembled/Tested (02-4701-0) $595 .	515
TV Dazzler, Kit (02-3501-0) List $215	182
Assembled/Tested (02-4501-0) $350 .	297

Multiple User Basic Now Available!

NORTH STAR



NOW DOUBLE DENSITY FLOPPY DISK — NO INCREASE IN PRICE

	CASH PRICE
Complete Minifloppy Disk System w/BASIC and drive, Kit, order 01-7735-0 List $699	$589
Assembled/Tested (01-7745-0) $799 . .	689
Horizon 1 Kit List $1599	$1349
Assembled/Tested List $1899	1599
Horizon 2 (w/2 drives) Kit, List $1999 .	1699
Assembled/Tested List $2349	1939
4 MHz Z-80 CPU Board, Kit $199 . .	$169
Assembled/Tested List $259	209

TERMINALS

Hazeltine 1500	$1049
Soroc 120	849
Lear Siegler ADM3A	849
Intertube®	784

MEMORY BOARDS

North Star RAM-16-A 16K Dynamic RAM Board, Kit, (order 01-3216-0) List $399 .	329
Assembled/Tested (01-4216-0) $459 . . .	379
Morrow 16K Static Board, 450ns Kit, (order 08-3216-0) List $299	264
Dynabyte New MSC1625, 250ns (no write protect) Order 03-4216-3	359
Dynabyte 1625 16K Static RAM Module, 250ns (order 03-4216-0) List $555	444
Dynabyte 3225 32K Static RAM Module 250ns (order 03-4232-0) List $995	796

All Dynabyte assembled and tested, guaranteed one year.

MICROPOLIS

1041 MacroFloppy® in enclosure (order as 04-7701-0) List $695	$625*
1042 MacroFloppy® w/case and AC power supply (04-7702-0) $795 . . .	709
1053 Dual MetaFloppy® (order as 04-7705-0) List $1895	1695

* power source and regulator board required.

TELETYPE 33, 35, 43

Bargains on used Teletype 33's and 35's. Most models of new Teletype 43's available for immediate shipment.

IMSAI

Check with Mini Micro Mart for your IMSAI needs — most popular items now in stock at special savings.

WRITE FOR FREE CATALOG

MiniMicroMart, Inc.

1618 James Street, Syracuse, NY 13203 (315) 422-4467

FOR SHIPPING, add $2 for boards, $5 for floppy disk systems, $10 for Horizons, $15 for Cromemco Z-2 and Z-2D kits. Assembled Cromemco systems are shipped freight collect.

All prices are based on cash purchases. Higher prices apply to credit card and institutional purchase orders. Minimum deposit 25% on COD orders. All prices subject to change without notice.

Circle 250 on inquiry card.

A Multiuser Data Network

Communicating Over VHF Radio

Robert E Bruninga
907 Ninovan Rd
Vienna VA 22180

One of the more exasperating aspects of getting started in microcomputing is the cost of suitable IO devices. There is a rather large group of amateurs who already have terminals up and running; however, they have no microcomputer to play with. Our local computer club was such a group, having formed from a group of radio amateurs already connected via a VHF radio informal teletypewriter network. The availability of these inexpensive surplus Baudot code teletypewriters has been frequently espoused in the literature, and articles covering ASCII to Baudot conversion are plentiful.

While our club matured, and with interest in microcomputing mushrooming at the same time, the idea to design a microcomputer into the network was a natural step. It was soon apparent that developing hardware and software to provide all of the desired functions was impossible for any gainfully employed individual. To spread out the effort and to get as much participation as possible, it was decided to first bring up the microcomputer with a simple monitor system to enable anyone on the net to have full access to memory, and to help in writing software for any desired routines or pet projects. It is this microprocessor system which is to be described herein. But first we will give a description of the teletypewriter net and of the usual operating practices govern-

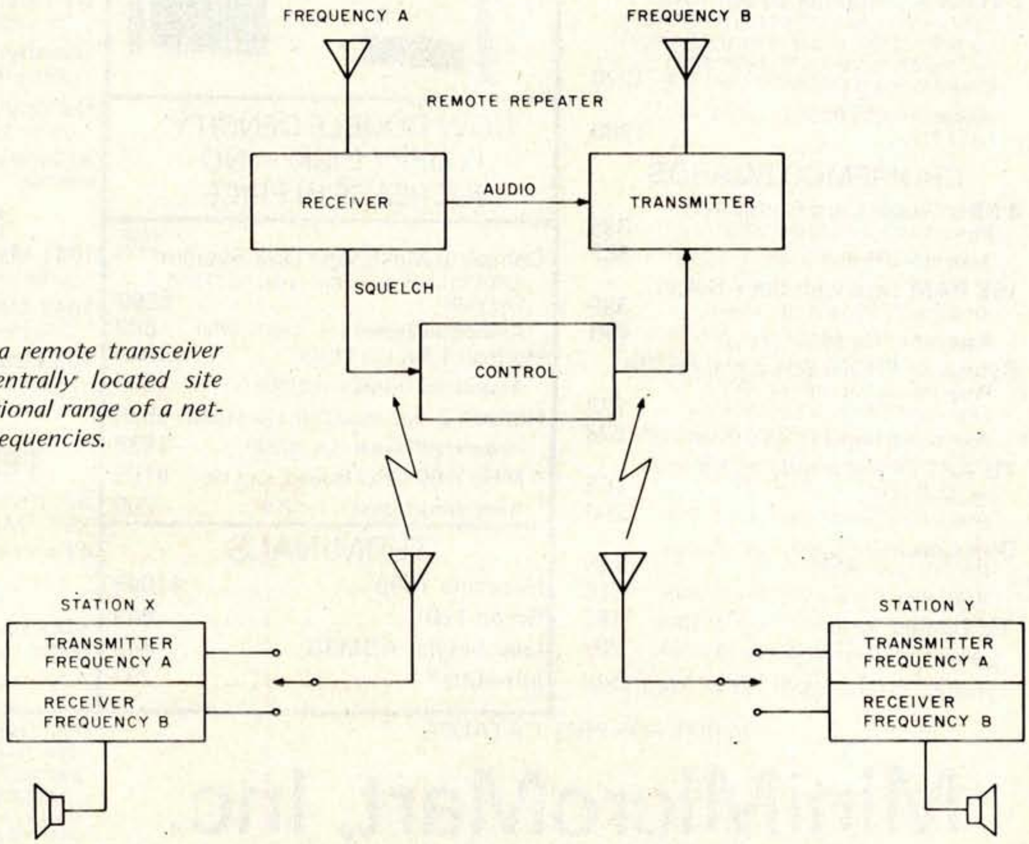


Figure 1: The use of a remote transceiver at an advantageous centrally located site greatly increases operational range of a network of users at VHF frequencies.

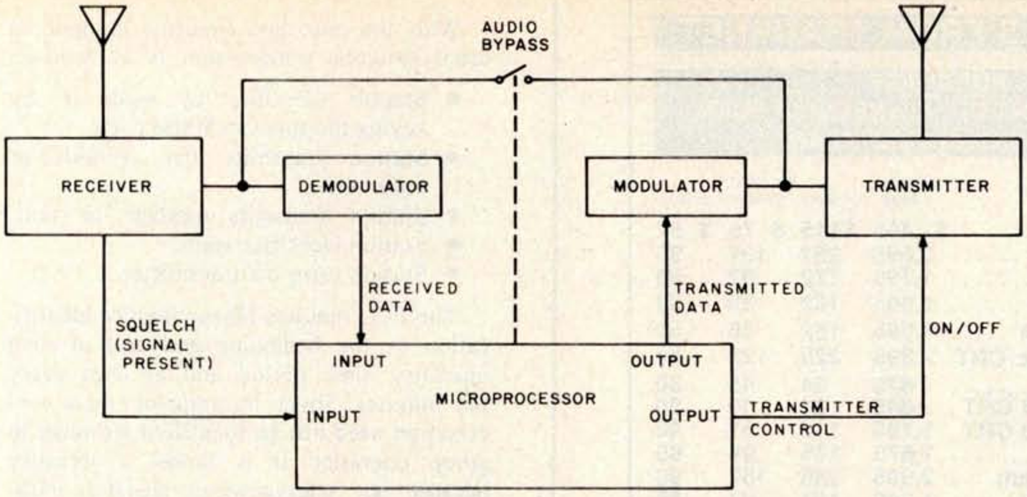


Figure 2: Having the microcomputer at the repeater site has the advantage of full duplex operation if user stations overcome the complexity of full duplex operation at their own stations.

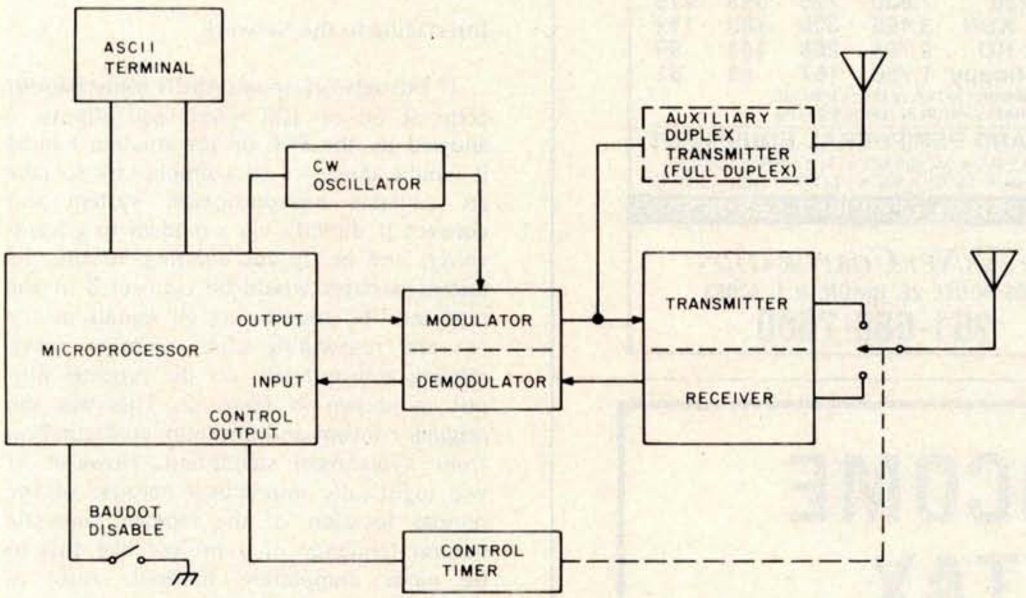


Figure 3: Having the microcomputer at a user station requires half duplex operation, but allows network expansion to include multiprocessors at any user station without conflict.

ing the participation of microcomputers.

Repeater Teletypewriter Network

The use of a remote transceiver or repeater, centrally located at the highest elevation possible, is the key to network coverage of a large area. The repeater relays signals omnidirectionally from any transmitting station to all receiving locations, as shown in figure 1. In addition to improving the range, it assures reliability of coverage to all users in the net since each station need only transmit a powerful enough signal to be clear at the repeater, and be able to receive the output of the repeater without error. Depending on terrain, reliable coverage of such a repeater is from 20 to 75 miles (32 to 120 km) in radius.

The repeater operation is fully automatic and almost transparent to the user. The presence of a signal on the repeater's receiving frequency turns on the repeater transmitter to relay the signals with the advantage of its high power and advantageous location. If no one is transmitting within five seconds of the last signal, the repeater transmitter will turn off.

Teletypewriter signals are transmitted using audio frequency shift keying (AFSK) with 2125 Hz indicating a mark and 2295 Hz indicating space. Since teletypewriter traffic is innately hard copy and is usually not of a real time nature, many network users have their stations configured for fully automatic unattended reception. This feature, called *autostart*, is simply a tone detector which latches a relay on receipt of a continuous mark tone and turns on the teletypewriter equipment. Usual practice requires from three to five seconds of mark tone to activate the autostart; it will then remain on for up to 30 seconds after the last tones are heard.

With the autostart circuitry in mind, a usual network transmission is as follows:

- Station identifies by voice or by keying the tones in Morse code.
- Station transmits five seconds of mark tone.
- Station transmits message or text.
- Station identifies again.
- Station turns off transmitter.

The FCC requires Morse or voice identification at the beginning and end of each operating time period and at least every ten minutes. Short transmissions in a conversation need not be identified although in group operation it is almost a necessity because one teletypewriter signal is indistinguishable from another.

Interfacing to the Network

If the network spoke ASCII (only Baudot code at 60 or 100 words per minute is allowed by the FCC on the amateur bands) it would appear to be a simple task to take an available microprocessor system and connect it directly via a modem to a transceiver, and be up and running. Ideally the microcomputer would be connected at the repeater site, monitoring all signals in the net and responding when asked by transmitting appropriately on the repeater output as shown in figure 2. This was the original concept and optimum configuration from a technical standpoint. However, it was logistically impractical because of the remote location of the repeater and the natural tendency of a project like this to be never completely finished. Also, as microprocessors become more and more of a common fixture about the house, we felt that a centrally located computer serving the whole network alone would be an obsolete idea in the near future.

For this reason, the microcomputer began to evolve as just another station on the net, transmitting and receiving in a half duplex mode as shown in figure 3. At present it is the only microcomputer that can be remotely accessed and controlled by all users on the net, although there are a few other microcomputers being used by stations for their own local functions. This type of operation allows the network to grow into a fully coordinated common data network as more and more users interface their microcomputers into the system and provide for remote access in whatever fashion their particular software packages allow.

The microcomputer monitors all signals on the repeater and responds whenever its call sign is recognized. To isolate the microcomputer from the identity of the owner or operator, a convention of placing a slant bar (/) prior to the call sign has been instituted to identify the microcomputer. In this manner, users in the network may communicate with the operator or talk about him using his call sign without accidentally initiating a response.

Remote operation of the microcomputer is very simple. To gain access from any station on the network, just transmit the microcomputer's call sign, WD4IWG/WB-4APR (these are the combined call signs of the computer club and the station at which the microcomputer is connected) and wait for the computer's response. A flowchart of the standby and ready states is shown in figure 4. Once the microcomputer recognizes its call sign, it comes up with five seconds of mark tone and transmits a prompt (–). After the prompt it goes into the receive mode and waits for a valid command sequence. After each command sequence is executed, the microcomputer returns with the prompt sequence and waits for more commands. If the sign off command, NNNN, is executed, the microcomputer becomes inactive, responding to nothing until it again recognizes its call sign.

The monitor is designed to interpret the input commands on a character-by-character basis as shown in the flowchart of figure 4. In this way, commands are not limited in length or structure as they would be if the processor did not begin interpretation or execution until an entire command and all parameters had been entered into the buffer. Once the single letter command has been interpreted, the software can help in the formatting of the remaining command parameters by providing spaces, line feeds or additional prompting as necessary. A space is used as the delimiter of command parameters, and a carriage return will abort the command if required. The judicious use of carriage returns is due to the large number of low budget hobbyists on autostart who must buy their own paper.

After each command is executed, control returns to the monitor entry point where the transmitter is turned off and the microcomputer waits for the next command input. All invalid command characters are ignored. Originally, only a valid command character was required to initiate execution, but this was prone to occasional errors. The program was changed to require a /$ sequence (the Baudot equivalent of the ASCII escape character), a single command character, and then a space before the command is recognized. Failure to provide all four conditions causes a return to the wait state for the proper escape sequence. The escape character is just one of 15 additional

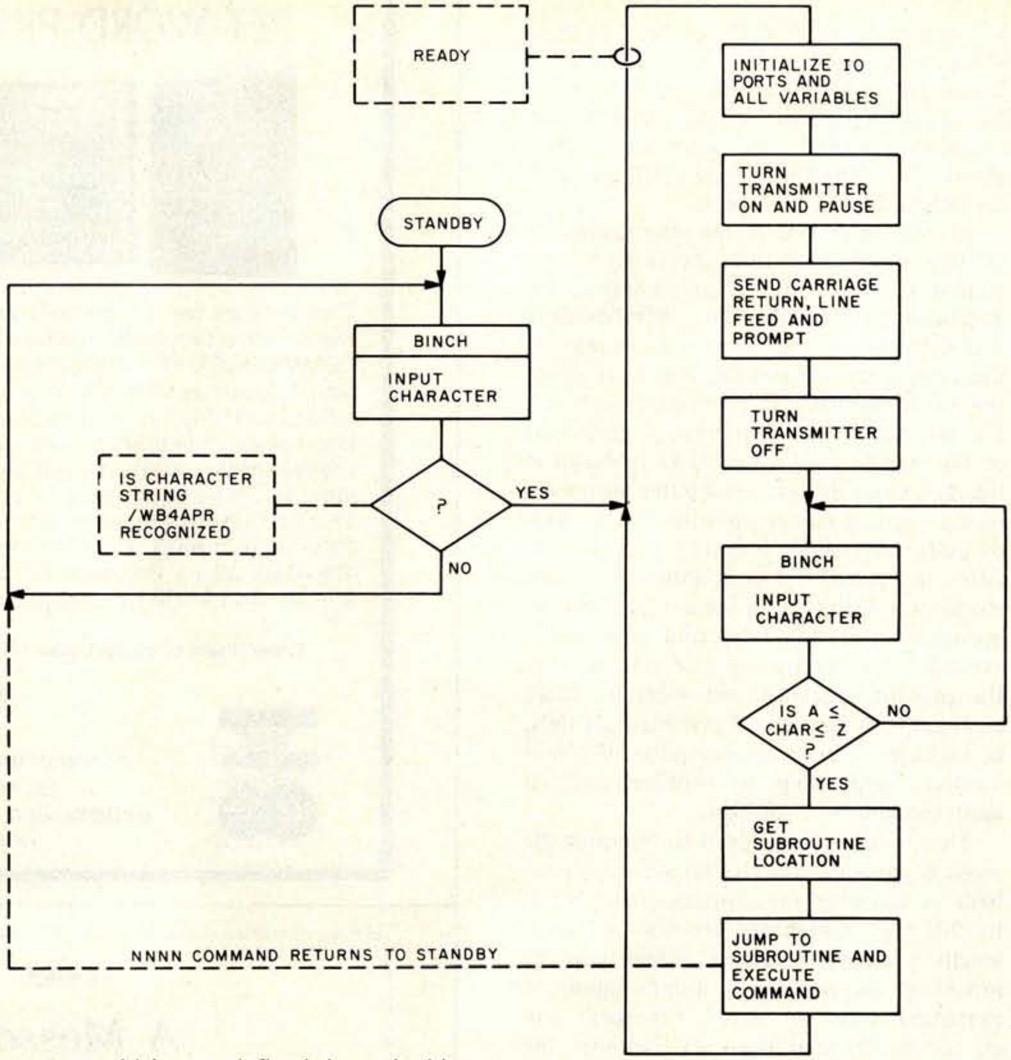


Figure 4: Flowchart of the system monitor showing the standby and ready states.

characters which are defined in a double shift case.

The structure of the character input (BINCH) and character output (BOUCH) subroutines allows simultaneous output in both ASCII and Baudot of any character available at the Baudot or ASCII input. The structure of BINCH and BOUCH is shown in figure 5. Because all routines written for the network system use these two subroutines for input and output, the remote user at the Baudot port has the same software relationship to the computer as the local programmer except that the output may be disabled.

Control Routines

The monitor commands are relatively straightforward except for the requirement that the microcomputer operate in a half duplex mode. The subroutine responsible for coordinating the microcomputer response in the half duplex mode is PLEX (figure 5). PLEX turns the transmitter and transmission tones on and waits the delay time found in location ECHO before returning to the calling routine. PLEX is called by every monitor routine which generates output back to the remote user. The delay in ECHO is roughly in tenths of a second; usually, a value of 50 is used to allow reliable transmit and receive switching. The value 01 implies immediate response, and the duplex transmitter line is therefore turned on as well. ECHO can be changed by the monitor command E, allowing the user to tailor the response to a particular hardware configuration. To allow the user to regain control when an error occurs, the transmit line drives a timer that cuts off the transmitter after two minutes. A second timer will hardware reset the microprocessor if a space tone of greater than two seconds is received.

Command

The monitor commands in table 1 are designed to provide as much assistance to the machine language programmer as possible while also recognizing the special needs of the users on the network. The basic

What's Your A.I.Q.?*

Learn About Artificial Intelligence & Robots

INTRODUCTION TO ARTIFICIAL INTELLIGENCE by Philip C Jackson Jr.
☐ This book surveys the field beginning with Turing's test, the mathematical description of phenomena, finite state machines and limits to computational ability, followed by chapters on problem solving, game playing, pattern perception, theorem proving, semantic information processing, parallel processing, evolutionary systems, robots, and a look at the future of the field. This thoughtful and unusual book will make a useful addition to your library. A 50-page bibliography is included. 453 pp. $18.50 hardcover (no photo).

ARTIFICIAL INTELLIGENCE by Patrick Henry Winston.
☐ This book is designed for use in a course on artificial intelligence, but should prove invaluable to the newcomer to the topic, as well as being useful to the experienced as a reference text. Part One is set up to be used as an introduction to the key ideas of the subject, including the basics of elementary representation, symbolic constraint exploitation, search, and control. In addition, understanding natural language, representing common sense meaning, computer vision, and expert problem solving are covered. Part Two covers the programming language LISP, its organization and uses. Profusely illustrated. Only $10.00 hardcover. 444 pp.

ADAPTIVE INFORMATION PROCESSING, AN INTRODUCTORY SURVEY by Jeffrey R Sampson.
☐ Even the most complex of today's computers is dwarfed by the capabilities of its designer's brain. Imitation of the capabilities and adaptive nature of biological information processing systems has long been a goal of computer designers. Jeffrey R Sampson presents an introductory survey of this field. From the fundamentals of information and information processing entities, he examines the richness of biological information processing and goes on to cover the application of computers to artificial intelligence. His is a view which unifies the diverse aspects of the subject of artificial intelligence, emphasizing adaptive behavior. 214 pp. $14.80 hardcover.

COMPUTER MODELS OF THOUGHT AND LANGUAGE edited by Roger C Schank & Kenneth Mark Colby.
☐ Part of the current work in AI involves the modeling of human mental processes in computer programs. Here is a collection of articles, which deal with natural language models, models of belief systems, and models of memory and cognition. Required reading for AI enthusiasts. 450 pp. $23.50 hardcover.

HOW TO BUILD A COMPUTER-CONTROLLED ROBOT by Tod Loofbourrow.
☐ This book combines the dream of robotics—to create an intelligence other than human—with the reality, by providing both hands-on experience with robotics and an application of a microprocessor. It details the step-by-step directions for building a robot, named "Mike," controlled by a KIM-1 microprocessor, with the complete control programs clearly written out. 132 pp. $7.95.

ROBOTS ON YOUR DOORSTEP by Nels Winkless and Iben Browning.
☐ "This book will amaze you, frighten you, nauseate you, excite you. . .it will probably make you think about things you have never contemplated before," states the introduction to this clever and well-written account of robots: past, present, and future. Intelligence, artificial intelligence, brain structure and simulation, and characteristics of robots are only a few of the areas explored. 179 pp. $6.95.

BUILD YOUR OWN WORKING ROBOT by David L Heiserman.
☐ This book will introduce you to the problems of making a robot mobile device called Buster III, using pre-microprocessor TTL integrated circuits for all logic functions. It is a must book for background reading, but much of the logic can be extremely simplified using today's microprocessor technology. Use this book as a first look at these problems from which you can build further and more elaborate solutions. $5.95.

THE THINKING COMPUTER: MIND INSIDE MATTER by Bertram Raphael.
☐ With a minimum of technical jargon, this book discusses the capabilities of modern digital computers and how they are being used in contemporary AI research. It discusses the progress of AI, the goals, and the variety of current approaches to making the computer more intelligent. $6.95.

COMPUTER POWER AND HUMAN REASON by Joseph Weizenbaum.
☐ If you're presently a programmer by trade or skill, you'll see a philosophy of computer use and abuse propounded. If you're a novice, the tutorial and explanatory chapters of this book serve as an excellent and eminently readable background source. This includes an outstanding, low level explanation of what an algorithm is, and how computers execute effective algorithms. $5.95.

BITS™ inc Books to erase the impossible
POB 428, 25 Route 101 West, Peterborough NH, 03458

For your convenience in ordering, please use this page plus the order form on page 111. You may photocopy this page.
DIAL YOUR CHARGE CARD ORDERS TOLL FREE (800) 258-5477.

**Artificial Intelligence Quotient*

Table 1: Table of commands used by the computer on the communications network. All addresses are given in hexadecimal notation.

A XXXX *string*	Stores an ASCII character string at location XXXX beginning and ending with quotes.
B XXXX YYYY	Calculate branch instruction offset from address XXXX to address YYYY.
C XXXX	Sends character string at address XXXX in Morse code.
D XXXX YYYY	Display hexadecimal bytes from address XXXX to YYYY. If E command is odd, display 16 bytes per line, 32 bytes per line if even.
E XX	Set echo delay to X.X seconds (approximate).
F XXXX YYYY *string*	Find *string* by searching from address XXXX to YYYY. Reply with starting and ending address.
G XXXX	Go to address XXXX and begin execution of a subroutine.
H XXXX	Convert hexadecimal value XXXX to decimal.
I XXXX hh hh	Input hexadecimal bytes sequentially beginning at address XXXX and ending with a carriage return.
J XX	Jump to additional monitor commands.
K	Reserved.
L XXXX YYYY	List by instruction in 1, 2 or 3 byte format all code between addresses XXXX and YYYY.
M XXXX YYYY ZZZZ	Move all bytes between addresses XXXX and YYYY to contiguous memory starting at ZZZZ.
NNNN	Sign off procedure.
O	Return to AMI monitor.
P XXXX YYYY ZZZZ	Punch a hexadecimal tape in MIKBUG format located from XXXX to YYYY with offset ZZZZ.
Q *string*	Quest search through all memory for first occurrence of *string*. Back up and print starting address of first quotation mark, then print all following characters until a second quotation mark is found.
R XXXX	Read character string at address XXXX.
S *string*	Save. In message storage area, store *string* at next available address and update pointer to point to next available address. Print pointer value.
T XXXX YYYY ZZZZ	Tape. Load hexadecimal tape into memory between addresses XXXX and YYYY with offset ZZZZ.
U	Reserved.
V *string*	Void first string encountered which contains *string*. All characters between limiting quotes are eliminated including quotes. All memory from ending quote to pointer is moved up to location of the beginning quote. Pointer is updated and printed.
W XXXX *string*	Wedge. Insert *string* at location XXXX. Push memory from XXXX to pointer down and update and print pointer.
X	Transmit (Xmt) prompt.
Y	Reserved.
Z XXXX YYYY HH	Fill memory space from address XXXX to YYYY with hexadecimal value HH.

monitor commands for machine language programming are the DISPLAY, GO, INPUT, LIST, PUNCH and TAPE commands. Additional aids for programming are the ASCII STRING, BRANCH, FIND, MOVE, READ, WEDGE and ZERO commands. The monitor functions related directly to network applications are the CW, ECHO, NNNN and XMT commands. The special string manipulation commands SAVE, QUEST and VOID can be used with call signs as key words to form a message store and retrieval system for network operation; or line numbers may be used as key words, and text editing is possible. In text editing mode corrections, insertions and deletions are performed using ASCII STRING, FIND, READ and WEDGE commands.

The system described thus far is fully operational in an unattended mode. It has been more or less continuously available during the evening hours from about 1800 to 2300 hours local time since December 23 1977 (except for occasional visits by "Mr Murphy"). During this time the original monitor has not been altered, so users would not be confused by frequent system changes. However, several changes are being considered to improve system reliability in the presence of noise and user typographical errors, to expand monitor command functions and to upgrade protocols in anticipation of other computers on the network. The changes under consideration are:

- Provide prompting at sign on to allow uninformed users to request operating instructions.
- Provide a carrier present input line to allow the microcomputer to test for a clear channel prior to transmission.
- Rebuild restart circuitry for more reliability.
- Change all string manipulations to use quotes for delimiters. Currently, figure shift H is used, which causes some Baudot machines to stop.
- Add additional monitor commands to allow unaddressed message storage and retrieval by call signs alone.
- Add line delete code recognition into Baudot input routine to allow escape to command mode at any time.
- Consider buffering all input commands to allow for more flexibility and reliability by using line and character delete codes.
- Restructure BINCH and BOUCH routines for complete transparency. All ASCII codes needed for BASIC will have one-to-one mapping into Baudot characters by using the figures slash shift alphabet. The usual ASCII con-

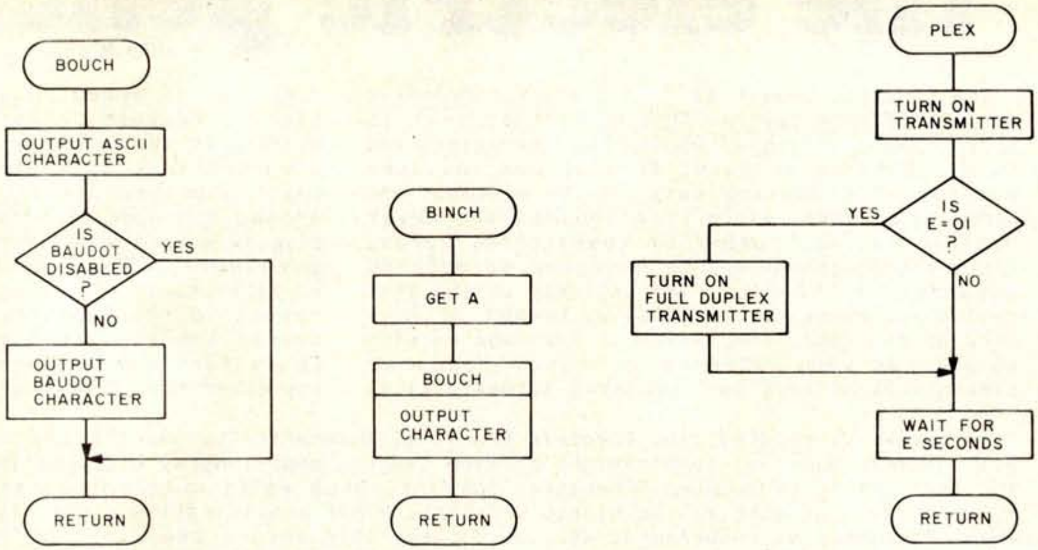


Figure 5: The character input and character output subroutines are BINCH and BOUCH. PLEX controls half duplex operation.

trol characters will be available to allow terminal control, escape sequences and other control functions.

ASCII Considerations

The most important change which is foreseen is the addition of an ASCII input/output port and modem at 300 bps to the system when ASCII transmission is finally permitted by the FCC. Although Baudot operation will continue, the evolution of computer-to-computer operation will be greatly simplified. It is anticipated that all network teletypewriter transmissions will remain at 60 words per minute Baudot due to the availability of the inexpensive terminals. Therefore the 2125 Hz mark and 2295 Hz space tones will still be reserved for Baudot use. All ASCII transmission will be at the commercial modem frequency pair of 1270 Hz mark and 1070 Hz space. The Baudot and ASCII transmissions will not interfere with one another due to the separation in frequencies.

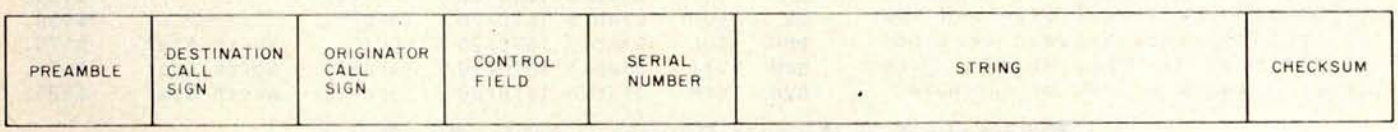
Due to the capture effect on FM transmissions in which the stronger of two signals at the receiver will capture the detector and the weaker station will not be heard, simultaneous transmission by two separate stations (one in ASCII and one in Baudot) is not possible. To overcome this difficulty, a secondary receiver will be placed at the repeater which has its audio output in parallel with the original receiver in driving the transmitter. This secondary receiver will allow simultaneous access to any two system users.

Because all stations will be monitoring the repeater output channel, simultaneous ASCII and Baudot transactions can take place between separate and independent users. Since the microcomputer operating system already communicates through both its ASCII and Baudot ports at different rates simultaneously, upgrading the system for ASCII is a simple matter. A second modem at the 1070/1270 Hz frequency pair is interfaced in parallel with the local terminal. With the present structure of routines BINCH and BOUCH, automatic translation between ASCII and Baudot will be accomplished. The only change to the monitor will be to allow remote access to the Baudot disable line so that full speed ASCII can be used when a Baudot user is not in the system. Additionally, various procedural subroutines will be made available to handle computer-to-computer protocols.

Protocol

The first protocol problem to consider is the separation of messages directed to a computer from messages directed to other users or to the operator. In this case all computers should allow a *hand-holding* mode in which the remote user can talk directly to the operator or to other stations on the net even while the computer may be expecting some input.

Figure 6: Protocol for transactions over the network.



"My Structured Systems business software has paid for itself in labor hours saved alone."

Mr. Ken Tunnah, Colloid-A-Tron Inc., Buffalo, New York

Ken Tunnah is one of many innovators bringing the micro revolution to the small business. As a programmer, he knows computers and their languages. As a businessman, he knows business and its languages. And when Mr. Tunnah decided to micro-computerize the accounting function at Colloid-A-Tron, he turned to Structured Systems software.

Says Mr. Tunnah: *"The program is designed from a CPA standpoint, for multiple corporations, which we have. It is flexible and gives me the ability to change reporting by profit centers easily. It is up and running quickly, and it just keeps on running. I think it's the best business software available."*

The best software available. That's what Structured Systems Group set out to create.

Structured Systems offers three sophisticated accounting systems. Our General Ledger software is big enough for multi-client write-up by the CPA, or multi-corporate reporting for the business, but small enough for the micro budget. The very comprehensive Accounts Receivable and Accounts Payable packages will operate independently, or they will coordinate with the General Ledger.

Our systems record transactions easily and correctly, and provide an audit trail from source document to financial statements. And they will maintain monthly and year-to-date information in dollars and in percentages. And they are reliable.

The three systems interact with the user to set up parameters such as format and headings, account titles and numbering, automatic billing or reminder notices, credit limits, sales reports, a check register, and much more.

The software is designed to run on an 8080 or Z-80 CPU with 48K of memory, dual disks with CP/M®, printer, keyboard, and CRT. To make it all work for you, we have provided the most extensive documentation and support in the industry.

We provide the capability to computerize complex accounting functions on relatively inexpensive micro-computer equipment. Ken Tunnah has told us what that means: *"I've bucked some trends. I looked around, and decided that with the right software, I could get a micro to outperform a $45,000 mini. I'm satisfied. It's simple economics."*

We can refer you to a growing number of sophisticated retailers experienced in Structured Systems Group business systems. Or we can work directly with you. We'd be happy to provide you with more information on our product line, which includes QSORT (a sort/merge program), CBASIC (a business BASIC), NAD (a mailing and addressing system), and our General Ledger, Accounts Receivable, and Accounts Payable packages.

®CP/M is a registered trademark of Digital Research

Structured Systems Group

5208 Claremont Avenue
Oakland, CA 94618
Call us at (415) 547-1567. Or write.

Circle 351 on inquiry card.

The exclamation point has been suggested as a command indicating that the following line is to be ignored. All characters following the exclamation point are ignored until a carriage return is encountered. In the case of the system just described, the operator whose keyboard is essentially in parallel with the remote user can make comments in return, again without disturbing the computer.

Once the microcomputer is signed on at present, it uses the communication channel as a private line assuming all commands are directed exclusively to it. Because other computers are interfaced to the net, this point of view will have to be modified to allow many processors to share the channel and respond only to commands addressed specifically to them.

The FCC-assigned call letters of all users on the net are considered to be a convenient addressing string for just this purpose. Call letters consist of from four to six alphanumeric characters. Using the call signs, a possible protocol for all transactions might be as shown in figure 6. All system messages would consist of a preamble, the call sign of the destination station, the call sign of the originating station, a control field, a serial number, the message or data string itself and a checksum.

The control field could identify the length and nature of the string; whether binary, hexadecimal or ASCII characters were in the message; whether a checksum was generated or necessary; and other data necessary to tell the receiving station how to handle the particular string. The preamble would be an appropriate series of characters to activate and initialize all monitoring stations. A suggested preamble has been a carriage return, line feed, five null characters and a line delete code. Optional call signs and other characters could precede the line delete code, because its main function is to establish the channel and prepare the microcomputer to recognize all following characters.

As more and more microcomputers join the net, another convenient feature would be the capability of polling the net to identify which stations are in a ready status. This problem of polling, a complex one due to the dynamic nature of the net, has a wide range of solutions depending on the particular structure assumed for the net. It is introduced here as a topic for further study.

One possibility of providing the polling capability is through the assignment of some central responsibility for all transactions on the net. Once a significant degree of sophistication is attained, it might be desirable to assign one microcomputer at any one time to be a control station. The computer would be responsible for maintaining an up-to-date status of all systems on the net and a directory of all available programs and data. All a user would have to do is query the control station to get a complete picture of the present net configuration. The function of the control station can be passed along from one station to another as new stations log on and other stations log off. The status data would be coded very efficiently for speed of transfer from one station to another.

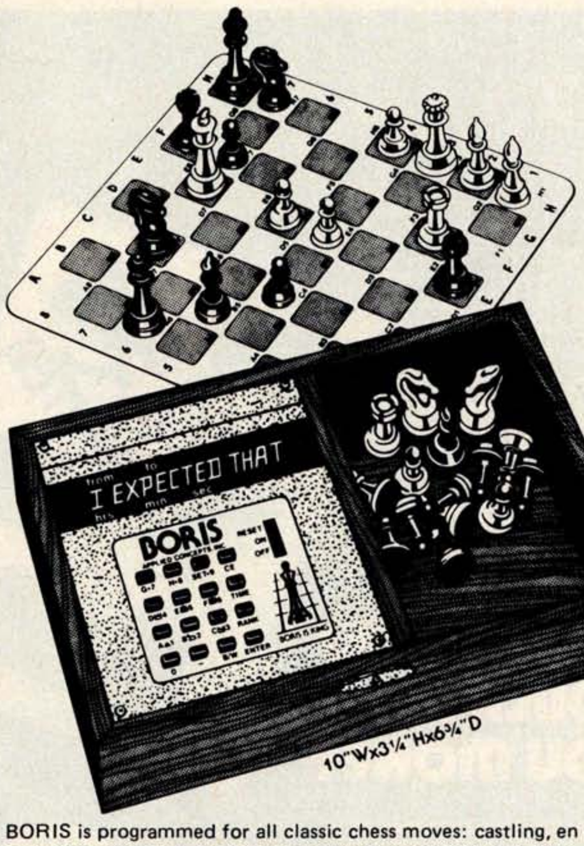
It has not been the intention of this article to make firm proposals on network protocol or operating procedures, because so little experience is available to date on the nature or expectations for a computer experimenter's network. Rather, it is hoped that sharing the little experience we have gained in interfacing the computer to an existing radio-teletypewriter network may be of use in further discussions of this goal.

We are at the birth of something great and dynamic. Writing this article has been difficult due to the daily increase of ideas and the conflicting desire to implement each idea as it occurs. We invite others to share their similar experiences and conjectures.■

MIRAGE

This could be the most intriguing, innovative conversation piece you've ever seen. It is composed of two concave mirrors; the top mirror contains a small opening in the middle of it. A small object placed at the bottom of the lower dish appears above the opening of the top mirror totally reconstructed in full 3-D. MIRAGE is a breakthrough in three-dimensional real imagery. It uses ambient light and can be viewed from any direction. The mirror coating (5/1,000,000th of an inch thick) is identical to that used to surface the 200-inch telescope at California's Mt. Polomar Observatory. MIRAGE combines art, illusion, science, and fun. Each true-to-life vision is actually a scientific phenomenon. Manufacturing tolerances are critical; the resulting imagery is alive with depth, surpassing holographs in clarity and resolution. Yet it's so simple it can fool anyone. $24.95 plus $1.50 insured UPS shipping and handling.

Full 3-Dimensional Illusion



The Classic Game of Chess Will Never Be the Same

BORIS is the world's most advanced portable chess computer. Match him against any other computer in his class, and he defeats them all. For the millions who play chess, plus those who want to learn the game, BORIS provides the challenge and flexibility that until now were found only in expensive laboratory computers. Like his big brothers, BORIS even talks to his opponents, with messages programmed into his almost-human brain.

BORIS was created to be the perfect chess opponent; always ready at a moment's notice, with a game and strategy that's unpredictable. If your chess skills are wasting away because you can't find an opponent, BORIS will be a challenging adversary, and there's never a fear of being embarrassed if you make a careless blunder. BORIS plays black or white; he'll even play himself. Set him to play at a beginner's level, or let him loose, and he'll play a game that will keep the attention of a Master.

BORIS is unmatched as a chess teacher. It's easy to find scores of books for beginners, but it's never as easy to find someone with the time and patience to teach a novice. BORIS knows every rule and move of the game, and has all the patience in the world. Even if you've never played the game before, you can learn from BORIS in less than an hour. If you're his student and unsure of your next move, simply ask BORIS to suggest the best one. When BORIS backs you into a corner, just change places with him, and see how he battles his way out of his own trap. You can alter his level of play at any time to match your advancement.

BORIS keeps you honest by displaying "ILLEGAL MOVE," if you attempt one. He concedes defeat with "CONGRATULATIONS," if you're good enough to beat him, even "STALEMATE," if you play him to a draw. His vocabulary includes a variety of phrases which will add flavor to your game. He can certainly unnerve you at any time with a comment like, "I EXPECTED THAT" or "IS THIS A TRAP?"

BORIS is programmed for all classic chess moves: castling, en passant, queening, etc.; plus, he solves all mate in two problems. Also, BORIS is programmed for random play, so you'll never get the same game twice. BORIS comes in a hand-crafted, solid walnut case, 10" x 3¼" x 6¾", with chess pieces and folding chess board. You can use your own board and pieces if you prefer.

Whether you're an experienced player, a beginner, or you need a special gift for a chess buff, BORIS is definitely for you. When you sit down to challenge BORIS, be prepared for the game of your life. You'll quickly see for yourself why BORIS IS KING. $299.95 plus $4.00 for insured UPS shipping and handling. Includes a 90-day limited warranty.

IF YOU ARE NOT TOTALLY SATISFIED WITH **MIRAGE** *OR* **BORIS**, *RETURN WITHIN 15 DAYS FOR A FULL REFUND.*

For your convenience in ordering, please use this page plus the order form on page 111.
You may photocopy this page.

DIAL YOUR CHARGE CARD ORDERS TOLL FREE (800)-258-5477.

BITS inc POB 428, 25 Route 101 West, Peterborough NH, 03458

BORIS

Book Reviews

Etudes for Programmers
by Charles Wetherell
Prentice-Hall, Englewood Cliffs NJ 1977
200 pages paperback
$12.95

My dictionary defines an etude as a composition for the practice of techniques (from the French *étude,* meaning study). In *Etudes for Programmers,* Mr Wetherell dwells on the craft of programming. A craftsman, in the guilds of Renaissance Europe, spent many years doing menial tasks, mastering one facet at a time. Mastery of computer science is also a long process. "The difficulty with teaching programming," the author states, "is that it cannot be taught." He continues, "the difficulty with learning programming is that it is so much work."

Today, *programmer* is a generic term which applies to practitioners of widely different skills. Many of us learn piecemeal, working with code someone gives us to write or complete. Thus, a "guiding hand" to round out our training is lacking. It is like trying to cut down a giant redwood with a coping saw.

This splendid new book is intended to aid moderately competent programmers in becoming master craftspeople. I stress the moderately competent, for these are not beginners' problems.

The text is designed for a graduate course in computer science. There are 27 projects offering exercises in all aspects of our craft. Automatic text formatting, puzzles and mazes, and cryptanalysis are just some of the offerings. Four of the projects, suitable for a compiler course, deal with such topics as loaders and large computer simulators. Each task is given a suggested manpower level, such as five weeks for one person or three weeks for two people. Although no particular language is used, suggestions are given in each assignment. The variety of applications will show the reader the power and the poverty of each language used. Thus students can broaden their familiarity with several languages.

Etudes for Programmers improves one's skills by developing good techniques. The problems are interesting and challenging. As their accumulated computer lore increases, readers develop a greater ability to write and document clear concise code. I feel this book will help any programmer of some experience to attain a more professional level of competency.

Each chapter has references to aid in the further pursuit of a given topic, and there is a handy index. To pick a few nits, I feel the price is a bit steep for a paperback, although these days a textbook under $20 is a buy. Also, many programmers don't have the different compilers that may be available at a college, so a few of the assignments may not suit everyone. Yet the scope of the book is so broad that everyone can learn from it. Once a given project is finished, be it a game or a means of home bookkeeping, the programmer has a useful game or tool to be used in the future.

Noel K Julkowski
18755 Van Buren St
Salinas CA 93906

Program Style, Design, Efficiency, Debugging, and Testing
by D Van Tassel
Prentice-Hall Inc
Englewood Cliffs NJ 07632
256 pages
$13.50

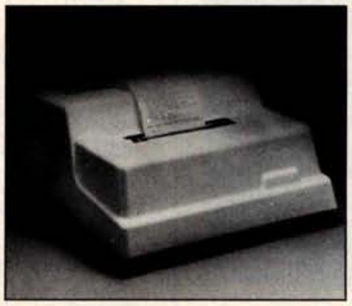
There is a lot in this book for the computer hobbyist. The first chapter is about program style. Program style can be useful even in BASIC. For example, FOR loops can be indented three spaces to aid in reading the program. Also, REMARKS can be used to indicate what the variables stand for and how the algorithm being used operates. This is especially helpful in BASIC because BASIC uses only short variable names.

Chapter 3 discusses program efficiency. This is especially important to small computer users. Both storage efficiency and time efficiency are discussed. The author points out that the selection of the problem algorithm is the most important factor for an efficient program. One of the suggestions that will help BASIC users is optimization of program loops. According to Van Tassel, typical programs spend 50 percent of their time in 5 percent of the code. This 5 percent often consists of nested loops. Thus, to make a program more efficient, inner loops must be optimized. The book lists numerous methods of code optimization.

The debugging chapter provides a wealth of hints on how to find program errors. For example, the first thing the user needs is a printout of the program. Too often people try to debug a program without a current listing. If the listing is not enough to find the error, the rule is: divide and conquer. Start putting print statements in the program to trace the logic and calculations.

It usually does little good to work for long periods of time with the program listing because the hobbyist often traces the program logic with how it *should work* in mind instead of how it actually *is working.* The added debugging print statements will help you locate the program errors. The book indicates how and where to place debugging statements and discusses common errors.

In the last chapter the author points out that programs are often just debugged



and not tested. Sample tests and hints on testing are provided.

The book has many exercises and programming problems which are not only for BASIC users. Examples are given in a variety of programming languages. Most of the techniques mentioned are good in any language. *Program Style, Design, Efficiency, Debugging, and Testing* will be a useful addition to your programming library.

F Sokolowski
956 Scott
Montebello CA 90640■

(Microcomputer) Problem Solving Using PASCAL
by Kenneth L Bowles
Springer-Verlag, New York 1977
563 pages
$9.80

(Microcomputer) Problem Solving Using PASCAL: the title sounds like a wish, but it is fact. Prof Bowles of the University of California at San Diego (UCSD) has a full Pascal (with extensions for graphics, character string manipulation and direct access files) running on interactive single user microcomputer systems such as: Digital Equipment LSI-11s, Zilog Z-80s, and 8080 based machines. Soon he expects to have Motorola 6800 and MOS Technology 6502 Pascal systems as well. Almost all of the software for these systems is written in machine independent Pascal.

The text is an integral part of a revolutionary environment for computing education. As Bowles says, "Pascal is clearly the best language now in widespread use for teaching . . . structured programming at the introductory level." By using Pascal, Bowles is able to introduce algorithm development and problem solving as components of top down, stepwise design. Procedures are introduced right from the start. Flow of control is presented in terms of modern programming principles (sequence, selection and iteration). Recursion is presented as an obvious extension of the procedure mechanism. Data structures are explained fully and clearly.

The use of graphics and text processing programming examples keeps the text from becoming bogged down in numerical methods. At UCSD, students from the arts, humanities and business disciplines are able to do just as well writing programs for nonnumeric applications as are more mathematically sophisticated students.

In addition to the compiler, interpreter and editor software, the UCSD system includes a complete computer aided and managed instruction system. The system's lessons parallel the text and permit easy management of very large introductory classes. The system is simple and complete in and of itself and could also be used effectively by high schools and community colleges to provide low cost interactive student computing.

Kenneth Bowles has revolutionized the teaching of introductory computing at UCSD. The publication of his book and the release of UCSD Pascal will permit others to follow his lead.

Richard J Cichelli
Software Consulting Services
901 Whittier Dr
Allentown PA 18103
P A Easton
Dept of Mathematics, Lehigh University
Bethlehem PA 18001■

Circle 400 on inquiry card.

Puzzle

Hextext

There once was a little (1) ___. As he flew (2) ___ to (3) ____, he encountered an (4) ___, who was going to smash his (5) _____. Calling to his friends for (6) ___, he zoomed in on the (7) ___, and made contact (8) _ times. "Good (9) ____," cried his friends, "now you are an (10) __ _."

Thomas J Dunsheath
2828 Knollwood Ln
Glenview IL 60025

(1) 101111101110 Binary
(2) 377 Octal
(3) 1111111011101101 Binary
(4) 175 Decimal
(5) 700638 Decimal
(6) 2589 Decimal
(7) 257 Octal
(8) 0101 Binary
(9) 1101111011101101 Binary
(10) 5316 Octal

This story is basically a learning tool that teaches the user how to convert numbers from one number base to another. If the binary, octal and decimal clues given for each blank are translated into their hexadecimal equivalents, the entire story will become readable. In some cases initial zeros will be required to understand the word.

See page 8B hexadecimal for the solution.

Transmission of Digital Data Over Twisted Pair Lines

Technical Forum

Edward Beebe
3341 NE 13th Av
Pompano Beach FL 33064

Having owned a Digital Group system for almost three years, I have come across many problems in interfacing peripheral equipment in the home environment. One of the biggest problems is induced noise in long lengths of cable connecting printers and terminals placed in different rooms. This should become an increasing problem with the many people placing systems in small business environments. Even though small businesses will not be plagued with the vacuum cleaner adding bits to the printer, the long run to the stock room terminal can produce unexpected results.

The first cure I recommend is to use an RS-232 interface. The RS-232 works well when several lines of handshaking are required and the cable is run less than 200 feet (60 meters). The large voltage swing and "negative" true logic reduces noise spikes very well.

An alternative approach to the problem is differential transmission of the data. In this system, two wires are used for each direction of data flow (four wires would be used with a terminal, two coming and two going). When a logic 1 is sent to the peripheral device, one line will be high and the other will be low as shown in figure 1. When a logic 0 is sent to the device, the states of the two lines switch (high going low and vice versa).

The two integrated circuits used as line drivers and receivers are the Texas Instruments 75183 and 75182. Mounting one of each on a printed circuit card at both ends of a 4 wire twisted pair will give you communications with a terminal up to 1000 feet (300 meters) away.

Figure 1 is self-explanatory. The two capacitors in the receiver are optional. The 0.002 pF capacitor between R_T and the inverted input may be used for DC isolation of the line terminating resistor. The value of this capacitor will vary with the data rate. The 100 pF capacitor between the response time constant pin and ground is used to control the response time.

More information on this and related techniques can be found in the Texas Instruments data handbook on memory and peripheral interfacing.■

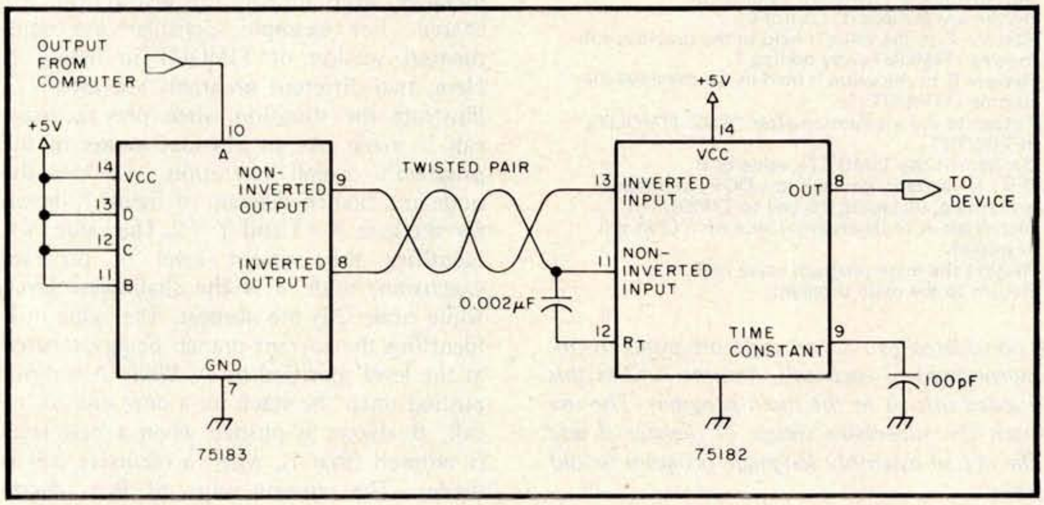


Figure 1.

Solution to Machine Language Puzzler (see page 74):

```
TIMOUT1: PUSH  B0        ; Save the main program value of B.
         MVI   B1, #X    ; Load B with X, one of the loop constants.
         DCR   A         ; A was loaded in the main program and is the second
                           delay constant Y.
LOOP1:   CPI   0         ; Is A = 0?
         CNZ   TIMOUT2   ; If not, call TIMOUT2.
TIMOUT2: PUSH  B1        ; Save B from TIMOUT1;
         MVI   B2, #X    ; Reset B to its initial value.
         DCR   A         ; Decrement A again (A is not reset in this new pro-
                           gram environment).
LOOP2:   CPI   0         ; Is A = 0?
         CNZ   TIMOUT3   ; If not, call TIMOUT3.
         DCR   B2        ; Decrement the TIMOUT2 value of B.
         JNZ   LOOP2     ; Decrement B more if it is not 0.
         INR   A         ; Restore A to the value it held in the previous sub-
                           routine (TIMOUT1) by adding 1.
         POP   B1        ; Restore B to the value it held in the previous sub-
                           routine (TIMOUT1).
         RET             ; Return to the instruction after "CNZ TIMOUT2"
                           in TIMOUT1.
         DCR   B1        ; Decrement the TIMOUT1 value of B.
         JNZ   LOOP1     ; If B1 is not zero, go back to LOOP1 and repeat
                           everything, including the call to TIMOUT2.
         INR   A         ; Increment A to its original value of Y (2 in this
                           example).
         POP   B0        ; Restore the main program value of B.
         RET             ; Return to the main program.
```

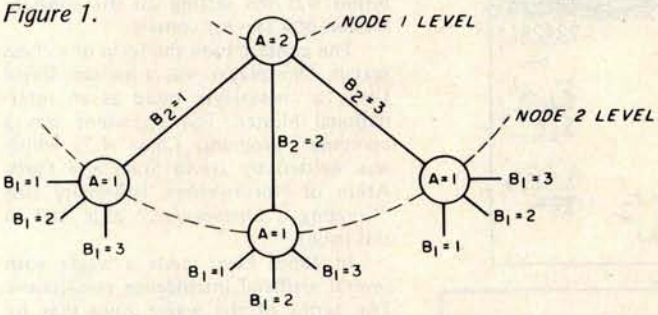
Listing 1: The TIMOUT routine considered as two separate programs to emphasize the different program environment of each call. Assume A=2 in this example, which is the Y value loaded into A in the main program. The use of subscripts distinguishes between the successive values of register B and labels of TIMOUT and LOOP. The actual assembly language program would not appear with subscripts, of course.

The most important feature of this subroutine is that it is recursive: that is, it calls itself. Recursive programs can be difficult to follow unless one remembers to place each subroutine call in a unique environment different from all other subroutine calls. This means that each call of TIMOUT should be thought of as beginning a different subroutine in a different location of memory, even though the instructions are shared. For example, consider the commented version of TIMOUT in listing 1. Here, two different programs are shown to illustrate the situation when one recursive call is made. As an abstract model of the program's overall operation, consider the node and branch diagram of figure 1, drawn for the case X =3 and Y = 2. The value in A identifies the current level of program execution; node 1 is the shallowest level, while node 2 is the deepest. The value in B identifies the current branch being executed at the level specified by A. While A is never pushed onto the stack on a new subroutine call, B always is pushed when a new level is entered (that is, when a recursive call is made). The current value of B is decremented independently of previous values of B.

The most common mistake in analyzing TIMOUT is to return, following RET, to the beginning of the previous call to TIMOUT rather than returning to the instruction following the CNZ at that level.

A completely different way of viewing the operation of TIMOUT is to think of decrementing a Y digit number, modulo X. In this case, the current value of A (originally loaded with Y) specifies the digit being decremented and the current value of B represents the present value of that digit. The stack reaches its maximum depth almost instantly and holds an image of the entire Y digit number.

Figure 1.



Answers:

(a)

X	Y	Delay
1	1	75 states
2	1	105 states
1	2	150 states
2	2	315 states

(b) The basic delay period is determined by:

```
CPI  0
CNZ  TIMOUT
DCR  B
JNZ  LOOP
```

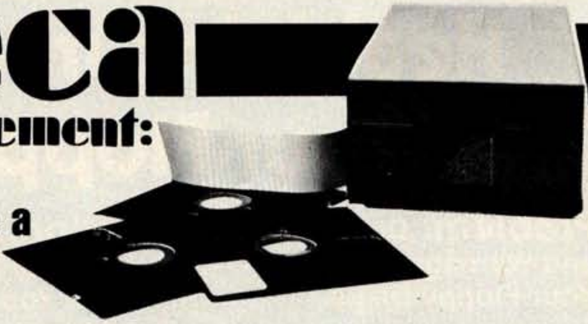
It lasts 30 states.

(c) The stack would grow to 58 bytes deep.

(d) The delay is approximately $Q(X^Y)$. ■

Answers to Hextext from page 135.

(1) BEE
(2) OFF
(3) FEED
(4) OAF
(5) ABODE
(6) AID
(7) OAF
(8) 5
(9) DEED
(10) ACE



BYTE's Bits

Department of Missing Authors:

An article entitled "Procedures for Large Numbers," written by Michael P Finerty, whose last known address was 2626 E 4th, Tucson AZ 85716, is presently in a state of suspended animation in our files. The article was produced, author's proofs sent out, and the envelope, shown below, documents the post office's attempts to get the mail through. Thus we'd like to request that Michael Finerty contact us with his present address.■

Change of Address

In the August 1978 BYTE we carried an announcement of a new RF video adapter line, produced by AVA Electronics Corp (see What's New, page 177). Immediately after publication Jason Van Hollander of that firm contacted us with a new address. You may now reach AVA Electronics Corp at 4000 Bridge, Drexel Hill PA 19026. Fortunately we were able to change the address in our files early enough that those readers who made use of our inquiry card should experience no delays.■

Levy Sweats, Wins Bet

Computer People turned their attention toward Toronto during the last weeks of August and early September this year. The Canadian National Exhibition was the setting for the consummation of a 10 year contest.

The contest took the form of a chess match. One player was a human, David Levy, a chessplayer rated as an International Master. The opponent was a computer program, Chess 4.7, which was written by David Slate and Larry Atkin of Northwestern University (see "Creating a Chessplayer," page 162 in this issue).

In 1968, Levy made a wager with several artificial intelligence researchers. The terms of the wager were that by August 1978 no computer would be able to defeat Levy in chess competition under tournament conditions. After several revisions, the amount of money at stake stood at £1250.

The first game was played on August 26 1978. The computer surprised Levy after ten moves with a 12 move combination. Levy had to struggle against great disadvantage for 30 moves. The game ended in a draw; this was the first time a computer had done so well against a top human player.

The second and third games were won by Levy, who cleverly took advantage of the computer's weaknesses. In the fourth game, both players sprang surprises. Levy played an unusual opening that led to a tactical sort of position in which the computer plays well. The computer obtained an advantage, and Levy resigned after 56 moves. This game represents the present high water mark of computer chess.

Levy easily won the fifth game, ending the match with a score of 3½ to 1½, and therefore winning the bet. We have not heard the end of this tale, however. David Levy has renewed his bet for five more years. He expects steady progress in computer chess, and has a particular interest in programs run on small computers. He stated, "This is the beginning of a 10 year boom, and many advances will be made both in microprocessor chess and general computer chess."

Future BYTEs will report in greater detail on this match and on other events in this fascinating field.■

Computers for Kydes

The San Juan Unified School District (Carmichael CA) has just received a grant called Kyde Tyme to develop computer assisted instruction software for microcomputers. The public is encouraged to develop and submit teaching programs.

Project priorities include all basic skills (language, reading and math) and support to all other curriculum areas (science, learning games, etc). Grade levels range from preschool through high school. Currently, submissions for inclusion and publication should be written in BASIC and on hard copy, including a sample run of the program. The source code is needed to evaluate and test the program. After evaluation, suggested revisions will be discussed with the author, if needed. Programs will not be published without the author's expressed permission.

The Kyde Tyme project will provide teaching programs to the public at reproduction cost. Contact Ted Perry, project manager, 2331 St Marks Way, room 17, Sacramento CA 95825, (916) 482-4306.■

Papers on Computing in the Humanities

The Fourth International Conference on Computing in the Humanities has issued a call for papers. The conference will be held at Dartmouth College in Hanover NH on August 19 thru 22 1979. Papers may be on any topic involving computing in the humanities. Languages of the conference will be English, German and French.

Abstracts of 500 to 1000 words should be submitted by December 1. Acceptances will be announced by April 1. For further information contact Stephen V F Waite, Kiewit Computation Center, Dartmouth College, Hanover NH 03755.■

A Puzzling Switch

One of the more interesting methods of interchanging the contents of two memory locations involves the use of the Boolean exclusive OR function. For example: suppose we have two memory locations labeled X and Y and we wish to transpose the contents of the two locations. The following series of instructions will perform the operations:

$$X = X \oplus Y$$
$$Y = X \oplus Y$$
$$X = X \oplus Y.$$

As an example: suppose location X contains hexadecimal 37 and location Y contains hexadecimal AC. Then:

Operation	Location X	Location Y
	37	AC
X = X⊕Y	9B	AC
Y = X⊕Y	9B	37
X = X⊕Y	AC	37

Trying to determine the logic behind this is a good test of your knowledge of Boolean algebra. . . .RGAC■

Continued from page 10

responsible is Ron. Would you please convey my personal thank you to him.

Theron Wierenga
POB 2007
Holland MI 49423

BITS Inc is a separate company from BYTE Publications Inc, but we are friends with Floyd, Ron, Jeremy, and the rest of the people at BITS, and your thanks have been conveyed. . . .CM

MANUSCRIPT EDITING PARADISE

In your editorial "Don't Ignore the High End. . .or My Search for Manuscript Editing Paradise" (March 1978 BYTE, page 6) you specified some features that you'd like to see in a "manuscript editing paradise." One feature you listed as unavailable was a video display that could convey an entire page of information in a single frame. You added that you might be able to accomplish this goal by using two 24 by 80 displays. There is a simpler way.

For about six months now, I've been using a VT-4800 video display board made by Video Terminal Technology. The basic display is 48 lines of 80 characters each. All ASCII graphics are supported, including upper and lower case. I'm using the board as part of a timesharing terminal, with the display on a surplus video monitor. On my 12 inch screen, the display is quite dense but still entirely legible.

To use the board in my application I had to add only a keyboard, monitor, and modem; all other electronics were included on the VT-4800. The 4800 also seems to support a large number of other features (parallel IO, inverse video, direct cursor addressing, etc) that might be handy in your application although they are not supported by my host system.

I understand the board currently costs about $500; I was lucky enough to purchase mine for $350 when it was still in the planning stages. If you'd like to look into the VT-4800 further, you may wish to call Ralph Butler at Video Terminal Technology – (408) 255-3001. I don't have their current address.

Roger Sward
13347 Pastel Ln
Mountain View CA 94040

TOO MANY EXPERTS?

I feel it is going to be a long time before BASIC is displaced as a hobby computer language. Just contrast the readability of the BASIC program listing

Circle 33 on inquiry card.

4 on page 173 with the Pascal program listing 5 on page 174 in "Pascal versus BASIC: An Exercise," (August 1978 BYTE, page 168). I think that convertibility is a myth. Look at what has happened to FORTRAN, locked into old ideas, old methods, and old equipment. BASIC should grow and change until it spawns a new language. Like all fields, computing has too many experts.

Milo Ketchum
58 Willowbrook Rd
Storrs CT 06268

You've raised an interesting point about readability. True, for a short program it seems easier to read the BASIC version, but then you are used to reading BASIC listings. Also, try to add modifications to a section of the BASIC version without affecting any other section of the program. This is easy in a modular language such as Pascal . . . . RGAC

READER Cs PASCAL ALTERNATIVE

As a firm advocate of good programming style, I was happy to see the August 1978 BYTE give Pascal the attention and support that it deserves. However, I think that something should be done to correct the impression conveyed by that issue that Pascal is the sole existing alternative for microcomputer users who want to program in a decent language.

Another alternative is C, developed at the Bell Laboratories. Pascal and C are different in several important respects, but they hold this in common: they are both simple, elegant, and flexible languages that encourage the writing of clear and concise programs.

Programmers should realize their good fortune in having more than one state-of-the-art language to choose from. As a case in point, Charles Forsyth and Randall Howard, the authors of the article, "Compilation and Pascal on the New Microprocessors," (August 1978 BYTE, page 50) state (page 61) that

> *. . . .we have found the C language a pleasant and effective language for developing programs, but it does not, of course, follow that everyone would.*

In truth, Pascal is simply not an option for most home computerists at this particular point of microcomputer evolution. My reading of *Pascal News* indicates that current Pascal implementations require hardware not existing today in most home systems. Specifically, they require floppy disks and "big" chunks of memory. Microcomputer users who desire to program in a state-of-the-art language on minimal hardware systems should know about the current availability of Tiny C.

Tiny C is an *interpreter* for a stripped down version of C, designed specifically for the microcomputer environment. It runs in the same "small" environments as does BASIC. All indications so far point to its being a thoroughly professional and reliable product. It is inexpensive. It comes with a program preparation system which includes an editor. Its users' manual is superbly informative, complete and readable.

I hope that BYTE will soon give C and Tiny C the attention they deserve. In the meantime, readers who wish to gain the flavor of C should consult *The C Programming Language*, by Kernighan and Ritchie (Prentice-Hall, Englewood Cliffs NJ, 1978). And readers who want information about Tiny C should contact

Tiny C Associates
POB 269
Holmdel NJ 07733.

Frank Nussbaum
Dept of Math Sciences
Loyola University of Chicago
6525 N Sheridan Rd
Chicago IL 60626

STORED PROGRAMS: A REPLY

Keith Reid-Green, in his "Short History of Computing" (July 1978 BYTE, page 84) states correctly that John von Neumann suggested putting programs into the memory of the ENIAC computer. However, the implication that previous programs were all hard-wired is not true.

In fact, there was a series of six models of Bell Laboratories Relay computers, the first of which (1939) antedated the ENIAC by seven years and the Mark I by five. All but one of the series used readily changeable programs in punched tapes — none were hard-wired.

Model II of the Bell Series was a small computer designed primarily for digital filtering and for subtabulation of a mass of data. A dozen or two programs were written and punched in tape for it. Changing a program took about 20 seconds. Most of the programs were cyclic and, therefore, were pasted into loop form.

Later models (but still antedating the ENIAC and Mark I) had more capability and more sophisticated programs. Thus Model V (circa 1945-46) had numerous program "positions," any one of which could be called up by the computer with a jump instruction in the main program. Such jumps were controlled by a "logic computer" which made decisions on the basis of manipulations of Boolean algebraic variables.

George R Stibitz
Prof Emeritus Physiology
Dept of Physiology
Dartmouth Medical School
Hanover NH 03755

Dr Stibitz, a research mathematician and pioneer in the development of the relay computer at Bell Labs in the 1930s and 1940s, was the recent recipient of the IEEE's Emanuel R Piore award "for pioneering contributions to the develop-

ment of computers, utilizing binary and floating point arithmetic, memory indexing, operation from a remote console, and program controlled computations."

THE 2650 BECOMES MORE AVAILABLE

I read with interest Mr Frenzel's article "How to Choose a Microprocessor," in July 1978 BYTE, page 124. Particularly gratifying was his mention of the Signetics 2650 as a processor with an unusually powerful instruction set and minicomputerlike architecture. However, its status as an item of limited availability to the experimenter is changing rapidly.

As Mr Frenzel's article was going to press, Signetics was preparing initial advertising for the Instructor 50^{tm} desktop computer, which uses the 2650 as its central processing element. This product, a complete hardware and software package, is particularly well-suited to the student and hobbyist. Thus, it is no longer true that experimenters who wish to use the 2650 in a personal computing system must develop their hardware and software from scratch.

J E Doll
Signetics Corp
POB 9052
811 E Arques Av
Sunnyvale CA 94086

ROUNDING OUT SOME CALCULATIONS

Your articles in the August 1978 BYTE are great! I am a Pascal fan already, and a UCSD Pascal fan on top of that. I've never written a program with Pascal, but it looks like a fulfillment of a wish-list of useful data and control structures. I've written programs in FORTRAN, BASIC, COBOL, APL, PDP-11 assembler and INTERCOM-500, an "assembler" for a Bendix G-15 computer. It has features and efficiencies I've long wanted in a high level language. I have heard of Pascal before, but never paid it much attention. Now that I see that it has a chance of becoming much more available, and after learning about the language and its method of implementation, I'm ready to help pull the bandwagon, not just hop on. Having written interactive business programs in BASIC and FORTRAN, I also appreciate the extensions implemented by UCSD, as described in the article "Pascal versus COBOL" on page 122.

I enjoyed the articles on antique "computers" ("Philadelphia's 179 Year Old Android," page 90, and "Antique Mechanical Computers, Part 2," page 96, August 1978 BYTE) as well as those on Pascal. I think that the estimates of storage capacity in the "Antique Mechanical Computers, Part 2" article are in error. As an example, I will discuss the calculations done on pages 104 and 105.

The author makes a telling remark: "A point which is obscure to me is that the letter forming cams are alleged to operate on a polar coordinate system." His calculations are then based on XY coordinates. This analysis would be correct if the machine produced a dot matrix picture with the accuracy and precision estimated. However, this is not the mechanism the machine uses.

I will assume that each letter is composed of a series of moves, and that each move is described by a distance and a direction to be moved from the previous position of the pen. Polar coordinates would work like this, or in a similar manner. With polar coordinates, a point is defined on a plane by giving a distance (radius) and direction relative to the origin, or (0,0) point on a Cartesian (XY) plane.

The actual method used by the machine and precise information theory may not be represented by what follows, but I believe my calculations will more closely approximate the actual storage capacity of the machine than those in the article. The three cams that define each letter will be assumed to specify the height or pressure of the pen, and the distance and angle made for each of a sequence of moves. The following table gives the *information content* needed to attain the precision found in the results of the machine:

Parameter	Number of Bits	Range of Values
Height and pen pressure	2	Up; light, medium, or heavy pressure.
Distance	6	0.1 mm to 6 mm, in 0.1 mm steps (60 discrete values).
Angle	7	3° to 360° in 3° steps (120 discrete values).
Total bits per move	15	
x moves per letter	12	(A reasonable guess, based on the y).
Total bits per letter	180	This is different from the author's estimate
x number of letters	52	by a factor of 80.
	9360 =	Total number of bits to encode the entire alphabet.

The estimate of minimum distance, different from the author's, is based on the assumption that a single move will tend to be fairly straight due to some mechanical damping while drawing a line between two points. Any of the parameters might vary by one or two bits from the given estimates, but I believe them to be fairly close given the assumptions used.

One other factor to take into account is the "noise" introduced in translating cam shape to pen movement. The cams would need twice as many bits of storage if the mechanism's noise obscured half of their precision.

The machine discussed obviously had other "memory" as well as the memory that stored the shape of the letters. For instance, the mechanism that determined the sequence of letters would have a capacity equal to about six bits (enough to store 52 choices) times the number of letters.

Sigurd Andersen
48 Prospect Av
Newark DE 19711

ON BUSES AND LANGUAGES

I have two questions for your readers: why Pascal? Pascal and BASIC were both written for education and both need extensions to be of use in serious (large) programming experiments and production. What we need is a language that simplifies structured programming like ALGOL, that is complete like PL/I and which compiles like HP-BASIC during editing to a compact code for further compilation or interpretation. It should be adapted to systems from batch via disk operating system (DOS) to real time. In short: it needs everything for everybody on a microcomputer, not a patchwork of languages that were never intended for amateur use. My next question is about all those buses that our poor chips are struggling with. Why don't we (the users, not the manufacturers) define a bus for all processors? It can be done! The Dutch Hobby Computer Club (page 148 in this issue) is currently using one.

B J Besseling
Tolstraat 401
Amsterdam HOLLAND

Just to reiterate, Pascal has a lot of interest going for it based on availability of implementations—and enthusiasm on the part of those who are into the language as a means of expression. We find machines from the microprocessor up through the Cray-1 using it as a systems programming tool, and with extensions of the sort UCSD has been implementing, it works as well for many applications areas.

MORE PIXELS PER PIXE?

Concerning "Convert Your TV Set to a Video Monitor" (May 1978 BYTE, page 22), Dan Fylstra stated that using a Pixe-Verter radio frequency (RF) unit with a video board, one would only be able to generate usable line lengths of 32 characters.

That may be the case when using a black and white television. I, however, have been using a Pixe-Verter, Processor Technology VDM board, and a 15 inch Sony KV1510 color set with excellent results. I am able to use this configuration for 64 character lines very easily. The quality of the display is not as high as available on a direct connection of the video board and a video monitor, but the display is very usable.

Using a color set may seem like overkill, but I, like many people, have a color set available and no black and white set. Of course color sets are much more likely to be "non-portable," but the smaller color sets may save one from the hazards of playing with a television set's innards.

W Czyzynski
1618 Home Av
Berwyn IL 60402

ABACUS QUERY

In reference to the alleged Chinese abacus shown in "A Short History of Computing," by Reid-Green, (July 1978 BYTE, page 84), I have been informed by supposedly reliable sources that an "abacus" with five beads below and two beads above the bar is really a Japanese device called a *soroban*, and that a genuine Chinese abacus has four and one beads, respectively.

My references show that the abacus has the 4/1 configuration. The device I own, which I thought was an abacus, has a 5/2 configuration; but it is clearly marked "Japan" and *soroban* in small print on the side, confirming what I had been told about the 5/2 "abacus."

Do any of BYTE's readers know if both types were in use in both countries and whether there are any other configurations?

William B Adams
POB 34683
Bethesda MD 20034

ON OTHER HARDWARE WAYS TO MULTIPLY

I just finished reading the article "How to Multiply in a Wet Climate: Design Details" (May 1978 BYTE, page 104) by Jack Bryant and Manot Swasdee. I thought the article was great. However the authors made mention of only one chip from TRW (the MPY-16AJ-M).

There is one other 8 bit by 8 bit multiply integrated circuit chip (see "Whats New" December 1977 BYTE, page 210). It uses less power than the one the authors Bryant and Swasdee use, and is faster. Also there are transistor-transistor logic (TTL) 4 bit by 4 bit multiplication integrated circuits (the 74284 and the 74285). The two circuits must be used together. They will do 4 bit by 4 bit multiplication in under 50 ns. I am currently using a pair to do binary coded decimal (BCD) multiplications. I would like to see more of this hardware type of article in BYTE.

FTB3 Allan W Bathe
W-2 Division
USS Canopus (AS-34)
FPO New York 09501

But authors Bryant and Swasdee actually used the chip they concentrate on . . . . CH ■

BYTE's Bugs

Sweets for KIM Spurned

Dan Fylstra's useful low calorie text editor and assembler for KIM-1, "SWEETS for KIM, a Low Calorie Text Editor," February 1978 BYTE, page 62, contains a bug on page 68 which limits the size of programs to be handled. The assembler code at address hexadecimal 1796 should read E5[SBC ENDAD+1] and similarly at address hexadecimal 179C should read E5[SBC ENDAD].

Ernst Schumacher
Dept of Chemistry
University of Bern
CH-3000 Bern 9, Freiestrasse 3
SWITZERLAND■

A Minor Mode

I very much enjoyed Bill Struve's article concerning computers and music, "A $19 Music Interface (and Some Music Theory for Computer Nuts)" in the December 1977 BYTE, page 48. It was well-written and very explanatory. However, there is one small error I wish to call to your attention: table 2 on page 56 contains the wrong solfeggio syllables. The syllables to which I am referring are *mi*, *la*, and *ti*. They should be *me*, *le*, and *te* because they are in the minor mode of the A scale. Keep up the good work!

Scott Claybaugh
RD #1, Box 136B
East Berlin PA 17316■

Irresistible Capacitor

A small bug appeared in figure 5 of "How to Get Your Tarbell Going," by Larry Weinstein (July 1978 BYTE, page 164): capacitor C16 was incorrectly drawn as a resistor. The 0.0022 μF value is correct, however.■

Univac Had It First

W F Steagall Sr, System Development Corp, 2500 Colorado, Santa Monica CA, has informed us that the figure 8 caption on page 92 of the August 1978 BYTE ("A Short History of Computing" by Keith S Reid-Green) is in error, and that the Univac I featured simultaneous input and output (IO) and processing as early as 1951.■

Translation Error

In "BYTE's Bugs" February 1978 BYTE, page 93, John D Leasia writes to point out an error on the pseudorandom number generator, first published in the November 1977 BYTE. However, the original object code as supplied by Daniel Grieser, was totally correct. Indeed, if the *corrections* suggested by John D Leasia are implemented, one will create the very *bug* he is trying to eradicate.

The confusion must arise over the difference in architecture between the 6800 and 6500 type machines, the former possessing two accumulators while the latter has only one. The original, as written for the 6800, makes use of both accumulators and is, as stated in the article, a direct implementation of the identity $13 \times N = N \times 2^3 + N \times 2^2 + N$. If, however, we were unaware of the dual accumulators being used, and assumed all the math to be going on in one accumulator (somehow), then the equation that *appears* to be implemented is: $[N \times 4) + N] \times 2 + N = 11N$. If, as suggested by John D Leasia, the op codes at 0005 and 0006 are interchanged, the equation that is actually implemented is: $N + 2N + 2^3 \times N$ (which again equals 11N and, as John points out, this does not generate all possible numbers and repeats after only 128 numbers).

When using a 6500 type machine with its single accumulator, one must adopt a slightly different approach. The program supplied by John for his KIM-1 would, I believe, run perfectly; it is in fact using the identity:

$$13 \times N = [(N \times 2) + N] \times 4 + N = 3N \times 4 + N.$$

A good point to remember is that, to try achieving a semiliteral translation of assembled object code from one machine to another, is *not that easy*, especially when the exact mode of action of one of the instruction sets involved is likely to be unknown.

J C Burchell
24 Worrin Rd
Shenfield, Essex ENGLAND
CM15 8DE■

Pascal a Wrong Number?

Due to a transcription error in a telephone number, listing where UCSD Pascal may be ordered, a slight rewiring of the UCSD telephone system had to be temporarily performed to intercept all phone calls. The *correct telephone number* for the UCSD Pascal project is (714) 452-4526. Ken Bowles tells us that the incorrect number published in August 1978 BYTE, page 132, will be discontinued as soon as possible.■

A Looming Bug

July 1978 BYTE's articles on the history of data processing and computers added a welcome dimension to your journal. However, I have one question. According to Dr James M Williams in "Antique Mechanical Computers, Part 1: Early Automata" (on page 57):

In 1741, he [Vaucanson] *devised the system of punched cards that controlled the looms in the Jacquard tapestry factory.*

According to Keith S Reid-Green in "A Short History of Computing" (on page 89):

Although Jacquard is commonly thought to have orginated the use of cards, it was actually done first by Falcon in 1728. Falcon's cards, which were connected together like a roll of postage stamps, were used by Jacquard to control the first fully automatic loom in France. . . .

Would you care to comment?

Ms Kemp Vicik
7752 Chalmette Dr
Hazelwood MO 63042

Keith S Reid-Green Replies

The following quote is taken from A History of Technology, *volume IV, Singer Holmyard et al, editors, Oxford University Press, 1958. It should help to explain matters:*

Prior to the 19th Century an automatic loom required considerable manual assistance. In the most advanced models of the early 1700s, woven patterns were controlled by a cylindrical perforated paper roll. In 1728, Falcon replaced the paper roll with connected punched cards and a mechanism to read them. In 1745, Vaucanson transferred the mechanism from the side of the loom to an overhead position and added a perforated cylinder rotated directly from the loom. Jacquard, in 1801, is credited with mechanizing out of existence the job of draw-boy and thus fully automating the loom. A patent to replace the draw-boy with a machine was issued before Jacquard was born — in 1687 to Joseph Mason and a later one in 1779 to William Cheape.

Jacquard may get the credit because "Cheape Looms" probably didn't sell. For more details, see A History of Technology.

Keith S Reid-Green
Director, Software System Development
Educational Testing Service
Princeton NJ 08540■

Clubs and Newsletters

MITS Minifloppy Disk Users Group

The formation of a users group for the owners of the MITS minifloppy disk system has been announced. The group is nonprofit and will act as a clearing house for software for the new minifloppy. Those interested should send a self-addressed stamped envelope to AAA Computer Services, POB 2742, Appleton WI 54911.

Cromemco Users Pool

The Cromemco User, Systems and Software Pool (CUssP) has been formed for users of Cromemco systems and boards. Its purpose is the exchange of hardware and software operating notes, bugs and their "cures," product evaluations and user written software. Three initial newsletters are planned, each highlighting a different piece of Cromemco software: 16 K BASIC, assembler and FORTRAN. Any user input is welcome. The membership fee including the three initial newsletters is $10. For more information, user input, or membership write to David Dameron, 402 E O'Keefe St, Apt 27, E Palo Alto CA 94303.

The New England Computer Society

The New England Computer Society (NECS) meets the first Wednesday of each month for the purpose of exchanging computer hobbyist information. NECS is the largest club in the area with 170 paid members. Within the club there are PET, TRS-80, M6800 and 6502 user groups. They are also fostering a PC Net interest group with a planned bulletin board system. Membership dues are $6 per year which include a newsletter. The meetings start at 7 PM and are held at the Mitre Corp cafeteria, Route 6, east of Route 3, Bedford MA. For additional information, write to the New England Computer Society, POB 198, Bedford MA 01730.

Attention Singapore Readers

There is a move to set up a Singapore Microcomputer Society which would include home and hobby computer users as well as professional system designers. If you live in Singapore and are interested in this society, you should get in touch with Bernard Tan, Physics Dept, University of Singapore, Bukit Timah Rd, SINGAPORE 10.

New Club Supports the RCA 1802 COSMAC

Owners of Elf, Super Elf, Elf II, COSMAC VIP, COSMAC Development System or Homebrew 1802 will find many programs, applications and experiments for their microcomputers in each issue of *Questdata*. This publication will discuss the RCA instruction set and how to build interesting programs for graphics, control, games and business purposes. Coverage will be given to Tiny BASIC, Elf expansion possibilities (memory, cassette input/output, etc), light pens, reader questions and music programs. To obtain this monthly publication, send $12 to *Questdata*, POB 4430, Santa Clara CA 95054.

Dutch and Belgian Hobby Computer Club

This past summer Rob Bronckers from the Dutch Hobby Computer Club stopped by the BYTE offices to give us information about this European club. It is a rather large club with eight chapters in The Netherlands and two in Belgium

— an approximate membership of 1000. The club offers a hardware service and has a software library. The membership dues are HFL 15= (15 Dutch Florin) including the newsletter, *HCCN*, which is published every two months. For more information about this club, you can contact Rob Bronckers at Delfsekade 12, Leidschendam, NETHERLANDS 030-713568.

Blackhawk Bit Burners Computer Club

Rockford IL is the location of this new club which meets at 7:15 PM on the second Wednesday of every month. They have over 40 members and are anxious to increase that number. Membership is $12 a year which includes a newsletter. Interested individuals should write to Frank D Dougherty, 325 Beacon Dr, Belvidere IL 61008, or call (815) 544-5206 after 7 PM.

New Tandy Computer Users Group

The National Capitol Chapter of the Tandy Computer Users Group has been formed. General membership meetings are held on the last Wednesday of each month. The group is open to any and all interested persons. For more details on group activities, write to Rod Wright, 8205 Chivalry Rd, Annandale VA 22003, (703) 560-5854.

North Star Users Group Releases Program Library

The North Star Users Group has released more than 300 programs on 20 diskettes which are available for a small copying charge plus the price of a new disk. The disks are designed to run in microcomputer systems that utilize the North Star DOS and North Star BASIC, a high level interpreter.

The library includes business and finance programs, mailing list programs, math programs, programming utilities, a Pilot interpreter, a Palo Alto Tiny BASIC interpreter and a large number of games. Further information can be obtained by sending a self-addressed stamped envelope to J Dvorak, NSUG Program Library Distributor, 704 Solano Av, Albany CA 94706.

San Fernando Valley KIM-1 Users Club

The San Fernando Valley KIM-1 Users Club is off and running. The first meeting was a great success and they are trying to attract as many new members as possible. The general purpose of the club is to foster informal communication among fellow KIM-1 users. Meetings will be on the second Wednesday of each month. For more information contact Jim Zuber, 20224 Cohasset, number 16, Canoga Park CA 91306.

Washington DC Amateur Computer Society

Through the monthly newsletter, *JWACS*, we are kept posted on the current happenings of the Washington DC Amateur Computer Society. This informative newsletter is available to Society members; nonmembers may subscribe to the journal at the rate of $6 per year. The club is interested in exchanging newsletters with other organizations to further the interchange of hobbyist information. If you live in the Washington area and are interested in becoming a member, send a self-addressed stamped envelope to Washington Amateur Computer Society, 4201 Massachusetts Av, Washington DC 20016 for meeting details.

Tulsa Computer Society

The Tulsa Computer Society is an active computer club that meets on the last Tuesday of every month at 7:30 PM. The meeting place is the Tulsa Vocational-Technical School seminar room at 3420 E Memorial Dr. Membership in TCS is $6 annually, which includes a one year's subscription to its newsletter, *The I/O Port*. Correspondence should be sent to The Tulsa Computer Society, POB 1133, Tulsa OK 74101.■

Hobbyist Computerized Bulletin Board

Ward Christensen
688 E 154th St #3D
Dolton IL 60419

Randy Suess
1930 Bradley
Chicago IL 60613

Bulletin board systems could become nodes in a communication network.

Note:

*This project was a collaboration of Ward Christensen and Randy Suess. Each had a particular part to uphold. The first part of this article describing the purpose of the bulletin board is written by Ward. The part describing the hardware details is written by Randy. . . .*RGAC

The Computerized Hobbyist Bulletin Board System is a personal computer based system for message communication among experimenters. People with terminals or computers equipped with modems call in to leave and retrieve messages. It was conceived, designed, built, programmed, tested, and installed in a 30 day period (January 16 1978 to February 16 1978) by the two of us. In an effort to generate material for our computer club's newsletter, I first thought of the idea and discussed it with Randy on January 16 1978.

We laid out the hardware requirements: an 8080 processor with 24 K bytes of memory, single floppy disk, modem interface, and some sort of local keyboard and display. Randy scoured the computer stores and purchased a mother board and two 4 K byte memory boards at a reasonable price. I talked with Lloyd Smith and Bill Bassett, who operate DMA Inc, a manufacturer of floppy disk drive systems based on the Tarbell controller and the Innovex (now Innotronics) floppy disk drive. DMA offered to donate 40 percent of the cost of a controller and floppy disk drive to the project. I purchased the floppy disk drive, controller and CP/M license, and loaned 24 K bytes of memory to the project, pending receipt of 16 K bytes offered by DMA. Randy donated his D C Hayes modem board, PolyMorphics VTI, SwTPC keyboard, power supply, chassis, IMSAI 8080 processor card and Vector memory board.

We started with the monitor for the system, but found that it was difficult to make the VTI keyboard port work because the VTI keyboard data bus shares the bus carrying the characters being displayed. To solve this, Randy bought a Processor Technology 3P+S board and interfaced the keyboard to it. This also allowed us to have a sense switch port using the 3P+S. Since this board has a serial port, Randy later decided to add a Teletype to the system for logging incoming data. This completed the configuration.

Programming

In the first week of the project, I wrote a mock-up of the software using MITS 8 K BASIC. The input/output (IO) drivers could be switched to my modem under sense switch control, so I had people call in and critique the system. Many good suggestions were made.

By now Randy had the computer far enough along to need some programming, so a monitor was put into read only memory. Since the system was to run under CP/M, a Teletype compatible scroll routine was also put into the read only memory for both the monitor's and CP/M's use. Additions were made later to support the 3P+S board for keyboard input.

After the BASIC mock-up of the system was close to what we wanted, programming started on the assembler version of the software.

Assembler language was chosen over BASIC for the implementation language because of size and speed efficiency, and to maintain control over such functions as *control-K* to terminate (kill) the current function and return to the main menu. The program now consists of the pieces of assembler source, shown in table 1, which are combined to produce the final assembler source program.

In addition to this application program, a modified BASIC IO system (BIOS) for CP/M had to be written. We wanted the system to be able to be started from a "cold boot" bootstrap program and come up running the bulletin board program, not CP/M. This would make the system more crash resistant since the ringing of the phone would activate it and the person calling in would not be able to gain access to CP/M itself. I studied the functions of the CP/M console command processor and modified the cold start routine to be able to load in a program and branch to it. This was done selectively under control of the sense switches. If we wanted to load CP/M instead, we just set the remote mode sense switch off.

About 30 days after the project was conceived, it went on the air via a new telephone line Randy had installed in his basement. It was used for a week by our friends and was then announced at the February meeting of CACHE, the Chicago Area Computer Hobbyist's Exchange. A poll was taken to see how many people could call. We expected perhaps five or ten, but 25 hands were raised!

During the weeks that followed the CACHE meeting, many people tried the system. We informed Dennis (D C) Hayes and PCNET of it, and they both tried it. Most people were able to communicate with the system. Those who couldn't had either bad telephone lines or modems that put out weak signals, which the system seems quite intolerant of. People who left messages saying they had some information of interest and those who said they needed information discovered that other people using the system contacted them. We were pleased to find the system working this way, because that was one of its purposes.

Getting good feedback about the operation of the system is important to us. That is why the *goodbye* function allows you to leave comments. Many people contributed good suggestions which were soon implemented. For example, a new version of the software was being put up on the system every few days to send nulls after carriage returns and line feeds for the many people calling in using Texas Instruments Silent 700 terminals.

We will continue to improve the system, although we consider the project to be close to completion. We might add date and time, or if we get ambitious, a second telephone line, but we are quite satisfied with the system as it exists now. We would like to see other experimenters or clubs implement such a system.

These bulletin board systems could then become *nodes* in a communication network of automated message and program switching.

Hardware Details

My objective for the hardware of the bulletin board system was to get the most functions at the least cost. To this end I started with my IMSAI 8080 processor card, having replaced it with a Cromemco Z-80 in my personal system. I found an old Vector mother board for free and two reasonably priced 4 K byte memory boards at a computer store. I also bought a PolyMorphics video terminal interface (VTI) card.

Name	Function
CE00	Mainline routine: sign on, menu, etc.
CE10	Goodbye function, comments function.
CE20	Message summary retrieval function.
CE30	Message entry (input, edit, save).
CE40	Message retrieval.
CE50	System functions (erasing messages, etc).
CE80	Disk subroutines (read, write, extend, etc).
CE90	Nondisk subroutines (modem IO, etc).
CE99	Constants and work area.

Table 1: Assembler routines used in the bulletin board program.

My objective was to get the most functions at the least cost.

We burned a monitor for a memory board to support the video terminal interface. Adding an SwTPC keyboard completed the local console for the system.

We are now running an 8 K Vector Graphic memory board and two 8 K byte boards designed by Forrest Duston (a local hobbyist and design engineer), which were donated by Lloyd Smith and Bill Bassett of DMA Inc.

A floppy disk was necessary for storing the messages, so Ward bought a Tarbell controller board and Innovex drive from DMA, who graciously offered it at 40 percent off.

I put these components into a chassis and home-built card cage and started testing the system. I found some incompatibilities in the hardware, such as between the new VTI and the memory board (pressing reset would not give control to the monitor in read only memory). Switching to an older design video terminal interface solved the problem.

I then found problems with the keyboard. The sharing of the interface's on board bus between the display data and keyboard status data caused the display to break up badly. Putting a delay loop in the software didn't help, because the nonlatched data would easily be missed. I solved this by purchasing a Processor Technology 3P+S board for the keyboard input port. This turned out to be very useful, first to allow a sense switch input port, and later to drive a logging Teletype for all incoming messages.

I had also purchased a D C Hayes modem board to be used for exchanging files and programs with other CP/M users, but decided to use it in this system because of its ability to select the transmission rate (110 or 300) and to answer the phone. Hooking this through a data access arrangement (DAA) to the phone line completed the *standard* hardware of the system.

Ward needed some way for the system to recover from possible software and hardware crashes, so I built a simple interface between the modem board and the hard reset of the system so that, every time the telephone rang, the system would do a cold boot to CP/M. A 555 timer was added to the reset circuitry so that after the first ring, subsequent rings would not interfere with the CP/M booting process. I did not use the ring detect feature of the Bell data access arrangement, because the modem board's setup was easier to control and detect under software. After some preliminary problems with the modem board (Dennis Hayes was more than helpful in resolving these) and the Tarbell controller, the hardware has been functioning as expected.

To prolong the motor life and minimize power consumption, I installed a solid state relay that turns the drive motor on when the telephone rings. The 555 then provides 30 seconds during which ring detect is disabled to prevent subsequent rings from interrupting the booting, allowing CP/M to come in and answer the phone. The Tarbell controller board is designed to boot CP/M when the system is reset, so this hardware, with Ward's software, gets the system going.

That's how the hardware now stands, except that I sold the Teletype and am going to use an SwTPC PR-40 for the logging device. The system has now been operating very successfully since mid February 1978.

Programming Details

The following information is supplied for those who want the details of the structure of the files on the system, what modifications were made to CP/M to support the system, or the design of the bulletin board program itself.

File Structure

CP/M is capable of storing 240 K bytes of information on a diskette. The directory is capable of keeping track of up to 64 files. I would like to keep about 200 to 300 active messages on the system. This means that each message cannot have its own directory entry. I decided to group messages in quantities of ten per directory entry. Thus filename MESSAGE.00X contains messages 1 thru 9, MESSAGE.01X contains messages 10 thru 19, etc. Assuming the number of active messages to be about one half the highest message number implies that there will be five messages per file. If there are 40 active files (such as messages from 1 to 400), then there will be approximately 200 messages. If each message averages eight lines of 50 characters, this is 400 bytes. The total disk space for the messages themselves will be 80,000 bytes. Thus 200 messages can easily be held. The directory keeps itself clean: when the last message in a group of ten is erased, the file itself is automatically erased from the directory.

There are also message summary files. They contain the following fields for each message:

Message number.
Number of lines.
Date entered.
From (author)
To (recipient)
Subject.
Password (must be supplied to erase the message).

The summary files are grouped like the message files, but by 100s. Thus, summaries 001 thru 099 are in a file called "SUMMARY.0XX", etc. This keeps the message erase function simple: it can read the entire file into memory, delete the summary, and rewrite the entire file.

There are a number of text files on the disk. They are used when predefined information is to be given, such as the welcome message. There are two types of files: straight text files, and question files.

Question files are used in such functions as Help, in which the user is asked a question, "Want help with summary function?" for example, and is then allowed to answer Y or N. If Y is typed, the text associated with the question is typed. If N is typed, the text of the message is skipped. Table 2 contains the text and question files used by the system, as well as other files not previously mentioned.

CP/M Modifications

When an unmodified CP/M system is initially loaded, control transfers to the "cold boot" entry in the BASIC IO system (BIOS). Its function is to:

- Build the required jumps in low memory to CP/M.
- Transfer control to the console command processor.
- Reset the disk, log in drive A, and begin communicating with the console.

Since we don't want CP/M to get control when the system answers the phone, a modification was made to the cold start routine so that the sense switches were examined to determine if the remote switch was on. If not, control proceeded to normal CP/M operation. However, if the remote switch was on, the system performed the reset disk system and log in drive A functions normally done by the console command processor. The normal CP/M functions of OPEN and READ are performed to load the bulletin board program into memory. Note that all of this happens before the telephone is answered, so that if the program can't be found, or has an error loading, a call is not wasted. When the program is loaded, control transfers to it, the telephone is answered, the information transfer rate is determined, and communication begins.

Bulletin Board Program

The functions of the program are:

- Answer the telephone.
- Determine the information transmission rate.
- Process the user's requests.
- Hang up the telephone.

Feedback on the operation of the system is important.

Table 2: Descriptive files to aid users of the bulletin board system.

WELCOME: This file is typed when you first connect to the system. It welcomes you, and tells of the various control characters that can be used to control the system.

BULLETIN: This file may or may not exist. If it exists, it is typed after the welcome file. It contains bulletins such as changes in the operation of the system, club meeting announcements, etc.

LOG: When you first get on the system, it asks you for your first and last name. They are written to this file to give us a log of who uses the system. Also, if you are a first time user, the system asks where you are calling from, and logs this information after your name.

FIRSTTIME: This file is a question file, and it is interpreted if you reply Y to the question asking if this is your first time on the system. It prompts the new user about what can be done.

HELP: This is a question file. It is quite lengthy, going through the various system functions and, at each one, asking if you want help.

ENTINTRO: When you ask to enter a message into the system, you are asked if you would like to review the steps used to enter a message. If you say Y, the system types this file.

ENTRHELP: After you have entered your message, you have several options: aborting, editing, saving, or help. This file is the question file which is interpreted if you ask for help.

NEXT: This file contains the next message number to be entered into the system. It is incremented by one every time a message is saved.

COMMENTS: When you leave the system via the G (Goodbye) function, the system asks if you would like to leave any comments. If you say yes, it writes the lines you type into the comments file. This provides a technique much easier than message entry, so we will get feedback on the use of the system.

PASSWORD: Certain system functions are of such a nature that only the system operators (as opposed to users) may execute them. For example: erasing a message whose password is not known, exiting into CP/M for file maintenance, determining disk status, etc. This file contains the password which must be entered to get into this operator mode. It is a disk file, rather than being built into the program, to facilitate changes when the need arises.

```
         TERMINAL NEED NULLS?  TYPE CTL-N WHILE THIS TYPES:

                 ***   WELCOME TO CBBS/CHICAGO   ***
  ***  WARD AND RANDY'S COMPUTERIZED BULLETIN BOARD SYSTEM  ***

-----> CONTROL CHARACTERS ACCEPTED BY THIS SYSTEM:

      DEL/RUBOUT  ERASES LAST CHAR. TYPED (AND ECHOS IT)
      CTL-C       CANCEL CURRENT PRINTING
      CTL-K       'KILLS' CURRENT FUNCTION, RETURNS TO MENU
      CTL-N       SEND 5 NULLS AFTER CR/LF
      CTL-R       RETYPES CURRENT INPUT LINE (AFTER DEL)
      CTL-S       STOP/START OUTPUT (FOR VIDEO TERMINAL)
      CTL-U       ERASE CURRENT INPUT LINE

---------------------    BULLETIN  ---------------------
        PROBLEMS WITH THE SYSTEM??
HARDWARE: RANDY (SUESS), (312) 935-3356
SOFTWARE: WARD (CHRISTENSEN), (312) 849-6279
---------------------    BULLETIN  ---------------------
)
---------------------    BULLETIN  -   ----------------
--->  ALL USERS:  BE FAMILIAR WITH MESSAGES 3, 6, AND 60

                         N O T E
-----> AS OF 4/8/78, MESSAGES PACKED AND RENUMBERED <-----
---------------------    BULLETIN  ---------------------
```

Listing 1: Output of the bulletin board system when a user first logs onto the system. The welcome message is printed out followed by the important control characters. Any bulletin of general interest is also shown at this time. Bulletins can only be put on the system by the system operators. Individual users may not access this function.

Recovery from software and hardware crashes was needed.

The system answers the telephone by an output instruction to the D C Hayes modem board. It then has two subroutines, called SET110 and SET300, which set the transmission rate. Initially, the program gets control, calls SET110, which picks up the telephone, sends out the answer tone, and selects the 110 bps transmission rate with two stop bits and no parity.

The program then monitors the modem status for about 15 seconds to determine when a connection with another modem has been made. If no connection is made, a message is printed to the local console, the telephone is hung up, and control is transferred to the monitor. Note that the monitor does not take part in the bulletin board system except to supply the scroll routine for the video terminal interface. The ring detect and system startup is all done through hardware, for reliability. Thus the system is able to experience an unexpected power outage and still operate the next time the telephone rings.

If a connection is established, the system, at 110 bps, waits for a character from the line. If this character is not an ASCII carriage return, it switches to 300 bps and tries again. It tries each speed, alternately, ten times. If this technique does not result in a carriage return being seen, the system prints a message, hangs up the telephone, and jumps to the monitor.

The technique for establishing communications with the system is to:

- Dial the telephone.
- Wait for the answer tone.
- Put the modem on the telephone line.
- Press return repeatedly until the system responds.

When the data connection is established, the system types a carriage return and linefeed to indicate that it is connected. It then sends the contents of the disk file WELCOME to the caller, followed by any bulletins that may be of interest. This output sequence is shown in listing 1. Next, it asks for your first and last name, and logs them to disk. It asks if you are a first time user, and if so, asks where you are from, logs this to the disk, then gives you an introduction to the system.

The system also keeps track of how many people have used the system. This number is output at sign on. Mostly, it is just for interest, but helps to see if anyone has used the system since you last did.

The following prompt asks what you want to do:

```
FUNCTION: B,C,D,E,G,H,K,N,P,R,S,W,X, (OR
          ? IF NOT KNOWN)?
```

If you do not know or remember what the letters stand for, you type ?, in which case the system replies:

FUNCTIONS SUPPORTED:

B=PRINT BULLETIN
C=CASE SWITCH (UPPER/LOWER)
D=DUPLEX SWITCH (ECHO/NO ECHO)
E=ENTER MSG INTO SYSTEM
G=GOODBYE (LEAVE SYSTEM)
H=HELP WITH FUNCTIONS
K=KILL (ERASE) A MESSAGE
N=NULLS: SET 0 TO 9 AS REQUIRED
P=PROMPT SWITCH (BELL ON/OFF)
R=RETRIEVE MSG
S=SUMMARIZE MSGS
W=TYPE WELCOME
X=EXPERT USER

Details of these functions are given below.

B=PRINT BULLETIN

If you are using a printing terminal, you can simply look back to see what the bulletin said when you first connected to the system. If, however, you are using a video terminal, you may want to review what the bulletin said. The B function does this.

C=CASE SWITCH (UPPER/LOWER)

Most terminals are capable of handling any ASCII character that comes to them, whether it is lower or upper case, typically translating lower case to upper case if necessary. However, there are exceptions, notably the Heath H9 terminal, which displays garbage when sent lower case characters. For this reason, the system defaults to upper case only, until you use the C command to switch to lower and upper case mode. Using the command again will switch back to upper case only mode.

D=DUPLEX SWITCH (ECHO/NO ECHO)

Full duplex and half duplex refer to the ability for a medium (such as the telephone line) to have data going in both directions simultaneously (full duplex) or only in one direction, (half duplex) at a time. These terms have now taken on different meanings when referring to a computer terminal. Full duplex refers to a terminal which sends a character from its keyboard, then echos what comes back from the phone line. Half duplex refers to a terminal which echos its characters locally. Thus, the D function is used to determine whether the system echos characters back to you.

If you are using a Teletype, for example, which prints every character you type, you will use the D command to go to half duplex mode so you do not see every character twice (one from the Teletype, the other from the phone line). This does sacrifice some integrity, however, because you do not see the characters echoed back to you to verify that what you typed went in correctly.

E=ENTER MSG INTO SYSTEM

You must know: the date; who the message is addressed to (up to 20 characters); and the subject of the message (up to 30 characters). You will be asked if you want to be able to erase the message; if you reply Y, you may elect to protect it with a 4 character password, or to use no password. We recommend not using passwords so that it will be easier for the recipient to erase the message when he or she has read it. Also, you will be asked if you want to review the steps used in entering a message. This is useful the first few times you enter a message, but typically is not used after that.

G=GOODBYE (LEAVE SYSTEM)

When you use the G function, the system asks if you want to leave any comments. If you say Y, you will be able to type in comments, which are written to disk at the end of the COMMENTS file. Typically we find suggestions for system improvement, or congratulations, or requests to erase messages whose passwords have been forgotten, or any other information which the user wanted to send to the operators, but didn't want to leave as a message.

H=HELP WITH FUNCTIONS

The H function asks a series of questions such as, "Do you want help with the E (enter) function?" For any question which you answer Y to, you will be given some detailed information.

K=KILL (ERASE) A MESSAGE

The letter K (for kill) rather than E (for erase) was chosen because E was already used for enter. In order to keep disk utilization down, this command allows users to erase their own messages. The system asks for the password, and if correct, the message and its summary are removed. If users do not elect to be able to kill their own messages, a special system password NONE is put in the file. Then only the system operators who have entered another password will be able to kill it.

N=NULLS: SET 0 TO 9 AS REQUIRED

Some terminals (notably the Texas Instruments Silent 700 series) require nulls to be sent at the beginning of each line to allow the carriage time to return. This command sets the nulls to any value (0 to 9) necessary. Note that use of *control-N* any time the system is typing something to you will set the nulls to 5. Thus using *control-N* will allow the welcome and bulletin messages to be printed properly until the N (null) command can be used.

P=PROMPT BELL

The P command allows the prompting bell (normally sounded every time the system wants input) to be turned off or back on. It operates as a toggle and sets the bell response into whatever mode is not current.

R=RETRIEVE MESSAGE

To retrieve a message, you must know the message number, which is typically found through the use of the S (summary) command.

S=SUMMARIZE MESSAGES

The S command prints a 2 line summary of each message from a starting message number. *Control-C* cancels the printing of a particular summary, going on to the next. *Control-K* kills the printing entirely, returning to the function menu.

The summary function asks you for a starting message number. It then types the summaries for all active messages from that number on, stopping when there are no more. If you want to scan the summaries for a particular value, you can do so on the following fields: date, from, to, or subject. Any character string can be searched for. For example, to retrieve all summaries which contain CACHE (our local computer club) in the subject, type:

```
S;nnn,t=CACHE
```

which will scan the summary file, starting at message *nnn*, looking for the character string CACHE. All fields which can be scanned are stored in upper case only, and all scan requests are translated to upper case.

You can also use this summary scan to see if there are any messages to you. For example:

```
S;nnn,t=ward
```

checks if I have any messages. Other typical scans are:

s=pcnet	(any messages relating to PCNET).
s=sale	(find things for sale).
f=oper	(messages from system operators).

W=TYPE WELCOME

Since the welcome message gives a list of all control characters which can be used, you might want to review it to check them again. The W command will retype the welcome message.

X=EXPERT USER

This command specifies the expert user mode to the computer. All prompt messages are shortened and, in general, the system is much more pleasant to use by the frequent user.

Accessing the System

If you would like to try out the system, all you need is a 110 or 300 bps ASCII terminal and a modem, such as the type used to communicate with a timesharing system. Dial the system at (312) 528-7141. When you hear the answer tone, connect your modem, and press return several times until the system determines your transmission rate. If you have a reasonably good telephone line, and a moderately strong modem signal, you should be able to communicate with the system. We find that if the conditions are marginal, communications are possible, but such functions as *control-C* and *control-S* do not necessarily work. Feel free to hang up and try several times if you have problems.

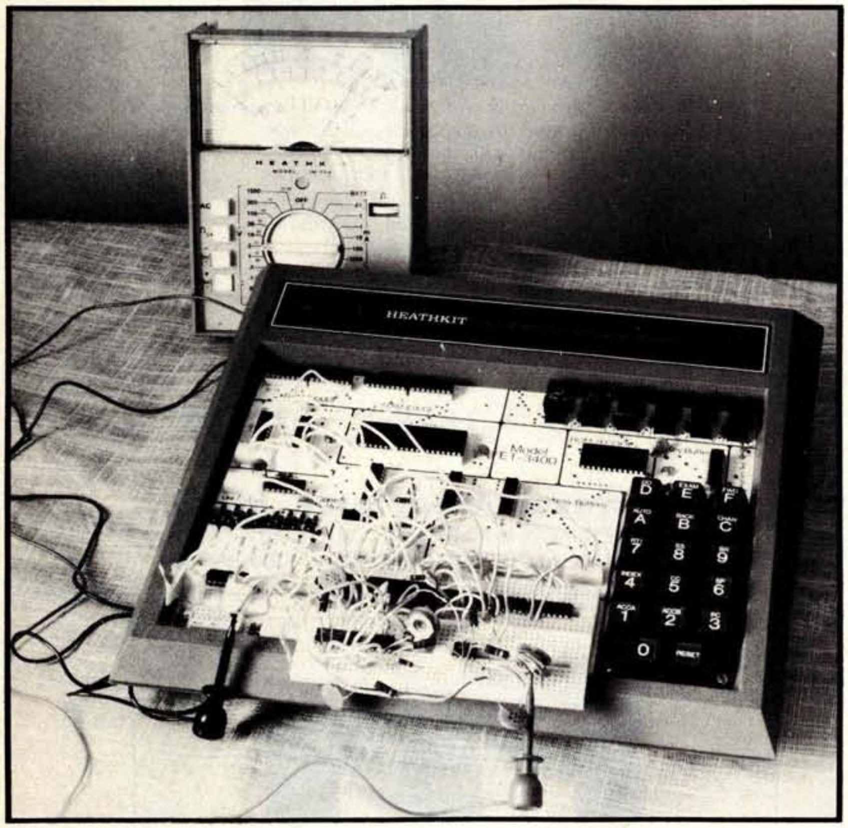
REFERENCES

DMA Inc is located at 530 Pierce Av, Dyer IN 46311.

CACHE is at POB 52, South Holland IL 60473.■

Product Description

Heath Microprocessor Training System



W N Hubin
719 Cuyahoga St
Kent OH 44240

How does a stranger gain entrance to the Valhalla of microprocessing? Who will provide bits of knowledge that logically string into byte capability? Herein lies the siren song sung by the new Heath 6800 based ET-3400 Trainer and the accompanying EE-3401 Microprocessor Course.

Heath aimed in my direction when they designed this trainer. It will appeal to anyone who has been enticed by microprocessors but has not yet bought one, a person who probably has done some high level computer programming, and one who has a working knowledge of digital circuits. Persons already caught in the web of programming are well known for their susceptibility to microprocessors, although such a background is neither assumed nor necessary for this course. I happen to be a professional proponent of the joys to be found in number crunching. Some background in digital electronics is needed (though several laboratory timer projects proved quite adequate in my case).

My order went out a few days after the Microprocessor Course and Trainer first appeared in the Heathkit catalog. The package arrived about 12 days later. The combination of course and trainer goes for $270. Don't even think of purchasing the 6800 trainer without the course unless you already can program in machine language.

There were a few clues that betrayed this as the first shipment. Some typos appeared in the text, which otherwise had the clear illustrations and clear assembly instructions that are properly associated with Heath products. Also, the holes in the printed circuit board were too small for the supplied key switches and had to be drilled larger; shortly afterwards Heath wrote to confirm the problem and the solution. Also, a memo included with the parts indicated that a few unneeded parts had been included. Compare this with the known fact that Heath normally expects a Heathkitite to show visible joy upon discovery of just a spare #8 lockwasher in his bag of goodies. Finally, the trainer refused to blink its "CPU UP" message to me after the assembly was completed. Now Heath provides a very complete troubleshooting procedure which uses the on board LEDs to check for continuity and shorts. This quickly showed up several minor assembly errors. The final problem was an address line that wasn't continuous (this ill was cured with a jumper wire). One week and about 3.5 days of work had elapsed. Another 100 plus hours of study were to pass over the dam before the course was finished.

The microprocessor course consists of:

- 469 pages of theory in a programmed learning format.
- 255 pages of concurrent experiments using the trainer.
- Two audio cassettes that go with a colorfully cartooned 230 page reference chart.
- 130 pages of manufacturers' product information sheets.

The theory is divided into sections devoted to:

- Number systems and codes.
- Microprocessor basics.
- Computer arithmetic.
- Introduction to programming.
- The processor.
- Interfacing.

It is a comprehensive and well integrated course of study. The explanations are clear and the text is replete with examples. The experiments were divided into one group of programming experiments in machine code using the hexadecimal keyboard (no assembler) and another group of interfacing experiments. The audio and chart combination is a brain tension reliever that is called on periodically to provide information on the organization of the course, various uses for and applications of microprocessors in industry, an overview of programming concepts and languages, a discussion of the various types of semiconductor memory devices, a breakdown of the steps involved in the design of a microprocessor based device, and a comparison of current microprocessors.

By the time the programming part is finished, the whole instruction set of the 6800 processor has been covered along with the provision for considerable practice in machine language programming. This is not to say that great programming proficiency was achieved from just finishing the course. No matter how cleverly I thought I had written a program (and they all ran), the official version always proved to be much more craftily constructed. The programming experiments include multiple precision and binary coded decimal arithmetic.

The interfacing section proved to be the most exciting for me. The theory and practice of interfacing with memory and displays comprise the early topics. Software key debouncing and display multiplexing are included. The interrupt capability of the processor is demonstrated by leading up to a line frequency triggered digital clock. The peripheral interface adapter (PIA) is discussed and then used to play tunes, to matrix decode a second keyboard, and to demonstrate parallel to serial conversion. A digital to analog converter is used to program various voltage waveforms and finally to simulate a digital voltmeter with a comparator circuit. This last experiment is the only one that requires additional equipment: a voltmeter and (optionally) an oscilloscope to view the pretty voltage waveforms. All the experiments worked as advertised.

As a professional educator, this course proved to be somewhat threatening. The material is so well presented that a teacher doesn't appear to be necessary. Yet real learning surely must draw brain pain, and some persons will always need a structured course with scheduled exams to force them to study, so perhaps my job is safe for a few years yet.

Minor Criticisms

The discussions after the experiments were very complete, but I would have liked more advance information before wiring a circuit on the breadboard. The integrated circuits being used should be labeled with their function (such as 3-IN NAND) as well as their TTL number. A good index would make the material much more useful for later reference.

In sum, Heath has developed an appealing and cost effective entree into the world of microprocessing. They have devised a system that is suitable for either self-instruction or individualized class instruction. The trainer lends itself to continued breadboarding of new ideas and to continued practice in machine language programming.■

A Cassette Interface Switching Box for the TRS-80

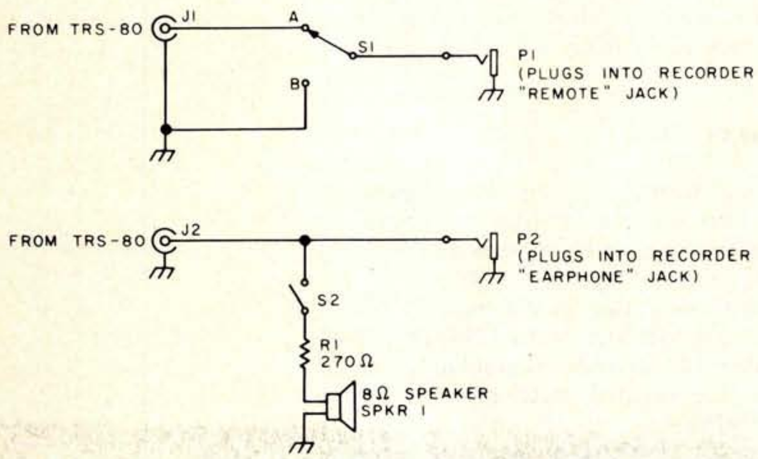


Figure 1: Cassette interface switching box for the TRS-80. The circuit enables the user to manually control the cassette recorder without having to unplug the recorder's remote jack, and allows audio monitoring of the tape while it is being loaded into the computer.

Craig Anderton
c/o BYTE

Recently, I had the chance to borrow a Radio Shack TRS-80 for the purpose of getting started in the art of writing programs. The aspect of its operation that pleased me the most was the incredibly easy start up, which involved locating a wall socket and opening the manual. In addition, I liked having a cassette recorder and interface included in the TRS-80 package, making it easy to store programs on cassette.

After working with the unit for a while, I noticed that I was saving programs with increasing frequency. Being new at the game, I would often develop a program to a certain point and then make a save. When, on subsequent passes, I wrote in some kind of garbage that crashed the program, or made some other grievous error, I always had the cassette as a point of reference for analyzing what went wrong and why.

Unfortunately, the cassette interface system has a few mechanical peculiarities that detract from the overall usefulness of the system. This article is about remedying these particular problems with a simple, very inexpensive and easy to use switching box that inserts between the recorder itself and the combined computer/keyboard unit. This box requires no modifications of any kind to the TRS-80, so you can keep your warranty intact too.

Let's identify the problems. Problem number 1 is that the recorder is controlled at all times by the computer. . .and the computer cannot tell it to do things like rewind, fast forward, etc. As a result, a means of manual control over the tape recorder is a necessity. Returning to manual control with the stock TRS-80 requires physically removing a plug which comes from the computer and mates with the recorder's "remote jack." I've never been much of an optimist regarding the reliability of tiny jacks and plugs; the repeated plugging and unplugging is not only time consuming and awkward, but it could lead to eventual reliability problems.

Problem number 2: there is no simple provision for audio monitoring of the cassette. Perhaps this doesn't seem like much of a problem—after all, the cassette is designed to talk to the computer, not to us. However, the audio monitoring is very helpful. (We'll explore this more towards the end of the article.) Monitoring with the TRS-80 involves unplugging another plug which comes from the computer and plugs into the recorder's earphone jack. Then you can listen to the cassette over the recorder's internal speaker. This means that you cannot monitor a tape while it is being loaded into the computer.

Figure 1 shows an economical solution to both problems. The cord from the keyboard that would normally feed the remote jack of the recorder plugs into J1; P1 plugs into the recorder's remote jack. With S1 in position A, the TRS-80 controls tape motion. In position B, the recorder is manually controlled.

In order to monitor, the cord from the keyboard that normally goes into the earphone jack on the recorder plugs into J2; P2 plugs into the recorder's earphone jack. S2 either connects or disconnects a small, 8 ohm transistor radio speaker from the line, with isolation from the line provided by a 270 ohm resistor. The added loading seems to have no effect on operation at all.

I constructed the circuit in a small metal box, tried to keep the leads reasonably short, and have used it for many hours with no problems. Following is an example of how to record a good cassette using this box.

Start with the cassette recorder under computer control. Before doing a save, however, I recommend identifying the program *by voice* on the tape. The TRS-80 recorder has a rubber dummy plug inserted into the microphone jack. Temporarily unplug this dummy and plug in a regular cassette microphone. Push "play" and "record": the tape won't start since the computer hasn't said anything; however, flip the switch to manual control, and the tape will run. Speak your introduction, ie: "This is the first take of computer graphics program number 5. It uses approximately 2 K bytes of memory." After recording this, flip the switch back to computer control, return the microphone jack to its original state, and type the usual CSAVE into the computer.

As the save process occurs, you will hear audio information being transferred to the cassette through the monitor speaker if S2 is set to monitor. It's interesting to compare what you hear going in with what you get out on playback. For example, you will find out about the quality of your cassette, and whether or not you're having dropout or recording problems.

Radio Shack recommends a second save for safety's sake on complex programs; a third save isn't a bad idea either. After the computer has finished saving the program, the video display signals with READY and the tape motor shuts off. Plug the microphone back in; flip S1 back to manual (without changing the record/play buttons; they are still depressed), and again, record an identification, such as, "This is take 2 of computer graphics program number 5." Save again as described above.

When it's time to load the program back into the computer to see if all went well, you'll find it's easy to find the right spot on the tape. Just flip to manual control, switch in the monitor, and juggle between rewind and play until you find the beginning of the tape program. You should hear the unambiguous voice identification, followed by the sound of data being transferred. The TRS-80 does print a couple of flashing asterisks to let you know the program is loading, but the extra audio feedback is very reassuring.

There are a couple of final notes. Due to the 270 ohm isolation resistor, you don't get much volume out of the speaker. You could lower the resistance, but that might also reduce the level of the signal going to the computer. A better option, if you need a loud signal, is to use a small audio amp integrated circuit (like the LM380). Unless you play with your computer in a very noisy environment, you probably won't have any complaints about the nice unobtrusive level you get with a 270 ohm series resistor.

I hope that if you have a Radio Shack computer, you will try this circuit. It can save a lot of time, avoid the plugging and unplugging blues, and lessen the number of human errors, all for only a few hours of work.■

Listing 1 (opposite): The first half of Chess 0.5, written in Pascal. The second half of the program will be presented in part 3 (December 1978 BYTE) of this series. The portion of the program presented here covers initialization of the program, variable declaration, manipulation of the "bit boards" (used to represent positions on the chessboard), user print routines and move generation. The second half of the listing will include procedures for evaluation of terminal positions, the look-ahead procedure, and user commands.

Creating a Chess Player Part 2: Chess 0.5

Part 1 of this series ("Creating a Chess Player," October 1978 BYTE, page 182) was an essay on human and computer skill. This month and next we present Chess 0.5, a program written in Pascal by Larry Atkin, who is coauthor with David Slate of the world championship computer chess program Chess 4.6. The program is readily adaptable to personal computers having Pascal systems such as the UCSD Pascal project software. Part 4 of the series will conclude with some thoughts about computer chess strategy.

Peter W Frey
Dept of Psychology
Northwestern University
Evanston IL 60201

Larry R Atkin
Health Information Services
542 Michigan Av
Evanston IL 60202

Note: *The Pascal subset described in "A 'Tiny' Pascal Compiler" (page 182) is not compatible with the more sophisticated Pascal used here. . . .*CM

We have attempted to incorporate several features which make the search process more efficient and others which increase the user's options. Both of these enhancements are important. The first set of features (incremental updating, iterative searching, staged move generation, etc) were described in general terms in part 1. These features reduce computation to the point where a move can be selected in a reasonable amount of time even with a full-width search. The second set of features (special control and print commands, accepting chess moves in standard notation) not only add to the pleasure of using the program, but also make the debugging process much easier. The price for these enhancements is a longer, more complicated program. We hope the length of our listing will not discourage the reader from becoming actively involved.

Pascal was developed to provide a logical and systematic higher level language which could produce reasonably efficient machine code for existing hardware. Computer programs can be conceptualized in terms of two essential parts, descriptions of data and descriptions of actions which are to be performed on the data. Pascal requires that every variable occurring in the program be introduced by a declaration statement which associates an identifier and a data type with that variable. The data type defines the set of values which may be assumed by the variable. Since a chess program involves a large number of variables, our program begins with a long list of declaration statements.

A constant definition introduces an identifier as a synonym for a constant. This is very useful since the value of the constant as stated in the declaration list can be changed at some later date, and this change will then be reflected throughout the program in every place where the constant is used. In the chess program, the values of some of the constants depend on the characteristics of the user's hardware. For example, the values of ZK (maximum search depth) and ZW (move stack limit) will reflect the amount of memory which is available on your system. On personal computers, ZX will generally be set at 7 if you have an 8 bit processor and at 15 if you have a 16 bit processor. Note also that the value of PZX8 depends on the value of ZX. To implement this program on a given computer, it is necessary to insert at the beginning of the program the appropriate values for these constants.

For the sake of clarity, specific data types are declared for a number of different chess concepts and for certain useful indices. The program also takes advantage of the different properties represented in Pascal's data structures: the set, array and record. It is unlikely that anyone will immediately memorize the names of all the variables. Therefore it is useful to have them listed at the beginning where they can easily be found for later reference.

There is a comment statement accompanying almost every instruction in the program. Although these brief statements

```
PROGRAM CHESS(INPUT,OUTPUT);

LABEL

  1,                                (* INITIALIZE FOR A NEW GAME *)
  2,                                (* EXECUTE MACHINES MOVE *)
  9;                                (* END OF PROGRAM *)


CONST

  AA = 1; ZA = 10;                  (* CHARACTERS IS A WORD *)
  AC = "A"; ZC = ",";               (* CHARACTER LIMITS *)
  AD = -21; ZD = +21;               (* DIRECTION LIMITS *)
  AJ = 0; ZJ = 73;                  (* CHARACTERS IN A STRING *)
  AK = 0; ZK = 16;                  (* SEARCH DEPTH LIMITS *)
  AKM2 = -2;                        (* AK-2 *)
  ZKP1 = 17;                        (* ZK+1 *)
  AL = 0; ZL = 119;                 (* LARGE BOARD VECTOR LIMITS *)
  AZL = -119; ZAL = 119;            (* LARGE BOARD DIFFERENCES
                                       LIMITS *)
  AN = 1; ZN = 30;                  (* MESSAGE LIMITS *)
  AS = 0; ZS = 63;                  (* BOARD VECTOR LIMITS *)
  AT = -1; ZT = 63;                 (* BOARD VECTOR LIMITS AND
                                       ANOTHER VALUE *)
  AV = -32767; ZV = +32767;         (* EVALUATION LIMITS *)
  AW = 1; ZW = 500;                 (* MOVE STACK LIMITS *)
  AX = 0; ZX = 31;                  (* SUBSETS OF SQUARES *)
  AY = 0; ZY = 1;                   (* ARRAY OF SUBSETS TO FORM A SET
                                       OF ALL SQUARES ON BOARD *)

  LPP = 20;                         (* LINES PER PAGE *)
  PZX8 = 16777216;                  (* 2^(ZX-7) *)

  SYNCF = 1;                        (* FIRST CAPTURE SYNTAX *)
  SYNCL = 36;                       (* LAST CAPTURE SYNTAX *)
  SYNMF = 37;                       (* FIRST MOVE SYNTAX *)
  SYNML = 47;                       (* LAST MOVE SYNTAX *)


TYPE

  (* SIMPLE TYPES *)

  TA = AA..ZA;                      (* INDEX TO WORDS OF CHAR *)
  TB = BOOLEAN;                     (* TRUE OR FALSE *)
  TC = CHAR;                        (* SINGLE CHARACTERS *)
  TD = AD..ZD;                      (* DIRECTIONS *)
  TE = (B1,B2,B3,B4,S1,S2,S3,S4,N1,N2,N3,N4,N5,N6,N7,N8);
                                    (* NUMBER OF DIRECTIONS *)
  TF = (F1,F2,F3,F4,F5,F6,F7,F8);   (* FILES *)
  TG = (PQ,PR,PN,PB);               (* PROMOTION PIECES *)
  TH = (H0,H1,H2,H3,H4,H5,H6,H7);
                                    (* TREE SEARCH MODES *)
  TI = INTEGER;                     (* NUMBERS *)
  TJ = AJ..ZJ;                      (* INDEX TO STRINGS *)
  TK = AK..ZK;                      (* PLY INDEX *)
  TL = AL..ZL;                      (* LARGE (10X12) BOARD *)
  TM = (LITE,DARK,NONE);            (* SIDES *)
  TN = AN..ZN;                      (* INDEX TO MESSAGES *)
  TP = (LP,LR,LN,LB,LQ,LK,DP,DR,DN,DB,DQ,DK,MT);
                                    (* PIECES: LIGHT PAWN, LIGHT
                                       ROOK, ... , DARK KING, EMPTY
                                       SQUARE *)
  TQ = (LS,LL,DS,DL);               (* QUADRANTS *)
  TR = (R1,R2,R3,R4,R5,R6,R7,R8);   (* RANKS *)
  TS = AS..ZS;                      (* SQUARES *)
  TT = AT..ZT;                      (* SQUARES, AND ANOTHER VALUE *)
  TU = (EP,ER,EN,EB,EQ,EK);         (* TYPES: PAWN, ROOK, ... ,
                                       KING *)
  TV = AV..ZV;                      (* EVALUATIONS *)
  TW = AW..ZW;                      (* MOVES INDEX *)
  TX = AX..ZX;                      (* SOME SQUARES *)
  TY = AY..ZY;                      (* NUMBER OF TX'S IN A BOARD *)
  TZ = REAL;                        (* FLOATING POINT NUMBERS *)

  (* SETS *)

  SC = SET OF AC..ZC;               (* SET OF CHARACTERS *)
  SF = SET OF TF;                   (* SET OF FILES *)
  SQ = SET OF TQ;                   (* SET OF CASTLING TYPES *)
  SR = SET OF TR;                   (* SET OF RANKS *)
  SX = SET OF TX;                   (* SET OF SOME SQUARES *)

  (* RECORDS *)

  RB = RECORD                       (* BOARDS *)
    RBTM : TM;                      (* SIDE TO MOVE *)
    RBTS : TT;                      (* ENPASSANT SQUARE *)
    RBTI : TI;                      (* MOVE NUMBER *)
    RBSQ : SQ;                      (* CASTLE FLAGS *)
    CASE INTEGER OF
      0: ( RBIS: ARRAY [TS] OF TP);     (* INDEXED BY SQUARE *)
      1: ( RBIRF: ARRAY [TR,TF] OF TP);(* INDEXED BY RANK AND FILE *)
    END;

  RA = PACKED ARRAY [TA] OF TC;     (* WORDS OF CHARACTERS *)
  RC = ARRAY [TS] OF TP;            (* BOARD VECTORS *)
  RN = PACKED ARRAY [TN] OF TC;     (* MESSAGES *)
  RJ = PACKED ARRAY [TJ] OF TC;     (* STRINGS *)

  RD = PACKED RECORD                (* SYNTAX DESCRIPTOR FOR
                                       SINGLE SQUARE *)
    RDPC : TB;                      (* PIECE *)
    RDSL : TB;                      (* / *)
    RDKQ : TB;                      (* K OR Q *)
    RDNB : TB;                      (* R, N, OR B *)
    RDRK : TB;                      (* RANK *)
    END;

  RK = RECORD                       (* KLUDGE TO FIND NEXT BIT *)
    CASE INTEGER OF
      0: (RKTB: SET OF 0..47);      (* BITS *)
      1: (RKTZ: TZ);                (* FLOATING POINT NUMBER *)
    END;

  RM = PACKED RECORD                (* MOVES *)
    RMFR : TS;                      (* FROM SQUARE *)
    RMTO : TS;                      (* TO SQUARE *)
    RMCP : TP;                      (* CAPTURED PIECE *)
    RMCA : TB;                      (* CAPTURE *)
    RMAC : TB;                      (* AFFECTS CASTLE STATUS *)
    RMCH : TB;                      (* CHECK *)
    RMMT : TB;                      (* MATE *)
    RMIL : TB;                      (* ILLEGAL *)
    RMSU : TB;                      (* SEARCHED *)
    CASE RMPR : TB OF               (* PROMOTION *)
      FALSE: (
        CASE RMOO : TB OF           (* CASTLE *)
          FALSE: (RMEP : TB);       (* ENPASSANT *)
          TRUE : (RMQS : TB);       (* QUEEN SIDE *)
             );
      TRUE : (RMPP : TG);           (* PROMOTION TYPE *)
    END;

  RS = RECORD                       (* BIT BOARDS *)
    CASE INTEGER OF
      0: (RSSS: ARRAY [TY] OF SX);  (* ARRAY OF SETS *)
      1: (RSTI: ARRAY [TY] OF TI);  (* ARRAY OF INTEGERS *)
    END;

  RX = ARRAY [TS] OF RS;            (* ATTACK MAPS *)

  RY = PACKED RECORD                (* MOVE SYNTAX DESCRIPTOR *)
    RYLS : RD;                      (* LEFT SIDE DESCRIPTOR *)
    RYCH : TC;                      (* MOVE OR CAPTURE *)
    RYRS : RD;                      (* RIGHT SIDE DESCRIPTOR *)
    END;

  RE = ARRAY [TW] OF TV;            (* ARRAY OF VALUES *)
  RF = ARRAY [TW] OF RM;            (* ARRAY OF MOVES *)


VAR

  (* DATA BASE *)

  BOARD : RB;                       (* THE BOARD *)
  NBORD : ARRAY [TS] OF TP;         (* LOOK-AHEAD BOARD *)
  ATKFR : ARRAY [TS] OF RS;         (* ATTACKS FROM A SQUARE *)
  ATKTO : ARRAY [TS] OF RS;         (* ATTACKS TO A SQUARE *)
  ALATK : ARRAY [TM] OF RS;         (* ATTACKS BY EACH COLOR *)
  TPLOC : ARRAY [TP] OF RS;         (* LOCATIONS OF PIECE BY TYPE *)
  TMLOC : ARRAY [TM] OF RS;         (* LOCATIONS OF PIECE BY COLOR *)
  MOVES : ARRAY [TW] OF RM;         (* MOVES *)
  VALUE : ARRAY [TW] OF TV;         (* VALUES *)
  ALLOC : ARRAY [TK] OF RS;         (* ALL PIECES *)
  BSTMV : ARRAY [TK] OF TW;         (* BEST MOVE SO FAR *)
  BSTVL : ARRAY [AKM2..ZKP1] OF TV; (* VALUE OF BEST MOVE *)
  CSTAT : ARRAY [TK] OF RS;         (* CASTLING SQUARES *)
  ENPAS : ARRAY [TK] OF RS;         (* ENPASSANT SQUARES *)
  GENPN : ARRAY [TK] OF RS;         (* PAWN ORIGINATION SQUARES *)
  GENTO : ARRAY [TK] OF RS;         (* MOVE DESTINATION SQUARES *)
  GENFR : ARRAY [TK] OF RS;         (* MOVE ORIGINATION SQUARES *)
  MBVAL : ARRAY [TK] OF TV;         (* MATERIAL BALANCE VALUES *)
  MVSEL : ARRAY [TK] OF TI;         (* COUNT MOVES SELECTED BY PLY *)
  INDEX : ARRAY [AK..ZKP1] OF TW;   (* CURRENT MOVE FOR PLY *)
  KILLR : ARRAY [TK] OF RM;         (* KILLER MOVES BY PLY *)
  LINDX : ARRAY [TK] OF TW;         (* LAST MOVE FOR PLY *)
  SRCHM : ARRAY [TK] OF TH;         (* SEARCH MODES *)
  GOING : TI;                       (* NUMBER OF MOVES TO EXECUTE *)
  LSTMV : RM;                       (* PREVIOUS MOVE *)
  MAXPS : TV;                       (* MAXIMUM POSITIONAL SCORE *)
  MBLTE : TV;                       (* MATERIAL BALANCE LITE EDGE *)
  MBPWN : ARRAY [TM] OF TI;         (* NUMBER OF PAWNS BY SIDE *)
  MBTOT : TV;                       (* TOTAL MATERIAL ON NBORD *)
  NODES : TI;                       (* NUMBER OF NODES SEARCHED *)

  JNTK  : TK;                       (* PLY INDEX *)
  JMTK  : TK;                       (* ITERATION *)
  JNTM  : TM;                       (* SIDE TO MOVE *)
  JNTW  : TW;                       (* MOVES STACK POINTER *)

  (* LETS *)

  FKPSHD : TI;                      (* KING PAWN SHIELD CREDIT *)
  FKSANQ : TI;                      (* KING IN SANCTUARY CREDIT *)
  FMAXMT : TI;                      (* MAXIMUM MATERIAL SCORE *)
  FNODEL : TI;                      (* NODE LIMIT FOR SEARCH *)
  FPADCR : ARRAY [TF] OF TI;        (* PAWN ADVANCE CREDIT BY FILE *)
  FPBLOK : TI;                      (* PAWN BLOCKED PENALTY *)
  FPCONN : TI;                      (* PAWN CONNECTED CREDIT *)
  FPFLNX : TI;                      (* PAWN PHALANX CREDIT *)
  FRDUBL : TI;                      (* DOUBLED ROOK CREDIT *)
  FRK7TH : TI;                      (* ROOK ON SEVENTH CREDIT *)
  FTRADE : TI;                      (* TRADE-DOWN BONUS FACTOR *)
  FTRDSL : TI;                      (* TRADE-DOWN TUNING FACTOR *)
  FTRPOK : TI;                      (* PAWN TRADE-DOWN RELAXATION *)
  FTRPWN : TI;                      (* PAWN TRADE-DOWN FACTOR *)
  FWKING : TI;                      (* KING EVALUATION WEIGHT *)
  FWMAJM : TI;                      (* MAJOR PIECE MOBILITY WEIGHT *)
  FWMINM : TI;                      (* MINOR PIECE MOBILITY WEIGHT *)
  FWPAWN : TI;                      (* PAWN EVALUATION WEIGHT *)
  FWROOK : TI;                      (* ROOK EVALUATION WEIGHT *)
  WINDOW : TI;                      (* SIZE OF ALPHA-BETA WINDOW *)

  (* SWITCHES *)

  SWEC : TB;                        (* ECHO INPUT *)
  SWPA : TB;                        (* PAGING *)
  SWPS : TB;                        (* PRINT PRELIMINARY SCORES *)
  SWRE : TB;                        (* REPLY WITH MOVE *)
  SWSU : TB;                        (* PRINT STATISTICS SUMMARY *)
  SWTR : TB;                        (* TRACE TREE SEARCH *)

  (* COMMAND PROCESSING DATA *)

  ICARD : RJ;                       (* INPUT CARD IMAGE *)
  ILINE : RJ;                       (* CURRENT COMMAND *)
  JMTJ  : TJ;                       (* CURRENT INPUT LINE POSITION *)
  JNTJ  : TJ;                       (* CURRENT COMMAND POSITION *)
```

Listing 1, continued:

```
   MOVMS : RM;                           (* MOVE MESSAGE *)

   (* TRANSLATION TABLES *)

   XSPB : ARRAY [TP] OF TB;              (* TRUE FOR SWEEP PIECES *)
   XFPE : ARRAY [TP] OF TE;              (* FIRST DIRECTION *)
   XLLD : ARRAY [AZL..ZAL] OF TD;        (* DIRECTION FOR LARGE BOARD
                                            SQUARE DIFFERENCES *)
   XLPE : ARRAY [TP] OF TE;              (* LAST DIRECTION *)
   XRFS : ARRAY [TF] OF RS;              (* BIT BOARD FOR FILES *)
   XRRS : ARRAY [TR] OF RS;              (* BIT BOARD FOR RANKS *)
   XNFS : ARRAY [TF] OF RS;              (* COMP BIT BOARD FOR FILES *)
   XNRS : ARRAY [TR] OF RS;              (* COMP BIT BOARD FOR RANKS *)
   XRSS : ARRAY [TS] OF RS;              (* BIT BOARD FOR 8X8 INDEX *)
   XRQM : ARRAY [TQ] OF RM;              (* MOVES FOR CASTLE TYPES *)
   XSQS : ARRAY [TQ] OF RS;              (* BIT BOARD FOR CASTLE TYPES *)
   XSSX : ARRAY [TS] OF SX;              (* SET ELEMENT FOR 8X8 INDEX *)
   XTBC : ARRAY [TB] OF TC;              (* CHARACTERS FOR BOOLEANS *)
   XTED : ARRAY [TE] OF TD;              (* DIRECTION NUMBER TO 10X12
                                            SQUARE DIFFERENCE *)
   XTGC : ARRAY [TG] OF TC;              (* CHARACTERS FOR PROMOTION *)
   XTGMP: ARRAY [TG,TM] OF TP;           (* PIECE FOR PROMOTION TYPE
                                            AND COLOR *)
   XTLS : ARRAY [TL] OF TT;              (* 8X8 INDEX FOR 10X12 INDEX *)
   XTMA : ARRAY [TM] OF RA;              (* WORDS FOR COLORS *)
   XTMQ : ARRAY [TM] OF TQ;              (* CASTLE TYPES FOR SIDE *)
   XTMV : ARRAY [TM] OF TV;              (* SCORE FACTOR FOR SIDE *)
   XTPC : ARRAY [TP] OF TC;              (* CHARACTERS FOR PIECES *)
   XTPM : ARRAY [TP] OF TM;              (* SIDES FOR PIECES *)
   XTPU : ARRAY [TP] OF TU;              (* TYPE FOR PIECE *)
   XTPV : ARRAY [TP] OF TV;              (* VALUES OF PIECES *)
   XTQA : ARRAY [TQ] OF RA;              (* WORDS FOR CASTLES *)
   XTQS : ARRAY [TQ] OF TS;              (* TO SQUARES FOR CASTLE TYPES *)
   XTRFS: ARRAY [TR,TF] OF TS;           (* 8X8 INDEX FOR RANK AND FILE *)
   XTSF : ARRAY [TS] OF TF;              (* FILES FOR SQUARES *)
   XTSL : ARRAY [TS] OF TL;              (* 10X12 INDEX FOR 8X8 INDEX *)
   XTSR : ARRAY [TS] OF TR;              (* RANKS FOR SQUARES *)
   XTSX : ARRAY [TS] OF TX;              (* ELEMENT NUMBER FOR 8X8
                                            INDEX *)
   XTSY : ARRAY [TS] OF TY;              (* ARRAY SUBSCRIPT INTO BIT BOARD
                                            FOR 8X8 INDEX *)
   XTUC : ARRAY [TU] OF TC;              (* CHARACTER FOR TYPE *)
   XTUMP: ARRAY [TU,TM] OF TP;           (* PIECE FOR TYPE AND SIDE *)

   XRQSO: ARRAY [TQ] OF RS;              (* UNOCCUPIED SQUARES FOR
                                            CASTLING *)
   XRQSA: ARRAY [TQ] OF RS;              (* UNATTACKED SQUARES FOR
                                            CASTLING *)

   EDGE : ARRAY [TE] OF RS;              (* EDGES IN VARIOUS DIRECTIONS *)
   CORNR: RS;                            (* KING SANCTUARY *)
   NULMV: RM;                            (* NULL MOVE *)
   OTHER: ARRAY [TM] OF TM;              (* OTHER COLOR *)
   SYNTX: ARRAY[SYNCF..SYNML] OF RY;     (* MOVE SYNTAX TABLE *)


FUNCTION MAX(A,B:TI):TI;                 (* LARGER OF TWO NUMBERS *)

BEGIN
  IF A > B THEN
    MAX := A
  ELSE
    MAX := B;
END;  (* MAX *)


FUNCTION MIN(A,B:TI):TI;                 (* SMALLER OF TWO NUMBERS *)

BEGIN
  IF A < B THEN
    MIN := A
  ELSE
    MIN := B;
END;  (* MIN *)


FUNCTION SIGN(A,B:TI):TI;                (* SIGN OF B APPLIED TO
                                            ABSOLUTE VALUE OF A *)

BEGIN
  SIGN := TRUNC(B/ABS(B)) * ABS(A);
END;  (* SIGN *)


PROCEDURE SORTIT                         (* SORT PRELIMINARY SCORES *)
  (VAR A:RE;                             (* ARRAY OF SCORES *)
   VAR B:RF;                             (* ARRAY OF MOVES *)
   C:TW);                                (* NUMBER OF ENTRIES *)

VAR
  INTB : TB;                             (* LOOP EXIT FLAG *)
  INTW : TW;                             (* OUTER LOOP INDEX *)
  INTI : TI;                             (* INNER LOOP INDEX *)
  INTV : TV;                             (* HOLD SCORE *)
  INRM : RM;                             (* HOLD MOVE *)

BEGIN
  FOR INTW := AW+2 TO C DO
  BEGIN
    INTI := INTW - 1;
    INTV := A[INTW];
    INRM := B[INTW];
    INTB := TRUE;
    WHILE (INTI > AW) AND INTB DO
      IF INTV < A[INTI] THEN
      BEGIN
        A[INTI+1] := A[INTI];
        B[INTI+1] := B[INTI];
        INTI := INTI - 1;
      END
      ELSE
```

```
        INTB := FALSE;                   (* EXIT *)
    A[INTI+1] := INTV;
    B[INTI+1] := INRM;
  END;
END;  (* SORTIT *)


PROCEDURE ANDRS                          (* INTERSECTION OF TWO BIT
                                            BOARDS *)
  (VAR C:RS;                             (* RESULT *)
   A, B:RS);                             (* OPERANDS *)

VAR
  INTY : TY;                             (* BIT BOARD WORD INDEX *)

BEGIN
  FOR INTY := AY TO ZY DO
    C.RSSS[INTY] := A.RSSS[INTY] * B.RSSS[INTY];
END;  (* ANDRS *)

PROCEDURE CLRRS                          (* REMOVE SQUARE FROM BIT
                                            BOARD *)
  (VAR C:RS;                             (* BIT BOARD *)
   A:TS);                                (* SQUARE TO REMOVE *)

BEGIN
  C.RSSS[XTSY[A]] := C.RSSS[XTSY[A]] - XSSX[A];
END;  (* CLRRS *)


PROCEDURE CPYRS                          (* COPY OF A BIT BOARD *)
  (VAR C:RS;                             (* RESULT *)
   A:RS);                                (* OPERAND *)

VAR
  INTY : TY;                             (* BIT BOARD WORD INDEX *)

BEGIN
  FOR INTY := AY TO ZY DO
    C.RSSS[INTY] := A.RSSS[INTY];
END;  (* CPYRS *)


PROCEDURE IORRS                          (* UNION OF TWO BIT BOARDS *)
  (VAR C:RS;                             (* RESULT *)
   A, B:RS);                             (* OPERANDS *)

VAR
  INTY : TY;                             (* BIT BOARD WORD INDEX *)

BEGIN
  FOR INTY := AY TO ZY DO
    C.RSSS[INTY] := A.RSSS[INTY] + B.RSSS[INTY];
END;  (* IORRS *)


PROCEDURE NEWRS                          (* CLEAR BIT BOARD *)
  (VAR A:RS);                            (* BIT BOARD TO CLEAR *)

VAR
  INTY : TY;                             (* BIT BOARD WORD INDEX *)

BEGIN
  FOR INTY := AY TO ZY DO
    A.RSSS[INTY] := [];
END;  (* NEWRS *)


PROCEDURE NOTRS                          (* COMPLEMENT OF A BIT BOARD *)
  (VAR C:RS;                             (* RESULT *)
   A:RS);                                (* OPERAND *)

VAR
  INTY : TY;                             (* BIT BOARD WORD INDEX *)

BEGIN
  FOR INTY := AY TO ZY DO
    C.RSSS[INTY] := [AX..ZX]-A.RSSS[INTY];
END;  (* NOTRS *)


FUNCTION NXTTS                           (* NEXT ELEMENT IN BIT BOARD *)
  (VAR A:RS;                             (* BIT BOARD TO LOCATE FIRST
                                            SQUARE, AND THEN REMOVE *)
   VAR B:TS                              (* SQUARE NUMBER OF FIRST SQUARE
                                            IN BIT BOARD *)
   ):TB;                                 (* TRUE IFF ANY SQUARES WERE SET
                                            INITIALLY *)

LABEL
  11;                                    (* RETURN *)

VAR
  INTX : TX;                             (* BIT BOARD BIT INDEX *)
  INTY : TY;                             (* BIT BOARD WORD INDEX *)
  X : RK;                                (* KLUDGE WORD *)

BEGIN
  FOR INTY := ZY DOWNTO AY DO            (* LOOP THRU BIT BOARD WORDS *)
    IF A.RSTI[INTY] <> 0 THEN
    BEGIN

(*** BEGIN CDC 6000 DEPENDANT CODE *)
(*** FOLLOWING CODE REQUIRES THE 'EXPO' FUNCTION TO RETURN
(*** THE EXPONENT FROM A FLOATING POINT NUMBER.  IT ALSO ASSUMES
(*** THAT FLOATING POINT NUMBERS HAVE 48 BIT COEFFICIENTS RIGHT-
(*** JUSTIFIED IN A WORD, AND THAT SETS ARE RIGHT-JUSTIFIED IN
(*** A WORD. *)
(*     X.RKTZ := A.RSTI[INTY];           (* FLOAT WORD *)
(*     B := EXPO(X.RKTZ) + INTY * (ZX+1);
(*                                       (* CONVERT TO SQUARE NUMBER *)
```

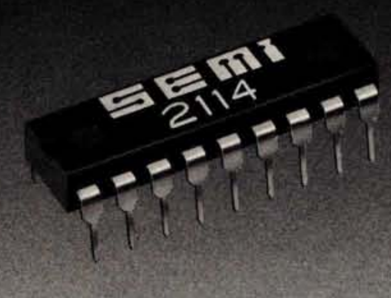
may not initially be very meaningful, we expect them to be helpful when the user becomes familiar with the program. Because Pascal requires that all procedures and functions be defined in the serial listing before they are called by another portion of the program, the procedures and functions which are first defined tend to be primitives. The main part of the program is concentrated at the end of the listing.

The most important part of the variable declaration list in terms of understanding the program is the portion which specifies the global data base. This includes the current board (BOARD, a record) and a number of important arrays. The look-ahead board (NBORD) is an array listing the piece occupying each square. The attacks emanating from each square are represented by ATKFR, an array which lists an 8 by 8 bit board for each of the 64 squares. The attacks to each square are represented by a similar array, ATKTO. The combined attacks for each side are represented by a 2 item array of 8 by 8 bit boards called ALATK.

The location of all pieces by type is represented by an array of 12 8 by 8 bit boards, TPLOC. The location of all pieces by color is represented by an array of two 8 by 8 bit boards, TMLOC. The moves are stored in an array (MOVES) of records. Each record (RM) contains information about the from square, to square; whether a capture is involved and the type of piece captured, whether the move affects castle status, involves check or mate, involves a piece promotion, and whether the move has been searched yet. Additional arrays provide information on castling squares, en passant squares, the location of all pieces, the location of pawns, etc. To be successful, a chess program must organize the data base in a logical manner and be able to manipulate it efficiently.

For reasons of efficiency, the program often stores the same information in two or more different ways. Because of this, it is necessary to be able to translate from one form to the other. These activities are handled by special arrays. For example, the XTPC array allows one to use a piece designator (LP, LQ, LK, DQ, etc) as an index and returns the corresponding character (1 thru 6 for Black pieces and A thru F for White pieces) which is used when a board representation is printed on the terminal.

There are several general purpose routines which are needed by the program. Two functions, MIN and MAX, provide the smaller or larger of two numbers upon request. A third function, SIGN, applies the sign of one number to the absolute value of another



Listing 1, continued:

```
(*     X.RKTB := X.RKTB - [47];            (* REMOVE MOST SIGNIFICANT BIT *)
(*     A.RSTI[INTY] := TRUNC(X.RKTZ);      (* INTEGERIZE *)
(*     NXTTS := TRUE;                      (* RETURN A BIT SET *)
(*     GOTO 11;                            (* RETURN *)
(*** END CDC 6000 DEPENDANT CODE *)

(*** BEGIN MACHINE INDEPENDENT CODE *)
    FOR INTX := ZX DOWNTO AX DO            (* LOOP THROUGH BITS IN WORD OF
                                              SET *)

      IF INTX IN A.RSSS[INTY] THEN
      BEGIN
        B := INTX+INTY*(ZX+1);             (* RETURN SQUARE NUMBER *)
        A.RSSS[INTY] := A.RSSS[INTY] - [INTX];
                                           (* REMOVE BIT FROM WORD *)
        NXTTS := TRUE;                     (* RETURN A BIT SET *)
        GOTO 11;                           (* RETURN *)
      END;
(*** END MACHINE INDEPENDENT CODE *)

    END;
  NXTTS := FALSE;                          (* ELSE RETURN NO BITS SET *)
11:  (* RETURN *)
END;  (* NXTTS *)


FUNCTION CNTRS                             (* COUNT MEMBERS OF A BIT
                                              BOARD *)
  (A:RS):TS;                               (* BIT BOARD TO COUNT *)

VAR
  INTY : TY;                               (* BIT BOARD WORD INDEX *)
  INTS : TS;                               (* TEMPORARY *)
  IMRS : RS;                               (* SCRATCH *)
  IMTS : TS;                               (* SCRATCH *)

BEGIN
  INTS := 0;

(*** BEGIN MACHINE INDEPENDENT CODE *)
  CPYRS(IMRS,A);
  WHILE NXTTS(IMRS,IMTS) DO
    INTS := INTS+1;                        (* COUNT SQUARES *)
(*** END MACHINE INDEPENDENT CODE *)

(*** BEGIN CDC 6000 DEPENDENT CODE *)
(*** FOLLOWING CODE REQUIRES THE 'CARD' FUNCTION TO
(*** COUNT THE MEMBERS IN A SET. *)
(*FOR INTY := AY TO ZY DO
(*  INTS := INTS + CARD(A.RSSS[INTY]);
(*** END CDC DEPENDENT CODE *)

  CNTRS := INTS;                           (* RETURN SUM *)
END;  (* CNTRS *)


PROCEDURE SETRS                            (* INSERT SQUARE INTO BIT
                                              BOARD *)
  (VAR C:RS;                               (* BIT BOARD *)
   A:TS);                                  (* SQUARE TO INSERT *)

BEGIN
  C.RSSS[XTSY[A]] := C.RSSS[XTSY[A]] + XSSX[A];
END;  (* SETRS *)


PROCEDURE SFTRS                            (* SHIFT BIT BOARD *)
  (VAR A:RS;                               (* RESULT *)
   B:RS;                                   (* SOURCE *)
   C:TE);                                  (* DIRECTION *)

VAR
  INRS : RS;                               (* SCRATCH *)
  INTS : TS;                               (* SCRATCH *)
  INTY : TY;                               (* BIT BOARD WORD INDEX *)

BEGIN

(*** BEGIN MACHINE INDEPENDENT CODE *)
  NEWRS(A);                                (* CLEAR NEW BIT BOARD *)
  WHILE NXTTS(B,INTS) DO
    IF XTLS[XTSL[INTS]+XTED[C]] > 0 THEN
                                           (* SHIFT EACH BIT *)
      SETRS(A,XTLS[XTSL[INTS]+XTED[C]]);
(*** END MACHINE INDEPENDENT CODE *)
(*** BEGIN CDC 6000 DEPENDENT CODE *)
(*** FOLLOWING CODE ASSUMES THAT MULTIPLICATION OR DIVISION
(*** BY A CONSTANT POWER OF 2 IS DONE WITH A SHIFT INSTRUCTION. *)
(*CASE C OF
(*S1:  BEGIN
(*        FOR INTY := AY TO ZY DO          (* SHIFT ONE PLACE *)
(*        BEGIN
(*          B.RSSS[INTY] := B.RSSS[INTY] - EDGE[S1].RSSS[INTY];
(*          A.RSTI[INTY] := B.RSTI[INTY] DIV 2;
(*        END;
(*     END;
(*S2:  BEGIN
(*        FOR INTY := AY TO ZY DO          (* SHIFT WORDS *)
(*        BEGIN
(*          B.RSSS[INTY] := B.RSSS[INTY] - EDGE[S2].RSSS[INTY];
(*          INRS.RSSS[INTY] := B.RSSS[INTY] * [ZX-7..ZX];
(*          A.RSSS[INTY] := B.RSSS[INTY] - [ZX-7..ZX];
(*          A.RSTI[INTY] := A.RSTI[INTY] * 256
(*        END;
(*        FOR INTY := AY+1 TO ZY DO        (* CARRY BETWEEN WORDS *)
(*          A.PSTI[INTY] := A.RSTI[INTY] + INRS.RSTI[INTY-1] DIV PZX8;
(*     END;
(*S3:  BEGIN
(*        FOR INTY := AY TO ZY DO          (* SHIFT ONE PLACE *)
(*        BEGIN
(*          A.RSSS[INTY] := B.RSSS[INTY] - EDGE[S3].RSSS[INTY];
(*          A.RSTI[INTY] := A.RSTI[INTY] * 2;
(*        END;
(*     END;
(*S4:  BEGIN
(*        FOR INTY := AY TO ZY DO          (* SHIFT WORDS *)
(*        BEGIN
(*          B.RSSS[INTY] := B.RSSS[INTY] - EDGE[S4].RSSS[INTY];
(*          INRS.RSSS[INTY] := B.RSSS[INTY] * [AX..AX+7];
(*          A.RSTI[INTY] := B.RSTI[INTY] DIV 256;
(*        END;
(*        FOR INTY := AY TO ZY-1 DO        (* CARRY BETWEEN WORDS *)
(*          A.RSTI[INTY] := A.RSTI[INTY] + INRS.RSTI[INTY+1] * PZX8;
(*     END;
(*B1:  BEGIN
(*        SFTRS(INRS,B,S1);
(*        SFTRS(A,INRS,S2);
(*     END;
(*B2:  BEGIN
(*        SFTRS(INRS,B,S2);
(*        SFTRS(A,INRS,S3);
(*     END;
(*B3:  BEGIN
(*        SFTRS(INRS,B,S3);
(*        SFTRS(A,INRS,S4);
(*     END;
(*B4:  BEGIN
(*        SFTRS(INRS,B,S4);
(*        SFTRS(A,INRS,S1);
(*     END;
(*N1:  BEGIN
(*        SFTRS(INRS,B,B1);
(*        SFTRS(A,INRS,S2);
(*     END;
(*N2:  BEGIN
(*        SFTRS(INRS,B,B2);
(*        SFTRS(A,INRS,S2);
(*     END;
(*N3:  BEGIN
(*        SFTRS(INRS,B,B2);
(*        SFTRS(A,INRS,S3);
(*     END;
(*N4:  BEGIN
(*        SFTRS(INRS,B,B3);
(*        SFTRS(A,INRS,S3);
(*     END;
(*N5:  BEGIN
(*        SFTRS(INRS,B,B3);
(*        SFTRS(A,INRS,S4);
(*     END;
(*N6:  BEGIN
(*        SFTRS(INRS,B,B4);
(*        SFTRS(A,INRS,S4);
(*     END;
(*N7:  BEGIN
(*        SFTRS(INRS,B,B4);
(*        SFTRS(A,INRS,S1);
(*     END;
(*N8:  BEGIN
(*        SFTRS(INRS,B,B1);
(*        SFTRS(A,INRS,S1);
(*     END;
(*END;
(*** END CDC 6000 DEPENDENT CODE *)

END;  (* SFTRS *)


FUNCTION INRSTB                            (* SQUARE IN BIT BOARD BOOLEAN *)
  (A:RS;                                   (* BIT BOARD *)
   B:TS):TB;                               (* SQUARE IN QUESTION *)

BEGIN
  INRSTB := XSSX[B] <= A.RSSS[XTSY[B]];
END;  (* INRSTB *)


FUNCTION NULRS                             (* NULL BIT BOARD *)
  (A:RS)                                   (* BIT BOARD TO CHECK *)
  :TB;                                     (* TRUE IF BIT BOARD EMPTY *)

VAR
  INTY : TY;                               (* BIT BOARD WORD INDEX *)
  INTB : TB;                               (* TEMPORARY VALUE *)

BEGIN
  INTB := TRUE;
  FOR INTY := AY TO ZY DO
    INTB := INTB AND (A.RSTI[INTY] = 0);
  NULRS := INTB;
END;  (* NULRS *)


FUNCTION NULMVB                            (* NULL MOVE BOOLEAN *)
  (A:RM)                                   (* MOVE TO TEST *)
   :TB;                                    (* TRUE IF NULL MOVE *)
BEGIN
  WITH A DO
    NULMVB := RMAC AND RMPR AND (NOT RMCA);
END;  (* NULMVB *)


PROCEDURE INICON;                          (* INITIALIZE GLOBAL CONSTANTS *)

VAR
  INTD : TD;                               (* DIRECTION INDEX *)
  INTE : TE;                               (* DIRECTION *)
  INTF : TF;                               (* FILE INDEX *)
  INTI : TI;                               (* SCRATCH *)
  INTL : TL;                               (* LARGE BOARD INDEX *)
  INTQ : TQ;                               (* CASTLE TYPE INDEX *)
  INTR : TR;                               (* RANK INDEX *)
  INTT : TT;                               (* SQUARE INDEX *)
  INTX : TX;                               (* SET ELEMENT INDEX *)
  INTY : TY;                               (* BIT BOARD WORD INDEX *)
  IMTI : TI;                               (* SCRATCH *)
  INRS : RS;                               (* SCRATCH *)
```

number. A general purpose sort routine, SORTIT, is also provided.

Manipulating the Bit Boards

There are a number of primitive operations which involve the manipulation of information represented in bit board form. A bit board is one or more computer words which have a bit set in specific locations to represent the occurrence or nonoccurrence of a particular event. For example eight 8 bit words can be used to represent the eight rows of a chessboard. Each bit corresponds to one square. To represent the location of all White pawns, a bit is set (ie: 1) in the proper locations and all other locations remain clear (ie: 0). This method for representing and manipulating information is very useful in chess programming. For this reason, the first actions defined by our chess program are a set of procedures and functions for manipulating bit boards.

The actions represented are:

(1) the intersection of two bit boards (ANDRS);
(2) the union of two bit boards (IORRS);
(3) the complement of a bit board (NOTRS);
(4) setting a bit in a bit board (SETRS);
(5) removing a bit from a bit board (CLRRS);
(6) counting the number of bits that are set on a bit board (CNTRS);
(7) making a copy of a bit board (CPYRS);
(8) setting all bits to 0 (NEWRS);
(9) shifting all bits in a particular direction (SFTRS);
(10) determining whether a particular bit is set (INRSTB);
(11) determining whether a bit board is empty, ie: has no bits set (NULRS); and
(12) finding and reporting integer value for a location where a bit is set (NXTTS).

Since these routines are used repeatedly by the program, you can decrease the move calculation time quite a bit by implementing these primitives in assembly language. You will note that the function NXTTS is written in two ways: machine independent code, and code which is compatible only with the Control Data 6000 series machines. There are a number of places in the program where execution time can be enhanced by substituting machine dependent code which takes advantage of one or more special features of the hardware you are using. It would be helpful, also, if functions in Pascal could return an array or record instead of just a single value. There are many places in the program where this type of function would be more logical and more efficient than using a procedure (ie: subroutine). If one were to consider the best of all possible worlds, it would be especially nice if the bit map manipulations could be compiled in line. With the Pascal arrangement, many of the procedure calls take as much time as the execution of the procedure.

Initial Steps

It is also necessary at the beginning of the program to provide values for the variables which define the chess environment, such as piece characteristics. For example, a White pawn is represented as LP for some purposes and as the letter A for other purposes. It has the color LITE, is not a sweep piece, and moves only in certain directions. It is necessary to initialize the translation tables, the constant and variable 8 by 8 bit boards, and a number of other tables. The three routines which are called to do this when the program is first activated are INISYN, INIXTP and INICON. A fourth procedure (INITAL) is called by the main program to get ready for a new game. It will be called

Listing 1, continued:

```
  PROCEDURE INISYN                           (* INITIALIZE MOVE SYNTAX
                                                TABLE ENTRY *)
    (A:RA);                                  (* MOVE SYNTAX *)

  BEGIN
    WITH SYNTX[INTI] DO
    BEGIN
      WITH RYLS DO
      BEGIN
        RDPC := TRUE;
        RDSL := A[AA+0] <> " ";
        RDKQ := A[AA+1] <> " ";
        RDNB := A[AA+2] <> " ";
        RDRK := A[AA+3] <> " ";
      END;
      RYCH := A[AA+4];
      WITH RYRS DO
      BEGIN
        RDPC := A[AA+5] <> " ";
        RDSL := A[AA+6] <> " ";
        RDKQ := A[AA+7] <> " ";
        RDNB := A[AA+8] <> " ";
        RDRK := A[AA+9] <> " ";
      END;
    END;
    INTI := INTI+1;
  END;  (* INISYN *)

  PROCEDURE INIXTP                           (* INITIALIZE PIECE TRANSLATION
                                                TABLES *)
    (A : TP;                                 (* PIECE TO BE TRANSLATED *)
     B : TC;                                 (* DISPLAY EQUIVALENT *)
     C : TM;                                 (* COLOR OF PIECE *)
     D : TU;                                 (* TYPE OF PIECE *)
     E : TB;                                 (* TRUE IF SWEEP PIECE *)
     F : TE;                                 (* FIRST DIRECTION OF MOVEMENT *)
     G : TE;                                 (* LAST DIRECTION OF MOVEMENT *)
     H : TV);                                (* VALUE OF PIECE *)

  BEGIN
    XTPC[A] := B;
    XTPM[A] := C;
    XSPB[A] := E;
    XFPE[A] := F;
    XLPE[A] := G;
    XTPU[A] := D;
    XTPV[A] := H;
    IF A <> MT THEN
      XTUMP[D,C] := A;
  END;  (* INIXTP *)

BEGIN  (* INICON *)

  (** INITIALIZE PIECE CHARACTERISTICS *)

  INIXTP(LP,"A",LITE,EP,FALSE,B1,B2,1*64);
  INIXTP(LR,"B",LITE,ER,TRUE ,S1,S4,5*64);
  INIXTP(LN,"C",LITE,EN,FALSE,N1,N8,3*64);
  INIXTP(LB,"D",LITE,EB,TRUE ,B1,B4,3*64);
  INIXTP(LQ,"E",LITE,EQ,TRUE ,B1,S4,9*64);
  INIXTP(LK,"F",LITE,EK,FALSE,B1,S4,0);
  INIXTP(DP,"1",DARK,EP,FALSE,B3,B4,-1*64);
  INIXTP(DR,"2",DARK,ER,TRUE ,S1,S4,-5*64);
  INIXTP(DN,"3",DARK,EN,FALSE,N1,N8,-3*64);
  INIXTP(DB,"4",DARK,EB,TRUE ,B1,B4,-3*64);
  INIXTP(DQ,"5",DARK,EQ,TRUE ,B1,S4,-9*64);
  INIXTP(DK,"6",DARK,EK,FALSE,B1,S4,0);
  INIXTP(MT,"-",NONE,EP,FALSE,B2,B1,0);

  XTGMP[PQ,LITE] := LQ;  XTGMP[PQ,DARK] := DQ;  XTGC[PQ] := "Q";
  XTGMP[PR,LITE] := LR;  XTGMP[PR,DARK] := DR;  XTGC[PR] := "R";
  XTGMP[PN,LITE] := LN;  XTGMP[PN,DARK] := DN;  XTGC[PN] := "N";
  XTGMP[PB,LITE] := LB;  XTGMP[PB,DARK] := DB;  XTGC[PB] := "B";

  XTUC[EK] := "K";
  XTUC[EQ] := "Q";
  XTUC[ER] := "R";
  XTUC[EN] := "N";
  XTUC[EB] := "B";
  XTUC[EP] := "P";

  (** INITIALIZE OTHER CONSTANTS *)

  XTBC[FALSE] := "-";
  XTBC[TRUE ] := "*";

  OTHER[LITE] := DARK;  XTMV[LITE] :=  1;
  OTHER[DARK] := LITE;  XTMV[DARK] := -1;
  OTHER[NONE] := NONE;

  XTMA[LITE] := "    WHITE ";
  XTMA[DARK] := "    BLACK ";
  XTMA[NONE] := "   NO ONE ";

  XTQA[LS] := "WHITE KING";
  XTQA[LL] := "WHITE LONG";
  XTQA[DS] := "BLACK KING";
  XTQA[DL] := "BLACK LONG";

  (** INITIALIZE 10X12 TO 8X8 AND 8X8 TO 10X12 TRANSLATION TABLES *)

  FOR INTL := AL TO ZL DO                    (* LOOP THROUGH LARGE BOARD *)
    XTLS[INTL] := -1;                        (* PRESET ARRAY TO OFF BOARD *)

  INTL := 21;                                (* INDEX OF FIRST SQUARE ON LARGE
                                                BOARD *)
  INTT := -1;                                (* INDEX OF FIRST SQUARE ON SMALL
                                                BOARD *)
  FOR INTR := R1 TO R8 DO                    (* LOOP THROUGH RANKS *)
  BEGIN
    FOR INTF := F1 TO F8 DO                  (* LOOP THROUGH FILES *)
    BEGIN
      INTT := INTT+1;                        (* ADVANCE SMALL BOARD INDEX *)
      XTRFS[INTR,INTF] := INTT;              (* SET MATRIX TO VECTOR
                                                TRANSLATION *)
      XTLS[INTL] := INTT;                    (* SET LARGE BOARD TRANSLATION
                                                TABLE WITH SMALL BOARD
                                                INDEX *)
      XTSL[INTT] := INTL;                    (* SET SMALL BOARD TRANSLATION
                                                TABLE WITH LARGE BOARD
                                                INDEX *)
      XTSR[INTT] := INTR;                    (* SET RANK OF SQUARE *)
      XTSF[INTT] := INTF;                    (* SET FILE OF SQUARE *)
      INTL := INTL+1;                        (* ADVANCE LARGE BOARD INDEX *)
    END;
    INTL := INTL+2;                          (* ADVANCE LARGE BOARD INDEX TO
                                                SKIP BORDER *)
  END;

  (** INITIALIZE 8X8 TO BIT BOARD TABLES *)

  INTT := -1;
  FOR INTY := AY TO ZY DO
  BEGIN
    FOR INTX := AX TO ZX DO
    BEGIN
      INTT := INTT+1;
      XTSX[INTT] := INTX;
      XTSY[INTT] := INTY;
      XSSX[INTT] := [INTX];
      NEWRS(XRSS[INTT]);
      XRSS[INTT].RSSS[INTY] := [INTX];
    END;
  END;

  (** INITIALIZE CONSTANT BIT BOARDS *)

  FOR INTR := R1 TO R8 DO
    NEWRS(XRRS[INTR]);

  FOR INTF := F1 TO F8 DO
    NEWRS(XRFS[INTF]);

  FOR INTR := R1 TO R8 DO
    FOR INTF := F1 TO F8 DO
    BEGIN
      SETRS(XRRS[INTR],XTRFS[INTR,INTF]);
      SETRS(XRFS[INTF],XTRFS[INTR,INTF]);
    END;

  FOR INTF := F1 TO F8 DO
    NOTRS(XNFS[INTF],XRFS[INTF]);

  FOR INTR := R1 TO R8 DO
    NOTRS(XNRS[INTR],XRRS[INTR]);

  (** INITIALIZE EDGES *)

  CPYRS(EDGE[S1],XRFS[F1]);
  CPYRS(EDGE[S2],XRRS[R8]);
  CPYRS(EDGE[S3],XRFS[F8]);
  CPYRS(EDGE[S4],XRRS[R1]);
  IORRS(EDGE[B1],EDGE[S1],EDGE[S2]);
  IORRS(EDGE[B2],EDGE[S2],EDGE[S3]);
  IORRS(EDGE[B3],EDGE[S3],EDGE[S4]);
  IORRS(EDGE[B4],EDGE[S4],EDGE[S1]);
  IORRS(EDGE[N1],EDGE[B1],XRRS[R7]);
  IORRS(EDGE[N2],EDGE[B2],XRRS[R7]);
  IORRS(EDGE[N3],EDGE[B2],XRFS[F7]);
  IORRS(EDGE[N4],EDGE[B3],XRFS[F7]);
  IORRS(EDGE[N5],EDGE[B3],XRRS[R2]);
  IORRS(EDGE[N6],EDGE[B4],XRRS[R2]);
  IORRS(EDGE[N7],EDGE[B4],XRFS[F2]);
  IORRS(EDGE[N8],EDGE[B1],XRFS[F2]);

  (** INITIALIZE CORNER MASK *)

  IORRS(INRS,XRRS[R1],XRRS[R2]);
  IORRS(INRS,INRS,XRRS[R7]);
  IORRS(INRS,INRS,XRRS[R8]);
  IORRS(CORNR,XRFS[F1],XRFS[F2]);
  IORRS(CORNR,CORNR,XRFS[F7]);
  IORRS(CORNR,CORNR,XRFS[F8]);
  ANDRS(CORNR,CORNR,INRS);

  (** INITIALIZE DIRECTION TABLE *)

                XTED[N1]:= 19;                   XTED[N2]:= 21;
  XTED[N8]:=  8;XTED[B1]:=  9;XTED[S2]:= 10;XTED[B2]:= 11;XTED[N3]:= 12;
                XTED[S1]:= -1;                   XTED[S3]:=  1;
  XTED[N7]:=-12;XTED[B4]:=-11;XTED[S4]:=-10;XTED[B3]:= -9;XTED[N4]:= -8;
                XTED[N6]:=-21;                   XTED[N5]:=-19;

  (** INITIALIZE SQUARE DIFFERENCE TO DIRECTION TABLE *)

  FOR INTI := AZL TO ZAL DO
    XLLD[INTI] := 0;
  FOR INTE := B1 TO S4 DO
  BEGIN
    INTD := XTED[INTE];
    FOR IMTI := 1 TO 7 DO
      XLLD[IMTI*INTD] := INTD;
  END;
  FOR INTE := N1 TO N8 DO
    XLLD[XTED[INTE]] := XTED[INTE];

  (** INITIALIZE CASTLING TRANSLATION TABLES *)

  IORRS(XSQS[LS],XRSS[XTRFS[R1,F8]],XRSS[XTRFS[R1,F5]]);
  IORRS(XSQS[LL],XRSS[XTRFS[R1,F1]],XRSS[XTRFS[R1,F5]]);
  IORRS(XSQS[DS],XRSS[XTRFS[R8,F8]],XRSS[XTRFS[R8,F5]]);
  IORRS(XSQS[DL],XRSS[XTRFS[R8,F1]],XRSS[XTRFS[R8,F5]]);

  IORRS(XRQSO[LS],XRSS[XTRFS[R1,F6]],XRSS[XTRFS[R1,F7]]);
  IORRS(XRQSO[LL],XRSS[XTRFS[R1,F4]],XRSS[XTRFS[R1,F3]]);
  IORRS(XRQSA[LS],XRSS[XTRFS[R1,F5]],XRQSO[LS]);
  IORRS(XRQSA[LL],XRSS[XTRFS[R1,F5]],XRQSO[LL]);
  IORRS(XRQSO[LL],XRSS[XTRFS[R1,F2]],XRQSO[LL]);

  IORRS(XRQSO[DS],XRSS[XTRFS[R8,F6]],XRSS[XTRFS[R8,F7]]);
  IORRS(XRQSO[DL],XRSS[XTRFS[R8,F4]],XRSS[XTRFS[R8,F3]]);
```

Circle 82 on inquiry card.

more than once if the user wishes to play more than one game.

During the development of the program, it is necessary to determine whether the individual procedures are functioning properly. To do this, it is helpful to have a few primitive print routines which can provide information about the internal workings in a form which is understandable to the programmer. These same routines are also called by the main input/output (IO) routine (READER) which appears later in the program.

One of these routines (PRIMOV) prints an internal representation of the machine's move. Another prints an 8 by 8 array representing the board (PRINTB). This consists of numbers for Black's pieces (Black pawn = 1; Black King = 6) and letters for White's pieces (White pawn = A; White King = F) with empty squares represented by a –. The PRINBB routine prints an 8 by 8 array representing a bit board. In this case an asterisk (*) stands for a square where a bit is set and a minus sign (–) stands for a square where a bit has not been set. An attack map is printed by PRINAM and this consists of 64 (one for each square) 8 by 8 bit maps in which an * stands for a bit which is set and a – stands for a clear bit.

Other useful print routines include one which permits a user controlled pause during printing (PAUSER) and one which informs the programmer of the status of particular control switches (PRISWI). Because of Pascal's serial requirement (ie: every procedure must be defined before it can be called by another procedure), these routines appear early in the program so that they can be used to test the procedures and functions which follow.

In part 1 we mentioned incremental updating as an important feature of an efficient chess program. It is necessary to apply an evaluation function to the terminal nodes of the look-ahead tree. These evaluations, if they are at all sophisticated, require a substantial amount of detailed information about the position. Although it is possible to calculate this information separately for each evaluation, this is not a very efficient procedure, because adjacent nodes are almost identical. Most of the information which would be calculated each time would be redundant. A more efficient alternative is to "update" and "downdate" the relevant data base incrementally as the program moves about in the look-ahead tree. This capability requires quite a bit of special programming.

Several primitive routines are very useful for this. If the move involves a capture, it

Circle 320 on inquiry card.

Listing 1, continued:

```
IORRS(XRQSA[DS],XRSS[XTRFS[R8,F5]],XRQSO[DS]);
IORRS(XRQSA[DL],XRSS[XTRFS[R8,F5]],XRQSO[DL]);
IORRS(XRQSO[DL],XRSS[XTRFS[R8,F2]],XRQSO[DL]);

FOR INTQ := LS TO DL DO
  WITH XRQM[INTQ] DO
  BEGIN
    RMCP := MT;
    RMCA := FALSE;
    RMAC := TRUE;
    RMCH := FALSE;
    RMMT := FALSE;
    RMIL := FALSE;
    RMSU := FALSE;
    RMPR := FALSE;
    RMOO := TRUE;
  END;

XRQM[LS].RMFR := XTRFS[R1,F5]; XRQM[LS].RMTO := XTRFS[R1,F7];
XRQM[LL].RMFR := XTRFS[R1,F5]; XRQM[LL].RMTO := XTRFS[R1,F3];
XRQM[DS].RMFR := XTRFS[R8,F5]; XRQM[DS].RMTO := XTRFS[R8,F7];
XRQM[DL].RMFR := XTRFS[R8,F5]; XRQM[DL].RMTO := XTRFS[R8,F3];

XRQM[LS].RMQS := FALSE;
XRQM[LL].RMQS := TRUE;
XRQM[DS].RMQS := FALSE;
XRQM[DL].RMQS := TRUE;

XTMQ[LITE] := LS;
XTMQ[DARK] := DS;

XTQS[LS] := XTRFS[R1,F8];
XTQS[LL] := XTRFS[R1,F1];
XTQS[DS] := XTRFS[R8,F8];
XTQS[DL] := XTRFS[R8,F1];

(** INITIALIZE NULL MOVE *)

WITH NULMV DO
BEGIN
  RMFR := AS;
  RMTO := AS;
  RMCP := MT;
  RMCA := FALSE;
  RMAC := TRUE;
  RMCH := FALSE;
  RMMT := FALSE;
  RMIL := FALSE;
  RMSU := FALSE;
  RMPR := TRUE;
  RMPP := PB;
END;

(** INITIALIZE COMMAND PROCESSING VARIABLES *)

JMTJ := ZJ;
ICARD[ZJ] := ";";
ILINE[ZJ] := ";";

(** INITIALIZE MOVES SYNTAX TABLE *)

INTI := SYNCF;
INISYN("    *P    ");
INISYN("    *P/  1");
INISYN("/  1*P    ");
INISYN("    *P/ R ");
INISYN("/ R *P    ");
INISYN("    *P/ R1");
INISYN("/ R1*P    ");
INISYN("    *P/KR ");
INISYN("/KR *P    ");
INISYN("    *P/KR1");
INISYN("/KR1*P    ");
INISYN("/  1*P/  1");
INISYN("/ R *P/ R ");
INISYN("/  1*P/ R ");
INISYN("/ R *P/  1");
INISYN("/ R1*P/  1");
INISYN("/  1*P/ R1");
INISYN("/ R1*P/ R ");
INISYN("/ R *P/ R1");
INISYN("/KR *P/  1");
INISYN("/  1*P/KR ");
INISYN("/KR *P/ R ");
INISYN("/ R *P/KR ");
INISYN("/  1*P/KR1");
INISYN("/KR1*P/  1");
INISYN("  R *P/KR1");
INISYN("/KR1*P/ R ");
INISYN("/ R1*P/ R1");
INISYN("/KR *P/ R1");
INISYN("/ R1*P/KR ");
INISYN("/KR *P/KR ");
INISYN("/KR1*P/ R1");
INISYN("/ R1*P/KR1");
INISYN("/KR1*P/KR ");
INISYN("/KR *P/KR1");
INISYN("/KR1*P/KR1");

INISYN("    -   R1");
INISYN("    -  KR1");
INISYN("/  1-   R1");
INISYN("/ R -   R1");
INISYN("/  1-  KR1");
INISYN("/ R -  KR1");
INISYN("/ R1-   R1");
INISYN("/KR -   R1");
INISYN("/ R1-  KR1");
INISYN("/KR -  KR1");
INISYN("/KR1-  KR1");

(** INITIALIZE LETS *)

FKPSHD := 10;
FKSANQ := 150;
FMAXMT := 256;
FNODEL := 10;
FPADCR[F1] := 0;
FPADCR[F2] := 0;
FPADCR[F3] := 5;
FPADCR[F4] := 10;
FPADCR[F5] := 15;
FPADCR[F6] := 5;
FPADCR[F7] := 0;
FPADCR[F8] := 0;
FPBLOK := 20;
FPCONN := 5;
FPFLNX := 12;
FRDUBL := 60;
FRK7TH := 120;
FTRADE := 36;
FTRDSL := 5156;
FTRPOK := 2;
FTRPWN := 8;
FWKING := 50;
FWMAJM := 1;
FWMINM := 200;
FWPAWN := 100;
FWROOK := 2;
WINDOW := 30;

(** INITIALIZE SWITCHES *)

SWEC := TRUE;
SWPA := TRUE;
SWPS := FALSE;
SWRE := TRUE;
SWSU := FALSE;
SWTR := FALSE;

(** INITIALIZE MAIN LOOP CONTROL VARIABLES *)

GOING := 0;

END;  (* INICON *)


PROCEDURE INITAL(VAR A:RB);                 (* INITIALIZE FOR A NEW GAME *)

VAR
  INTF : TF;                                (* FILE INDEX *)
  INTR : TR;                                (* RANK INDEX *)

BEGIN
  WITH A DO
  BEGIN
    RBTM := LITE;                           (* SIDE TO MOVE *)
    RBTS := -1;                             (* NO ENPASSANT SQUARE *)
    RBTI := 0;                              (* GAME HAS NOT STARTED *)
    RBSQ := [LS,LL,DS,DL];                  (* ALL CASTLING MOVES LEGAL *)
    FOR INTF := F1 TO F8 DO                 (* LOOP THROUGH ALL FILES *)
    BEGIN
      RBIRF[R2,INTF] := LP;                 (* SET LIGHT PAWNS ON BOARD *)
      FOR INTR := R3 TO R6 DO               (* LOOP THRU MIDDLE OF BOARD *)
        RBIRF[INTR,INTF] := MT;             (* SET MIDDLE OF BOARD EMPTY *)
      RBIRF[R7,INTF] := DP;                 (* SET DARK PAWNS ON BOARD *)
    END;
    RBIRF[R1,F1] := LR;                     (* SET REMAINDER OF PIECES ON
                                               BOARD *)
    RBIRF[R1,F2] := LN;
    RBIRF[R1,F3] := LB;
    RBIRF[R1,F4] := LQ;
    RBIRF[R1,F5] := LK;
    RBIRF[R1,F6] := LB;
    RBIRF[R1,F7] := LN;
    RBIRF[R1,F8] := LR;
    RBIRF[R8,F1] := DR;
    RBIRF[R8,F2] := DN;
    RBIRF[R8,F3] := DB;
    RBIRP[R8,F4] := DQ;
    RBIRF[R8,F5] := DK;
    RBIRF[R8,F6] := DB;
    RBIRF[R8,F7] := DN;
    RBIRF[R8,F8] := DR;

    MOVMS := " ENTER MOVE OR TYPE GO.        ";
    WRITELN(MOVMS);
    LSTMV := NULMV;                         (* INITIALIZE PREVIOUS MOVE *)
  END;
END;  (* INITAL *)


PROCEDURE PAUSER;                           (* PAUSE FOR CARRIAGE RETURN *)

BEGIN
  IF SWPA THEN
  BEGIN
    WRITELN(" PAUSING ");
    READLN;
  END;
END;  (* PAUSER *)


PROCEDURE PRIMOV(A:RM);                     (* PRINT A MOVE *)

BEGIN
  WITH A DO
  BEGIN
    WRITE(" FROM ",RMFR:2," TO ",RMTO:2);
    IF NULMVB(A) THEN
      WRITE(", NULL MOVE")
    ELSE
    BEGIN
      IF RMCA THEN
        WRITE(", CAPTURE ",XTPC[RMCP],",")
      ELSE
        WRITE(", SIMPLE,");
      IF NOT RMAC THEN
        WRITE(" NO");
      WRITE(" ACS");
      IF RMCH THEN
```

Circle 120 on inquiry card.

is necessary to change the material balance function. The actual scoring itself is handled by MBEVAL. This routine is called either by MBCAPT or MBTPAC when a piece is lost (update) or gained (downdate); or by MBPROM or MBMORP when a pawn is promoted (update); or when a newly promoted pawn is demoted (downdate). There are other changes which are required in the data base for both capture and noncapture moves. The new squares which are attacked by the piece need to be added to the attack maps (ATKFR, ATKTO, ALATK). This is done by ADDATK. The new square for the piece is added to the data base by ADDLOC. The attacks of sliding pieces which are blocked by the newly moved piece are recomputed by CUTATK. The attacks of sliding pieces which are unblocked by vacating the former square are recomputed by PRPATK. The attacks which emanated from the piece on its former square are deleted by DELATK. These primitive routines are called by LOSEIT when a capture is involved or by MOVEIT otherwise. If the move affects castling status, the necessary data base changes are made by PROACA and PROACS. If a pawn promotion is involved, PROMOT makes the necessary adjustments.

Move Generation

A major part of any chess program is the move generation module. Because of the complexity of the game, many programs simply ignore some of the more unusual moves, such as Queenside castling, en passant pawn captures, or promotion of a pawn to a piece other than a Queen (ie: underpromotion). This arrangement will suffice to play legal chess, but it may be costly if one of the omitted move types is highly desirable in a specific game situation. In addition, an incomplete move generation facility prevents the machine from checking the legality of its opponent's moves.

Rather than being satisfied with an approximate solution, we have heeded the old maxim, "If a job is worth doing, it is worth doing well," and have implemented a move generator which permits the program to play a complete game of legal chess. As you can see from the listing, this requires extensive programming.

The first step in move generation is to create the data base for the important features of the existing board configuration. This is done by CREATE. Once a move has been selected, it is necessary to change the data base. This is done by UPDATE which

Text continued on page 181

Circle 320 on inquiry card.

Listing 1, continued:

```
        WRITE(', CHECK');
      IF RMMT THEN
        WRITE(', MATE');
      IF RMIL THEN
        WRITE(', ILLEGAL');
      IF RMSU THEN
        WRITE(', SEARCHED');
      CASE RMPR OF
      FALSE:  (* NOT PROMOTION *)
        CASE RMOO OF
        FALSE:  (* NOT CASTLE *)
          IF RMEP THEN
            WRITE(', ENPASSANT');
        TRUE:  (* CASTLE *)
          BEGIN
            WRITE(', CASTLE ');
            IF RMQS THEN
              WRITE('LONG')
            ELSE
              WRITE('SHORT');
          END;
        END;
      TRUE:   (* PROMOTION *)
        BEGIN
          WRITE(', PROMOTE TO ');
          CASE RMPP OF
          PQ:  WRITE('QUEEN');
          PR:  WRITE('ROOK');
          PB:  WRITE('BISHOP');
          PN:  WRITE('KNIGHT');
          END;
        END;
      END;
    END;
  END;
  WRITELN('.');
END;  (* PRIMOV *)


PROCEDURE PRINTB(A:RC);                     (* PRINT A BOARD *)

VAR
  INTR : TR;                                (* RANK INDEX *)
  INTF : TF;                                (* FILE INDEX *)

BEGIN
  WRITELN;                                  (* WRITE A BLANK LINE *)
  FOR INTR := R8 DOWNTO R1 DO               (* LOOP DOWN THROUGH RANKS *)
  BEGIN
    WRITE (' ',ORD(INTR)+1:1,' ');          (* OUTPUT RANK LABEL *)
    FOR INTF := F1 TO F8 DO                 (* LOOP ACROSS THROUGH FILES *)
      WRITE (XTPC[A[XTRFS[INTR,INTF]]]);
                                            (* OUTPUT CONTENTS OF SQUARE *)
    WRITELN;                                (* WRITE OUT A RANK *)
  END;
  WRITELN (' W RNBQKBNR');                  (* WRITE OUT BOTTOM LABEL *)
END;  (* PRINTB *)


PROCEDURE PRINBB(A:RS);                     (* PRINT A BIT BOARD *)

VAR
  INTR : TR;                                (* RANK INDEX *)
  INTF : TF;                                (* FILE INDEX *)

BEGIN
  WRITELN;                                  (* WRITE OUT A BLANK LINE *)
  FOR INTR := R8 DOWNTO R1 DO               (* LOOP DOWN THROUGH RANKS *)
  BEGIN
    WRITE (' ',ORD(INTR)+1:1,' ');          (* OUTPUT RANK LABEL *)
    FOR INTF := F1 TO F8 DO                 (* LOOP ACROSS THROUGH FILES *)
      WRITE (XTBC[INRSTB(A,XTRFS[INTR,INTF])]);
                                            (* OUTPUT CONTENTS OF SQUARE *)
    WRITELN;                                (* WRITE OUT A RANK *)
  END;
  WRITELN (' W RNBQKBNR');                  (* WRITE OUT BOTTOM LABEL *)
END;  (* PRINBB *)


PROCEDURE PRINAM(A:FX);                     (* PRINT ATTACK MAP *)

VAR
  INTR, JNTR : TR;                          (* RANK INDICES *)
  INTF, JNTF : TF;                          (* FILE INDICES *)

BEGIN
  WRITELN;
  FOR INTR := R8 DOWNTO R1 DO
  BEGIN
    FOR JNTR := R8 DOWNTO R1 DO
    BEGIN
      FOR INTF := F1 TO F8 DO
      BEGIN
        WRITE(' ');
        FOR JNTF := F1 TO F8 DO
        BEGIN
          WRITE(XTBC[INRSTB(A[XTRFS[INTR,INTF]],XTRFS[JNTR,JNTF])]);
        END;
        WRITE(' ');
      END;
      WRITELN;
    END;
    WRITELN;
    IF INTR IN [R1,F3,R5,R7] THEN PAUSER;
  END;
END;  (* PRINAM *)


PROCEDURE PRISWI(A:FA;B:TB);                (* PRINT A SWITCH *)

BEGIN
  WRITE(' ',A[AA],A[AA+1]);
  IF B THEN
    WRITELN(' ON')
  ELSE
    WRITELN(' OFF');
END;  (* PRISWI *)


PROCEDURE MBEVAL;                           (* EVALUATE MATERIAL BALANCE *)

VAR
  INTI : TI;                                (* COUNT PAWNS OF WINNING SIDE *)

BEGIN
  IF MBLTE <> 0 THEN
    IF MBLTE > 0 THEN
      INTI := MBPWN[LITE]
    ELSE
      INTI := MBPWN[DARK]
  ELSE
    INTI := 0;

  MBVAL[JNTK] := SIGN(MIN(MIN(FMAXMT,ABS(MBLTE))
      +FTRADE*ABS(MBLTE)*(FTRDSL-MBTOT)*(4*INTI+FTRPOK)
      DIV (4*INTI+FTRPWN) DIV 262144,16320),MBLTE);

END;  (* MBEVAL *)


PROCEDURE MBCAPT                            (* EVALUATE MATERIAL AFTER
                                               CAPTURE *)
  (A:TP);                                   (* PIECE CAPTURED *)

BEGIN
  MBTOT := MBTOT - ABS(XTPV[A]);            (* TOTAL MATERIAL ON BOARD *)
  IF XTPU[A] = EP THEN
    MBPWN[XTPM[A]] := MBPWN[XTPM[A]] - 1;
                                            (* REMOVE PAWN IF NECESSARY *)
  MBLTE := MBLTE - XTPV[A];                 (* LITE ADVANTAGE *)
  MBEVAL;                                   (* EVALUATE MATERIAL *)
END;  (* MBCAPT *)


PROCEDURE MBTPAC                            (* REMOVE CAPTURE FROM
                                               MATERIAL BALANCE DATA.  THIS
                                               IS THE INVERSE OF MBCAPT *)
  (A:TP);                                   (* PIECE UNCAPTURED *)

BEGIN
  MBTOT := MBTOT + ABS(XTPV[A]);
  IF XTPU[A] = EP THEN
    MBPWN[XTPM[A]] := MBPWN[XTPM[A]] + 1;
  MBLTE := MBLTE + XTPV[A];
END;  (* MBTPAC *)


PROCEDURE MBPROM                            (* EVALUATE MATERIAL BALANCE
                                               CHANGE DUE TO PAWN
                                               PROMOTION *)
  (A:TP);                                   (* PIECE TO PROMOTE TO *)

BEGIN
  MBTOT := MBTOT + ABS(XTPV[A]-XTPV[XTUMP[EP,XTPM[A]]]);
                                            (* TOTAL MATERIAL ON BOARD *)
  MBPWN[XTPM[A]] := MBPWN[XTPM[A]] - 1;(* COUNT PAWNS *)
  MBLTE := MBLTE + XTPV[A]-XTPV[XTUMP[EP,XTPM[A]]];
  MBEVAL;                                   (* EVALUATE RESULT *)
END;  (* MBPROM *)


PROCEDURE MBMORP                            (* REMOVE PAWN PROMOTION
                                               FROM MATERIAL BALANCE DATA.
                                               THIS IS THE INVERSE
                                               OF MBPROM *)
  (A:TP);                                   (* PIECE PROMOTED TO *)

BEGIN
  MBTOT := MBTOT - ABS(XTPV[A]-XTPV[XTUMP[EP,XTPM[A]]]);
  MBPWN[XTPM[A]] := MBPWN[XTPM[A]] + 1;
  MBLTE := MBLTE - (XTPV[A]-XTPV[XTUMP[EP,XTPM[A]]]);
END;  (* MBMORP *)


PROCEDURE ADDATK                            (* ADD ATTACKS OF PIECE TO DATA
                                               BASE *)
  (A:TS);                                   (* SQUARE OF PIECE TO ADD
                                               ATTACK *)

VAR
  INTB : TB;                                (* LOOP CONTROL BOOLEAN *)
  INTD : TD;                                (* CURRENT DIRECTION OFFSET *)
  INTE : TE;                                (* CURRENT DIRECTION INDEX *)
  INTM : TM;                                (* COLOR OF CURRENT PIECE *)
  INTP : TP;                                (* CURRENT PIECE *)
  INTT : TT;                                (* RUNNING SQUARE *)

BEGIN
  INTP := NBORD[A];                         (* PIECE OF INTEREST *)
  INTM := XTPM[INTP];                       (* COLOR *)
  FOR INTE := XFPE[INTP] TO XLPE[INTP] DO
  BEGIN
    INTT := A;                              (* INITIALIZE RUNNING SQUARE *)
    INTB := XSPB[INTP];                     (* TRUE IF SWEEP PIECE *)
    INTD := XTED[INTE];                     (* OFFSET *)
    REPEAT
      INTT := XTLS[XTSL[INTT] + INTD]; (* STEP IN PROPER DIRECTION *)
      IF INTT >= 0 THEN
      BEGIN
        SETRS(ATKFR[A],INTT);
        SETRS(ATKTO[INTT],A);
        SETRS(ALATK[INTM],INTT);
        IF NBORD[INTT] <> MT THEN
          INTB := FALSE;
```

Listing 1, continued:

```
      END
      ELSE
        INTB := FALSE;
    UNTIL NOT INTB;
  END;
END;  (* ADDATK *)


PROCEDURE ADDLOC                          (* ADD PIECE TO DATA BASE *)
  (A:TS;                                  (* SQUARE WITH NEW PIECE ON IT *)
   B:TP);                                 (* NEW PIECE TO ADD *)


BEGIN
  CLRRS(TPLOC[MT],A);                     (* BIT BOARD OF EMPTY SQUARES *)
  SETRS(TPLOC[B],A);                      (* BIT BOARD OF ALL SAME PIECE *)
  SETRS(TMLOC[XTPM[B]],A);                (* BIT BOARD OF ALL SAME COLOR *)
  SETRS(ALLOC[JNTK],A);                   (* BIT BOARD OF ALL PIECES *)
  NBORD[A] := B;                          (* SET NEW PIECE ON BOARD *)
END;  (* ADDLOC *)


PROCEDURE CLSTAT;                         (* CLEAR POSITION STATUS *)

BEGIN
  WITH BOARD DO
  BEGIN
    RBTM := LITE;                         (* WHITE TO MOVE *)
    RBTS := -1;                           (* NO ENPASSANT *)
    RBSQ := [];                           (* NO CASTLING LEGAL *)
  END;
END;  (* CLSTAT *)


PROCEDURE CUTATK                          (* CUT ATTACKS THROUGH SQUARE *)
  (A:TS);                                 (* SQUARE *)

VAR
  INRS : RS;                              (* ATTACKING PIECES *)
  INTS : TS;                              (* ATTACKING PIECE SQUARE *)
  IMRS : RS;                              (* SCRATCH *)
  INTD : TD;                              (* STEP SIZE *)
  INTM : TM;                              (* ATTACKING PIECE SIDE *)
  INTL : TL;                              (* NO LONGER ATTACKED SQUARE *)
  INTT : TT;                              (* NO LONGER ATTACKED SQUARE *)

BEGIN
  CPYRS(INRS,ATKTO[A]);                   (* ALL PIECES ATTACKING SQUARE *)
  WHILE NXTTS(INRS,INTS) DO
    IF XSPB[NBORD[INTS]] THEN             (* IF SWEEP PIECE *)
    BEGIN
      INTD := XLLD[XTSL[A]-XTSL[INTS]];
                                          (* STEP SIZE ON 10 X 12 BOARD *)
      INTM := XTPM[NBORD[INTS]];          (* SIDE OF ATTACKING PIECE *)
      INTL := XTSL[A]+INTD;               (* FIRST SQUARE BEYOND PIECE *)
      INTT := XTLS[INTL];                 (* FIRST SQUARE BEYOND PIECE ON
                                             8X8 BOARD *)
      WHILE INTT > AT DO                  (* WHILE ON BOARD *)
      BEGIN
        CLRRS(ATKFR[INTS],INTT);          (* CLEAR ATTACK MAP *)
        CLRRS(ATKTO[INTT],INTS);
        ANDRS(IMRS,ATKTO[INTT],TMLOC[INTM]);
                                          (* OTHER ATTACKS ON SQUARE BY
                                             SAME SIDE *)
        IF NULRS(IMRS) THEN               (* IF NO ATTACKS BY THAT SIDE *)
          CLRRS(ALATK[INTM],INTT);        (* CLEAR ATTACKS BY SIDE *)
        IF NBORD[INTT] = MT THEN
        BEGIN
          INTL := INTL+INTD;              (* STEP BEYOND SQUARE *)
          INTT := XTLS[INTL];
        END
        ELSE
          INTT := AT;                     (* STOP SCAN *)
      END;
  END;
END;  (* CUTATK *)


PROCEDURE DELATK                          (* DELETE ATTACKS FROM SQUARE *)
  (A:TS);                                 (* SQUARE TO REMOVE PIECE *)

VAR
  INRS : RS;                              (* SQUARES ATTACKED BY PIECE ON
                                             SQUARE *)
  IMRS : RS;                              (* SCRATCH *)
  INTS : TS;                              (* SQUARE ATTACKED BY PIECE ON
                                             SQUARE *)
  INTM : TM;                              (* SIDE OF PIECE ON SQUARE *)

BEGIN
  CPYRS(INRS,ATKFR[A]);                   (* SQUARES ATTACKED BY PIECE
                                             ON SQUARE *)
  NEWRS(ATKFR[A]);                        (* CLEAR ATTACKS FROM SQUARE *)
  INTM := XTPM[NBORD[A]];                 (* SIDE OF PIECE ON SQUARE *)
  WHILE NXTTS(INRS,INTS) DO               (* LOOP THROUGH ALL ATTACKS BY
                                             PIECE *)
  BEGIN
    CLRRS(ATKTO[INTS],A);                 (* CLEAR ATTACK TO OTHER
                                             SQUARE *)
    ANDRS(IMRS,ATKTO[INTS],TMLOC[INTM]);
                                          (* OTHER ATTACKS BY SAME SIDE *)
    IF NULRS(IMRS) THEN
      CLRRS(ALATK[INTM],INTS);            (* CLEAR ATTACKS BY SIDE *)
  CLRRS(TPLOC[NBORD[A]],A);               (* CLEAR PIECE *)
  CLRRS(TMLOC[INTM],A);                   (* CLEAR PIECE FROM SIDE *)
  CLRRS(ALLOC[JNTK],A);                   (* CLEAR PIECE FROM ALL PIECES *)
  SETRS(TPLOC[MT],A);                     (* SET EMPTY *)
  NBORD[A] := MT;
  END;
END;  (* DELATK *)
```

```
PROCEDURE PRPATK                          (* PROPAGATE ATTACKS THROUGH
                                             SQUARE *)
  (A:TS);                                 (* SQUARE *)

VAR
  INRS : RS;                              (* ATTACKING PIECES *)
  INTS : TS;                              (* ATTACKING PIECE SQUARE *)
  INTD : TD;                              (* STEP SIZE *)
  INTM : TM;                              (* ATTACKING PIECE SIDE *)
  INTL : TL;                              (* NEW ATTACKED SQUARE *)
  INTT : TT;                              (* NEW ATTACKED SQUARE *)

BEGIN
  CPYRS(INRS,ATKTO[A]);                   (* ALL PIECES ATTACKING SQUARE *)
  WHILE NXTTS(INRS,INTS) DO
    IF XSPB[NBORD[INTS]] THEN             (* IF SWEEP PIECE *)
    BEGIN
      INTD := XLLD[XTSL[A]-XTSL[INTS]];
                                          (* STEP SIZE ON 10 X 12 BOARD *)
      INTM := XTPM[NBORD[INTS]];          (* SIDE OF ATTACKING PIECE *)
      INTL := XTSL[A]+INTD;               (* FIRST SQUARE BEYOND PIECE *)
      INTT := XTLS[INTL];                 (* FIRST SQUARE BEYOND PIECE ON
                                             8X8 BOARD *)
      WHILE INTT >= 0 DO                  (* WHILE ON BOARD *)
      BEGIN
        SETRS(ATKFR[INTS],INTT);          (* SET ATTACK MAP *)
        SETRS(ATKTO[INTT],INTS);
        SETRS(ALATK[INTM],INTT);          (* SET ATTACKS BY SIDE *)
        IF NBORD[INTT] = MT THEN
        BEGIN
          INTL := INTL+INTD;              (* STEP BEYOND SQUARE *)
          INTT := XTLS[INTL];
        END
        ELSE
          INTT := -1;                     (* STOP SCAN *)
      END;
    END;
END;  (* PRPATK *)


PROCEDURE GAINIT                          (* UNPROCESS CAPTURE MOVE *)
  (A:RM);                                 (* CAPTURE MOVE *)

BEGIN
  WITH A DO
  BEGIN
    ADDLOC(RMFR,NBORD[RMTO]);             (* PUT PIECE ON ORIGINAL
                                             SQUARE *)
    ADDATK(RMFR);
    CUTATK(RMFR);                         (* STOP ATTACKS AT THIS SQUARE *)
    DELATK(RMTO);                         (* REMOVE THEM FROM
                                             DESTINATION SQUARE *)
    ADDLOC(RMTO,RMCF);                    (* REPLACE CAPTURED PIECE *)
    ADDATK(RMTO);
    MBTPAC(NBORD[RMTO]);                  (* UPDATE SCORE *)
  END;
END;  (* GAINIT *)


PROCEDURE LOSEIT                          (* PROCESS CAPTURE MOVE *)
  (A:RM);                                 (* CAPTURE MOVE *)

BEGIN
  WITH A DO
  BEGIN
    MBCAPT(NBORD[RMTO]);                  (* UPDATE SCORE *)
    DELATK(RMTO);                         (* DELETE ATTACKS OF CAPTURED
                                             PIECE *)
    ADDLOC(RMTO,NBORD[RMFR]);             (* ADD PIECE TO DESTINATION
                                             SQUARE *)
    DELATK(RMFR);                         (* DELETE ATTACKS OF MOVING
                                             PIECE *)
    PRPATK(RMFR);                         (* PROPAGATE ATTACKS THROUGH
                                             FROM SQUARE *)
    ADDATK(RMTO);                         (* ADD ATTACKS OF MOVING PIECE *)
  END;
END;  (* LOSEIT *)


PROCEDURE MOVEIT                          (* PROCESS ORDINARY MOVE *)
  (A:RM);                                 (* ORDINARY MOVE *)

BEGIN
  WITH A DO
  BEGIN
    ADDLOC(RMTO,NBORD[RMFR]);             (* ADD PIECE TO NEW SQUARE *)
    CUTATK(RMTO);                         (* CUT ATTACKS THROUGH NEW
                                             SQUARE *)
    DELATK(RMFR);                         (* DELETE ATTACKS FROM OLD
                                             SQUARE *)
    PRPATK(RMFR);                         (* PROPAGATE ATTACKS THROUGH OLD
                                             SQUARE *)
    ADDATK(RMTO);                         (* ADD ATTACKS FROM NEW SQUARE *)
  END;
END;  (* MOVEIT *)


PROCEDURE RTRKIT                          (* UNPROCESS ORDINARY MOVE *)
  (A:RM);                                 (* THE MOVE TO RETRACT *)

BEGIN
  WITH A DO
  BEGIN
    ADDLOC(RMFR,NBORD[RMTO]);             (* PUT PIECE ON ORIGINAL
                                             SQUARE *)
    CUTATK(RMFR);                         (* CUT ATTACKS THROUGH ORIGINAL
                                             SQUARE *)
    DELATK(RMTO);                         (* DELETE ATTACKS FROM
                                             DESTINATION SQUARE *)
    PRPATK(RMTO);                         (* PROPAGATE ATTACKS THROUGH
                                             DESTINATION SQUARE *)
    ADDATK(RMFR);                         (* ADD ATTACKS FROM ORIGINAL
                                             SQUARE *)
  END;
END;  (* RTRKIT *)
```

Circle 324 on inquiry card.

ATTENTION SWTPC DISC SYSTEM USERS

All Ed Smith's **M6800 SOFTWARE TOOLS**

are now available in mini-FLEX format as well as Smoke Signal Broadcasting format. See July BYTE for diskette media prices. Disc users specify SSB OR SWTPCO.

M6800 RELOCATING ASSEMBLER AND LINKING LOADER software for rapid program development and debugging. Some of its features are RELOCATABLE CODE—FULL ALPHABETIZED CROSS REFERENCE LISTING—BOTH GLOBAL AND LOCAL LABELS—LISTING OF EXECUTION TIME—RELATIVE BRANCH TARGET ADDRESSES—8 CHARACTER LABELS—CORESIDENT EDITOR—ENGLISH ERROR MESSAGES—80 COLUMN LISTING USING PR-40—etc.

M68AS . . cassette **$50.00**
M68AS-D . . minifloppy **$55.95**

M6800 RELOCATABLE DISASSEMBLER AND SEGMENTED SOURCE TEXT GENERATOR. This software tool enables you to modify and adapt those large sized object programs and re-assemble them on your system without requiring enormous memory. This program will produce segmented source text files with all the external linkage information required for re-assembly using the above Relocating Assembler.

M68RS . . cassette **$35.00**
M68RS-D . . minifloppy **$40.95**

M68ASPK Above two programs on single diskette **$85.00**

The above two programs are furnished in relocatable formatted code with instruction manuals and commented assembly listings. The loader is also supplied in standard MIKBUG format on cassette or as a binary file on disc to initially load the linking loader at any desired address.

The linking loader is also available in EPROM on two 2708's for $45.00. Specify desired address and version, i.e. cassette, FLEX or SSB.

S-M-I-T-H-B-U-G

FIRMWARE from Ed Smith's SOFTWARE WORKS

A 2k Monitor Disassembler-Trace Debugging Tool in EPROM. Use as stand-alone monitor or as an adjunct to your Smartbug or Swtbug. Requires ACIA as control port. Provides all the monitor commands (23) you could ever ask for, plus the convenience of a mnemonic disassembler and single stepping disassembler-trace display. Interactive use of single or dual breakpoint with trace pickup at breakpoint, plus many new monitor functions and subroutines, makes this a super tool you will love to work with. Uses location $F800 to $FFFF. Furnished on two 2708's or one 2716. Includes Manual and commented assembly listing. Specify ACIA location and present monitor in order to keep $A000 RAM compatibility.

SMITHBUG ON 2708's $60.00
SMITHBUG ON 2716 . . . . $70.00

Get under Ed Smith's Software RUG (Relocating assembler Users Group). All RUG members will have use of a building software support library in relocatable format. Upcoming items are a Floating Point package and a Trig package.

Order direct by check. Specify system configuration if other than SwTPCo. California residents add 6% sales tax.

Ed Smith's SOFTWARE WORKS

P.O. Box 339, Redondo Beach, CA 90277, (213) 373-3350

Circle 3 on inquiry card.

AJA SOFTWARE

P. O. BOX 2528 • ORANGE, CA. 92669
(714) 774-1270

BASIC SOFTWARE
NORTH STAR TRS-80 APPLE

NORTH STAR BASIC on diskette, ready to run. North Star Tutorial I, Tutorial II, Tutorial III (Release 4) teaches North Star Basic, Word Processing, Accts. Payable, Accts. Receivable, Payroll, General Ledger, Medical/Prof. Accts., Rec. Billing, Inventory, Sales Analysis, Computer Chess, Mailing Labels, Personnel System, MANY MORE AVAILABLE. EACH INDIVIDUAL SYSTEM **$35.00**

TOTALLY INTEGRATED SYSTEM (A/R, A/P, GEN. LEDGER, PAY, LETTER GENERATOR, BUS. STATISTICS) in Binder Complete Doc. **$150.00**
Checkbook Bal., Budget Planner, Golf Handicapper, Game Disk, X-RATED Story Generator. MORE AVAILABLE. EACH INDIVIDUAL SYSTEM **$25.00**

IQ TEST, MEMORY Diagnostic, Pers. Finance, Bus. Statistics, Crypto Encoder, Sign Maker, Menu Planner **$150.00 EACH**

SINGLE or DUAL DRIVE, 8K Free Memory Required.

TRS-80 on cassette (Level 1). Numerous Games, Examples, STARTREK, WUMPUS, MORE **only $3.95 EACH**
BASIC TUTORIAL (Learn Basic Programming) **$9.95**
STOCK MARKET ANALYSIS **$9.95**
MANY MORE.

APPLE on Cassette. Many Games, Examples, WORLDPOWER, STARTREK, BOWLING, BINGO **$5.95 EACH**
BASIC TUTORIAL (Learn Basic) **$9.95**
STOCK MARKET ANALYSIS **$9.95**
BUDGET PLAN **$5.95**
APPLE DISK BUSINESS SYSTEMS, A/R, A/P, etc. **ONLY $35.00 PER SYSTEM**

Order by Phone (714) 774-1270, UPS COD, or Mail Order Today.

Over 1000 Systems Sold, Order Now . . . .
SOFTWARE BOOKS **SAME PRICE FOR ALL SYSTEMS**

Need Intelligence in your 488 Instrumentation System ???

Get it with your favorite S-100 computer and the P&T-488 Interface Board.

The P&T-488 Interface Board gives your S-100 computer the ability to be a talker, listener or controller on the IEEE-488 instrumentation bus. Easy to use software helps you integrate the P&T-488 into your system—all you do is pass a message string to the provided subroutines and they take care of the details. **NOW AVAILABLE:** Software to access the 488 bus directly from **North Star Basic** and **MBASIC.**

P&T-488 (assembled & tested) $325

PICKLES TROUT

PICKLES & TROUT

P.O. BOX 1206, GOLETA, CA 93017, (805) 967-9563

Circle 313 on inquiry card.

Having Reservations About Your Software?

HUNT NO MORE!

Smoke Signal Broadcasting presents the

TD-1 TRACE/DISASSEMBLER

Provides interactive tracing and disassembly for 6800 program debugging.

- TRACE allows programmer to control execution of program under test for examination of processor's internal registers and memory on instruction-by-instruction basis.
- DISASSEMBLY converts machine code to easy to read mnemonics.
- Hex code for the opcode and program counter displayed in addition to mnemonic, when instruction is listed.
- Contents of any register or any memory location may be changed at any time.

Only $19.95 (on cassette), $25.90 (on diskette)

We're the "CHIEF" in 6800 products software

SMOKE SIGNAL BROADCASTING

6304 Yucca/Hollywood, CA 90028/(213) 462-5652

Circle 320 on inquiry card.

Listing 1, continued:

```
PROCEDURE PAWNIT                                (* UNPROMOTE A PAWN *)
  (A:RM);                                       (* PROMOTION MOVE *)

BEGIN
  WITH A DO
  BEGIN
    MBMORP(NBORD[RMTO]);                        (* UPDATE SCORE *)
    NBORD[RMTO] := XTUMP[EP,XTPM[NBORD[RMTO]]];
  END;
END;  (* PAWNIT *)


PROCEDURE PROACA                                (* PROCESS CASTLE STATUS
                                                   CHANGES *)
  (A:TS);                                       (* SQUARE *)

VAR
  INRS : RS;                                    (* SCRATCH *)
  IMRS : RS;                                    (* SCRATCH *)

BEGIN
  CLRRS(CSTAT[JNTK],A);                         (* CLEAR THIS SQUARE *)
  ANDRS(INRS,CSTAT[JNTK],XRRS[XTSR[A]]);
                                                (* CASTLE BITS FOR THIS SIDE *)
  IF NOT INRSTB(INRS,XTRFS[XTSR[A],F5]) THEN
                                                (* IF KING MOVE *)
    ANDRS(CSTAT[JNTK],CSTAT[JNTK],XNRS[XTSR[A]]);
                                                (* CLEAR ALL CASTLE MOVES FOR
                                                   SIDE *)
  ANDRS(IMRS,INRS,XRFS[F8]);                    (* KING ROOK SQUARE *)
  ANDRS(INRS,INRS,XRFS[F1]);                    (* QUEEN ROOK SQUARE *)
  IORRS(INRS,INRS,IMRS);                        (* BOTH ROOK SQUARES *)
  IF NULRS(INRS) THEN                           (* IF BOTH ROOKS GONE *)
    ANDRS(CSTAT[JNTK],CSTAT[JNTK],XNRS[XTSR[A]]);
END;  (* PROACA *)


PROCEDURE PROACS                                (* PROCESS MOVES AFFECTING CASTLE
                                                   STATUS *)
  (A:RM);                                       (* MOVE WITH RMAC *)

BEGIN
  WITH A DO
  BEGIN
    IF INRSTB(CSTAT[JNTK],RMFR) THEN            (* FROM SQUARE *)
      PROACA(RMFR);
    IF INRSTB(CSTAT[JNTK],RMTO) THEN            (* TO SQUARE *)
      PROACA(RMTO);
  END;
END;  (* PROACS *)


PROCEDURE PROMOT                                (* PROCESS PROMOTION *)
  (A:RM);                                       (* PROMOTION MOVE *)

BEGIN
  WITH A DO
  BEGIN
    MBPROM(XTGMP[RMPP,JNTM]);                   (* UPDATE SCORE *)
    NBORD[RMFR] := XTGMP[RMPP,JNTM];
  END;
END;  (* PROMOT *)


PROCEDURE CREATE;                               (* CREATE GLOBAL DATA BASE *)

VAR
  INRS : RS;                                    (* SCRATCH BIT BOARD *)
  INTM : TM;                                    (* COLOR INDEX *)
  INTP : TP;                                    (* PIECE INDEX *)
  INTQ : TQ;                                    (* CASTLE TYPE INDEX *)
  INTS : TS;                                    (* SQUARE INDEX *)

BEGIN
  WITH BOARD DO
  BEGIN

    JNTW := AW+1;                               (* INITIALIZE MOVES STACK
                                                   POINTER *)

    JNTK := AK;                                 (* PLY INDEX *)
    JNTM := RBTM;                               (* SIDE TO MOVE *)

    NODES := 0;                                 (* INITIALIZE TOTAL NODES *)

    LINDX[JNTK] := JNTW;                        (* MOVES ARRAY LIMIT *)
    SRCHM[JNTK] := M0;                          (* SEARCH MODE *)

    FOR INTS := AS TO ZS DO
    BEGIN
      NEWRS(ATKFR[INTS]);                       (* CLEAR ATTACKS FROM *)
      NEWRS(ATKTO[INTS]);                       (* CLEAR ATTACKS TO *)
      NBORD[INTS] := MT;                        (* CLEAR LOOKAHEAD BOARD *)
    END;

    NEWRS(ALLOC[JNTK]);                         (* CLEAR ALL PIECE LOCATIONS *)

    FOR INTP := LP TO MT DO
      NEWRS(TPLOC[INTP]);                       (* CLEAR PIECE LOCATIONS *)

    FOR INTM := LITE TO NONE DO
    BEGIN
      NEWRS(TMLOC[INTM]);                       (* CLEAR COLOR LOCATIONS *)
      NEWRS(ALATK[INTM]);                       (* CLEAR COLOR ATTACKS *)
    END;

    MBTOT := 0;
    MBPWN[LITE] := 0;
    MBPWN[DARK] := 0;
    MBLTE := 0;
    FOR INTS := AS TO ZS DO
      IF RBIS[INTS] <> MT THEN
      BEGIN
        ADDLOC(INTS,RBIS[INTS]);
        MBTPAC(RBIS[INTS]);
      END
      ELSE
        SETRS(TPLOC[MT],INTS);

    MBEVAL;                                     (* EVALUATE MATERIAL *)

    CPYRS(INRS,ALLOC[JNTK]);                    (* COPY BIT BOARD OF ALL
                                                   PIECES *)
    WHILE NXTTS(INRS,INTS) DO
      ADDATK(INTS);                             (* ADD ATTACKS OF ALL PIECES *)

    NEWRS(CSTAT[JNTK]);                         (* INITIALIZE CASTLING SQUARES *)
    FOR INTQ := LS TO DL DO
      IF INTQ IN RBSQ THEN
        IORRS(CSTAT[JNTK],CSTAT[JNTK],XSQS[INTQ]);

    NEWRS(ENPAS[JNTK]);                         (* INITIALIZE ENPASSANT SQUARE *)
    IF RBTS >= 0 THEN
      SETRS(ENPAS[JNTK],RBTS);

    CPYRS(GENPN[JNTK],TPLOC[XTUMP[EP,JNTM]]);
    NOTRS(GENTO[JNTK],TMLOC[JNTM]);
    NOTRS(INRS,GENPN[JNTK]);
    ANDRS(GENFR[JNTK],TMLOC[JNTM],INRS);
  END;
END;  (* CREATE *)


PROCEDURE DNDATE                                (* DOWNDATE DATA BASE TO BACK
                                                   OUT A MOVE *)
  (A:RM);                                       (* THE MOVE TO RETRACT *)

VAR
  INTS : TS;                                    (* SCRATCH *)
  INTR : TR;                                    (* ROOK RANK FOR CASTLING *)
  INTF : TF;                                    (* ROOK FILE FOR CASTLING *)

  RKFR : TS;                                    (* ROOK FROM SQUARE *)
  RKTO : TS;                                    (* ROOK TO SQUARE *)

BEGIN
  WITH A DO
  BEGIN
    CASE ORD(RMCA)*4 + ORD(RMAC)*2 + ORD(RMPR) OF
      0:  (* ORDINARY MOVE *)
        RTRKIT(A);
      1:  (* PAWN MOVE AND PROMOTE *)
        BEGIN
          PAWNIT(A);
          RTRKIT(A);
        END;
      2:  (* MISCELLANEOUS ACS *)
        IF RMOO THEN
        BEGIN  (* CASTLE *)
          IF RMQS THEN
            INTF := F1                          (* ROOK ON QUEEN ROOK FILE *)
          ELSE
            INTF := F8;                         (* ROOK ON KING ROOK FILE *)
          INTR := XTSR[RMFR];                   (* ROOK FILE *)
          RKFR := XTRFS[INTR,INTF];             (* ROOK FROM SQUARE *)
          RKTO := (RMFR+RMTO) DIV 2;            (* ROOK TO SQUARE *)
          ADDLOC(RKFR,NBORD[RKTO]);             (* REPLACE ROOK *)
          DELATK(RKTO);
          PRPATK(RKTO);
          ADDATK(RKFR);
          RTRKIT(A);                            (* RETRACT KING MOVE *)
        END
        ELSE  (* NOT CASTLE *)
          RTRKIT(A);
      3:; (* NULL MOVE *)
      4:  (* CAPTURE *)
        IF RMEP THEN
        BEGIN  (* CAPTURE ENPASSANT *)
          INTS := XTRFS[XTSR[RMFR],XTSF[RMTO]];
          ADDLOC(INTS,RMCP);
          CUTATK(INTS);
          ADDATK(INTS);
          RTRKIT(A);                            (* RETRACT PAWN MOVE *)
          MBTPAC(NBORD[INTS]);                  (* ADD PIECE TO SCORE *)
        END
        ELSE  (* CAPTURE NOT ENPASSANT *)
          GAINIT(A);
      5:  (* CAPTURE AND PROMOTE *)
        BEGIN
          PAWNIT(A);                            (* UNPROMOTE *)
          GAINIT(A);                            (* UNCAPTURE *)
        END;
      6:  (* CAPTURE ACS *)
        GAINIT(A);                              (* UNCAPTURE *)
      7:  (* CAPTURE ROOK ACS, PROMOTE *)
        BEGIN
          PAWNIT(A);
          GAINIT(A);
        END;
    END;
    JNTW := LINDX[JNTK];                        (* RESET MOVE GENERATION
                                                   POINTER *)
    JNTK := JNTK-1;                             (* BACK UP PLY INDEX *)
    JNTM := OTHER[JNTM];                        (* SWITCH SIDE TO MOVE *)
  END;
END;  (* DNDATE *)


FUNCTION UPDATE                                 (* UPDATE DATA BASE FOR A MOVE *)
  (VAR A:RM)                                    (* THE MOVE *)
   :TB;                                         (* RETURNS TRUE IF MOVE IS
                                                   LEGAL *)

VAR
  INRS : RS;                                    (* SCRATCH *)
  IMRS : RS;                                    (* SCRATCH *)
  INTS : TS;                                    (* SCRATCH *)
  INTF : TF;                                    (* ROOK FILE FOR CASTLING *)
```

Circle 326 on inquiry card.

SoftSide™

" your BASIC software magazine "

The Pages Come Alive With The Touch Of Your Keyboard.

Published monthly
SoftSide Subscriptions
PO Box 68 Milford NH 03055

$15.00 per year $28.00 for two years

Circle 308 on inquiry card.

APPLE II TRS-80 & other

CASSETTE TAPES FOR SALE

We promote, advertise and sell software for free lance computer specialists all over the United States.

Get the latest issue of our book full of programs by return mail. Send $1.00 to:

Programmers Software Exchange
2110 N. 2nd Street
Cabot, Arkansas 72023

INTEGRATED SMALL BUSINESS MICROSOFTWARE IN 'BASIC'

MICROLEDGER – General Ledger
MICROPAY – Accounts Payable
MICROREC – Accounts Receivable
MICROPERS – Personnel/Payroll
MICROINV – Inventory Control

Please send me information on small business software:

☐ Also interested in MICROHOME, a collection of 8 personal and home computing programs.

NAME ____________________

ADDRESS ____________________

____

COMPUMAX ASSOCIATES

505 Hamilton Avenue
Palo Alto, California 94301
Telephone: (415) 321-2881

Circle 48 on inquiry card.

Having Reservations About Your Software?

HUNT NO MORE!

Smoke Signal Broadcasting presents the

SD-2 6800 BASIC COMPILER

For High Performance Process Control Programs and Business Applications

- Fast execution---syntax analysis of line not performed each time line is executed.
- More power in runtime package---no room needed for statement analysis, program editing and listing.
- Wider variety of facilities --- nested conditions, IF–THEN–ELSE, and others.
- Decimal floating point, accuracy to a minimum of 9 digits, dollar values to $99,999,999.99.
- PRINT USING for formatted output.
- Utilizes RANDOM ACCESS disk files.

$325.00 (manual only, $10.00)

We're the "CHIEF" in 6800 products software

SMOKE SIGNAL BROADCASTING

6304 Yucca/Hollywood, CA 90028/(213) 462-5652

Circle 320 on inquiry card.

Listing 1, continued:

```
    INTR : TR;                          (* ROOK RANK FOR CASTLING *)
    RKTO : TS;                          (* ROOK DESTINATION SQUARE *)
    RKFR : TS;                          (* ROOK ORIGIN SQUARE *)

BEGIN
  WITH A DO
  BEGIN
    JNTK := JNTK+1;                     (* ADVANCE PLY INDEX *)
    NEWRS(ENPAS[JNTK]);                 (* CLEAR ENPASSANT BIT BOARD *)
    CPYRS(CSTAT[JNTK],CSTAT[JNTK-1]);   (* INITIALIZE CASTLE STATUS *)
    CPYRS(ALLOC[JNTK],ALLOC[JNTK-1]);   (* INITIALIZE ALL LOCATIONS *)
    MBVAL[JNTK] := MBVAL[JNTK-1];       (* INITIALIZE MATERIAL SCORE *)
    LINDX[JNTK] := JNTW;                (* MOVES ARRAY LIMIT *)
    CASE ORD(RMCA)*4 + ORD(RMAC)*2 + ORD(RMPR) OF
      0:  (* ORDINARY MOVE *)
        IF RMEP THEN
        BEGIN                           (* PAWN MOVE 2 SPACES *)
          SFTRS(INRS,XRSS[RMTO],S1);
          SFTRS(IMRS,XRSS[RMTO],S3);
          IORRS(INRS,INRS,IMRS);        (* SQUARES NEXT TO DESTINATION *)
          ANDRS(INRS,INRS,TPLOC[XTUMP[EP,OTHER[JNTM]]]);
                                        (* INTERSECT WITH ENEMY PAWNS *)
          IF NOT NULRS(INRS) THEN
            SETRS(ENPAS[JNTK],(RMTO+RMFR) DIV 2);
                                        (* SET ENPASSANT SQUARE *)
          MOVEIT(A);                    (* MOVE PAWN *)
        END
        ELSE
          MOVEIT(A);                    (* MOVE PIECE *)
      1:  (* MOVE AND PROMOTE *)
        BEGIN
          PROMOT(A);                    (* PROMOTE PAWN *)
          MOVEIT(A);                    (* MOVE PROMOTED PIECE *)
        END;
      2:  (* MISCELLANEOUS ACS *)
        BEGIN
          IF RMOO THEN
          BEGIN  (* CASTLE *)
            IF RMQS THEN
              INTF := F1
            ELSE
              INTF := F8;               (* ROOK ON QUEEN ROOK FILE *)
                                        (* ROOK ON KING ROOK FILE *)
            INTR := XTSR[RMFR];         (* ROOK ON KINGS RANK *)
            RKFR := XTRFS[INTR,INTF];   (* ROOK ORIGIN SQUARE *)
            RKTO := (RMFR+RMTO) DIV 2;  (* ROOK DESTINATION SQUARE *)
            ANDRS(CSTAT[JNTK],CSTAT[JNTK],XNRS[INTR]);
                                        (* DISALLOW FURTHER CASTLING
                                           BY THIS SIDE *)
            ADDLOC(RKTO,NBORD[RKFR]);   (* PUT ROOK ON NEW SQUARE *)
            ADDATK(RKTO);               (* ADD ITS ATTACKS *)
            DELATK(RKFR);               (* DELETE FROM ORIGINAL SQUARE *)
            MOVEIT(A);                  (* MOVE KING *)
          END
          ELSE  (* NOT CASTLE *)
          BEGIN
            PROACS(A);                  (* PROCESS CASTLE STATUS MODS *)
            MOVEIT(A);                  (* MOVE TO OR FROM KING OR ROOK
                                           SQUARE *)
          END;
        END;
      3:; (* NULL MOVE *)
      4:  (* CAPTURE *)
        IF RMEP THEN
        BEGIN  (* CAPTURE ENPASSANT *)
          INTS := XTRFS[XTSR[RMFR],XTSF[RMTO]];
                                        (* CAPTURED PAWN SQUARE *)
          MBCAPT(NBORD[INTS]);          (* UPDATE SCORE *)
          DELATK(INTS);                 (* DELETE CAPTURED PAWN
                                           ATTACKS *)
          PRPATK(INTS);                 (* PROPAGATE ATTACKS THROUGH
                                           PAWN *)
          MOVEIT(A);                    (* MOVE CAPTURING PAWN *)
        END
        ELSE  (* CAPTURE NOT ENPASSANT *)
          LOSEIT(A);                    (* PROCESS CAPTURE *)
      5:  (* CAPTURE AND PROMOTE *)
        BEGIN
          PROMOT(A);                    (* PROMOTE PAWN *)
          LOSEIT(A);                    (* PROCESS CAPTURE WITH PROMOTED
                                           PIECE *)
        END;
      6:  (* CAPTURE ACS *)
        BEGIN
          PROACS(A);                    (* PROCESS CASTLE STATUS MODS *)
          LOSEIT(A);                    (* PROCESS ROOK CAPTURE *)
        END;
      7:  (* CAPTURE POOK ACS, PROMOTE *)
        BEGIN
          PROMOT(A);                    (* PROMOTE PAWN *)
          PROACS(A);                    (* CHANGE CASTLE STATUS *)
          LOSEIT(A);                    (* PROCESS ROOK CAPTURE *)
        END;
  END;

  (* INITIALIZE MOVE GENERATION *)

  JNTM := OTHER[JNTM];                  (* SWITCH SIDE TO MOVE *)
  CPYRS(GENPN[JNTK],TPLOC[XTUMP[EP,JNTM]]);
  NOTRS(GENTO[JNTK],TMLOC[JNTM]);
  NOTRS(INRS,GENPN[JNTK]);
  ANDRS(GENFR[JNTK],TMLOC[JNTM],INRS);

  (* DETERMINE IF MOVE LEAVES KING IN CHECK, OR MOVES
     KING INTO CHECK *)

  ANDRS(INRS,TPLOC[XTUMP[EK,JNTM]],ALATK[OTHER[JNTM]]);
  RMCH := NOT NULRS(INRS);
  ANDRS(INRS,TPLOC[XTUMP[EK,OTHER[JNTM]]],ALATK[JNTM]);
  RMIL := NOT NULRS(INRS);
  UPDATE := NOT RMIL;
  IF NOT RMIL THEN                      (* COUNT LEGAL MOVES *)
    MVSEL[JNTK-1] := MVSEL[JNTK-1] + 1;

  (* INITIALIZE MOVE SEARCHING *)

  SRCHM[JNTK] := H1;
    NODES := NODES+1;                   (* COUNT NODES SEARCHED *)
  END;
END;  (* UPDATE *)


PROCEDURE GENONE                        (* STACK ONE GENERATED MOVE *)
  (A:TT;                                (* FROM SQUARE *)
   B:TS);                               (* TO SQUARE *)

VAR
  INRS : RS;                            (* SCRATCH *)

BEGIN
  WITH MOVES[JNTW] DO
  BEGIN
    RMFR := A;                          (* FROM SQUARE *)
    RMTO := B;                          (* TO SQUARE *)
    RMCP := NBORD[B];                   (* CAPTURED PIECE *)
    RMCA := (NBORD[B] <> MT);           (* CAPTURE *)
    IORRS(INRS,XRSS[A],XRSS[B]);
    ANDRS(INRS,INRS,CSTAT[JNTK]);
    RMAC := NOT NULRS(INRS);            (* AFFECTS CASTLE STATUS *)
    RMCH := FALSE;                      (* CHECK *)
    RMMT := FALSE;                      (* MATE *)
    RMIL := FALSE;                      (* ILLEGAL *)
    RMSU := FALSE;                      (* SEARCHED *)
    RMPR := FALSE;                      (* PROMOTION *)
    RMOO := FALSE;                      (* CASTLE *)
    RMEP := FALSE;                      (* ENPASSANT *)
  END;
  VALUE[JNTW] := 0;                     (* CLEAR VALUE *)
  IF JNTW < ZW THEN
    JNTW := JNTW+1;                     (* ADVANCE MOVES STACK POINTER *)
END;  (* GENONE *)


PROCEDURE PWNPRO;                       (* GENERATE ALL PROMOTION
                                           MOVES *)

VAR
  INTG : TG;                            (* PROMOTION TYPE *)

BEGIN
  MOVES[JNTW-1].RMPR := TRUE;           (* SET PROMOTION *)
  MOVES[JNTW-1].RMPP := PQ;             (* PROMOTE TO QUEEN FIRST *)
  FOR INTG := PR TO PB DO               (* GENERATE OTHER PROMOTIONS *)
  BEGIN
    MOVES[JNTW] := MOVES[JNTW-1];       (* COPY LAST MOVE *)
    MOVES[JNTW].RMPP := INTG;           (* CHANGE PROMOTE TO PIECE *)
    JNTW := JNTW+1;                     (* ADVANCE MOVE INDEX *)
  END;
END;  (* PWNPRO *)


PROCEDURE GENPWN                        (* GENERATE PAWN MOVES *)
  (A:RS;                                (* PAWNS TO MOVE *)
   B:RS);                               (* VALID DESTINATION SQUARES *)

VAR
  INRS, IMRS : RS;                      (* SCRATCH *)
  INTS : TS;                            (* DESTINATION SQUARE *)

BEGIN
  IF JNTM = LITE THEN
  BEGIN                                 (* WHITE PAWNS *)
    SFTRS(INRS,A,S2);                   (* ADVANCE ONE RANK *)
    ANDRS(INRS,TPLOC[MT],INRS);         (* ONLY TO EMPTY SQUARES *)
    CPYRS(IMRS,INRS);                   (* SAVE FOR 2 SQUARE MOVES *)
    ANDRS(INRS,B,INRS);                 (* ONLY VALID DESTINATION
                                           SQUARES *)
    WHILE NXTTS(INRS,INTS) DO
    BEGIN
      GENONE(XTLS[XTSL[INTS]-XTED[S2]],INTS);
                                        (* GENERATE SIMPLE PAWN MOVES *)
      IF INTS >= XTRFS[R8,F1] THEN
        PWNPRO;                         (* PROCESS PROMOTION *)
    END;
    ANDRS(INRS,IMRS,XRRS[R3]);          (* TAKE ONLY PAWNS ON THIRD *.
    SFTRS(INRS,INRS,S2);                (* ADVANCE ONE MORE RANK *)
    ANDRS(INRS,INRS,TPLOC[MT]);         (* ONLY TO EMPTY SQUARES *)
    ANDRS(INRS,INRS,B);                 (* ONLY VALID DESTINATION
                                           SQUARES *)
    WHILE NXTTS(INRS,INTS) DO
    BEGIN
      GENONE(XTLS[XTSL[INTS]-2*XTED[S2]],INTS);
                                        (* GENERATE DOUBLE PAWN MOVES *)
      MOVES[JNTW-1].RMEP := TRUE;       (* FLAG AS TWO SQUARES *)
    END;

    SFTRS(INRS,A,B1);                   (* TRY CAPTURES TO THE LEFT *)
    IORRS(IMRS,TMLOC[OTHER[JNTM]],ENPAS[JNTK]);
                                        (* OPPONENT PIECES + EP SQUARE *)
    ANDRS(IMRS,IMRS,B);                 (* VALID DESTINATION SQUARES *)
    ANDRS(INRS,INRS,IMRS);              (* CAPTURE MOVES TO LEFT *)
    WHILE NXTTS(INRS,INTS) DO
    BEGIN
      GENONE(XTLS[XTSL[INTS]-XTED[B1]],INTS);
                                        (* GENERATE CAPTURE MOVE *)
      MOVES[JNTW-1].RMCA := TRUE;       (* FLAG CAPTURE *)
      MOVES[JNTW-1].RMEP := INRSTB(ENPAS[JNTK],INTS);
                                        (* FLAG ENPASSANT CAPTURE *)
      IF MOVES[JNTW-1].RMEP THEN
        MOVES[JNTW-1].RMCP := DP;       (* SET CAPTURED PIECE TYPE *)
      IF INTS >= XTRFS[R8,F1] THEN
        PWNPRO;                         (* PROCESS PROMOTION *)
    END;

    SFTRS(INRS,A,B2);                   (* TRY CAPTURES TO THE RIGHT *)
    IORRS(IMRS,TMLOC[OTHER[JNTM]],ENPAS[JNTK]);
                                        (* OPPONENT PIECES + EP SQUARE *)
    ANDRS(IMRS,IMRS,B);                 (* VALID DESTINATION SQUARES *)
    ANDRS(INRS,INRS,IMRS);              (* CAPTURE MOVES TO LEFT *)
    WHILE NXTTS(INRS,INTS) DO
    BEGIN
      GENONE(XTLS[XTSL[INTS]-XTED[B2]],INTS);
                                        (* GENERATE CAPTURE MOVE *)
```

Circle 321 on inquiry card.

Available for Immediate Delivery

TRS-80 OWNERS

SOFTWARE NOW AVAILABLE
CASSETTE OR DISC (DOS)

BUSINESS – Complete small business, inventory control, invoicing, accounts receivable, accounts payable, general ledger, mailing list, payroll.

GAMES – Exciting new games to make your TRS-80 more fun.

HOME – Complete home budget, checkbook, savings, loans.

OVER 100 EXCITING PROGRAMS
–MORE ADDED EVERY DAY–

EDUCATION – ELECTRONICS – MARKETING
PROGRAMS FOR PET AND APPLE
Programs available in Level I BASIC and Level II BASIC. All Level II programs available on disc. Printer option on many programs. We also buy original programs. Custom programs also available. Send for a complete list of programs now.

SOFTWARE 80

25469 Hardt Street, Loma Linda, CA 92354

1976/1977

BYTE

Cover Price * plus postage and handling

1976/1977 BYTE Back Issues For Sale

Back Issues for sale

The following issues available:

'76	'77	'78
July	March	January
August	May	February
October	June	March
November	July	April
December	August	May
	September	June
	October	July
	November	August
	December	September

* Cover price for all issues thru August 1977 is $1.50 plus $.25 postage and handling ($3.50 total foreign). September '77 thru current '78 issues are $2.00 plus $.50 postage and handling ($4.00 total foreign).

Send requests with payment to: **BYTE Magazine**
70 Main Street
Peterborough, NH 03458
Attn: Back Issues

To further improve service to our customers we have installed a toll-free WATS line in our Peterborough, New Hampshire office. If you would like to order a subscription to BYTE, or if you have a question related to a BYTE subscription, you are invited to call* (800)258-5485 between 8:00 AM and 4:30 PM Eastern Time. (Friday 8 AM - Noon).

*Calls from continental U.S. only.

BYTE's New Toll-free Subscriber W.A.T.S. Line

(800) 258-5485

We thank you and look forward to serving you.

9178

Having Reservations About Your Software?

HUNT NO MORE!

Smoke Signal Broadcasting presents the

SE-1 SUPER EDITOR TEXT EDITING SYSTEM

Content oriented with string search and block move capability

- Allows any or all occurances of a particular string to be changed with one instruction.
- Disk transfer capability allows editing of files larger than available memory.
- Specific lines can be referenced by particular line number, offset amount or string of characters within the line.
- Automatic line numbering.
- Designed for file transfers to and from Smoke Signal Broadcasting's BFD-68 disk system.
- Complete source listing included.

Only $29.00 (on diskette)

We're the "CHIEF" in 6800 products software

SMOKE SIGNAL BROADCASTING
6304 Yucca/Hollywood, CA 90028/(213) 462-5652

Circle 320 on inquiry card.

Listing 1, continued:

```
      MOVES[JNTW-1].RMCA := TRUE;       (* FLAG CAPTURE *)
      MOVES[JNTW-1].RMEP := INRSTB(ENPAS[JNTK],INTS);
                                        (* FLAG ENPASSANT CAPTURE *)
      IF MOVES[JNTW-1].RMEP THEN
        MOVES[JNTW-1].RMCP := DP;       (* SET CAPTURED PIECE TYPE *)
      IF INTS >= XTRFS[R8,F1] THEN
        PWNPRO;                         (* PROCESS PROMOTION *)
    END;
  END
  ELSE
  BEGIN                                 (* BLACK PAWNS *)
    SFTRS(INRS,A,S4);                   (* ADVANCE ONE RANK *)
    ANDRS(INRS,TPLOC[MT],INRS);         (* ONLY TO EMPTY SQUARES *)
    CPYRS(IMRS,INRS);                   (* SAVE FOR 2 SQUARE MOVES *)
    ANDRS(INRS,B,INRS);                 (* ONLY VALID DESTINATION
                                           SQUARES *)
    WHILE NXTTS(INRS,INTS) DO
    BEGIN
      GENONE(XTLS[XTSL[INTS]-XTED[S4]],INTS);
                                        (* GENERATE SIMPLE PAWN MOVES *)
      IF INTS <= XTRFS[R1,F8] THEN
        PWNPRO;                         (* PROCESS PROMOTION *)
    END;
    ANDRS(INRS,IMRS,XRRS[R6]);          (* TAKE ONLY PAWNS ON THIRD *)
    SFTRS(INRS,INRS,S4);                (* ADVANCE ONE MORE RANK *)
    ANDRS(INRS,INRS,TPLOC[MT]);         (* ONLY TO EMPTY SQUARES *)
    ANDRS(INRS,INRS,B);                 (* ONLY VALID DESTINATION
                                           SQUARES *)
    WHILE NXTTS(INRS,INTS) DO
    BEGIN
      GENONE(XTLS[XTSL[INTS]-2*XTED[S4]],INTS);
                                        (* GENERATE DOUBLE PAWN MOVES *)
      MOVES[JNTW-1].RMEP := TRUE;       (* FLAG AS TWO SQUARES *)
    END;

    SFTRS(INRS,A,B3);                   (* TRY CAPTURES TO THE LEFT *)
    IORRS(IMRS,TMLOC[OTHER[JNTM]],ENPAS[JNTK]);
                                        (* OPPONENT PIECES + EP SQUARE *)
    ANDRS(IMRS,IMRS,B);                 (* VALID DESTINATION SQUARES *)
    ANDRS(INRS,INRS,IMRS);              (* CAPTURE MOVES TO LEFT *)
    WHILE NXTTS(INRS,INTS) DO
    BEGIN
      GENONE(XTLS[XTSL[INTS]-XTED[B3]],INTS);
                                        (* GENERATE PAWN CAPTURE MOVE *)
      MOVES[JNTW-1].RMCA := TRUE;       (* FLAG CAPTURE *)
      MOVES[JNTW-1].RMEP := INRSTB(ENPAS[JNTK],INTS);
                                        (* FLAG ENPASSANT CAPTURE *)
      IF MOVES[JNTW-1].RMEP THEN
        MOVES[JNTW-1].RMCP := LP;       (* SET CAPTURED PIECE TYPE *)
      IF INTS <= XTRFS[R1,F8] THEN
        PWNPRO;                         (* PROCESS PROMOTION *)
    END;

    SFTRS(INRS,A,B4);                   (* TRY CAPTURES TO THE RIGHT *)
    IORRS(IMRS,TMLOC[OTHER[JNTM]],ENPAS[JNTK]);
                                        (* OPPONENT PIECES + EP SQUARE *)
    ANDRS(IMRS,IMRS,B);                 (* VALID DESTINATION SQUARES *)
    ANDRS(INRS,INRS,IMRS);              (* CAPTURE MOVES TO LEFT *)
    WHILE NXTTS(INRS,INTS) DO
    BEGIN
      GENONE(XTLS[XTSL[INTS]-XTED[B4]],INTS);
                                        (* GENERATE PAWN CAPTURE MOVE *)
      MOVES[JNTW-1].RMCA := TRUE;       (* FLAG CAPTURE *)
      MOVES[JNTW-1].RMEP := INRSTB(ENPAS[JNTK],INTS);
                                        (* FLAG ENPASSANT CAPTURE *)
      IF MOVES[JNTW-1].RMEP THEN
        MOVES[JNTW-1].RMCP := LP;       (* SET CAPTURED PIECE TYPE *)
      IF INTS <= XTRFS[R1,F8] THEN
        PWNPRO;                         (* PROCESS PROMOTION *)
    END;
  END;
END;  (* GENPWN *)


PROCEDURE GENFSL                        (* GENERATE ALL MOVES FROM
                                           A SET OF SQUARES *)
  (A:RS);                               (* ORIGIN SET OF SQUARES *)

VAR
  INRS : RS;                            (* OUTER LOOP BIT BOARD *)
  IMRS : RS;                            (* INNER LOOP BIT BOARD *)
  IPRS : RS;                            (* PAWN ORIGIN BIT BOARD *)
  INTS : TS;                            (* OUTER LOOP SQUARE NUMBER *)
  IMTS : TS;                            (* INNER LOOP SQUARE NUMBER *)

BEGIN
  ANDRS(INRS,A,GENFR[JNTK]);            (* ONLY VALID FROM SQUARES *)
  NOTRS(IMRS,A);
  ANDRS(GENFR[JNTK],GENFR[JNTK],IMRS);  (* REMOVE ORIGIN SQUARES *)
  ANDRS(IPRS,A,GENPN[JNTK]);            (* VALID PAWN FROM SQUARES *)
  ANDRS(GENPN[JNTK],GENPN[JNTK],IMRS);  (* REMOVE PAWNS *)

  WHILE NXTTS(INRS,INTS) DO             (* LOOP THROUGH ORIGINS *)
  BEGIN
    ANDRS(IMRS,ATKFR[INTS],GENTO[JNTK]);
                                        (* GET UNPROCESSED DESTINATION
                                           SQUARES *)
    WHILE NXTTS(IMRS,IMTS) DO           (* LOOP THROUGH DESTINATIONS *)
      GENONE(INTS,IMTS);                (* GENERATE MOVE *)
  END;
  GENPWN(IPRS,GENTO[JNTK]);             (* GENERATE PAWN MOVES *)
END;  (* GETFSL *)


PROCEDURE GENTSL                        (* GENERATE ALL MOVES TO A
                                           SET OF SQUARES *)
  (A:RS);                               (* TARGET SET OF SQUARES *)

VAR
  INRS : RS;                            (* OUTER LOOP BIT BOARD *)
  IMRS : RS;                            (* INNER LOOP BIT BOARD *)
  IPRS : RS;                            (* PAWN BIT BOARD *)
  INTS : TS;                            (* OUTER LOOP SQUARE NUMBER *)
  IMTS : TS;                            (* INNER LOOP SQUARE NUMBER *)

BEGIN
  ANDRS(INRS,A,GENTO[JNTK]);            (* ONLY VALID TO SQUARES *)
  NOTRS(IMRS,A);
  ANDRS(GENTO[JNTK],GENTO[JNTK],IMRS);  (* REMOVE DESTINATION SQUARES *)
  CPYRS(IPRS,INRS);                     (* SAVE FOR PAWN MOVES *)

  WHILE NXTTS(INRS,INTS) DO             (* LOOP THROUGH DESTINATIONS *)
  BEGIN
    ANDRS(IMRS,ATKTO[INTS],GENFR[JNTK]);
                                        (* GET PIECES OF SIDE TO MOVE *)
    WHILE NXTTS(IMRS,IMTS) DO           (* LOOP THROUGH ORIGINS *)
      GENONE(IMTS,INTS);                (* GENERATE MOVE *)
  END;
  GENPWN(GENPN[JNTK],IPRS);             (* GENERATE PAWN MOVES *)
END;  (* GENTSL *)


PROCEDURE GENCAP;                       (* GENERATE CAPTURE MOVES *)

VAR
  INRS : RS;                            (* DESTINATION SQUARES *)

BEGIN
  IORRS(INRS,ENPAS[JNTK],TMLOC[OTHER[JNTM]]);
  GENTSL(INRS);                         (* GENERATE MOVES TO
                                           ENEMY SQUARES *)
END;  (* GENCAP *)


PROCEDURE GENCAS;                       (* GENERATE CASTLE MOVES *)

VAR
  INTQ : TQ;                            (* CASTLE TYPE INDEX *)
  INRS : RS;                            (* OCCUPIED SQUARES TEST *)
  IMRS : RS;                            (* ATTACKED SQUARES TEST *)

BEGIN
  FOR INTQ := XTMQ[JNTM] TO SUCC(XTMQ[JNTM]) DO
  IF INRSTB(CSTAT[JNTK],XTQS[INTQ]) THEN
                                        (* IF CASTLING IS LEGAL *)
  BEGIN
    ANDRS(INRS,XRQSO[INTQ],ALLOC[JNTK]);
                                        (* CHECK OCCUPIED SQUARES *)
    ANDRS(IMRS,XRQSA[INTQ],ALATK[OTHER[JNTM]]);
                                        (* CHECK ATTACKED SQUARES *)
    IF NULRS(INRS) AND NULRS(IMRS) THEN
                                        (* IF CASTLING IS LEGAL AND
                                           POSSIBLE *)
    BEGIN
      MOVES[JNTW] := XRQM[INTQ];        (* GENERATE CASTLING MOVE *)
      VALUE[JNTW] := 0;
      JNTW := JNTW+1;
    END;
  END;
END;  (* GENCAS *)


PROCEDURE GENALL;                       (* GENERATE ALL LEGAL MOVES *)

BEGIN
  GENFSL(ALLOC[JNTK]);                  (* GENERATE SIMPLE MOVES *)
  GENCAS;                               (* GENERATE CASTLE MOVES *)
END;  (* GENALL *)


PROCEDURE LSTMOV;                       (* LIST LEGAL PLAYERS MOVES *)

VAR
  INTW : TW;                            (* MOVES INDEX *)

BEGIN
  CREATE;                               (* CREATE DATA BASE *)
  GENALL;                               (* GENERATE ALL MOVES *)
  FOR INTW := AW+1 TO JNTW-1 DO
  BEGIN
    IF UPDATE(MOVES[INTW]) THEN;        (* SET ILLEGAL FLAG *)
    DNDATE(MOVES[INTW]);
  END;
END;  (* LSTMOV *)


PROCEDURE THEMOV                        (* MAKE THE MOVE FOR REAL *)
  (A:RM);                               (* THE MOVE TO MAKE *)

VAR
  INTB : TB;                            (* SCRATCH *)
  INRS : RS;                            (* SCRATCH *)
  INTQ : TQ;                            (* CASTLE TYPE INDEX *)
  INTS : TS;                            (* SCRATCH *)

BEGIN
  LSTMV := A;                           (* SAVE AS PREVIOUS MOVE *)
  INTB := UPDATE(A);                    (* UPDATE THE DATA BASE *)
  WITH BOARD DO                         (* AND COPY ALL THE RELEVANT DATA
                                           BACK DOWN *)
  BEGIN
    RBTM := JNTM;                       (* SIDE TO MOVE *)
    CPYRS(INRS,ENPAS[JNTK]);
    IF NXTTS(INRS,INTS) THEN            (* FIND ENPASSANT SQUARE *)
      RBTS := INTS
    ELSE
      RBTS := AT;
    IF JNTM = DARK THEN
      RBTI := RBTI+1;                   (* ADVANCE MOVE NUMBER *)
    FOR INTQ := LS TO DL DO
      IF INRSTB(CSTAT[JNTK],XTQS[INTQ]) THEN
        RBSQ := RBSQ+[INTQ]             (* CASTLE LEGAL *)
      ELSE
        RBSQ := RBSQ-[INTQ];            (* CASTLE NOT LEGAL *)
    FOR INTS := AS TO ZS DO
      RBIS[INTS] := NBORD[INTS];        (* COPY POSITION *)
  END;
END;  (* THEMOV *)
```

Text continued from page 171

makes use of the routines which were just described (eg: ADDATK, CUTATK, ADDLOC, CLSTAT, PRPATK, DELATK, MOVEIT, LOSEIT). The move is placed on the move stack by GENONE. Special routines exist for generating moves which involve the promotion of a pawn (PWNPRO) and for generating the standard pawn moves (GENPWN). When a move is tried and produces an α-β cutoff, the program backs down the look-ahead tree and begins to explore moves at a different node. Several procedures are employed to downdate the data base. These include the main routines RTRKIT and DNDATE, which are essentially the complement of MOVEIT and UPDATE. Two other procedures are also needed, one to unpromote a pawn (PAWNIT) and one to resurrect a captured piece (GAINIT). This set of routines permits the program to move about the look-ahead tree and incrementally update or downdate the data base.

The executive routines which are responsible for move generation are GENFSL, which generates all legal moves from a set of squares, and GENTSL, which generates all legal moves to a set of squares. The rationale for having two routines is that we wish to generate the moves in stages. For example, captures should be searched first at each node (ie: the capture heuristic). To do this, we identify the square locations of the opponent's pieces, and then call GENTSL to generate all capturing moves. These moves are searched before any other moves are generated. If one of these produces a cut-off the rest of the moves need not be generated at all. A third executive routine (GENCAS) generates all castling moves. These moves are generated after the captures if castling is still legal.

A fourth executive routine for move generation is GENALL. This procedure generates all legal moves and is used by the program to check the legality of the opponent's move. It is called by LSTMOV which makes a list of all the legal moves and each of these are compared with the opponent's move by YRMOVE (presented later). If the opponent's move is not on the list, the machine prints "illegal move." If the opponent's move is compatible with more than one of the moves on the list (eg: P-R3 could be either P-QR3 or P-KR3), the machine prints the message, "ambiguous move." When the machine has completed its own move selection or has determined that the opponent's move is legal and not ambiguous, the move is actually made by THEMOV.■

A "Tiny" Pascal Compiler

Part 3: P-Code to 8080 Conversion

Kin-Man Chung
124 Scottswood Dr
Urbana IL 61801

Herbert Yuen
POB 2591 Station A
Champaign IL 61820

In part 1 of this series (September 1978 BYTE, page 58) we defined a Pascal subset language in terms of syntax diagrams. The p-machine and its instruction set and a p-code interpreter were also described. In part 2 (October 1978 BYTE, page 34) we presented the design and implementation of the p-compiler. The subject matter for this part is the translation of p-codes to executable 8080 machine codes. We will also discuss the implementation of run time support routines and code optimization.

Compiler-Interpreter Systems

To understand why we need a p-code to 8080 translator, we should first take a brief look at the different structures of compiler-interpreter systems. The most widely used structure for microcomputers is the *interpreter*. Since interpreters are written in the target computer's assembly language, their memory size is small. They are self-contained in the sense that they include an editor for creating source programs and run time routines to do all computations. Memory storage for source programs is also small. The only disadvantage is speed. Execution time for a typical BASIC program is estimated to be about 300 to 1000 times the execution time of the same program written in assembly language. Interpreters may spend more than 70 percent of their time scanning source symbols character by character, parsing the syntax and checking errors. No matter how many times a program statement is executed, the parsing procedure is repeated every time.

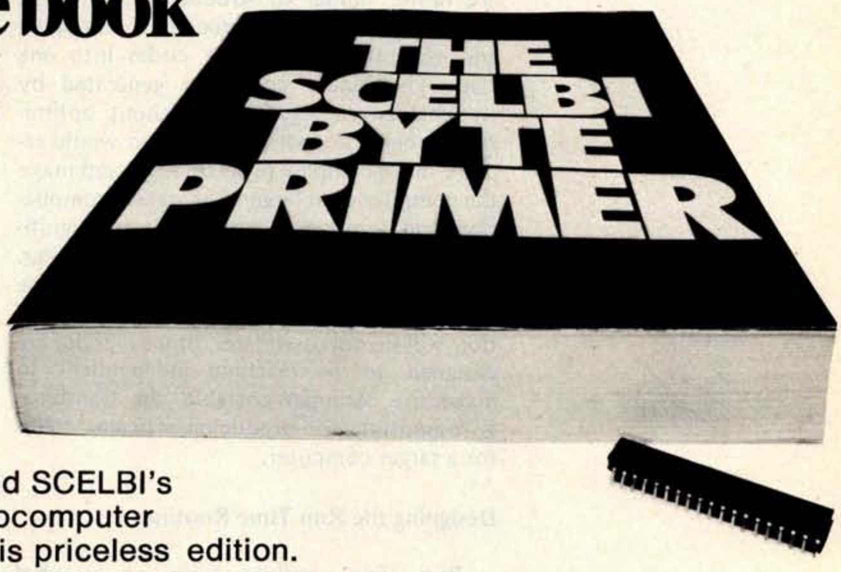
This problem can be readily solved by separating the parsing and execution steps. Before execution, the source program is compiled and intermediate code is generated. Thus scanning and parsing are done only once for each program statement. This is the so-called *compiler-interpreter* scheme used in some BASIC compilers. Execution of the intermediate codes is by interpretation. The gain in speed over a pure interpreter is a factor of approximately 2 to 10. However, the gain in speed is paid for by extra memory storage needed for intermediate codes.

The *compile-go* and *compile-link-go* approaches are commonly used for many high level language compilers in mainframe computer systems. These compilers generate relocatable binary codes. The compile-link-go approach has the advantage of linking together different modules of programs that are compiled separately, such as those in a subroutine library. This is done by a linking loader. However, due to limited system resources like memory and peripheral devices in microcomputers, these two structures are rarely used. Further, since Pascal is designed for fast compilation, linkage of program modules may be done at the source language level.

Among those four structures just mentioned, the compiler-interpreter seems to be most appropriate for implementation on microcomputers. However, execution speed is still slow because intermediate codes are interpreted rather than executed directly by the computer. An obvious solution to this problem is to translate the intermediate codes into executable machine codes. Thus, each intermediate code is decoded once by a program which we call a *translator*. The translated machine code can be expected to run about two to five times faster than interpreted intermediate codes. Therefore, the overall gain in speed, compared with a pure interpreter, is a factor of approximately 10 to 50. (Preliminary test runs in

Everything you've ever wanted to know about microcomputers in ONE complete book for only $10.95

Over 400 pages. Full 8½" x 11" size.



The ultimate book about microcomputers. Written by experts ... SCELBI and BYTE. Over 400 pages. A collector's item, featuring The Basics from the first 16 issues of BYTE and SCELBI's classic library of books. Your microcomputer bookshelf is incomplete without this priceless edition.

You can't buy information organized like this anywhere. This is the book that everyone who is into microcomputers needs for reference, for ideas, for clues to problem solving. It is a truly authoritative text, featuring easy-to-read, easy-to-understand articles by more than 50 recognized professional authors, who know and love microcomputers from the ground up. Logical and complete, it features many glossaries, and is illuminated with profuse illustrations and photographs.

The Scelbi/BYTE Primer is divided into four logical sections, that take you from point "0" through building and programming your own computer . . . step-by-step-by-step.

What can you do with a microcomputer? Checkbook balancing. Recipe converting and food inventory. Heating and air conditioning control. Home and business security and management. Playing the ponies. Analysis of the stock market. Maintaining massive data banks. Self-instruction. Toys and games. Small business accounting and inventory. And lots, lots more.

How does a microcomputer do it? Lots of "how to" theory. Introducing you to microcomputer operation. 6800, 6502, Z80 CPU chip capabilities. RAM and ROM memories. Addressing methods.

THE SCELBI/BYTE PRIMER

Over 400 pages. Selected articles from BYTE and SCELBI books. Profusely illustrated. Many photographs. $10.95, plus $1 shipping and handling.

How to control peripherals. Transmission of information to and from computers. Magnetic recording devices for bulk storage. Analog to digital conversion. How a computer can talk. Other I/O techniques. And more.

Order your copy today!

SCELBI COMPUTER CONSULTING INC.	BITS
Post Office Box 133 PP STN	25 Route 101 West
Dept. B	P.O. Box 428
Milford, CT 06460	Peterborough, NH 03458
	1-800-258-5477

Prices shown for North American customers. Master Charge. Postal and bank Money Orders preferred. Personal checks delay shipping up to 4 weeks. Pricing, specifications, availability subject to change without notice.

All about building a microcomputer system. Over 12 complete construction articles. Flip-flops. LED devices. Recycling used ICs. Modular construction. Making your own p.c. boards. Prototype board construction. Make your own logic probes. Construction plans for 6800 and Z80 computers. Building plans for I/Os — TV and CRT displays, cassette interfaces, etc. Mathematics functions. ROM programmer. Plus much, much more.

How to program a microcomputer. Programming for the beginner. Assembling programs by hand. Monitoring programs. Number conversions. Game of Hexpawn. Design your own assembler. Lots more.

And that's only the beginning! Others have spent millions acquiring the type of microcomputer information found within the 400 pages of **The Scelbi/BYTE Primer.** But, it costs you only $10.95, plus $1 for postage and handling, complete! You know the quality of Scelbi and BYTE. This is your assurance of excellence throughout this MUST text. Order your copy today! And, get one for a friend!

our system show that Pascal programs run about 15 times faster than the same program written in BASIC.) We call this structure *compile-translate-go.*

The five compiler-interpreter structures we discussed above are summarized in table 1. The compile-go and compile-translate-go are rather similar in structure. Compile-go actually combines the process of compiling and generating executable codes into one step. The binary codes are generated by straightforward algorithms without optimization, because code optimization would require more complex program logic and make the compiler even larger. Separating compilation and translation into two steps significantly reduces the size of the compiler. Local optimization techniques can also be applied during translation. Code optimization will be discussed later. Since p-codes are designed to be machine independent to make the compiler portable, the translator is responsible for producing efficient codes for a target computer.

Designing the Run Time Routines

Run time routines form an essential part of all compiler-interpreter systems in microcomputers. Large computers can do fixed point, floating point and decimal arithmetic with 32 bit or larger word sizes in single instructions. Many microcomputers, on the other hand, can do only basic integer arithmetic with 8 bit words (bytes). Therefore, multiple instructions are needed to implement 16 bit operations like multiply, divide, subtract, logical operations and multibit shifts. The run time routines, sometimes referred to as *run time support package*, are a collection of subroutines written in assembly language that can be called by an interpreter or any program to perform various arithmetic and logical operations. Usually they include subroutines for IO conversion between ASCII and binary data.

The design of run time routines for our compiler system is based on three principles:

- **Fast implementation and clarity**: A straightforward approach is followed so that the overall package can be debugged and tested quickly and modified easily.
- **Speed**: The best known algorithms are used for computer arithmetic to achieve fastest execution speed possible. However, tricks such as self-modifying code are not used.
- **Memory storage**: The package is expected to be fairly compact. Since p-codes are translated mostly into subroutine calls, the number of instructions to set up arguments to be passed to the subroutine should also be minimal.

As described in part 1, the p-machine has a data stack and four registers: stack pointer T, base register B, program counter P, instruction register I. Since the translator takes care of the program counter and p-code instructions are not needed after translation, all we need are the stack pointer and base register. In the current version of our run time routines, contiguous memory storage is used to represent the data stack. For the sake of program clarity and easy debugging, the 8080 machine stack is not used, although using it for dual purposes as a data stack and temporary storage for normal program logic is possible and probably more efficient.

Figure 1 shows the structural differences between the p-machine stack which we implement and the 8080 machine stack. Since

Table 1: Summary of different structures of compiler-interpreter systems.

Structure	Example	Step	Input	System software	Output	Remarks
interpreter	BASIC, APL interpreter	1	source program	interpreter (execution)		Most popular for microcomputers. Advantage: conserves memory space. Disadvantage: very slow execution speed.
compiler-interpreter	BASIC-E, Pascal compiler	1 2	source program intermediate code	compiler interpreter (execution)	intermediate code	The interpreter may overlay the compiler to save memory space. Advantage: faster execution speed.
compile-go	WATFIV,PL/C compiler	1	source program	compiler	executable code	Only used in large computers. Disadvantage: size is too big for microcomputers.
compile-link-go	FORTRAN IV, PL/I, COBOL compiler	1 2	source program binary code	compiler linking loader	binary code executable code	Widely used in large computers. Advantage: fast execution speed. Disadvantage: requires more system resources.
compile-translate-go	Pascal compiler (by authors)	1 2	source program p-code	compiler translator	p-code executable 8080 code	Advantage: size of compiler is reduced, fast execution speed, increased portability, easy implementation.

integer data is stored as pairs of 8 bit bytes (character strings are stored as single dimensional arrays, two bytes to each element and only the low order byte is used; see descriptions in part 1), each load instruction increments the stack pointer by 2. The order of the byte pair is arranged as high-low because it is more convenient to use than low-high. The stack pointer always points to the low order byte of the 16 bit integer, which is on top of the stack.

Register pair D,E is dedicated for use as the stack pointer, while registers H and L are mainly used for 16 bit operations such as DAD, LHLD, SHLD and PCHL. When needed, register pairs D,E and H,L can be easily exchanged using the XCHG instruction. Since the base address remains unchanged within a procedure block, a 2 byte fixed memory location (with symbolic name BB) is used to represent the base register. The LHLD and SHLD instructions are used to retrieve and update the base address value. A summary of register assignments for implementation of the p-machine is shown in table 2.

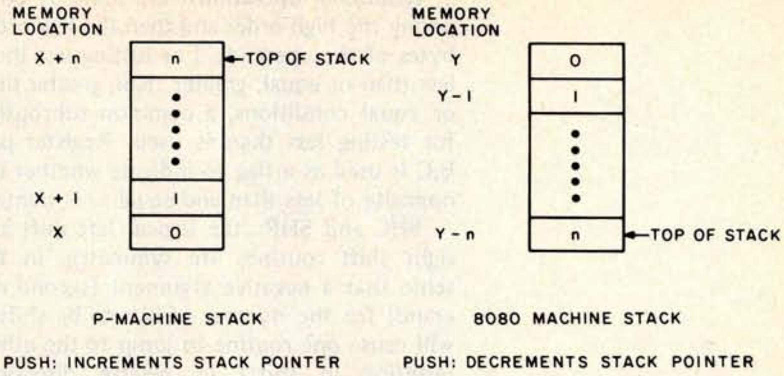


Figure 1. Differences between p-machine and 8080 stacks. This figure shows n+1 entries on each of the stacks.

P-machine	8080 run time routines
P: program counter	PC
T: stack pointer	D,E register pair
B: base register	memory location BB (16 bits)
I: instruction register	—
data stack	memory storage

Table 2: Register and storage assignment for runtime routines.

Coding the Run Time Routines

Most of the subroutines are easily understandable. The routines for load, store, call and load constant are coded by direct translation from the interpreter program to 8080 assembly language, keeping in mind that each stack element (one data item) occupies two bytes. The routines for arithmetic and logical operations and IO conversions require more programming effort. In general, single operand functions such as negate, logical not and increment are performed one byte at a time in register A. Double operand operations such as add, divide and logical or are performed with register pairs H,L and B,C. The entire runtime package occupies about 1 K bytes of memory. The following are remarks on coding some of the not-so-trivial subroutines.

PUSH and **POP**: for most double operand functions, subroutine POP is called first to get the two operands from the stack (memory) and put in register pairs H,L (first operand) and B,C (second operand). After the operations, subroutine PUSH is called to put the result from H,L back onto the stack.

Add and subtract: since DAD (double precision add) is the only 8080 instruction for double operand 16 bit operation, subtraction is done by adding the 2's complement of the second operand to the first. A message will be issued if overflow occurs and execution continues without any corrective action. The condition for overflow is detected by the rule:

if [sign(arg.1) $\oplus$ sign(arg.2) $\oplus$ carry $\oplus$ sign (result)] = 1, then overflow; otherwise nothing.

MULT16: 16 bit signed multiplication is done in two stages using an 8 bit multiplication routine. First, multiply the second operand by the high order byte of the first operand; the result is in register pair H,L. Second, continue the multiplication (left shift and double add) with the low order byte of the first operand; the result is in register pair H,L. This method is very efficient. In comparison, conventional 16 bit multiplication routines require more PUSH, POP and XCHG instructions because there are not enough registers to shift two 16 bit words and also update a loop counter. Overflows are ignored, as this is the usual practice for integer multiplication.

DIV16: 16 bit signed division is one of the most difficult routines to implement. First the signs of both operands are saved on a stack and are then converted to positive integers (actually the divisor is made negative in 2's complement because subtraction is done with a double add instruction). The divisor is also checked for zero value, and if so, a DIVIDE CHECK message is issued and the routine returns. Division is carried out as a sequence of subtraction and shifts. At the end, the signs of the quotient and remainder are corrected according to the original signs of the operands. The same routine is also used for calculation of the MOD function.

Relational operations: are done by comparing the high order and then the low order bytes of the operands. For testing less than, less than or equal, greater than, greater than or equal conditions, a common subroutine for testing less than is used. Register pair B,C is used as a flag to indicate whether the opposite of less than and equal to is wanted.

SHL and SHR: the logical left shift and right shift routines are symmetric in the sense that a negative argument (second operand) for the number of bits to be shifted will cause one routine to jump to the other, resulting in shifts in reverse direction.

INNUM: the conversion subroutine for input integers allows leading zeros and blanks and may optionally be preceded by a plus or minus sign (+ or −). It also checks for the absolute magnitude of the integer, which must be less than 32,768.

OUTNUM: conversion of binary integers to ASCII is done by repeated division by 10. The 16 bit divide routine is utilized.

P-code Translation

In general, p-codes are translated to subroutine call instructions which jump to the appropriate entry points in the run time routines. Output from the translator is an 8080 machine language program containing mostly subroutine call instructions. Some p-codes, such as load and store, require additional instructions to set up the arguments to be passed. Address offsets are always placed in register pair B,C and the static level difference is placed in register A. The jump instruction in p-code simply becomes a JMP instruction in 8080 with the correct address determined by the translator. The p-code addresses in CAL and JPC instructions are similarly taken care of by the translator. The complete list of 8080 code corresponding to each p-code is shown in table 3.

Hexadecimal Op code	P-code	8080 Mnemonic	Commentary
00	LIT 0,n	LXI B,n CALL LIT	
01	OPR 0,0	JMP P00;	procedure return routine
	OPR 0,n	CALL Pn ;	one of the 21 arithmetic/logical routines
02	LOD v,d		
	a) v=0	LXI B,2d CALL LOD	
	b) v>0	LXI B,2d MVI A,v CALL LOD1	
	c) v=255	CALL LODA;	load absolute address
12	LODX v,d		
	a) v=0	LXI B,2d CALL LODX	
	b) v>0	LXI B,2d MVI A,v CALL LODX1	
03	STO v,d		
	a) v=0	LXI B,2d CALL STO	
	b) v>0	LXI B,2d MVI A,v CALL STO1	
	c) v=255	CALL STOA;	store absolute address
13	STOX v,d		
	a) v=0	LXI B,2d CALL STOX	
	b) v>0	LXI B,2d MVI A,v CALL STOX1	

Hexadecimal Op code	P-code	8080 Mnemonic	Commentary
04	CAL v,a		
	a) v=0	CALL CAL JMP x	
	b) v>0	MVI A,v CALL CAL1 JMP x	
	c) v=255	CALL CALA;	machine language subroutine interface
05	INT 0,n	LXI H,2n CALL INT	
06	JMP 0,a	JMP x	
07	JPC 0,a	LDAX D; DCX D DCX D; RAR ; JNC x	get conditional code decrement stack pointer test conditional code
	JPC 1,a		(same as JPC 0,a except JC x)
08	CSP 0,n (n=0...5)	CALL SYSn;	one of the 6 conversion routines
	for n=8:		(output a string)
	LIT 0,c_1 LIT 0,c_2 . . . LIT 0,c_n LIT 0,n CSP 0,8	MVI C,n; CALL SYS8 DB c_1 DB c_2 . . . DB c_n	# of char.

Table 3: P-code to 8080 translation. LIT, LOS, STOX1, INT, LODA, etc, are used as symbolic entry points in the runtime routines. There are 22 routines for the OPR instructions: P00, P01, . . ., P21. There are seven standard routines for IO conversion: SYS0, SYS1, . . ., SYS5 and SYS8. The variable x *is used as the memory address in the translated 8080 code corresponding to p-code address* a *in a call and jump instruction.*

The 2 Pass Translator

The structure of the translator is similar to that of the interpreter. Both programs read p-codes from memory and decode them. The interpreter calls a simulator to execute the p-codes. The translator writes translated 8080 code in memory. The major difference between them is that the translator needs three additional tables to keep track of p-code and 8080 addresses. Since all p-code addresses are relative to the starting p-code of the program, the program is relocatable. The memory address corresponding to p-code address for any backward and forward referenced jumps can be calculated easily because all p-codes are four bytes long. The number of 8080 instructions generated per p-code is also not constant as shown in table 3. Therefore, it is necessary to build a table of 8080 addresses corresponding to p-code addresses to be used in jump and call instructions. However, it is not practical to build a table of 8080 addresses for every p-code because it will take too much memory storage for large programs. Only the addresses of those p-codes that are being referenced need be entered into the table.

P-code to 8080 machine code translation is done in two passes. During the first pass, p-code addresses in CAL, JMP and JPC instructions are entered into a table. The table is sorted after the completion of the first pass. Actual translation is carried out in the second pass. P-codes are fetched one by one from memory and decoded. The address of each p-code is checked with those in the address table. If it indicates that the current p-code is being referenced, the current 8080 address is entered to the corresponding 8080 address table. Then 8080 machine codes are produced according to the translation rules shown in table 3.

For CAL, JMP and JPC instructions, the p-code address in the instruction is looked up in the address table using a binary search. If the corresponding 8080 address has already been entered, it is output in the translated code; otherwise it is a forward referenced address. When the latter case occurs, it is necessary to record the current 8080 address in a *forward reference* table. Then, instead of the 8080 address (which is not yet known), its position in the table is output in the translated code. At the end of the second pass the forward referenced addresses are fixed up by the following procedure:

a) Get the 8080 address from the forward reference table (call it P).
b) Get the table entry (call it J) at address P in the translated program.
c) Get the updated 8080 address (call it A) at table entry J.

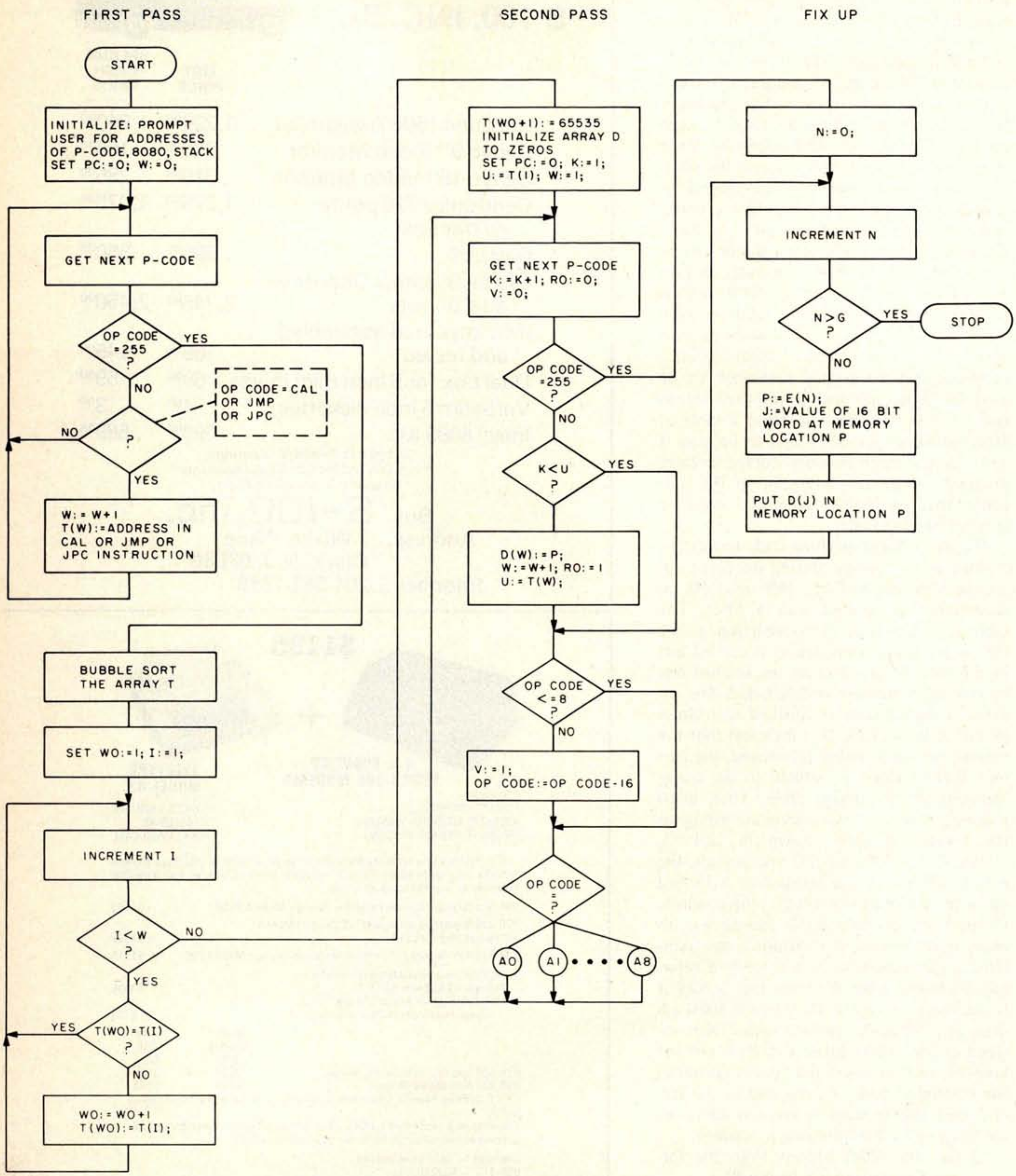


Figure 2: A simplified flowchart of the translator. A0, A1, . . .A8 are program segments for generating 8080 code for the p-code with peephole optimization as illustrated by the rules in table 4. Refer to table 5 for a description of the variables.

d) Write the correct address A back to memory location P.

Figure 2 is a simplified flowchart of the translator. The part for code generation is not shown, but it can be easily understood by referring to tables 3 and 4. Table lookup is done by binary search through the sorted table. The table elements are entered sequentially during the first pass. A simple bubble sort algorithm is used to sort the table. This method works fine for small Pascal programs. For larger programs, and thus more referenced addresses, the bubble sort algorithm is too slow because the number of comparisons is of order n^2 for n elements. A binary tree sorting algorithm with order $n \log n$ will be used for our next version of the translator.

The various entry points in the runtime routines are initialized in the translator as a series of string constants. These hexadecimal addresses are converted to integers and placed in arrays so they can be accessed very easily later on.

When execution begins, the program prompts the user for starting addresses of the p-code program, the output 8080 code, and starting and ending addresses of the data stack. The following three instructions are generated to initialize the data stack and pointer:

LXI H,STK1	starting address of data stack.
LXI D,STK2	2's complement of stack ending address.
CALL #1A00	runtime routine (initialization)

The program then begins its first pass. The number of address references and actual number of referenced addresses are displayed at the end of the first pass. During the second pass, cross references of p-code and 8080 addresses, which may be useful for future references, are listed in hexadecimal form. At the end of the translation, sizes of the p-code program and 8080 code are displayed.

Code Optimization

Code optimization is a technique employed by most compilers to improve the object code produced. Many sophisticated code optimization techniques are known today but are outside the scope of this article. We shall describe only one form of local optimization technique which is being used in our project. Local optimization

Table 4: Summary of peephole optimization. The goal is to reduce the size of the object program. The optimized code is more efficient than the unoptimized 8080 code. For the redundant store fix, the load instruction cannot be referenced elsewhere in the program.

Source of optimization	Example	P-code	8080 code	Optimized 8080 code
Redundant jump instructions	beginning of a procedure without inner procedure	n: JMP 0,n+1	JMP x	no code generated
Redundant loads and stores	J:=J+5; A[J]:=X;	STO v,d * LOD v,d	(as usual) (as usual)	(as usual) INX D; increment stack INX D; pointer
Repeated load of the same variable	A[J]:=A[J]+Y;	LOD v,d LOD v,d	(as usual) (as usual)	(as usual) CALL P21; copy
INT instruction with small constant	procedure call without parameter	INT 0,0	LXI H,#0000 CALL INT	no code generated
	procedure call	INT 0,n ($-3 \leq n \leq 2$)	LXI H,2n CALL INT	INX D INX D (repeat n times, $n > 0$) DCX D DCX D (repeat n times, $n < 0$)
Load negative constants	B:= −20;	LIT 0,n OPR 0,1	LXI B,n CALL LIT CALL P01	LXI B,−n CALL LIT
Add and subtract small constants ($n \leq 3$)	array subscripts A[J+2]:= B[K−1]:= L:=L+1;	LIT 0,n OPR 0,2 LIT 0,n OPR 0,3	LXI B,n CALL LIT CALL P02 LXI B,n CALL LIT CALL P03	CALL P19; increment (repeat n times) CALL P20; decrement (repeat n times)
Load zeros	P:=0;	LIT 0,0	LXI B,#0000 CALL LIT	XRA A INX D STAX D INX D STAX D

*Must be an unreferenced p-code

S-100 MAINFRAME $200

is done within a straight line block of code with no jumps into or out of the middle of the block. *Peephole* optimization is one form of local optimization which examines only small pieces of object code.

Since most code optimization techniques are difficult to build in a syntax directed code generation algorithm, peephole optimization is particularly useful in improving the intermediate code. Each improvement may lead to opportunities for further improvements. The technique can be applied repeatedly to get maximum optimization. In our translator, peephole optimization is applied only once during the second pass.

The goal of optimization is to minimize the size of the translated 8080 code and to increase execution speed without sacrificing a lot of time during translation. The peephole technique is quite simple. It examines only a single code or two consecutive codes. Some redundant p-codes are obvious and can be easily recognized. For example, the JMP instruction generated at the beginning of a procedure block which does not contain inner blocks is redundant. Similarly, the p-code INT 0,0 (increment stack pointer) generated after a procedure call with no arguments can be eliminated. The biggest benefit comes from optimizing redundant load and store instructions, because they are relatively slow in the current implementation. For example, a LOD instruction immediately following a STO instruction of the same variable can be replaced by an increment stack pointer instruction, because the variable is still on the stack. However, if the LOD instruction has a label, ie: is being referenced somewhere in the program, we cannot be sure that the STO instruction is always executed immediately before the LOD instruction.

Other sources of peephole optimization are the replacement of specific operations by more efficient instructions. Addition and subtraction of small constants (less than 4) occur frequently in array subscripts and loop counters. They can be replaced by repeated increment or decrement instructions. Some p-codes are translated into in line 8080 code instead of a call to runtime routines. Table 4 is a summary of peephole optimization used in the translator. Note that the optimized code always takes less memory space than the unoptimized code.

An Example

The various modules of the compiler system have been described. Now let us look at a complete program example. Listing 1 shows the compilation, translation and execution of a sample Pascal program. The program is stored in a disk file with file name T4. It is a sorting program that uses a binary tree algorithm. As mentioned before, it is more efficient than a bubble sort algorithm. The two subroutines in this program will be used in our next version of the translator (written in Pascal). The main program begins by asking the user to input an integer K (K must be less than 110) for the number of items to be sorted. It then reads the K+1 bytes of data starting from hexadecimal memory location 1A00 (the location where runtime routines are stored). The data items are read one at a time and procedure ENTER is called to build a binary tree with these items. Procedure TRAV is then called to traverse the tree recursively in the "left subtree..root..right subtree" fashion and the data with sorted order is placed in array S. Finally, array S is printed.

The p-compiler generates 145 p-codes (0 to 144) for this program. Afterwards, it uses a CHAIN statement (North Star BASIC) to load the translator program from disk, and overlays the compiler. The translator begins by asking the user to input memory addresses of runtime routines, p-code program, output 8080 code and data stack. At the end of the first pass, 20 address references are recorded. After sorting, it is found that there are only 15 actual labels. Output from the second pass of the translator is a cross-reference of p-code program counter and memory addresses of the corresponding translated 8080 code. The leftmost column is the p-code program counter. Hexadecimal memory addresses are printed in groups of 15 per line. With the exception of the first one, only the two low order hexadecimal digits are printed. At the end of the second pass, 11 forward references are recorded. A total of 766 bytes of 8080 code are generated. Compared to the size of the p-code program, the translated code is 1.32 times larger. This ratio usually ranges between 1.05 and 1.35, depending on program structure and the types of statements used.

After translation is completed, control is transferred to the disk operation system (DOS). The runtime routines are loaded from the disk file, PAS.LIB, to hexadecimal memory location 1A00. Then execution may begin by typing a JPxxxx command (jump to xxxx), where xxxx is the starting hexadecimal memory address of the translated code. In listing 1, two separate runs are shown: the first one sorts eight numbers (K+1 with K = 7) and the second sorts 21 numbers. The user may get back to BASIC by typing JP2A04, where 2A04 is the entry point of BASIC. (The command !CHR$(129)

T$	—	table of p-code address labels
D$	—	table of 8080 address corresponding to address labels in array T$
E	—	table of forward references
W	—	count of address references
WO	—	count of actual labels
G	—	count of forward references
K	—	p-code instruction counter
X	—	memory location of current p-code
P	—	8080 program counter of the translated code
F	—	current op code
V	—	=1 means indexed load or store
RO	—	=1 means current p-code is being referenced
U	—	program counter of the next referenced p-code

Table 5: Table of important variables and arrays in the translator program shown in flowchart form in figure 2.

```
P-CODES STARTS AT 0000
WANT CODE PRINTED?N
   0 ?$T4
   0  ( PGM -- SORTING BY BINARY TREE )
   0 VAR I,J,K,N,NEW:INTEGER;
   1 T,L,R,S:ARRAY[110] OF INTEGER;
   1  .
   1 PROC ENTER(N);
   1  VAR J:INTEGER;
   2  BEGIN J:=0;
   5   REPEAT
   5    IF N<=T[J] THEN
   9      IF L[J]<>0 THEN J:=L[J]
  17      ELSE BEGIN L[J]:=NEW;J:=0 END
  24    ELSE IF R[J]<>0 THEN J:=R[J]
  32          ELSE BEGIN R[J]:=NEW;J:=0 END
  39   UNTIL J=0;
  43   T[NEW]:=N;NEW:=NEW+1
  48  END;
  51
  51 PROC TRAV(J);  ( TRAVERSE THE TREE )
  51  BEGIN IF L[J]<>0 THEN TRAV(L[J]);
  62   S[K]:=T[J];K:=K+1;
  70   IF R[J]<>0 THEN TRAV(R[J])
  79  END;
  80
  80 BEGIN (MAIN)
  80  T[0]:=255;NEW:=0;
  86  READ(K#);WRITE(13,10);
  92  FOR I:=0 TO K DO BEGIN
  99   L[I]:=0;R[I]:=0; ENTER(MEM[I+%1A00]) END;
 116  K:=0; TRAV(0);
 121  FOR I:=0 TO K-1 DO WRITE(' ',S[I]#);
 140  WRITE(13,10)
 144 END.
INTERPRET(I), OR TRANSLATE(T)?T

*** P-CODE TO 8080 TRANSLATION ***
ADDR (HEX) OF PAS.LIB:1A00
ADDR (HEX) OF P-CODE:0000
ADDR (HEX) OF OUTPUT 8080 PGM:0800
STACK START ADDR (HEX):5000
STACK END ADDR (HEX):7FFF
 20 REFERENCES
 15 ACTUAL LABELS

   0  0809 0C 0C 12 17 1D 23 29 31 34 3B 41 49 4E 51
  15  0858 5E 66 6C 6F 75 7D 85 8A 90 93 99 A1 A6 A9
  30  08B0 B6 BE C4 C7 CD D5 DD E2 E8 EE F3 F6 FD 05
  45  090B 13 1B 21 1E 26 29 29 2F 35 3D 42 45 4C 52
  60  095A 62 64 6C 72 7A 82 8A 90 8D 95 9B A3 A8 AB
  75  09B2 B8 C0 C8 CA CD D3 D8 DE E4 E9 EF F2 F8 FE
  90  0A01 07 0A 0F 15 1B 1E 24 27 2E 34 39 3F 45 4A
 105  0A50 56 5C 5F 62 68 6A 70 73 79 7C 7E 83 89 8E
 120  0A94 96 9B A1 A7 AD AA AD B3 B6 BD C3 C6 CC D2
 135  0AD5 DB DE E4 E7 E9 EF F2 F8 FB FE
   11 FORWARD REFERENCES
P-CODE.. 145 INSTRUCTIONS
8080.. 766 BYTES
P-CODE:8080 = 1.3206897
* END TRANSLATION *
BYE
*LF PAS.LIB 1A00
*JP0800
?7
 29 34 34 35 43 43 235 242
*JP0800
?20
 0 1 5 25 29 29 32 33 34 34 35 35 40 43 43 112 113 201 235 242 244
*JP2A04
READY
!CHR$(129)
```

Listing 1: Compilation and translation of a sample Pascal program. At the end of the translation, the ratio of p-code to 8080 code is determined for reference purposes.

is an immediate BASIC statement used to turn off the printer.)

Summary

Compilers for high level languages are large, nontrivial programs. Their implementation usually requires a significant amount of computer system resources and human effort. Although our available system resources were limited, both in hardware and software, we managed to finish the bootstrap compiler within a relatively short time period. The reason is obvious: The Pascal subset we implemented is small. We followed the same approach professionals use for implementing portable Pascal compilers on mainframe computers. Syntax diagrams, which define the subset language, are used to construct the syntax directed, top-down parser of the compiler. The generation of p-code is also syntax directed. P-code is relocatable and portable, and its interpreter can be easily implemented on most microcomputers.

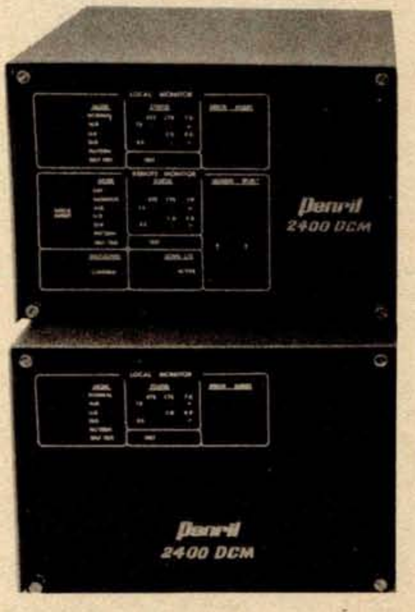
There are several features that are unique to our compiler project. First, the bootstrap compiler was written in BASIC (North Star disk BASIC). Although BASIC is not an appropriate language for compiler writing, it is the only high level language available in our system. Its ability to perform recursive function calls proved essential in simplifying the implementation of the compiler. Secondly, instead of writing a p-code interpreter in assembly language, a p-code to 8080 machine code translator was written in BASIC. The translated code can be expected to run more than twice as fast as interpreting p-codes. A p-code interpreter with debug facilities was also written (in Pascal). It can be used to debug p-code programs. Thirdly, minor extensions to the subset language were implemented. Absolute addressing of memory locations and machine language interface are desirable features for microcomputer systems. The availability of hexadecimal constants and IO conversions provides much user convenience.

Presently, the bootstrap compiler is very slow. It compiles at the rate of about eight lines per minute for a very dense Pascal program (using North Star BASIC with a 2 MHz 8080 processor). With some refinement in the compiler and runtime routines, the Pascal version of the compiler can be expected to run 25 times faster, or approximately 200 lines per minute.

Completion of the bootstrap compiler is only a milestone in our compiler project. There are many tasks still to be done. Logically the next step is to write the translator and then the p-compiler in the Pascal subset and compile them using the BASIC version of the compiler. Since the compiler source and p-codes are big, there may be a minor problem in memory management. It may be necessary to write the p-codes onto disk to save memory. After these two programs have been debugged, any further development can be done in Pascal without the BASIC interpreter. It would be quite interesting to have the compiler (in object code) compile itself (in source code) and use the output object code to compile itself again. After each compilation, the object code could be compared with the previous one to provide a means of verification.

More Pascal features or extensions can be implemented one step at a time. They may include character type and pointer type variables, disk IO capabilities, floating point arithmetic, multidimensional arrays and built-in functions. It is also necessary to improve the error diagnosis and recovery scheme of the compiler. Further development should be aimed at user convenience. A dynamic debugging package that can display and alter the values of variables as specified by name at runtime would be desirable. Ultimately, we hope to see a Pascal system that is as convenient and easy to use as an interactive BASIC system.■

What's New?



New Modem Provides Cost Effective Network Control

The 2400 DCM, Diagnostic Controlled Modem, provides control of a communications network. The diagnostics are contained within the master and remote site modems and provide a secondary channel with command mode procedures. Test and control of the remote site modems are performed over this secondary channel without interference to mainstream transmission.

Standard features include large scale integration logic, built-in test pattern generator and detector, EIA standard RS-232 interface, and on line compatibility with Bell system 201B and 201C data sets.

The 2400 DCM lines of modems range in price from $800 to $1600. Contact Penril Corp, 5520 Randolph Rd, Rockville MD 20852. ■

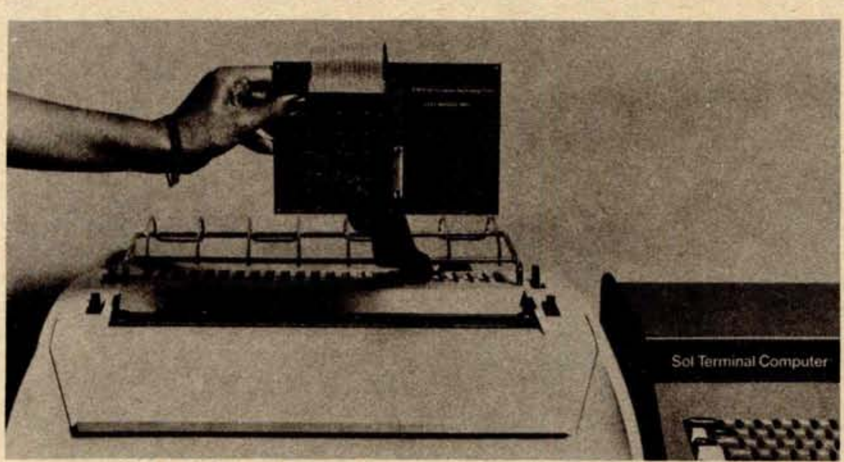
Circle 583 on inquiry card.

New Modem Assembly Houses Two ComData Series 330 Modems

The 330-2R-0 modem assembly houses two ComData Series 330 modems and provides display of four control functions and two data functions. The assembly may be used to house two 300 bps modems, one 300 and one 1200 bps modem, two 1200 bps modems, or one 1200 bps modem with reverse channel. The 300 Series modems are Bell 113B or 202 compatible and are available for leased line or auto answer applications. Since the assembly is compatible with the 330 rack mount series, its use at the terminal end permits maximum interchangeability of modems at both ends of the communications link. The 330-2R-0 is priced at $150. The 330 Series modem prices range from $150 to $225. Contact ComData Corp, 8115 N Monticello Av, Skokie IL 60076. ■

Circle 582 on inquiry card.

Printer Interfaces for SOL Computer



Two new printer interfaces for the SOL computer have been announced by Processor Technology Corp, 7100 Johnson Industrial Dr, Pleasanton CA 94566.

The SOL Hytype I mounts inside any Diablo Series 1200 printer, connecting it directly to the back of the SOL. Similarly, the SOL Hytype II printer interface works with the Diablo Series 1300 printer. The installation package includes a fully assembled, tested and burned in printed circuit board, software, all cables, and mounting hardware. The printer can be restored to its original condition if required.

Hytype driver software is included on a CUTS format cassette along with a source listing. The driver software may be modified to suit a particular application.

The price for both the Hytype I and Hytype II is $150. ■

Circle 580 on inquiry card.

Speech Recognition System

The SR/8 is a microprocessor speech recognition system that identifies up to 16 spoken words or phrases and which can be interfaced to any computer system. The SR/8 is trained for individual speakers and vocabularies by hearing each of the vocabulary words or phrases several times. From this training data, the SR/8 computes a prototype pattern describing each voiced input. These prototype patterns are then used to identify new spoken inputs via a decision process.

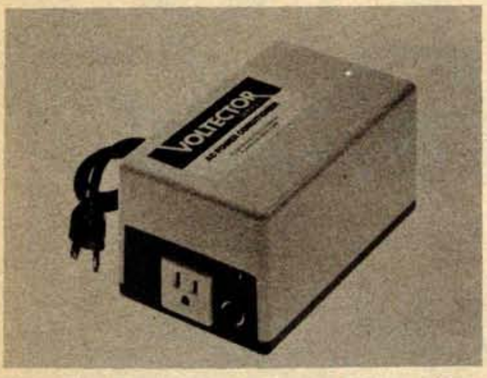
The system also has the capability to reject voice inputs that are not acoustically close to the prototype patterns derived during training, such as commands not in the vocabulary set or extraneous noises.

The system is available for $550 or as a self-contained, stand alone system with power supply and control panel for $975. Contact Phonics Inc, POB 62275, Sunnyvale CA 94087. ■

Circle 581 on inquiry card.

What's New?

New AC Power Conditioner



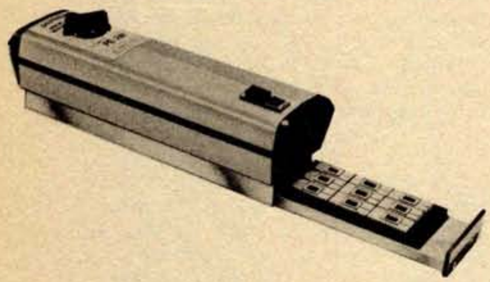
The new Voltector Series 5 AC power conditioner protects microcomputers and other sensitive electronic equipment from power line noise and high energy transients, saving the cost of a dedicated line. The Series 5 provides both common and transverse mode protection against RF (radio frequency) noise and against the destructive surges, spikes and transients that enter a building on the primary power line servicing it.

The conditioner is equipped with a green pilot light to indicate ready. It limits up to 2500 V spikes to safe levels and provides attenuation of frequencies above the line frequency. The Series 5 is internally fused and rated for 5, 10, 15 and 20 A at 120 V from 50 to 400 Hz.

Prices are: $169 5 A 0.6 kVA, $195 10 A 1.2 kVA, $248 15 A 1.8 kVA, and $327 20 A 2.4 kVA. Units are covered by a 5 year warranty. Contact Pilgrim Electric Company, 29 Cain Dr, Plainview NY 11803.■

Circle 614 on inquiry card.

New EROM-Erasing Ultraviolet Lamp



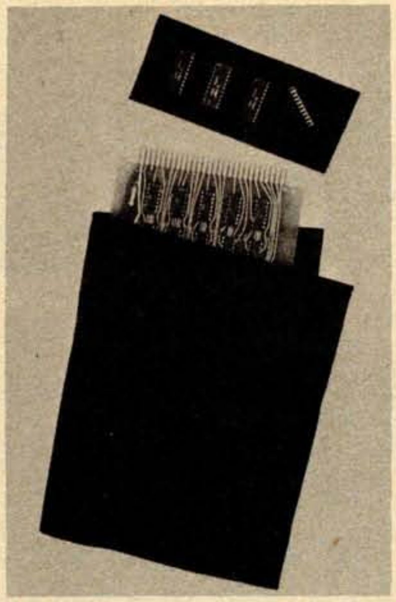
The introduction of a new erasable read only memory erasing ultraviolet lamp has been announced by Spectronics Corp, 956 Brush Hollow Rd, Westbury NY 11590. Designated the Spectroline PE-24T, the unit will erase up to nine integrated circuits in 12 minutes. The double tube construction and specular reflector provide broad, intense and uniform ultraviolet distribution.

Safety features include two high intensity ultraviolet tubes fully shielded within an anodized aluminum housing, and a safety interlock to prevent the unit from operating when the tray is not fully inserted.

A conductive foam pad holds the integrated circuits in place during exposure and prevents electrostatic buildup while protecting the integrated circuits from possible static change. A 60 minute timer is included for automatic shutoff of the unit.■

Circle 615 on inquiry card.

Conductive Foam Protects Static Sensitive Components

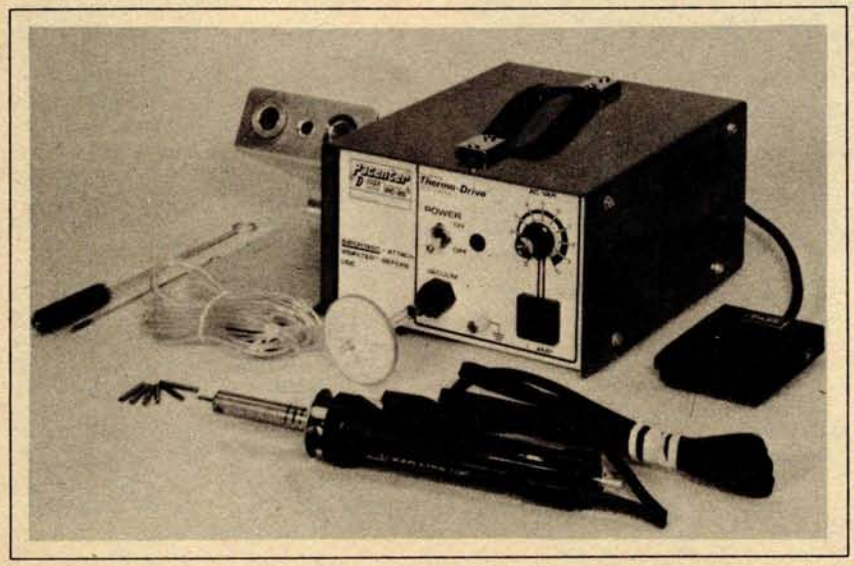


Electrically conductive foam for packaging integrated circuits, hybrid circuits, CMOS, microprocessor chips and other static sensitive devices has been introduced by Charleswater Products Inc, 3 Walnut Park, Wellesley MA 02181.

The foam is an electrically conductive packaging material that protects devices affected by both physical shock and static discharge. Available in low density for cushioning and in high density for inserting leads directly, the foam is unaffected by relative humidity. This conductive foam is available in thicknesses from 0.125 to 2.25 inches (0.32 to 5.72 cm) in 25 by 75 inch sheets (63.5 by 190 cm). Prices start at $18.50 per sheet. Literature and samples are available on request.■

Circle 616 on inquiry card.

Self-Contained Power Desoldering System

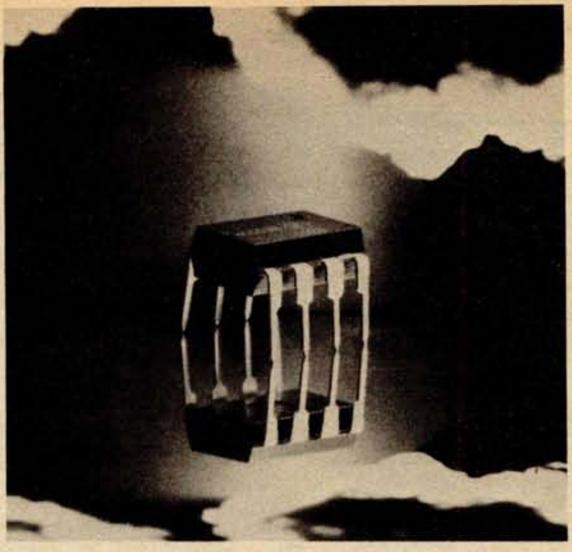


The new Pacenter series of equipment has been announced by Pace Inc, 9329 Fraser St, Silver Spring MD 20910. The Model PC-20 D-Sodr System is the first equipment to be marketed in this series.

The PC-20 is a lightweight, self-contained power desoldering system. Incorporating Pace's "whisper quiet" minipump, the under 7½ pound (3.4 kg) system provides low cost operation with high performance.

The PC-20 features Pace's Sodr-X-Tractor handpiece with "Thermo Drive" heat control. The power source provides a variable AC output for tip temperature control; vacuum supply with quick connect vacuum output; dual filtration system including plug-in filter; auxiliary ground lug; and foot pedal controlled operation. A hot cubby is provided for holding the handpiece and an assortment of Pace Super Tips is furnished with each system.■

Circle 617 on inquiry card.



THE BEST FOR BOTH WORLDS

Factory prime electronic equipment for both the discerning hobbyist and the professional

The New Hobby World Catalog

Your source for factory prime, professional quality equipment. Computers, add-on boards, IC's, sockets, resistors, supplies, tools, test equipment, books, and more. Shop your buy list at Hobby World. You'll find what you want, and at a solid savings.

This month's specials.

16K MEMORY ADD-ON FOR APPLE OR TRS-80
Hobby World price is only **$128.00** (specify when ordering)

NEW FROM SSM. THE CB-1 8080 CPU BOARD AT A HOBBY WORLD SUPER SPECIAL. (See SSM ad for specs) It's Loaded. It's Blue. IT'S ONLY **$119.95**

THE NEW ELENCO 3½ DIGIT SOLID STATE MULTIMETER
The ultimate in performance: measures resistance to .01 ohms, voltage to 100 micro-volts, current to one micro-amp. Assembled and tested, with 2-Year warranty. Lists at **$99.95.** Hobby World: **$74.95**

WAHL ISO-TIP CORDLESS SOLDERING IRON
Includes ni-cad batteries and wall plug transformer. Lists at **$19.95,** but Hobby World has it for **$14.95!**

S-100 WIRE WRAP BOARD
Over 3600 holes, for super versatility. Provisions for 4 regulators. Double-sided, plate-through. All S-100 pins labelled for number and signal. The PT-1. Hobby World Price: Only **$30.00**

Hobby World
19355 Business Center Drive #6
Northridge, CA 91324

TELEPHONE ORDERS
Inside Calif: 213 886-9200
Outside Calif: 800 423-5387

Hobby World Carries TAB Books! Affordable books for the hobbyist. Hobby World helps out with a 10% discount. (Order by number, and don't forget the 10% discount)

		LIST PRICE
Computer Circuits and How They Work	#538	**$5.95**
Beginner's Guide to Computer Logic	#548	**$5.95**
Computer Technician's Handbook	#554	**$9.95**
Build Your Own Working Robot	#841	**$5.95**
Programming Microprocessors	#985	**$6.95**
Beginner's Guide to Computer Programming	#574	**$9.95**
Digital Electronics: Principles and Practice	#585	**$5.95**
Microprocessor/Micro-programming Handbook	#785	**$6.95**
Computer Programming Handbook	#752	**$9.95**

Order by mail. Just write. Or order by phone. Toll free outside California.

SEND ME A CATALOG!

Name ____________________

Address ____________________

City ____________ State ____________

Satisfaction 100% Guaranteed

BANKAMERICARD VISA

master charge THE INTERBANK CARD

SEND ME THE FOLLOWING SPECIALS!

Item	Quantity	Price
	Postage/Handling	$1.50

California Residents add 6% Sales Tax
Charge My MC
VISA # ___ — ___ — ___

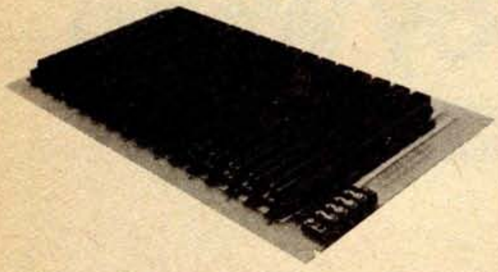
Signature ____________________

Expiration ____________________

Circle 170 on inquiry card.

What's New?

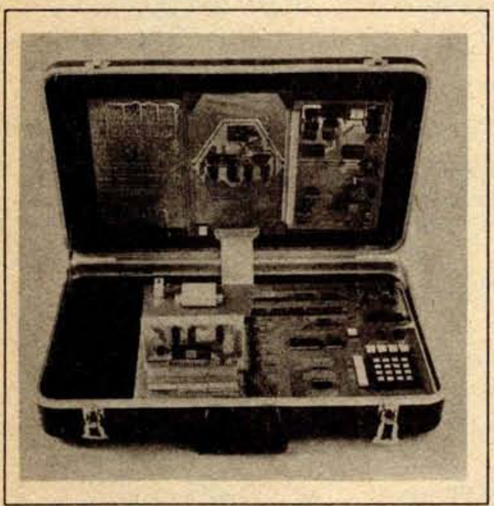
Shielded Motherboard Compatible with S-100 Computer Systems



Compatible with the S-100 computer systems, the Artec shielded motherboard is made of 0.125 inch (0.32 cm) FR4 glass epoxy. The board is designed to eliminate bus noise caused by unwanted coupling between adjacent lines on the motherboard in S-100 computer systems. The use of substantial ground traces between adjacent bus lines on this board eliminates line to line coupling. The motherboard comes in a 16 slot configuration, fits any standard chassis and requires no soldering. Price for the motherboard is $190 assembled and tested and $150 in kit form. For more information contact Artec, 605 Old County Rd, San Carlos CA 94070.■

Circle 637 on inquiry card.

Microcomputer Training and Development System



The TDS-M68 microcomputer training and development system offers the personal computer enthusiast an integrated system complete with microcomputer, self-teaching lab manual, textbook and experimenter's lab package. This complete and self-contained system brings together all the ingredients needed for a hands-on training course aimed at those who need to develop working capability with microcomputers.

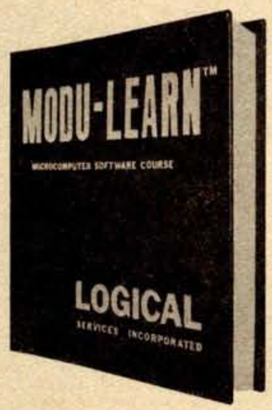
The system uses the M6800 microcomputer and features 2 K bytes of memory (expandable to 8 K bytes by simply adding the memory integrated circuits), a lab manual that guides the user step by step through the learning process, a lab package that includes computer devices (LEDs, relays, small motors, etc) that are interfaced to and controlled by the computer, plus a practical, applications oriented textbook covering programming, interfacing and application concepts. An instructor's guide and overhead transparencies are also available.

Six general purpose IO ports, utilizing the SwTPC bus configuration, accept a line of interface boards that includes parallel and serial IO ports, cassette interface, interrupt timer, calculator module, erasable read only memory programmer, etc.

All items are available separately or as a total package. A descriptive brochure is available from Technical Devices Ltd, 1138 Main St, Winnipeg, Manitoba CANADA R2W 3F3.■

Circle 638 on inquiry card.

Microcomputer Programming Home Study Course



A home study course in microcomputer programming for beginners is available from Logical Services Inc, 1080H E Duane Av, Sunnyvale CA 94086. The Modu-Learn course presents systematic software design techniques and structured programming in ten instructional lessons complete with problems, solutions, and practical examples in 8080 and 8050 assembly language. Background material on microcomputer architecture, hardware and software tradeoffs and useful reference tables are included in this book of over 500 pages. It is conveniently bound in a notebook for easy reference.

The price is $49.95.■

Circle 639 on inquiry card.

SD5 Modular Solder and Desolder System

The SD5 modular solder and desolder system consists of a 1 pound (0.45 kg) spool of hermetically sealed MIL spec solder with a D5 desolder wick dispenser tool snapped into the core of the spool. The SD5 is a simple, practical solution to the problem of alternating between soldering and desoldering. The unit is completely refillable.

The design of the removable D5 desoldering tool allows it to retreat or snap in or out of the solder spool for the convenience of the user. The tool has a 2½ inch (6.3 cm) heat resistant Teflon probe permitting pinpoint wick application even in densely packed electronics.

All components of the SD5 system are available packaged for sale separately as well as complete units. Solder is provided in 16, 18 and 21 gauge in the following alloys: 63/37 (eutectic), 60/40, 50/50 and 40/60. The wick is available in two gauges, 0.06 inch and 0.10 inch (0.15 and 0.25 cm) to cover all desoldering requirements.

For more information contact Chemtronics Inc, Solder Products Division, 45 Hoffman Av, Hauppauge NY 11787.■

Circle 640 on inquiry card.

New Wire Wrap Panel and Card Catalog

This 28 page brochure covers EECO's newest pin in board type ALA wire wrap panels, cards, drawers and frames. Products are documented with photos, descriptions, specifications, outline drawings and prices.

Write EECO, 1441 Chestnut Av, Santa Ana CA 92701 for this brochure.■

Circle 641 on inquiry card.

JADE

PLACE ORDERS TOLL FREE:

800/421-5809 Continental U.S.
800/262-1710 inside California

THE DISCONTINUED VECTOR GRAPHIC KITS ARE STILL AVAILABLE WHILE SUPPLY LASTS!

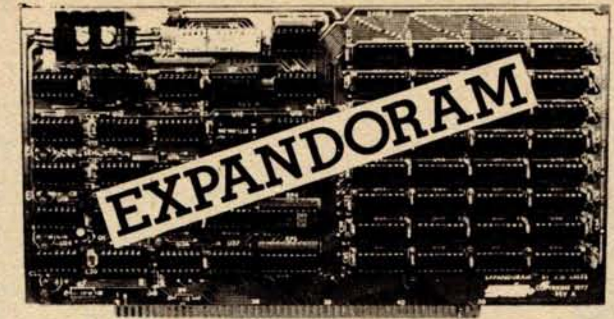
MICROPROCESSORS	
F8	16.95
Z80	20.00
Z80A	25.00
CDP1802CD	17.95
2650	19.95
AM2901	20.00
6502	11.95
6800	16.95
6802	25.00
8008-1	12.95
8035	20.00
8035-8	21.00
8080A	10.00
8085	23.00
TMS9900TL	49.95
8080A SUPPORT DEVICES	
8212	2.90
8214	4.65
8216	2.75
8224	4.30
8224-4	9.95
8226	2.75
8228	6.40
8238	6.40
8243	8.00
8251	7.50
8253	20.00
8255	6.45
8257	20.00
8259	20.00
8275	75.00
8279	18.50
USRT	
S2350	10.95
UARTS	
AY5-1013A	5.25
AY5-1014A	8.25
TR1602B	5.25
TMS6011	5.95
IM6402	9.00
IM6403	9.00
6800 PRODUCT	
6810P	4.00
6821P	6.60
6828P	11.25
6834P	16.95
6850P	8.65
6852P	11.00
6860P	9.25
6862P	12.00
6871P	28.75
6875P	8.75
6880P	2.50
68B10P	6.00
KIM	
6102	8.00
6502	11.95
6520	10.00
6522	9.25
6530	15.95
6530-002	15.95
6530-003	15.95
6530-004	15.95
6530-005	15.95
6532	17.95

CHARACTER GENERATORS		
2513	Upper (-12±5)	6.75
2513	Lower (-12 ±5)	6.75
2513	Upper (5volt)	9.75
2513	Lower (5 volt)	10.95
MCM6571	Up Scan	10.95
MCM6571A	Down Scan	10.95

PROM'S	
1702A	5.00
2708	8.95
2716(5+12)TI	25.00
2716(5v) INTL	38.00
2758(5v)	23.40
DYNAMIC RAMS	
416D/4116	16.00
2104	4.00
2107B-4	3.95
TMS4027	4.00
MM5270	4.50
MM5280	3.60

STATIC RAMS	1–24	25-99
21L02	1.50	1.20
21L02	1.75	1.50
2101-1	2.95	2.60
2111-1	3.25	3.00
2112-1	2.95	2.65
2114L	9.50	8.00
2114L	10.50	9.00
TMS4044	8.95	8.00
TMS4044	9.95	8.75
4200A	10.95	9.25

FLOPPY DISC CONTROLLER	
1771B01	39.95
1781	69.95
KEYBOARD CHIPS	
AY5-2376	13.75
AY5-3600	13.75
MM5740	18.00
MM5743	18.00



EXPANDO-32 KIT
Uses 4115 (8Kx1) Dynamic RAM's, can be expanded in 8K increments up to 32K.

8K	$179.00
16K	$255.00
24K	$325.00
32K	$400.00

EXPANDO-64 KIT
Uses 4116 (16Kx1) Dynamic RAM's, can be expanded in 16K increments up to 64K.

16K	$260.00
32K	$579.00
48K	$757.00
64K	$925.00

apple II computer
One of the best "Total Package" home and business computers on the market. "Basic" in ROM, Color Graphics, Floating Point Basic Package, etc.
16K version only **$1,095.00**

4116 (16K x 1, 300ns) MEMORY EXPANSION KIT
Dynamic RAM chip can be used for expanding APPLE II or TRS-80. Instructions incl.
8 for $98.00

EPROM BOARD KITS

RAM'N'ROM (16K any EPROM)	$117.00
MR-8 (1K RAM, uses 2708)	$99.50
MR-16 (1K RAM, uses 2716)	$99.50
EPM-1 (uses up to 4K of 1702)	$59.95
JG8/16 (uses 2708 or 2716)	$59.95

THE PIGGY IS COMING!

JADE PARALLEL/SERIAL INTERFACE
S-100 compatible, 2 serial I/O ports, 1 parallel I/O.

Kit JG-P/S	$124.95
Ass. & Tested JG-P/SA	$179.95
Bare Board with manual	$30.00

FLOPPY DISK INTERFACE
JADE FLOPPY DISK (Tarbell board)

Kit	$175.00
Assm. & Tested	$250.00

S.D. Computer Products "Versa Floppy"

Kit	$159.95
Assm. & Tested	$189.95

MOTHER BOARDS, S-100 STYLE
9-Slot "Little Mother"

Kit	$85.00
Assm. & Tested	$99.00
Bare Board	$35.00

13-Slot with front panel slot

Kit	$95.00
Assm. & Tested	$110.00
Bare Board	$40.00

22-Slot
Assm. & Tested $149.00

GOLD PLATED S-100 EDGE CONNECTORS
Soldertail $3.25 each, 10 for $30.00
Wire Wrap $4.50 each, 10 for $40.00

JADE
Computer Products

4901 W. ROSECRANS AVENUE
Department "B"
HAWTHORNE, CALIF. 90250

master charge · VISA · AMERICAN EXPRESS
Cards Welcome

Cash, Checks, Money Orders, and Credit Cards accepted. Add freight charge of $2.50 for orders under 10-lbs. Add 6% sales tax on all parts delivered in California. Discounts available at OEM quantities.

WRITE FOR OUR FREE CATALOG !

JADE 8080A
with full documentation

Kit	$100.00
Assm. & Tested	$149.95
Bare Board	$30.00

JADE Z80
with provisions for ONBOARD 2708 and POWER ON JUMP

2 MHz	
Kit	$135.00
Assm. & Tested	$185.00
4 MHz	
Kit	$149.95
Assm. & Tested	$199.95
Bare Board	$35.00

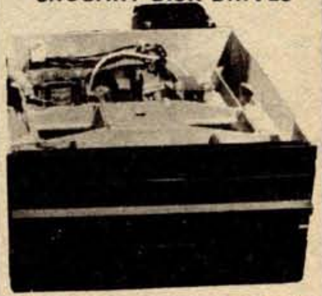
MD-690a CPU BOARD
S-100 Compatible 6800 MPU 1K x 8 RAM, PROM expandable to 10K.
$199.95

TU-1
Convert your T.V. set into a Video Monitor
Kit **$8.95**

JADE VIDEO INTERFACE
S-100 Compatible Serial Interface with Sockets Included.

Kit	$117.95
Assm. & Tested	$159.95
Bare Board with manual	$35.00

SHUGART DISK DRIVES



SA 400 **$295.00**
Single-density 5¼", 35 track drive.

SA801R **$495.00**
Single-sided 8" floppy disc drive.

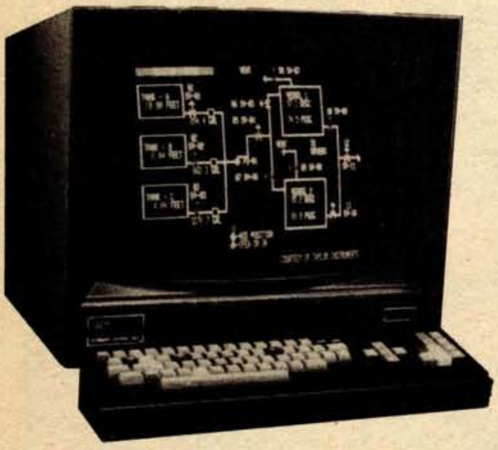
DM 2700-S **$750.00**
Includes SA801R disk drive, 10" x 10" x 16" cabinet, power supply, data cable, fan, AC line filter.

STATIC RAM BOARDS
JADE 8K

Kit JG8K (450ns)	$125.95
Assm. & Tested JG8KA (250ns)	$139.75
Kit JG8K (250ns)	$149.75
Assm. & Tested JG8KA (250ns)	$169.75
Bare Board without parts	$25.00
16K -Uses 2114's (lo power)	
Assm & Tested RAM 16 (250ns)	$375.00
Assm. & Tested 16B (450ns)	$325.00
Mem-2 Kit (250ns)	$285.00
16K Static with memory management	
Assem.& Tested RAM 65 (250ns)	$390.00
Assm. & Tested RAM 65B (450ns)	$350.00
Seals 32K	
Assm. & Tested JG32 (250ns)	$795.00
Assm. & Tested JG32B (450ns)	$725.00
Kit JG32K (250ns)	$575.00

Circle 195 on inquiry card.

New Color Graphics Data Terminal



This standard graphic data terminal, designated 8001G, is an addition to the Intecolor 8001 product series. The standard features of the terminal are: complete graphic software, selection of eight foreground and background colors, 19 inch color video tube, selectable bps rate to 9600, RS-232C connection, 48 lines and 80 characters per line, page roll, insert and delete — character or line, and 64 ISA special characters. The graphic software allows the user to plot bar graphs in both horizontal and vertical directions, plot points, lines and vectors by specifying the coordinates in a matrix of 160 by 192. Each graphic mode is specified by codes that can be initiated by keyboard or host computer through RS-232C. The 8001G is priced at $2750 in quantities of one to 24 or $1925 on a cash basis or in large quantities. For further information contact Intelligent Systems Corp, 5965 Peachtree Corners E, Norcross GA 30071.■

Circle 571 on inquiry card.

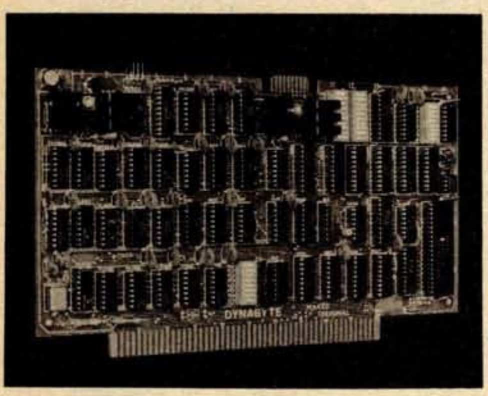
Automatic Answer and Dial Modem for LSI-11

An automatic answer and dial, low speed modem for the Digital Equipment Corp (DEC) LSI-11, LSI-11/2, and PDP-11/03 computer families is available from Nortek Inc, 2432 NW Johnson, Portland OR 97210. The complete system, contained on a dual width board, provides computer controlled answering and origination of data communication functions when used in conjunction with a TELCO CBS type DAA unit. Software selectable transmission rates include 110, 134.5, 300, and 600 bps. Number of data bits and parity are also software selectable, enabling use with most available data communications terminals. Emulating a DEC DLV-11E serial interface, the modem is software transparent to the TR-11 V3 and TSX operating systems when used in automatic answer mode.

The basic unit includes interconnecting cable for the DAA, software for an automatic dialing device handler, and is priced at $650. Additional software is available for intelligent terminal use.■

Circle 572 on inquiry card.

Complete Dumb Terminal on an S-100 Board



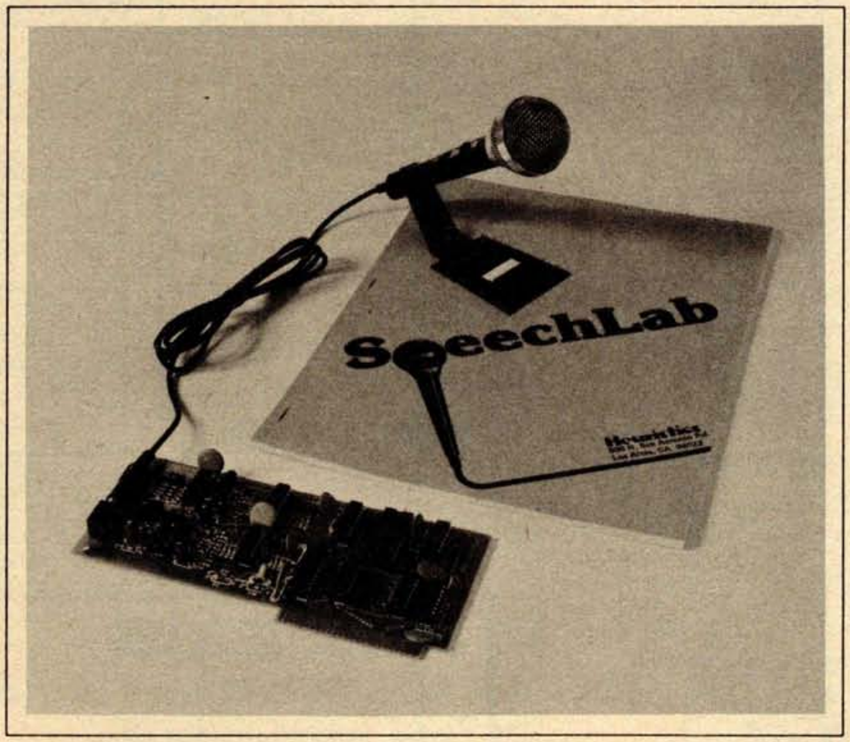
The Naked Terminal is an S-100 module that functions with a keyboard and video monitor. It contains a microprocessor with memory, software drivers and internal bus and displays 80 characters by 24 lines, using both upper and lower case characters in a 5 by 7 font. It features half duplex, full duplex and a block mode which allows editing before transmission. Editing is aided by an addressable cursor. Switch selectable features include black on white or white on black, blinking or nonblinking underline cursor, and variable bps rates. Software is not required.

The terminal can be configured by a dual-in-line package switch to drop into an existing system, replacing the serial input and output (IO) card and stand alone terminal without making any changes to software.

The Naked Terminal is completely assembled, socketed, tested and burned in, and carries a full year guarantee. For more information about this terminal, write to Dynabyte Inc, 4020 Fabian, Palo Alto CA 94303.■

Circle 573 on inquiry card.

Voice Input For Apple II Computer



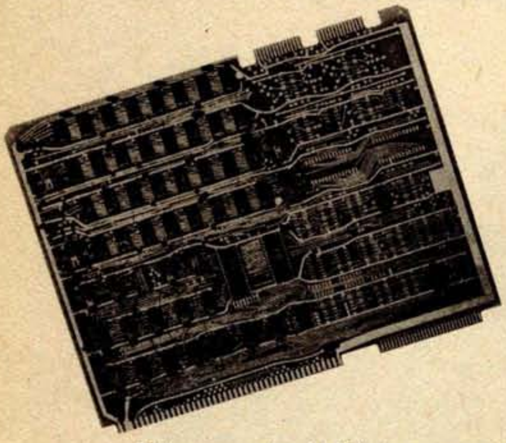
A new voice data input unit for the Apple II computer has been announced by Heuristics Inc, 900 N San Antonio Rd, Los Altos CA 94022. Known as Speechlab Model 20A, it features a 32 word vocabulary, fast real time response and the capability of multiple training samples for high accuracy.

The unit complies with Apple II computer "smart" peripheral conventions and interfaces directly with user written BASIC programs. The program is contained on an on board programmable read only memory which is automatically executed by the Apple II monitor program when speech input is desired.

A high fidelity microphone and a user manual with six demonstration programs (including Mastermind, blackjack and Shooting Stars) written in Apple BASIC are included. The price is $189 and the unit can be obtained from Heuristics, 900 N San Antonio Rd, Los Altos CA 94022.■

Circle 574 on inquiry card.

8010 Bus Compatible 64 K Byte Programmable Memory Board Saves Space



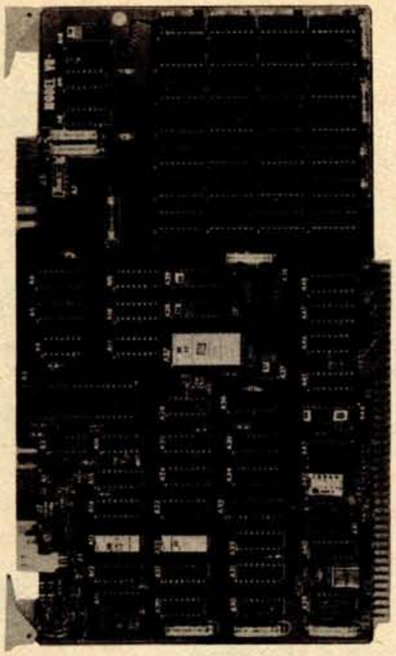
This space saving 64 K byte programmable memory board has been introduced by GSI Systems, 223 Crescent St, Waltham MA 02154. The GSI Systems 10046 64 K byte programmable memory board is a direct replacement for four Intel or National Semiconductor SBC 8010 16 K byte programmable memory boards and frees space for boards with other functions. Providing a 475 ns access time and a 650 ns refresh, the 10046 is compatible with a standard SBC 8010 backplane. Two serial IO ports capable of providing RS232C or current loop interface may be incorporated to permit the board to communicate with various peripherals, terminals and modems.

The board provides all of the necessary logic and electronics to perform accessing, reading, writing, transparent refresh, time out refresh, and direct memory access. Its logic resolves conflicts between refresh and normal or direct memory access operations so that both cannot occur simultaneously. The 10046 can be driven by any Intel or National Semiconductor 8010 processor board; 64 K byte address starts at 0000, page selectable. Voltages required are ±5 VDC and ±12 VDC. It can also be strapped for 16 K byte operation.

The system sells for $1795; the 16 K byte version sells for $495.■

Circle 648 on inquiry card.

Software Controlled Display Formats



The Datacube VR-106 video programmable memory board provides 80 by 24, 80 by 12, 40 by 24 or 40 by 12 character display formats under software control. The display consists of 128 upper case, lower case and Greek characters in a 7 by 7 or 7 by 9 font. A bipolar programmable read only memory character generator can be used for reverse video, underline, half intensity, blinking or graphics block.

Compatible with Intel Multibus SBC series computers, the VR-106 appears as memory and occupies 2048 locations in binary addressed models and 4096 in XY addressed models. A base address for the board is programmed on a 5 position dual-in-line switch. An 8 bit input port accommodates an optional keyboard, and two video output ports are provided: direct drive and composite video.

The Datacube VR-106 video programmable memory board is priced from $540. Additional information is available from Datacube/SMK-I, 670 Main St, POB 405, Reading MA 01867.■

Circle 650 on inquiry card.

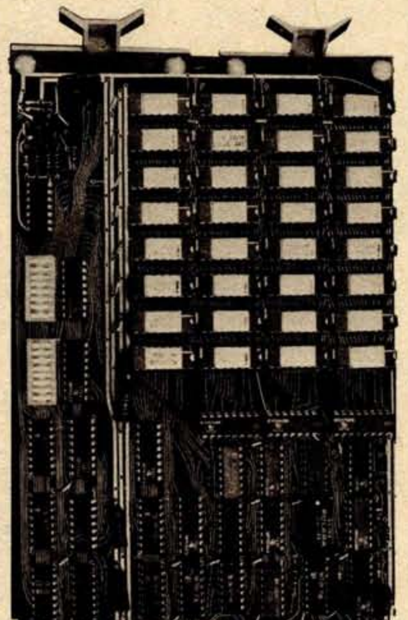
External Memory Expands PET 2001

The PEM-8K external memory stand alone unit supplies an additional 8 K bytes of external programmable memory compatible with either the 4 K byte or the 8 K byte version of the PET 2001 computer series. Connection to the computer's existing memory is made through a 3 foot interface cable and an edge connector plug that mates with the PET's memory expansion connector. A self-contained, fused and regulated power supply furnishes all necessary operating voltages. Power is controlled by a rear mounted switch.

The PEM-8K is housed in a woodgrained finished cabinet that complements the PET's appearance. As an introductory bonus, a 10 K byte software package will be supplied that allows the PET to perform financial and investment calculations, hyperbolic trigonometric functions, factorials and more. The PEM-8K is priced at $279 assembled, tested and ready for use. A special model wired for use with 200 VAC power is available on request. For more information write to International Technical Systems Inc, POB 264, Woodbridge VA 22194.■

Circle 649 on inquiry card.

64 K Bytes Memory for the Heathkit/Digital H11 Computer



The CI 1103 memory module is designed specifically for the Heathkit H11 computer and the PDP 11/03 microcomputers. The new memory features easy expansion from 8 K bytes to 32 K bytes by simply interchanging the 4027 4 K by 1 bits dynamic memory devices with their 16 K byte equivalents, with no further modification to the board. Available in 8, 16, 24 or 32 K byte versions, the unit plugs directly into the Heathkit/Digital H11, LSI-11, PDP 11/03 or LSI-11/2.

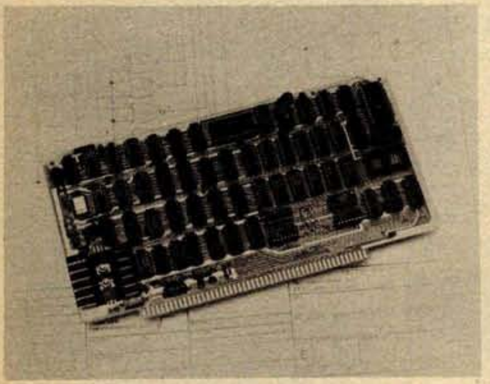
The CI 1103 is available with either on board distributed refresh or external refresh control logic. Data access time is 350 ns and cycle time is 525 ns. On board memory select is available in 2 K byte increments up to 128 K words of memory. Power consumption is under 7 W.

Single quantity price is $390 for 8 K by 16 bits and $995 for 32 K by 16 bits. For further information contact Chrislin Industries Inc, 31312 Via Colinas #102, Westlake Village CA 91361.■

Circle 651 on inquiry card.

Circle 39 on inquiry card.

64 Character by 16 Lines Video Display Board



Requiring +8 VDC at 1.2 A, the Flashwriter generates a video display of 1024 characters arranged as 16 lines of 64 characters each and uses a 7 by 9 dot matrix to produce a high quality, high resolution display image. The board also has an 8 bit parallel port with latched strobe that may be used as a keyboard port.

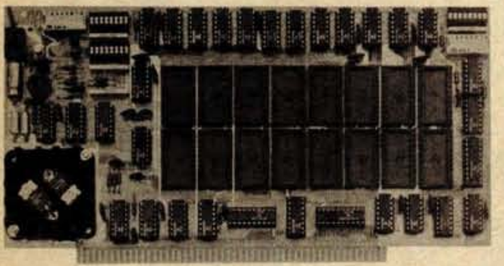
In addition to alphanumeric displays, the Flashwriter can generate character-by-character, reversed video, reduced intensity, block and line graphics. It contains its own screen refresh memory and is designed to operate with 4 MHz processor clock rates.

The Flashwriter is fully compatible with most S-100 bus microcomputers. Its video output conforms to RS-170 requirements and is available as composite video or separate video and sync.

It is priced at $195 for the kit or $235 assembled and tested. For more information, contact Vector Graphic Inc, 790 Hampshire Rd, Westlake Village CA 91361.■

Circle 529 on inquiry card.

Memory Board with 2716 Programmable Read Only Memory Programmer



Cromemco's 32 K byte Bytesaver card provides an on board 2716 programmable read only memory programmer. Information can be stored permanently by a simple, one time write of the desired data into an erased programmable read only memory with the on board programmer turned on. The card also provides a full 32 K byte capacity of nonvolatile storage for read only memory intensive applications.

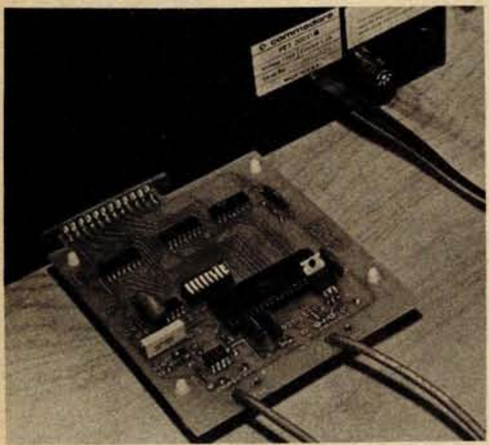
The 32 K byte Bytesaver holds up to 16 of the Intel 2716 memories or equivalent. Switches are provided to: protect and unprotect programmable read only memories individually or in groups for programming; shadow read only memory socket pairs, thus allowing external programmable memory to overlap portions of read only memory address space, select card address, and control the Bank-Select and direct memory access input and output features.

The 32 K byte Bytesaver is designed for use with the S-100 bus and is compatible with Cromemco's System Two and System Three computers.

The card is available in kit form for $195 and assembled and tested for $295. Contact Cromemco Inc, 280 Bernardo Av, Mountain View CA 94040.■

Circle 530 on inquiry card.

RS-232 Printer Adapter for the Commodore PET

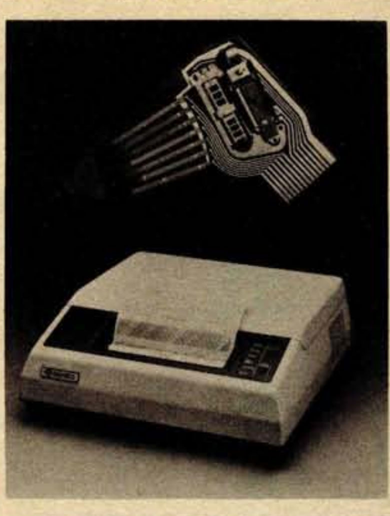


A line of peripheral adapters for the Commodore PET has been announced by Connecticut microComputer, 150 Pocono Rd, Brookfield CT 06804. The Pet ADApter Model 1200 drives an RS-232 printer from the PET IEEE-488 bus. The PET ADA 1200 allows the PET owner to obtain hard copy program listings and to type letters, manuscripts, mailing labels, tables of data, etc, using a standard RS-232 printer.

The PET ADA Model 1200 is available assembled and tested without power supply, case or RS-232 connector for $98.50 or complete for $169. Add $5 for shipping and handling. Specify data transmission rate when ordering (300 bps is supplied unless otherwise requested).■

Circle 531 on inquiry card.

New Nonimpact Ink Jet Printer



Called Quietype, this nonimpact ink jet printer prints 180 characters per second and is well suited for desktop operation in offices, libraries, hospitals and other environments where minimum noise levels are desirable.

Quietype operates bidirectionally, employing a motion minimization algorithm to maximize throughput. It prints 80 characters per line as standard format, using a 96 character ASCII set, and can also switch to 132 characters per line in compressed format.

The ink delivery system uses a pressurized disposable ink cartridge that supplies ink to the print head through a flexible tube. A simple pressure regulator in the head controls replenishment of the ink supply.

The Quietype has been designed for video hardcopy, minicomputer and microcomputor outputs, and message switching applications. Printing is done on a self-contained roll of class II Teletype paper.

The single unit price for the printer with an RS-232 interface is $2495. Disposable ink cartridges capable of printing six million characters (about 3000 pages) are priced at $17.50 each. Contact Silonics Inc, 525 Oakmead Pky, POB 9025, Sunnyvale CA 94086.■

Circle 532 on inquiry card.

Serial IO for Apple II

A serial IO board for the Apple II has been announced by Electronic Systems, POB 9641, San Jose CA 95157. The board comes with software for BASIC IO programs and monitor to Teletype or other serial device and a program for using the Apple II for a video terminal. The board has switch selectable parity, number of stop bits and jumper selectable address. Data rate goes to 30 k bps. The RS-232 IO is available as an assembled and tested unit for $62, a kit with parts for $42 or circuit board for $15. Full documentation and software are included. For information about additional kits, contact the company.■

Circle 533 on inquiry card.

Circle 100 on inquiry card.

What's New?

Hewlett-Packard Announces New Line of Calculators

A family of five new hand held calculators designed for scientific and financial applications has been introduced by the Hewlett-Packard Company. The five calculators, collectively entitled Series E, include the HP-31E, HP-32E and HP-33E scientific models and the HP-37E and HP-38E business models. These calculators replace much of the company's present low end hand held calculator product line.

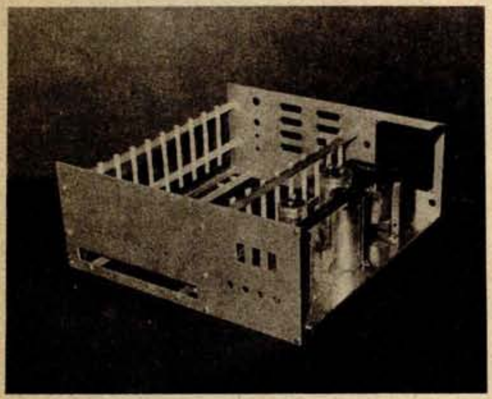
New features include a diagnostic error code system whereby the operator is informed of the error through a code number. There are as many as nine such numbers (depending on the model), each representing a particular type of operating or programming mistake. Each calculator automatically inserts commas in displayed numbers as they are needed. A low battery indicator light on the display warns the operator that the batteries are low in time to recharge.

Each Series E calculator comes with a new modular documentation system. There is an introductory manual that provides a basic outline of the capabilities common to all models in the family.

Each calculator is priced considerably less than its predecessor. The prices range from $60 to $120. Contact Inquiries Manager, Hewlett-Packard Company, 1507 Page Mill Rd, Palo Alto CA 94304.■

Circle 534 on inquiry card.

S-100 Cardframe Construction Kit

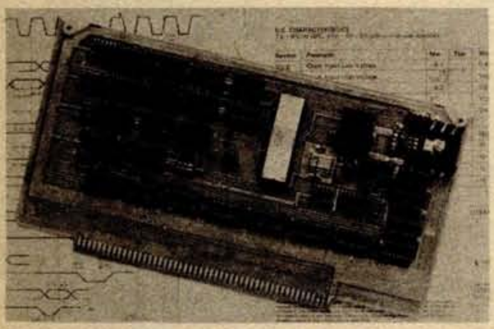


A starter set for the construction of an S-100 cardframe and power supply has been announced by Objective Design Inc, POB 20325, Tallahassee FL 32304. The kit allows the beginner or advanced computerist to construct a quality, durable cardframe with room for 22 cards. The kit contains front and rear panels specially prepunched for maximum versatility, support bars, structural bars, motherboard supports, a set of ten card guides, chassis plate and a ±16 V at 2 A power supply. Panels include cutouts for switches, power cords, connectors and motherboard extensions. No motherboard or fan is provided, although the cardframe will accept most popular versions of the S-100 motherboard. The frame is designed to allow the addition of a front panel, cover and bottom plate; these items are not supplied.

The cardframe kit is available with power supply for $154.50 and without power for $89.50 plus 5% US and Canadian shipping. Individual frame parts are available.■

Circle 535 on inquiry card.

Vector Graphic Z-80 Processor Board

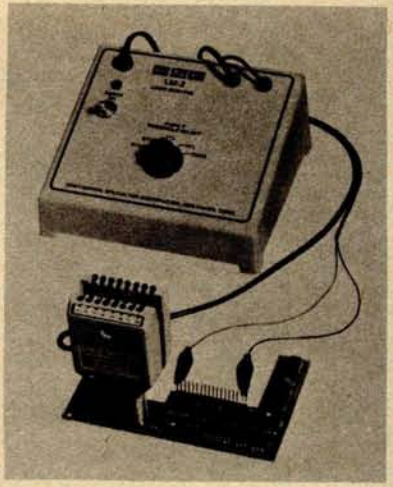


This new Z-80 processor board is offered assembled or in kit form. The board offers fully blocked design with on board wait state select, and is jumper selectable for operation at 2 MHz or 4 MHz. The board will operate standard 8080 software without modification. All Z-80 lines are fully buffered.

It is available for $175 as a kit or $215 assembled. Technical data covering the Z-80 processor board and other products may be obtained from Vector Graphic Inc, 790 Hampshire Rd, Westlake Village CA 91361.■

Circle 536 on inquiry card.

LM-2 Self-Powered Logic Monitor



The LM-2 is a self-powered version of Continental Specialties' earlier LM-1 logic monitor. A rotary switch selects the proper threshold for monitoring logic levels in RTL/DTL, TTL/HTL and CMOS circuits. The built-in power supply prevents undue circuit loading by the logic monitor itself. A separate cable for CMOS circuits uses the voltage of the circuit under test to determine the logic threshold level. It operates to a maximum useful input frequency of 30 kHz (at 50% of duty cycle). The LM-2, complete with built-in 117 VAC, 50/60 Hz power supply, is priced at $129.95. A 220 VAC, 50/60 Hz model is available at 10% more. For additional information contact Continental Specialties Corp, 70 Fulton Ter, New Haven CT 06509.■

Circle 537 on inquiry card.

Kit Introduces Students to Computers

A 7 piece Computer Project Kit (CPK) for students in elementary or secondary school and adults has been introduced by Edu-Pac Publishing Company, POB 27101-BK, Minneapolis MN 55427.

Designed as a motivational device for career education, this kit provides the students with not only a brief chronological history of the computer and answers to a list of most frequently asked questions, but also provides a hands on approach for entering data using actual data input devices.

The Computer Project Kit can be used as part of a mathematics or science course, as an introductory unit to a computer course or as a unit in itself.

Included in this color coded kit is a card decoder, prepunched with alphanumeric characters; a mark sense card on which the student can code his or her name and address; a printed Teletype tape decoder sheet; and actual samples of magnetic and prepunched Teletype 8 level tapes.

The kits cost 20¢ each in minimum quantities of 50 (shipping charges are added and billed). A single preview copy is available for 50¢.■

Circle 538 on inquiry card.

S A V E

COMPUTER INTERFACES & PERIPHERALS

S A V E

For free catalog including parts lists and schematics, send a self-addressed stamped envelope.

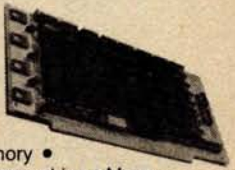
APPLE II SERIAL I/O INTERFACE *

Part no. 2
Baud rate is continuously adjustable from 0 to 30,000 • Plugs into any peripheral connector • Low current drain. RS-232 input and output • On board switch selectable 5 to 8 data bits, 1 or 2 stop bits, and parity or no parity either odd or even • Jumper selectable address • SOFTWARE • Input and Output routine from monitor or BASIC to teletype or other serial printer. • Program for using an Apple II for a video or an intelligent terminal. Also can output in correspondence code to interface with some selectrics. Board only — $15.00; with parts — $42.00; assembled and tested — $62.00.

T.V. TYPEWRITER

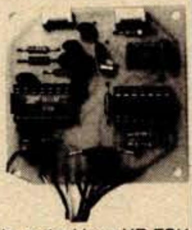
Part no. 106
• Stand alone TVT • 32 char/line, 16 lines, modifications for 64 char/line included • Parallel ASCII (TTL) input • Video output • 1K on board memory • Output for computer controlled curser • Auto scroll • Non-destructive curser • Curser inputs: up, down, left, right, home, EOL, EOS • Scroll up, down • Requires +5 volts at 1.5 amps, and -12 volts at 30 mA • All 7400, TTL chips • Char. gen. 2513 • Upper case only • Board only $39.00; with parts $145.00

8K STATIC RAM



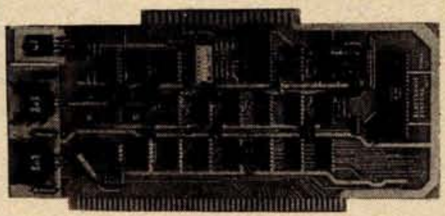
Part no. 300
• 8K Altair bus memory • Uses 2102 Static memory chips • Memory protect • Gold contacts • Wait states • On board regulator • S-100 bus compatible • Vector input option • TRI state buffered • Board only $22.50; with parts $160.00

MODEM *



Part no. 109
• Type 103 • Full or half duplex • Works up to 300 baud • Originate or Answer • No coils, only low cost components • TTL input and output-serial • Connect 8 ohm speaker and crystal mic. directly to board • Uses XR FSK demodulator • Requires +5 volts • Board $7.60; with parts $27.50

TIDMA *



Part no. 112
• Tape Interface Direct Memory Access • Record and play programs without bootstrap loader (no prom) has FSK encoder/decoder for direct connections to low cost recorder at 1200 baud rate, and direct connections for inputs and outputs to a digital recorder at any baud rate. • S-100 bus compatible • Board only $35.00; with parts $110.00

RF MODULATOR *

Part no. 107
• Converts video to AM modulated RF, Channels 2 or 3. So powerful almost no tuning is required. On board regulated power supply makes this extremely stable. Rated very highly in Doctor Dobbs' Journal. Recommended by Apple. • Power required is 12 volts AC C.T., or +5 volts DC • Board $7.60; with parts $13.50

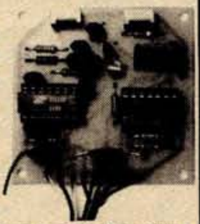
DC POWER SUPPLY *

Part no. 6085
• Board supplies a regulated +5 volts at 3 amps., +12, -12, and -5 volts at 1 amp. • Power required is 8 volts AC at 3 amps., and 24 volts AC C.T. at 1.5 amps. • Board only $12.50; with parts excluding transformers $42.50

RS 232/TTY * INTERFACE

Part no. 600
• Converts RS-232 to 20mA current loop, and 20mA current loop to RS-232 • Two separate circuits • Requires +12 and -12 volts • Board only $4.50, with parts $7.00

TAPE INTERFACE *



Part no. 111
• Play and record Kansas City Standard tapes • Converts a low cost tape recorder to a digital recorder • Works up to 1200 baud • Digital in and out are TTL-serial • Output of board connects to mic. in of recorder • Earphone of recorder connects to input on board • No coils • Requires +5 volts, low power drain • Board $7.60; with parts $27.50

UART & BAUD RATE GENERATOR *

Part no. 101
• Converts serial to parallel and parallel to serial • Low cost on board baud rate generator • Baud rates: 110, 150, 300, 600, 1200, and 2400 • Low power drain +5 volts and -12 volts required • TTL compatible • All characters contain a start bit, 5 to 8 data bits, 1 or 2 stop bits, and either odd or even parity. • All connections go to a 44 pin gold plated edge connector • Board only $12.00; with parts $35.00 with connector add $3.00

RS 232/TTL* INTERFACE

Part no. 232
• Converts TTL to RS-232, and converts RS-232 to TTL • Two separate circuits • Requires -12 and +12 volts • All connections go to a 10 pin gold plated edge connector • Board only $4.50; with parts $7.00 with connector add $2.00

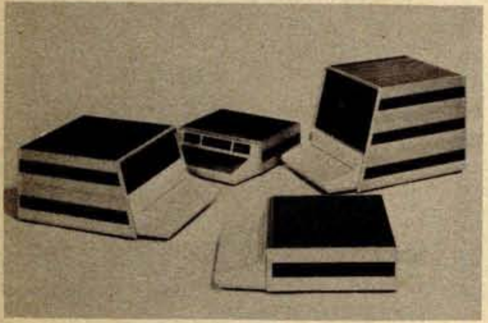
ELECTRONIC SYSTEMS Dept. B, P.O. Box 21638, San Jose, CA. USA 95151

To Order: Mention part number and description. For parts kits add "A" to part number. In USA, shipping paid for orders accompanied by check, money order, or Master Charge, BankAmericard, or VISA number, expiration date and signature. Shipping charges added to C.O.D. orders. California residents add 6.5% for tax. Outside USA add 10% for air mail postage, no C.O.D.'s. Checks and money orders must be payable in US dollars. Parts kits include sockets for all ICs, components, and circuit board. Documentation is included with all products. All items are in stock, and will be shipped the day order is received via first class mail. Prices are in US dollars. No open accounts. To eliminate tariff in Canada boxes are marked "Computer Parts." Dealer inquiries invited.
24 Hour Order Line: (408) 226-4064

✱ Circuits designed by John Bell

What's New?

Computer Related Additions to Buckeye Stamping Case Line



The Buckeye Stamping Company, 555 Marion Rd, Columbus OH 43207, has announced an expansion of its instrument case line by offering cases with keyboard holders and video display canopies.

These computer related additions, fabricated of extruded aluminum, are offered in standard 17 inch (43 cm) widths. The new cases are now offered with blue or black vinyl tops with matching feature stripes or with teakwood vinyl as an option.

Cases sell in a price range of $100 to $215 complete.■

Circle 652 on inquiry card.

Floppy Disk Preservers

Protect your flexible disks from magnetic degradation, erasure, or physical damage. This item will also protect 9 inch (22.86 cm) diameter tape reels. The case is designed for storage, shipment and hand carrying. A wide choice of models and capacities are also available for standard reels, disks, disk packs and standard cassettes. For further information contact Magnetic Shield Division, 740 N Thomas Dr, Bensenville IL 60106 and request their TP-5 catalog.■

Circle 653 on inquiry card.

New Catalog from PAIA Electronics Inc

PAIA Electronics' newest 24 page catalog lists dozens of kits for the musician or experimenter. Featured in the catalog are the most recent additions to the PAIA line: digital computer controlled electronic music synthesizers, orchestral string synthesizer, low cost video display module and single board computer. The Gnome microsynthesizer and a wide variety of special effects devices are also shown. The catalog is available without charge by writing: PAIA Electronics Inc, 1020 W Wilshire Blvd, Oklahoma City OK 73116.■

Circle 654 on inquiry card.

3M Introduces Compatible Data Cartridge with 50% Greater Capacity

An extra length (450 feet of tape) data cartridge that provides 50% more data capacity for users of the Scotch brand DC 300A data cartridge is now available from 3M Company's Data Products Mincom Division, Dept 89, 3M Company, POB 33600, St Paul MN 55133. The Scotch brand DC 300XL cartridge is suitable for backup of disk data systems and in applications where extensive logging is involved. It can be used in applications which have been using the DC 300A data cartridge. The basic price for the DC 300XL is $23.■

Circle 655 on inquiry card.

Circle 25 on inquiry card.

IN STOCK NOW

The EW-2001 A "Smart" VIDEO BOARD KIT At A "Dumb" Price!

A VIDEO BOARD + A MEMORY BOARD + AN I/O BOARD – ALL IN ONE!

- STATE OF THE ART TECHNOLOGY USING DEDICATED MICROPROCESSOR I.C.
- NUMBER OF I.C.s REDUCED BY 50% FOR HIGHER RELIABILITY
- MASTER PIECE OF ENGINEERING
- FULLY SOFTWARE CONTROLLED

Priced at ONLY **$199.95** Basic Software Included

SPECIAL FEATURES:
- S-100 bus compatible
- Parallel keyboard port
- On board 4K screen memory (optional)* relocatable to main computer memory
- Text editing capabilities (software optional)
- Scrolling: up and down through video memory
- Blinking characters
- Reversed video
- Provision for on board ROM
- CRT and video controls fully programmable (European TV)
- Programmable no. of scan lines
- Underline blinking cursor
- Cursor controls: up, down, left, right, home, carriage return
- Composite video

*Min. 2K required for operation of this board.

DISPLAY FEATURES:
- 128 displayable ASCII characters (upper and lower case alphanumeric, controls)
- 64 or 32 characters per line (jumper selectable)
- 32 or 16 lines (jumper selectable)
- Screen capacity 2048 or 512
- Character generation: 7 x 11 dot matrix

OPTIONS:

Sockets	$10.00
2K Static Memory (with Sockets)	$45.00
4K Static Memory (with Sockets)	$90.00
Complete unit, assembled and tested with 4K Memory	$335.00
Basic software on ROM	$20.00
Text editor on ROM	$75.00

DEALER INQUIRIES WELCOMED

8080 SUPPORT	
8212	$3.00
8214	7.95
8216	3.50
8228	5.95
8251	7.95
8255	8.50

8080A CPU $7.75

RAM-2114 1Kx4 450ns $8.00

GROUNDED A.C. CORDS 6 Ft. – $1.19

WIRE WRAP WIRE KYNAR 28 AWG $2.95/100 Feet Blue, Red, White

Computer Grade Capacitors 5000 mfd 60VDC $1.50

EDGE CONNECTOR WIRE WRAP PINS
44 Pin $1.25 – 72 Pin $1.75 – 100 Pin (S-100) $5.45

MISC. IC's	
DM8810	3/$1
DM8210	$2 ea.
N8T15	$1 ea.
9024	2/$1.50
93LO8	$1.50 ea.
93LO9	2/$1.50
93L24	$1 ea.

CMOS	
4011	6/$1.00
4022	2/$1.70
4023	4/$1.00
4071	5/$1.00

LINEAR I.C.'s	
LM320K-5	.99
LM320-12	.99
LM709N	11/$1
LM710H	6/$1
LM711H	6/$1
LM741M	7/$1

Transistors & Diodes	
2N3906	7/$1
2N4400	9/$1
2N4403	9/1
2N2222A	4/$1
1N4003	12/$1
1N4005	10/$1
1N4148	14/$1
SCR 400V 4A, TO220	$ .80

SHIPPING: Keyboard and Video Board: $3.50; OTHERS 125

California residents add 6% sales tax

ELECTRONICS WAREHOUSE Inc.
1603 AVIATION BLVD.
REDONDO BEACH, CA. 90278
TEL. (213) 376-8005

VISA — master charge

WRITE FOR FREE CATALOG
Minimum Order: $10

ASCII KEYBOARD KIT $74.00

Additional Improvements: Double Size Return Key
Control Characters Molderd on Key Caps

- Power: +5V 275mA
- Upper and Lower Case
- Full ASCII Set
- 7 or 8 Bits Parallel Data
- Optional Serial Output
- Selectable Positve or Negative Strobe, and Strobe Pulse Width
- 2 Key Roll-Over
- 3 User DEfineable Keys
- P.C. Board Size: 17-3/16" x 5"

OPTIONS:

Metal Enclosure Painted Blue and White	$27.50
18 Pin Edge Con.	$ 2.00
I.C. Sockets	$ 4.00
Serial Output Provision (Shift Register)	$ 2.00
Upper Case Lock Switch for Capital Letters and Nos.	$ 2.00
Assembled (on Sockets) and Tested	$90.00

APPLE II I/O BOARD KIT

Plugs Into Slot of Apple II Mother Board

18 Bit Parallel Output Port (Expandable to 3 Ports)

1 Input Port

15mA Output Current Sink or Source

Can be used for peripheral equipment such as printers, floppy discs, cassettes, paper tapes, etc.

1 free software listing for SWTP PR40 or IBM selectric

PRICE:

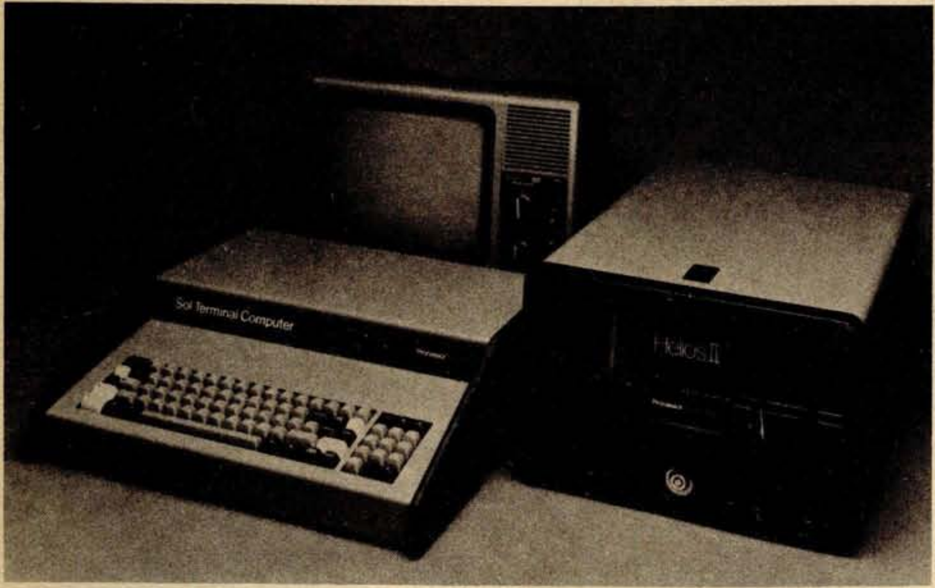
1 Input and 1 Output Port for $49.00

1 Input and 3 Output Ports for $64.00

Dealer Inquiries Invited

Circle 130 on inquiry card.

Four Disk Microsystem Introduced by Processor Technology



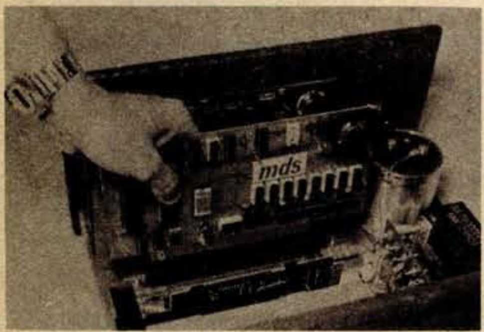
This integrated small computer system with four full-size floppy disks on line has been introduced by Processor Technology Corp, 7100 Johnson Industrial Dr, Pleasanton CA 94566. The new system, Sol System IV, includes the company's Sol-20 mainframe with 50,176 8 bit words of programmable memory, a Helios II Model 4 disk memory system, PTDOS disk operating system, Extended Disk BASIC, a video monitor and complete documentation. Total mass storage capability on four formatted disks is 1.5 million bytes.

The PTDOS disk operating system offers complex editors, assembler, device-independent files and random indexed files. The video display can be addressed randomly to any position on the screen. Extended BASIC includes string and advanced file functions, timed input, complete matrix algebra, base 10 and rational logarithms, trigonometric functions, exponential numbers and 8 digit precision.

In addition to Extended BASIC, Disk FORTRAN and Disk PILOT are available as options. The price for the Sol System IV, fully assembled and tested, is $7995.■

Circle 539 on inquiry card.

The New MD-690, an S-100 Compatible 6800



The MD-690 is an S-100 compatible processor board featuring the Motorola 6800 processor. This processor uses the MC6802 which combines the instruction set of the 6800 with internal programmable memory and crystal controlled clock.

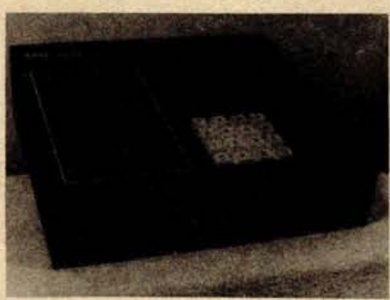
The board comes complete with Monbug, a 1 K byte monitor program which is software compatible with the standard Motorola MIKBUG monitor, although it is designed to interface with most ultrafast memory mapped video and graphics cards.

The processor card also features interrupt driven keyboard input for fast IO and an on card 2400 bps Manchester cassette interface. There are 1 K bytes of user available programmable memory on the card and provision for a second expanded monitor. A 2 MHz option is also available.

The price for the S-100 bus MC6802 processor card, complete with the 2400 bps cassette interface, 1 K byte monitor and 1152 bytes of programmable memory, is $198 in kit form. The processor is also available in a complete system, the MDS-2, which includes a case, power supply, motherboard, video graphics card and custom keyboard, priced at $579 in kit form and $798 assembled. An extensive line of software which mates the video graphics capability of the MD-690 processor is also available. Contact MDS, POB 36051, Los Angeles CA 90036.■

Circle 540 on inquiry card.

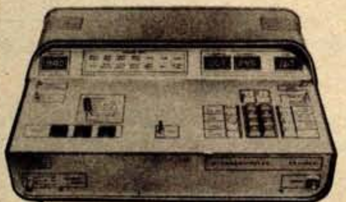
Attention Designers: A 6800 in a Box



This ready to use microcomputer features the 6800 microprocessor and is based on Motorola's MEK6800D2 evaluation kit. The Rank 68/01 is housed in a metal cabinet and provides 256 bytes of user programmable memory and 1 K bytes of read only memory containing Motorola's JBUG monitor program which permits the user to examine and alter memory locations; save and load programs to and from cassette tape; examine processor registers; insert and remove breakpoints and step one instruction at a time. The 68/01 comes complete with 6 character hexadecimal display and keyboard, cassette interface and three user manuals. The computer board has sockets for 256 bytes of extra programmable memory, two 2708 erasable read only memory parts and buffers. Rank also offers the unit with an in circuit emulator feature for testing and debugging other 6800 circuits, called the 68-01E. Prices are $359 for the 68/01 and $715 for the 68/01E from Rank Peripherals of Canada, 4998 Blvd de Maisonneuve W, Suite 1020, Montreal PQ CANADA H3Z 1N2.■

Circle 542 on inquiry card.

Microcomputer Trainer



The microcomputer trainer, Model MCT-1, is a microcomputer useful in the design of microprocessor systems. It is intended to highlight various activities of the microprocessor while executing programs. The microcomputer trainer provides the required support to learn software, debugging, computing concepts, peripheral interfacing through the sequence of events associated in program execution. This hands-on experience reinforces the basic concepts of hardware and software tradeoff.

The unit comes fully documented with illustrative user-oriented software library. It costs $720 from Allied Computers, B-58, Ashok Nagar, Madras-600 083 INDIA.■

Circle 541 on inquiry card.

7400 TTL

Part	Price	Part	Price	Part	Price
SN7400N	.16	SN7470N	.29	SN74160N	.89
SN7401N	.18	SN7472N	.29	SN74161N	.89
SN7402N	.18	SN7473N	.35	SN74162N	1.95
SN7403N	.18	SN7474N	.35	SN74163N	.89
SN7404N	.18	SN7475N	.49	SN74164N	.89
SN7405N	.20	SN7476N	.35	SN74165N	.89
SN7406N	.29	SN7479N	5.00	SN74166N	1.25
SN7407N	.29	SN7480N	.50	SN74167N	1.95
SN7408N	.20	SN7482N	.99	SN74170N	1.59
SN7409N	.20	SN7483N	.59	SN74172N	6.00
SN7410N	.18	SN7485N	.79	SN74173N	1.25
SN7411N	.25	SN7486N	.35	SN74174N	.89
SN7412N	.25	SN7489N	1.75	SN74175N	.79
SN7413N	.40	SN7490N	.45	SN74176N	.79
SN7414N	.70	SN7491N	.59	SN74177N	.79
SN7416N	.25	SN7492N	.43	SN74179N	1.95
SN7417N	.25	SN7493N	.43	SN74180N	.79
SN7420N	.20	SN7494N	.65	SN74181N	1.95
SN7421N	.29	SN7495N	.65	SN74182N	.79
SN7422N	.39	SN7496N	.65	SN74184N	1.95
SN7423N	.25	SN7497N	3.00	SN74185N	1.95
SN7425N	.29	SN74100N	.89	SN74186N	9.95
SN7426N	.29	SN74107N	.35	SN74188N	3.95
SN7427N	.25	SN74109N	.59	SN74190N	1.25
SN7429N	.39	SN74116N	1.95	SN74191N	1.25
SN7430N	.20	SN74121N	.35	SN74192N	.79
SN7432N	.25	SN74122N	.39	SN74193N	.79
SN7437N	.25	SN74123N	.49	SN74194N	.89
SN7438N	.25	SN74125N	.49	SN74195N	.69
SN7439N	.25	SN74126N	.49	SN74196N	.89
SN7440N	.20	SN74132N	.75	SN74197N	.89
SN7441N	.89	SN74136N	.75	SN74198N	1.49
SN7442N	.49	SN74141N	.79	SN74199N	1.49
SN7443N	.75	SN74142N	2.95	SN74S200	4.95
SN7444N	.75	SN74143N	2.95	SN74251N	1.79
SN7445N	.75	SN74144N	2.95	SN74279N	.79
SN7446N	.69	SN74145N	.79	SN74283N	2.25
SN7447N	.59	SN74147N	1.95	SN74284N	3.95
SN7448N	.79	SN74148N	1.29	SN74285N	3.95
SN7450N	.20	SN74150N	.89	SN74365N	.69
SN7451N	.20	SN74151N	.59	SN74366N	.69
SN7453N	.20	SN74152N	.59	SN74367N	.69
SN7454N	.20	SN74153N	.59	SN74368N	.69
SN7459A	.25	SN74154N	.99	SN74390N	1.95
SN7460N	.20	SN74155N	.79	SN74393N	1.95
		SN74156N	.79		
		SN74157N	.65		

20% Discount 100 pcs combined order 25% -1000 pcs combined order.

C/MOS

Part	Price	Part	Price	Part	Price
CD4000	.23	CD4028	.89	CD4070	.55
CD4001	.23	CD4029	1.19	CD4071	.23
CD4002	.23	CD4030	.49	CD4072	.49
CD4006	1.19	CD4035	.99	CD4076	1.39
CD4007	.25	CD4040	1.19	CD4081	.23
CD4009	.49	CD4041	1.25	CD4082	.23
CD4010	.49	CD4042	.99	CD4093	.99
CD4011	.23	CD4043	.89	CD4098	2.49
CD4012	.25	CD4044	.89	MC14409	14.95
CD4013	.39	CD4046	1.79	MC14410	14.95
CD4014	1.39	CD4047	2.50	MC14411	14.95
CD4015	1.19	CD4048	1.35	MC14419	4.95
CD4016	.49	CD4049	.69	MC14433	19.95
CD4017	1.19	CD4050	.49	MC14506	.75
CD4018	.99	CD4051	1.19	MC14507	.99
CD4019	.49	CD4053	1.19	MC14562	14.50
CD4020	1.19	CD4056	2.95	MC14583	3.50
CD4021	1.39	CD4059	9.95	CD4508	3.95
CD4022	1.19	CD4060	1.49	CD4510	1.39
CD4023	.23	CD4066	.79	CD4511	1.29
CD4024	.79	CD4068	.39	CD4515	2.95
CD4025	.23	CD4069	.45	CD4518	1.29
CD4026	2.25			CD4520	1.29
CD4027	.69			CD4566	2.25

74C00

Part	Price	Part	Price	Part	Price
74C00	.39	74C85	3.95	74C163	3.00
74C02	.55	74C90	3.00	74C164	3.25
74C04	.75	74C93	2.00	74C173	2.60
74C08	.75	74C95	2.00	74C192	3.49
74C10	.65	74C107	1.25	74C193	2.75
74C14	3.00	74C151	2.90	74C195	2.75
74C20	.65	74C154	3.00	74C922	9.95
74C30	.65	74C157	2.15	74C923	8.95
74C42	2.15	74C160	3.25	74C925	14.95
74C48	4.75	74C161	3.25	74C926	11.95
74C73	1.50			80C95	1.50
74C74	1.15			80C97	1.50

LINEAR

Part	Price	Part	Price	Part	Price
78MG	1.75	LM340T-6	1.25	LM741CN/H	.35
LM300H	.80	LM340T-8	1.25	LM741-14N	.39
LM301CN/H	.35	LM340T-12	1.25	LM747N/H	.79
LM302H	.75	LM340T-15	1.25	LM748N/H	.39
LM304H	1.00	LM340T-18	1.25	LM1303N	.90
LM305H	.60	LM340T-24	1.25	LM1304N	1.19
LM307CN/H	.35	LM350N	1.00	LM1305N	1.40
LM308CN/H	1.00	LM351CN	.65	LM1307N	.85
LM309H	1.10	LM370N	1.15	LM1310N	2.95
LM309K	1.25	LM373N	3.25	LM1351N	1.65
LM310CN	1.15	LM377N	4.00	LM1414N	1.75
LM311N/H	.90	LM380N	1.25	LM1458CN/H	.59
LM312H	2.00	LM380CN	.99	MC1488N	1.95
LM317K	6.50	LM381N	1.79	MC1489N	1.95
LM318CN/H	1.50	LM382N	1.79	LM1496N	.95
LM319N	1.30	NE501N	8.00	LM1556V	1.75
LM320K-5	1.35	NE510A	6.00	MC1741SCP	3.00
LM320K-5.2	1.35	NE529A	4.95	LM2901N	2.95
LM320K-12	1.35	NE531H/V	3.95	LM3053N	1.50
LM320K-15	1.35	NE536T	6.00	LM3065N	.69
LM320K-18	1.35	NE540L	6.00	LM3900N(3401)	.49
LM320K-24	1.35	NE544N	4.95	LM3905N	.89
LM320T-5	1.25	NE550N	1.30	LM3909N	1.25
LM320T-5.2	1.25	NE555V	.39	MC5558V	.59
LM320T-8	1.25	NE556N	.99	LM7525N	.90
LM320T-12	1.25	NE560B	5.00	LM7534N	.75
LM320T-15	1.25	NE561B	5.00	8038B	4.95
LM320T-18	1.25	NE562B	5.00	LM75450N	.50
LM320T-24	1.25	NE565N/H	1.25	75451CN	.39
LM323K-5	5.95	NE566CN	1.75	75452CN	.39
LM324N	1.80	NE567V/H	.99	75453CN	.39
LM339N	.99	NE570N	10.50	75454CN	.39
LM340K-5	1.35	LM703CN/H	.45	75491CN	.79
LM340K-6	1.35	LM709N/H	.29	75492CN	.89
LM340K-8	1.35	LM710N	.79	75494CN	.89
LM340K-12	1.35	LM711N	.39	RC4136	1.25
LM340K-15	1.35	LM723N/H	.55	RC4151	5.95
LM340K-18	1.35	LM733N	1.00	RC4194	5.95
LM340K-24	1.35	LM739N	1.19	RC4195	4.49
LM340T-5	1.25				

74LS00 TTL

Part	Price	Part	Price	Part	Price
74LS00	.23	74LS51	.23	74LS139	.69
74LS01	.23	74LS54	.23	74LS151	.69
74LS02	.23	74LS55	.23	74LS155	.69
74LS03	.23	74LS73	.35	74LS157	.69
74LS04	.29	74LS74	.35	74LS160	.89
74LS05	.29	74LS75	.49	74LS161	.89
74LS08	.23	74LS76	.35	74LS162	.89
74LS09	.29	74LS83	.75	74LS163	.89
74LS10	.23	74LS85	.99	74LS164	.99
74LS13	.49	74LS86	.35	74LS175	.79
74LS14	.99	74LS90	.49	74LS181	2.49
74LS15	.29	74LS92	.59	74LS190	.89
74LS20	.23	74LS93	.59	74LS191	.89
74LS21	.29	74LS95	.79	74LS192	.89
74LS22	.29	74LS96	.89	74LS193	.89
74LS26	.29	74LS107	.35	74LS194	.89
74LS27	.29	74LS109	.35	74LS195	.89
74LS28	.29	74LS112	.35	74LS253	.79
74LS30	.23	74LS123	.99	74LS257	.69
74LS32	.29	74LS132	.79	74LS260	.55
74LS37	.35	74LS136	.39	74LS279	.59
74LS40	.29	74LS138	.69	74LS367	.59
74LS42	.69			74LS368	.59
74LS47	.69			74LS670	1.95

BUGBOOK®

Continuing Education Series

BUGBOOKS I and II — $17.00 per set
by Peter R. Rony, David G. Larsen, WB4HYJ
Sold as a set these two books outline over 90 experiments designed to teach the reader all he will need to know about TTL logic chips to use them in conjunction with microprocessor systems. You'll learn about the basic concepts of digital electronics including gates, flip-flops, latches, buses, decoders, multiplexers, demultiplexers, LED displays, RAM's, ROM's, and much, much more.

BUGBOOK IIa — $5.00
by Peter R. Rony, David G. Larsen, WB4HYJ
This volume will introduce you to the fabulous UART chip — that all important interface between data terminals, etc., and your microcomputer. It also covers current loops, and the RS 232C interface standard. Particularly recommended for any RTTY enthusiast.

THE 555 TIMER APPLICATIONS SOURCEBOOK WITH EXPERIMENTS — $6.95
by Howard M. Berlin W3HB
This book shows you what the 555 timer is and how to use it. Included are over 100 various design techniques, equations and graphs to create "ready-to-go" timers, generators, power supplies, measurement and control circuits, party games, circuits for the home and automobile, photography, music and Amateur Radio.

BUGBOOK III — $15.00
by Peter R. Rony, David G. Larsen, WB4HYJ, Jonathan A. Titus
Here is the book that puts it all together. Besides having much valuable text there are a series of experiments in which the reader completely explores the 8080 chip pin by pin and introduces you to the Mark 80 microcomputer, a unique easily interfaced system. It is recommended that you have the background on the BUGBOOKS I & II before proceeding with BUGBOOK III.

INSTRUCTOR'S MANUAL — $3.00
Necessary for instruction of Bugbook I and II. Answers questions regarding experiments, suggestions for further reading, philosophy of authors approach to digital electronics. A must for self-teaching individuals.

BUGBOOK V and VI — $19.00 per set
by David G. Larsen, Peter R. Rony, Jonathan A. Titus
Experiments in digital electronics, 8080A microcomputer programming and 8080A microcomputer interfacing. An integrated approach to self-instructed basic digital electronics, breadboarding and 8080A interfacing/programming. Bugbook VI integrates the digital concepts of Bugbook V into a treatment of 8080A microcomputer programming and interfacing. Detail & laboratory experiments included with each book.

OP AMP MANUAL by Howard M. Berlin W3HB — $9.00
An experiment guide to application of operational amplifiers. Over 25 experiments on all phases of Op Amps.

DBUG — $5.00
8080 interpretive debugger. A program for entering, debugging and storing assembly language programs.

CMOS-M — DESIGNERS PRIMER AND HANDBOOK New expanded version — $8.50
Starts at basic structure of CMOS devices through integration into MSI.

COMPLETE MANUAL FOR DIGITAL CLOCKS by John Weiss and John Brooks — $3.95
Familiarizes technician or hobbyist with basic theories behind digital clocks. Includes trouble shooting guides, basic characteristics of clocks, soldering techniques, clock component data sheets and construction tips.

DISCRETE LEDS

.125" dia.			.185" dia.			.200" dia.			.190" dia.		
XC209	Red	5/$1	XC526	Red	5/$1	XC556	Red	5/$1	XC111	Red	5/$1
XC209	Green	4/$1	XC526	Red	100/$8	XC556	Red	100/$8	XC111	Green	4/$1
XC209	Yellow	4/$1	XC526	Green	4/$1	XC556	Green	4/$1	XC111	Yellow	4/$1
			XC526	Yellow	4/$1	XC556	Yellow	4/$1	XC111	Clear	4/$1
			XC526	Clear	4/$1	XC556	Clear	7/$1			

.200" dia: XC22 Red 5/$1; XC22 Green 4/$1; XC22 Yellow 4/$1

.085" dia.: MV50 - Red - 6 $1
.170" dia.: MV10 Red - 4/$1
INFRA-RED LED ¼" x ¼" x 1 16 Flat 5/$1.00

DISPLAY LEDS

TYPE	POLARITY	HT	PRICE	TYPE	POLARITY	HT	PRICE
MAN 1	Common Anode-red	.270	2.95	MAN 6730	Common Anode-red ± 1	.560	.99
MAN 2	5 x 7 Dot Matrix-red	.300	4.95	MAN 6740	Common Cathode-red-D.D.	.560	.99
MAN 3	Common Cathode-red	.125	.25	MAN 6750	Common Cathode-red ± 1	.560	.99
MAN 4	Common Cathode-red	.187	1.95	MAN 6760	Common Anode-red	.560	.99
MAN 52	Common Anode-green	.300	1.25	MAN 6780	Common Cathode-red	.560	.99
MAN 72	Common Anode-red	.300	.99	DL701	Common Anode-red ± 1	.300	.99
MAN 74	Common Cathode-red	.300	1.25	DL704	Common Cathode-red	.300	.99
MAN 82	Common Anode-yellow	.300	.99	DL707	Common Anode-red	.300	.99
MAN 84	Common Cathode-yellow	.300	.99	DL741	Common Anode-red	.600	1.25
MAN 3620	Common Anode-orange	.300	.99	DL746	Common Anode-red ± 1	.630	1.49
MAN 3630	Common Anode-orange ± 1	.300	.99	DL747	Common Anode-red	.600	1.49
MAN 3640	Common Cathode-orange	.300	.99	DL749	Common Cathode-red ± 1	.630	1.49
MAN 4610	Common Anode-orange	.300	.99	DL750	Common Cathode-red	.600	1.49
MAN 4640	Common Cathode-orange	.400	.99	DL338	Common Cathode-red	.110	.35
MAN 4710	Common Anode-red ± 1	.400	.99	FND70	Common Cathode	.250	.69
MAN 4730	Common Anode-red	.400	.99	FND359	Common Anode	.350	.75
MAN 4740	Common Cathode-red	.400	.99	FND503	Common Cathode (FND500)	.500	.99
MAN 4810	Common Anode-yellow	.400	.99	FND507	Common Anode (FND510)	.500	.99
MAN 6610	Common Anode-orange-D.D.	.560	.99	5082-7730	Common Anode-red	.300	1.30
MAN 6630	Common Anode-orange	.560	.99	HDSP-3400	Common Anode red	.800	2.10
MAN 6640	Common Cathode-orange-D.D.	.560	.99	ADSP-3403	Common Cathode red	.800	2.10
MAN 6650	Common Cathode-orange ± 1	.560	.99	5082-7300	4 x 7 Sgl. Digit-RHDP	.600	19.95
MAN 6660	Common Anode-orange	.560	.99	5082-7302	4 x 7 Sgl. Digit-LHDP	.600	19.95
MAN 6680	Common Cathode-orange	.560	.99	5082-7304	Overrange character (±1)	.600	15.00
MAN 6710	Common Anode-red-D.D.	.560	.99	5082-7340	4 x 7 Sgl. Digit-Hexadecimal	.600	22.50

RCA LINEAR

Part	Price	Part	Price
CA3013	2.15	CA3082	2.00
CA2023	2.56	CA3083	1.60
CA3035	2.48	CA3086	.85
CA3039	1.35	CA3089	3.75
CA3046	1.30	CA3130	1.39
CA3059	3.25	CA3140	1.25
CA3060	3.25	CA3160	1.25
CA3080	.85	CA3401	.49
CA3081	2.00	CA3600	3.50

CALCULATOR CHIPS AND DRIVER

Part	Price
FCM3817 (1998A)	$ 5.00
MM5725	2.95
MM5738	2.95
DM8864	2.00
DM8865	1.00
DM8887	.75
DM8889	.75

CLOCK CHIPS

Part	Price
MM5309	$ 9.95
MM5311	4.95
MM5312	4.95
MM5314	4.95
MM5316	6.95
MM5318	9.95
MM5369	2.95
MM5387	4.95
MM5841	9.95

9374 - 7-segment LED driver common anode LEDs $ .99

IC SOLDERTAIL — LOW PROFILE (TIN) SOCKETS

	1-24	25-49	50-100		1-24	25-49	50-100
8 pin LP	$.17	.16	.15	22 pin LP	$ .37	.36	.35
14 pin LP	.20	.19	.18	24 pin LP	.38	.37	.36
16 pin LP	.22	.21	.20	28 pin LP	.45	.44	.43
18 pin LP	.29	.28	.27	36 pin LP	.60	.59	.58
20 pin LP	.34	.32	.30	40 pin LP	.63	.62	.61

SOLDERTAIL STANDARD (TIN)

	1-24	25-49	50-100		1-24	25-49	50-100
14 pin ST	$.27	.25	.24	28 pin ST	$ .99	.90	.81
16 pin ST	.30	.27	.25	36 pin ST	1.39	1.26	1.15
18 pin ST	.35	.32	.30	40 pin ST	1.59	1.45	1.30
24 pin ST	.49	.45	.42				

SOLDERTAIL STANDARD (GOLD)

	1-24	25-49	50-100		1-24	25-49	50-100
8 pin SG	$.30	.27	.24	24 pin SG	$ .70	.63	.57
14 pin SG	.35	.32	.29	28 pin SG	1.10	1.00	.90
16 pin SG	.38	.35	.32	36 pin SG	1.75	1.40	1.26
18 pin SG	.52	.47	.43	40 pin SG	1.75	1.59	1.45

WIRE WRAP SOCKETS (GOLD) LEVEL #3

	1-24	25-49	50-100		1-24	25-49	50-100
8 pin WW	$.40	.38	.35	22 pin WW	$ .95	.85	.75
10 pin WW	.45	.41	.37	24 pin WW	1.05	.95	.85
14 pin WW	.39	.38	.37	28 pin WW	1.40	1.25	1.10
16 pin WW	.43	.42	.41	36 pin WW	1.59	1.45	1.30
18 pin WW	.75	.68	.62	40 pin WW	1.75	1.55	1.40

50 PCS. RESISTOR ASSORTMENTS $1.75 PER ASST.

ASST. 1	5 ea.	10 OHM / 27 OHM	12 OHM / 33 OHM	15 OHM / 39 OHM	18 OHM / 47 OHM	22 OHM / 56 OHM	1/4 WATT 5%	50 PCS.
ASST. 2	5 ea.	68 OHM / 180 OHM	82 OHM / 220 OHM	100 OHM / 270 OHM	120 OHM / 330 OHM	150 OHM / 390 OHM	1/4 WATT 5%	50 PCS.
ASST. 3	5 ea.	470 OHM / 1.2K	560 OHM / 1.5K	680 OHM / 1.8K	820 OHM / 2.2K	1K / 2.7K	1/4 WATT 5%	50 PCS.
ASST. 4	5 ea.	3.3K / 8.2K	3.9K / 10K	4.7K / 12K	5.6K / 15K	6.8K / 18K	1/4 WATT 5%	50 PCS.
ASST. 5	5 ea.	22K / 56K	27K / 68K	33K / 82K	39K / 100K	47K / 120K	1/4 WATT 5%	50 PCS.
ASST. 6	5 ea.	150K / 390K	180K / 470K	220K / 560K	270K / 680K	330K / 820K	1/4 WATT 5%	50 PCS.
ASST. 7	5 ea.	1M / 2.7M	1.2M / 3.3M	1.5M / 3.9M	1.8M / 4.7M	2.2M / 5.6M	1/4 WATT 5%	50 PCS.

ASST. 8R Includes Resistor Assortments 1-7 (350 PCS.) **$9.95 ea.**

$5.00 Minimum Order – U.S. Funds Only
California Residents – Add 6% Sales Tax

Spec Sheets – 25¢
1979 A Catalog Available—Send 41¢ stamp

NEW 1979 Catalog

Jameco ELECTRONICS
a division of JAMES ELECTRONICS of California

PHONE ORDERS WELCOME (415) 592-8097

MAIL ORDER ELECTRONICS – WORLDWIDE
1021 HOWARD AVENUE, SAN CARLOS, CA 94070
Advertised Prices Good Thru November

WIRE-WRAP KIT — WK-2-W

WRAP • STRIP • UNWRAP

- Tool for 30 AWG Wire
- Roll of 50 Ft. White or Blue 30 AWG Wire
- 50 pcs. each 1", 2", 3" & 4" lengths — pre-stripped wire.

$12.95

WIRE WRAP TOOL WSU-30

WRAP • STRIP • UNWRAP -$6.95

WIRE WRAP WIRE — 30 AWG

25 ft. min. $1.25 50 ft. $1.95 100 ft. $2.95 1000 ft. $15.00
SPECIFY COLOR — White - Yellow - Red - Green - Blue - Black

WIRE DISPENSER — WD-30

- 50 ft. roll 30 AWG KYNAR wire wrap wire **$3.95 ea.**
- Cuts wire to desired length
- Strips 1" of insulation **Specify** — Blue-Yellow-White-Red

REPLACEMENT DISPENSER SPOOLS FOR WD 30
Specify blue, yellow, white or red **$1.98/spool**

EXAR

JE2206A $14.95 Function Generator Kit (includes chip, P.C. Board and instructions)

JE2206B $19.95 Function Generator Kit (includes all components, P.C. Board and instructions)

XR-L555 $1.50
Micro-Power version of the popular 555 Timer and directly interchangeable. Dissipates 1/15th the power and operates down to 2.7 volts. Perfect for battery operation and CMOS circuits.

XR2242CP $1.50
Precision timing circuit for generating timing pulses in **minutes, hours** and days or up to 1 year by using two. Reduces cost of time delay circuits. Basic 555 Timer with built-in 8-bit Counter.

Part	Price	Part	Price	Part	Price
XR205	$ 8.40			XR2556	$ 3.20
XR210	4.40	XR1489	1.39	XR2567	2.99
XR215	4.40	XR1800	3.20	XR3403	1.25
XR320	1.55	XR2206	4.40	XR4136	1.25
XR555	.39	XR2207	3.85	XR4151	2.85
XR556	.99	XR2208	5.20	XR4194	1.45
XR567CP	.99	XR2209	1.75	XR4202	3.60
XR567CT	1.25	XR2211	5.25	XR4212	2.05
XR1310P	1.30	XR2212	4.35	XR4558	.75
XR1468CN	3.85	XR2240	3.45	XR4739	1.15
XR1488	1.39	XR2264	4.25	XR4741	1.47

ZENERS — DIODES — RECTIFIERS

TYPE	VOLTS	W	PRICE	TYPE	VOLTS	W	PRICE
1N746	3.3	400m	4/1.00	1N4005	600 PIV	1 AMP	10/1.00
1N751A	5.1	400m	4/1.00	1N4006	800 PIV	1 AMP	10/1.00
1N752	5.6	400m	4/1.00	1N4007	1000 PIV	1 AMP	10/1.00
1N753	6.2	400m	4/1.00	1N3600	50	200m	6/1.00
1N754	6.8	400m	4/1.00	1N4148	75	10m	15/1.00
1N959	8.2	400m	4/1.00	1N4154	35	10m	12/1.00
1N965B	15	400m	4/1.00	1N4305	75	25m	20/1.00
1N5232	5.6	500m	.28	1N4734	5.6	1w	.28
1N5234	6.2	500m	.28	1N4735	6.2	1w	.28
1N5235	6.8	500m	.28	1N4736	6.8	1w	.28
1N5236	7.5	500m	.28	1N4738	8.2	1w	.28
1N456	25	40m	6/1.00	1N4742	12	1w	.28
1N458	150	7m	6/1.00	1N4744	15	1w	.28
1N485A	180	10m	5/1.00	1N1183	50 PIV	35 AMP	1.60
1N4001	50 PIV	1 AMP	12/1.00	1N1184	100 PIV	35 AMP	1.70
1N4002	100 PIV	1 AMP	12/1.00	1N1185	150 PIV	35 AMP	1.70
1N4003	200 PIV	1 AMP	12/1.00	1N1186	200 PIV	35 AMP	1.80
1N4004	400 PIV	1 AMP	12/1.00	1N1188	400 PIV	35 AMP	3.00

SCR AND FW BRIDGE RECTIFIERS

C36D	15A @ 400V	SCR(2N1849)	$1.95
C38M	35A @ 600V	SCR	1.95
2N2328	1.6A @ 300V	SCR	.50
MDA 980-1	12A @ 50V	FW BRIDGE REC	1.95
MDA 980-3	12A @ 200V	FW BRIDGE REC	1.95

TRANSISTORS

Part	Price	Part	Price	Part	Price
C106B1	.50			2N3904	4/1.00
MPSA05	.30	2N3055	.89	2N3905	4/1.00
MPSA06	5/1.00	MJE3055	1.00	2N3906	4/1.00
TIS97	6/1.00	2N3392	5/1.00	2N4013	3/1.00
TIS98	6/1.00	2N3398	5/1.00	2N4123	6/1.00
TIS133	5/1.00	PN3567	3/1.00	PN4249	4/1.00
TIS135	5/1.00	PN3568	4/1.00	PN4250	4/1.00
40409	1.75	PN3569	4/1.00	2N4400	4/1.00
40410	1.75	MPS3638A	5/1.00	2N4401	4/1.00
40673	1.75	MPS3702	5/1.00	2N4402	4/1.00
2N918	4/1.00	2N3704	5/1.00	2N4403	4/1.00
2N2219A	3/1.00	MPS3704	5/1.00	2N4409	5/1.00
2N2221A	4/1.00	2N3705	5/1.00	2N5086	4/1.00
2N2222A	5/1.00	MPS3705	5/1.00	2N5087	4/1.00
2N2369	5/1.00	2N3706	5/1.00	2N5088	4/1.00
2N2369A	4/1.00	MPS3706	5/1.00	2N5089	4/1.00
MPS2369	5/1.00	2N3707	5/1.00	2N5129	5/1.00
2N2484	4/1.00	2N3711	5/1.00	PN5134	5/1.00
2N2906	4/1.00	2N3724A	.65	PN5138	5/1.00
2N2907	5/1.00	2N3725A	1.00	2N5139	5/1.00
2N2925	5/1.00	2N3772	2.25	2N5210	5/1.00
MJE2955	1.25	2N3823	1.00	2N5449	3/1.00
2N3053	2/1.00	2N3903	4/1.00	2N5951	3/1.00

CAPACITOR CORNER

50 VOLT CERAMIC DISC CAPACITORS

	1-9	10-99	100		1-9	10-99	100
10 pf	.05	.04	.03	.001μF	.05	.04	.035
22 pf	.05	.04	.03	.0047μF	.05	.04	.035
47 pf	.05	.04	.03	.01μF	.05	.04	.035
100 pf	.05	.04	.03	.022μF	.06	.05	.04
220 pf	.05	.04	.03	.047μF	.06	.05	.04
470 pf	.05	.04	.035	.1μF	.12	.09	.075

100 VOLT MYLAR FILM CAPACITORS

	1-9	10-99	100		1-9	10-99	100
.001mf	.12	.10	.07	.022mf	.13	.11	.08
.0022	.12	.10	.07	.047mf	.21	.17	.13
.0047mf	.12	.10	.07	.1mf	.27	.23	.17
.01mf	.12	.10	.07	.22mf	.33	.27	.22

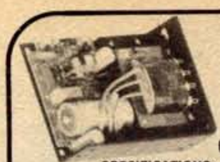
+20% DIPPED TANTALUMS (SOLID) CAPACITORS

	1-9	10-99	100		1-9	10-99	100
.1/35V	.28	.23	.17	1.5/35V	.30	.26	.21
.15/35V	.28	.23	.17	2.2/25V	.31	.27	.22
.22/35V	.28	.23	.17	3.3/25V	.31	.27	.22
.33/35V	.28	.23	.17	4.7/25V	.32	.28	.23
.47/35V	.28	.23	.17	6.8/25V	.36	.31	.25
.68/35V	.28	.23	.17	10/25V	.40	.35	.29
1.0/35V	.28	.23	.17	15/25V	.63	.50	.40

MINIATURE ALUMINUM ELECTROLYTIC CAPACITORS

Axial Lead	1-9	10-99	100	Radial Lead	1-9	10-99	100
.47/50V	.15	.13	.10	.47/25V	.15	.13	.10
1.0/50V	.16	.14	.11	.47/50V	.16	.14	.11
3.3/50V	.14	.12	.09	1.0/16V	.15	.13	.10
4.7/25V	.15	.13	.10	1.0/25V	.16	.14	.11
10/25V	.15	.13	.10	1.0/50V	.16	.14	.11
10/50V	.16	.14	.12	4.7/16V	.15	.13	.10
22/25V	.17	.15	.12	4.7/25V	.15	.13	.10
22/50V	.24	.20	.18	4.7/50V	.16	.14	.11
47/25V	.19	.17	.15	10/16V	.14	.12	.09
47/50V	.25	.21	.19	10/25V	.15	.13	.10
100/25V	.24	.20	.18	10/50V	.16	.14	.12
100/50V	.35	.30	.28	47/50V	.24	.21	.19
220/25V	.32	.28	.25	100/16V	.19	.15	.14
220/50V	.45	.41	.38	100/25V	.24	.20	.18
470/25V	.33	.29	.27	100/50V	.35	.30	.28
1000/16V	.55	.50	.45	220/16V	.23	.17	.16
2200/16V	.70	.62	.55	470/25V	.31	.28	.26

Circle 200 on inquiry card.



ELPAC POWER SUPPLIES

Completely Assembled

SPECIFICATIONS:
105-125/210-250 Vac, 47-440 Hz input:

Line Regulation	±0.1%
Load Regulation	±0.1%no-load to rated-load
Output Ripple and Noise	±0.1%p-p,dc to 10 MHz
Input/Output Isolation	100 megohm dc, 900 Vac
Short Circuit Current	35% rated current

PART NO.	RATINGS WATTS	VOLTS	AMPS	PRICE
SOLV15-5*	15	5	3	$39.95
SOLV15-12*	18	12	1.5	39.95
SOLV30-5	30	5	6	59.95
SOLV30-12	48	12	4	59.95
OVP1	over voltage protection for SOLV30-5,-12			9.95

*SOLV15-5, 12 includes OVP installed

SUP 'R' MOD II

UHF Channel 33 TV Interface Unit Kit

- Wide Band B/W or Color System
- Converts TV to Video Display for home computers, CCTV camera, Apple II, works with Cromeco Dazzler, SOL-20, IRS-80, Challenger, etc.
- MOD II is pretuned to Channel 33 (UHF).
- Includes coaxial cable and antenna transformer.

MOD II $29.95 Kit

CRYSTALS

THESE FREQUENCIES ONLY

PART NO.	FREQUENCY	CASE	PRICE
CY1A	1.000MHz	HC33	5.95
CY1.84	1.8432MHz	HC33	5.95
CY2A	2.000MHz	HC33	5.95
CY2.01	2.010MHz	HC33	1.95
CY2.50	2.500MHz	HC33	4.95
CY3.27	3.2768MHz	HC33	4.95
CY3.57	3.579545MHz	HC33	4.95
CY3A	4.000MHz	HC18	4.95
CY4.91	4.916MHz	HC18	4.95
CY7A	5.000MHz	HC18	4.95
CY5.18	5.185MHz	HC18	4.95
CY6.14	6.144MHz	HC18	4.95
CY6.40	6.400MHz	HC18	4.95
CY6.55	6.5536MHz	HC18	4.95
CY12A	10.000MHz	HC18	4.95
CY14A	14.31818MHz	HC18	4.95
CY19A	18.000MHz	HC18	4.95
CY18.43	18.432MHz	HC18	4.95
CY22A	20.000MHz	HC18	4.95
CY30A	32.000MHz	HC18	4.95

Custom Cables & Jumpers

NEW NEW

DB 25 Series Cables

Part No.	Cable Length	Connectors	Price
DB25P-4-P	4 Ft.	2-DP25P	$15.95 ea.
DB25P-4-S	4 Ft.	1-DP25P/1-25S	$16.95 ea.
DB25S-4-S	4 ft.	2-DP25S	$17.95 ea.

Dipped Jumpers Plugs

DJ14-1	1 ft.	1-14 Pin	$1.59 ea.
DJ16-1	1 ft.	1-16 Pin	1.79 ea.
DJ24-1	1 ft.	1-24 Pin	2.79 ea.
DJ14-1-14	1 ft.	2-14 Pin	2.79 ea.
DJ16-1-16	1 ft.	2-16 Pin	3.19 ea.
DJ24-1-16	1 ft.	2-24 Pin	4.95 ea.

For Custom Cables & Jumpers, See JAMECO 1979 Catalog for Pricing

1/16 VECTOR BOARD

	0.1" Hole Spacing Part No.	P-Pattern L	W	Price 1-9	10 up
PHENOLIC	64P44 062XXXP	4.50	6.50	1.72	1.54
	169P44 062XXXP	4.50	17.00	3.69	3.32
EPOXY	64P44 062WE	4.50	6.50	2.07	1.86
GLASS	84P44 062WE	4.50	8.50	2.56	2.31
	169P44 062WE	4.50	17.00	5.04	4.53
	169P84 062WE	8.50	17.00	9.23	8.26
EPOXY GLASS COPPER CLAD	169P44 062WEC1	4.50	17.00	6.80	6.12

CONNECTORS

25 Pin-D Subminiature

DB25P(as pictured)	PLUG	$2.95
DB25S	SOCKET	3.50
DB51226-1	Cable Cover for DB25 P or S	1.75

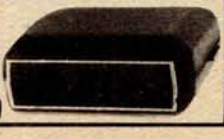
MOLEX CONNECTOR PINS

M-530-1 $1.95/100 pins (minimum order) $16.00/1000 pins

Pre-packaged in strips

INSTRUMENT/ CLOCK CASE

Injection molded unit.
Complete with red bezel.
4½" x 4" x 1-9/16"



$3.49

MICROPROCESSOR COMPONENTS

CDP 1802	CPU	$19.95
P8085	CPU	19.95
8080A	CPU	9.95
8212	8-Bit Input/Output	3.25
8214	Priority Interrupt Control	5.95
8216	Bi-Directional Bus Driver	3.49
8224	Clock Generator/Driver	3.95
8228	System Controller/BusDriver	5.95
8251	Prog. Comm. Interface	7.95
8253	Prog. Interval Timer	14.95
8255	Prog. Periph. Interface	9.95
8257	Prog. DMA Control	19.95
Z80(780C)	CPU	$19.95
Z80A(708-1)	CPU	24.95
2650	MPU	26.50
MC6800	MPU	14.95
MC6810API	128 x 8 Static Ram	5.95
MC6821	Periph. Interface Adapter(6820)	7.49
MC6830L8	1024 x 8 Bit ROM(68A30)	14.95
MC6850	Asychronous Comm Adapter	7.95
MC6852	Syn. Serial Data Adapter	9.95
MC6880	(MC8T26) Quad Tri State Bus Trans.	2.25
MC6802	MPU w/Clock & Ram	24.95

RAM'S

1101	256 x 1	Static	$1.49
1103	1024 x 1	Dynamic	.99
2101(8101)	256 x 4	Static	3.95
2102	1024 x 1	Static	1.75
2107/5280	4096 x 1	Dynamic	4.95
2111(8111)	256 x 4	Static	3.95
2112	256 x 4	Static	4.95
2114	1024 x 4	Static 450ns	9.95
2114L	1024 x 4	Static 450ns Low Power	10.95
2114-3	1024 x 4	Static 300ns	10.95
2114L-3	1024 x 4	Static 300ns Low Power	11.95
7489	16 x 4	Static	1.75
8599	16 x 4	Static	3.49
21L02	1024 x 1	Static	1.95
74S200	256 x 1	Static	4.95
93421	256 x 1	Static	2.95
MM5262	2K x 1	Dynamic	3/1.00
MK4027 (UPD414)	4K	DYNAMIC 16 PIN	4.95
MK4116 (UPD416)	16K	DYNAMIC 16 PIN	14.95
TMS4044-45NL	4K	STATIC	14.95

ROM'S

2513(2140)	Character Generator (upper case)	$9.95
2513(3021)	Character Generator (lower case)	9.95
2516	Character Generator	10.95
MM5230N	2048 Bit Read Only Memory	1.95

USER MANUALS

1802M	CDP1802 Manual	$7.50
Z80M	Z80 Manual	7.50
2650M	2650 Manual	5.00

PROMS

1702A	2048 x 1	Famous	$5.95
5203	2048 x 1	Famous	14.95
82S23	32 x 8	Open C	3.95
82S115	4096 x 1	Bipolar	19.95
82S123	32 x 8	Tristate	3.95
74S287	1024 x 1	Static	2.95
TMS2532	32K	EPROM	99.95
2708	8K	EPROM	10.95
2716 T.I.	16K	EPROM	29.95
2716 Intel (2516 T.I.)	16K	EPROM	49.95
6301-1	1024 x 1	Tri-State Bipolar	3.49
6330-1	256 x 1	Open C Bipolar	2.95

SHIFT REGISTERS

MM5016H	500/512 Bit Dynamic	.89
2504T	1024 Dynamic	3.95
2518	Hex 32 Bit Static	4.95
2522	Dual 132 Bit Static	2.95
2524	512 Dynamic	.99
2525	1024 Dynamic	2.95
2527	Dual 256 Bit Static	2.95
2528	Dual 250 Static	4.00
2529	Dual 240 Bit Static	4.00
2532	Quad 80 Bit Static	2.95
2533	1024 Static	2.95
3341	Fifo	6.95
74LS670	4 x 4 Register	1.95

UART'S

AY-5-1013	30K BAUD	$5.95

SPECIAL REQUESTED ITEMS

TELEPHONE KEYBOARD CHIPS		ICM CHIPS		NMOS READ ONLY MEMORIES		MISCELLANEOUS			
AY-5-9100	$14.95	ICM7045	$24.95	MCM6571	$13.50	11C90	$19.95	MK40240	$17.50
AY-5-9200	14.95	ICM7205	19.95	MCM6574	13.50	MC3061P	11.95	DS0026CH	3.75
AY-5-9500	4.95	ICM7207	7.50	MCM6575	13.50	MC1408L7	4.95	TIL308	10.20
AY-5-2376	14.95	ICM7208	19.95			MC1408L8	5.75	95H90	11.95
HD0165	7.95	ICM7209	6.95			LD110/111	$25.00/set		
74C922	9.95					MC4016(74416)	7.50		
						4N33	3.95		

TV GAME CHIP SET
AY-3-8500-1 Chip and 2.010 MH2 Crystal **$7.95**

The Sinclair PDM35. A personal digital multimeter for only $59.95

Now everyone can afford to own a digital multimeter

A digital multimeter used to mean an expensive, bulky piece of equipment.

The Sinclair PDM35 changes that. It's got all the functions and features you want in a digital multimeter, yet they're neatly packaged in a rugged but light pocket-size case, ready to go anywhere.

The Sinclair PDM35 is supplied completely assembled with test leads and prods, protective wallet and Operator's Manual.

The Sinclair PDM35 is tailor-made for anyone who needs to make rapid measurements. Development engineers, field service engineers, lab technicians, computer specialists, radio and electronic hobbyists will find it ideal.

Technical specification

DC Volts (4 ranges) Range: 1 mV to 1000 V. Accuracy of reading 1.0% ± 1 count. Note: 10 MΩ input impedance.

AC Volts (40 Hz-5 kHz) Range: 1 V to 500 V. Accuracy of reading 1.0% ± 2 counts.

DC Current (6 ranges) Range: 1 nA to 200 mA. Accuracy of reading 1.0% ± 1 count. Note: Max resolution 0.1 nA.

Resistance (5 ranges) Range: 1Ω to 20 MΩ. Accuracy of reading 1.5% ± 1 count. Also provides 5 junction-test ranges.

Dimensions: 6 in x 3 in x 1½ in.

Weight: 6½ oz.

Power supply: 9 V battery or Sinclair AC adapter (battery not incl.)

Sockets: Standard 4 mm for resilient plugs.

Options: AC adapter for 117 V 60 Hz power. De-luxe padded carrying wallet.

PART NO.	DESCRIPTION	PRICE
PDM35	Digital Multimeter (Completely Assembled)	$59.95
PDM-AC	II7volt AC Adapter	6.95
PDM-DP	Deluxe padded carrying case	6.95

B&K PRECISION

3½-Digit Portable DMM

- Overload Protected
- .3" high LED Display
- Battery or AC operation
- Auto Zeroing
- 1mv, 1Va, 0.1 ohm resolution
- Overange reading
- 10 meg input impendence
- DC Accuracy 1% typical

Ranges: DC Voltage - 0-1000V
AC Voltage: 0-1000V
Freq. Response: 50-400 HZ
DC/AC Current: 0-100mA
Resistance: 0-10 meg ohm
Size: 6.4" x 4.4" x 2"

Model 2800 $99.95

Comes with test leads, operating manual and spare fuse.

Accessories:

AC Adapter BC-28	**$9.00**
Rechargeable Batteries BP-26	**20.00**
Carrying Case LC-28	**7.50**

CSC

100 MHz 8-Digit Counter

- 20 Hz-100 MHz Range
- .6" LED Display
- Crystal-controlled timebase
- Fully Automatic
- Portable — completely self-contained
- Size — 1.75" x 7.38" x 5.63"
- Four power souces, i.e. batteries, 110 or 220V with charger 12V with auto lighter adapter and external 7.2-10V power supply

MAX-100 $134.95

ACCESSORIES FOR MAX 100:

Mobile Charger Eliminator use power from car battery — **Model 100 — CLA $3.95**

Charger/Eliminator use 110 V AC — **Model 100 — CAI $9.95**

KEYBOARDS

63-Key Unencoded

This is a 63-key, terminal keyboard newly manufactured by a large computer manufacturer. It is unencoded with SPST keys, unattached to any kind of PC board. A very solid molded plastic 13 x 4" base suits most application. IN STOCK **$29.95/each**

Hexadecimal Key Pad Unencoded

19-key pad includes 1-10 keys, ABCDEF and 2 optional keys and a shift key. **$10.95/each**

The Incredible "Pennywhistle 103"

$139.95 Kit Only

The **Pennywhistle 103** is capable of recording data to and from audio tape without critical speed requirements for the recorder and it is able to communicate directly with another modem and terminal for telephone "hamming" and communications. In addition, it is free of critical adjustments and is built with non-precision, readily available parts.

Data Transmission Method	Frequency-Shift Keying, full-duplex (half-duplex selectable).
Maximum Data Rate	300 Baud.
Data Format	Asynchronous Serial (return to mark level required between each character).
Receive Channel Frequencies	2025 Hz for space; 2225 Hz for mark.
Transmit Channel Frequencies	Switch selectable: Low (normal) – 1070 space, 1270 mark; High – 025 space, 2225 mark.
Receive Sensitivity	-46 dbm accoustically coupled.
Transmit Level	-15 dbm nominal. Adjustable from -6 dbm to -20 dbm.
Receive Frequency Tolerance	Frequency reference automatically adjusts to allow for operation between 1800 Hz and 2400 Hz.
Digital Data Interface	EIA RS-232C or 20 mA current loop (receiver is optoisolated and non-polar).
Power Requirements	120 VAC, single phase, 10 Watts.
Physical	All components mount on a single 5" by 9" printed circuit board. All components included.

Requires a VOM, Audio Oscillator, Frequency Counter and/or Oscilloscope to align.

The Original the 3rd Hand

$9.95 each

- Leaves two hands free for working
- Clamps on edge of bench, table or work bench
- Position board on angle or flat position for soldering or clipping
- Sturdy, aluminum construction for hobbyist, manufacturer or school rooms

TRS-80 16K Conversion Kit

Expand your 4K TRS-80 System to 16K. Kit comes complete with:
* 8 each UPD416 (16K Dynamic Rams)
* Documentation for conversion

TRS-16K $115.00

Special Offer - Order both your TRS-16K and the Sup'R' MOD II Interface kit together (retail value $144.95) **for only $139.95**

IMC

3½ DIGIT DPM KIT

- New Bipolar Unit
- Auto Zeroing
- .5" LED
- Auto Polarity
- Low Power
- Single IC Unit

Model KB500 DPM Kit	**$49.00**
Model KB503 5V Power Kit	**$17.50**

JE701 Clock Kit

6¾" X 3 ⅛" X 1¾"

$19.95 per Kit

- 6 Digit, .300 Height Common Cathode Display
- Uses MM5314 Clock Chip
- Push Button Switches for setting Hours, Minutes & Hold.
- Easily Viewable to 20 Feet
- Simulated Walnut Case
- 115 VAC Operation
- 12 or 24 Hour Format
- Includes all Components & Case

JE803 PROBE

The Logic Probe is a unit which is for the most part indespensible in trouble shooting logic families TTL, DTL, RTL, CMOS. It derives the power it needs to operate directly off of the circuit under test drawing a scant 10 mA max. It uses a MAN3 readout to indicate any of the following states by these symbols: (HI) 1 (LOW) 0 (PULSE) P. The Probe can detect high frequency pulses to 45 MHz. It can't be used at MOS levels or circuit damage will result.

$9.95 Per Kit

printed circuit board

T_2L 5V 0.5A Supply

This is standard TTL power supply using the well known LM309K regulator IC to provide a solid 0.5 AMP of current at 5 volts. We try to make things easy for you by providing everything you need in one package including the hardware for only.

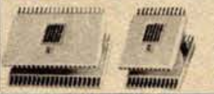
JE225 $9.95 Per Kit

CSC CONTINENTAL SPECIALTIES CORPORATION

PROTO BOARDS

PROTO BOARD 6
$15.95
(6" long X 4" wide)

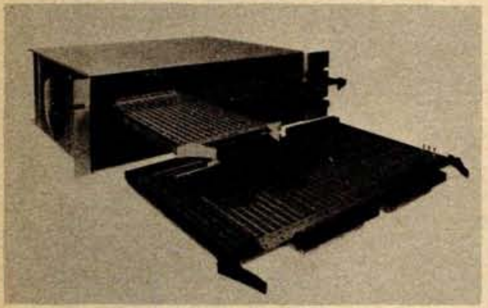
PB100 - 4.5" x 6"	$19.95
PB101 - 5.8" x 4.5"	22.95
PB102 - 7" x 4.5"	26.95
PB103 - 9" x 6"	44.95
PB104 - 9.5" x 8"	54.95
PB203 - 9.75 x 6½ x 2¾	75.00
PB203A - 9.75 x 6½ x 2¾ (includes power supply)	124.95



PROTO CLIPS

14 PIN	$4.50
16 PIN	4.75
24 PIN	8.50
40 PIN	13.75

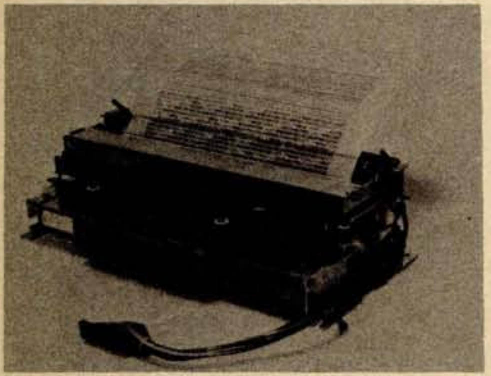
Rack Assembly for Big Prototypes



The Mupac Mixer is a modular packaging system that integrates a variety of panel sizes, IC densities and user selectable voltages into one rack assembly. Logic can be subdivided into multiple size functions. Three independent backplanes permit the modular separation of analog and digital grounds and voltage supply requirements for optimum noise immunity. Multilayer panel construction allows high frequency applications. The system can accommodate panels with 32 to 192 ICs ranging in size from 4.5 inch by 6.0 inch to 8.0 inch by 14.9 inch. For maximum flexibility, blank, copper-clad, cable and wire wrap panels can be mixed. Mupac panels contain from 108 to 540 IO pins to prevent restrictions and pin limitations. The system ranges in price from $400 to $500 for rack assemblies and from $100 to $500 for panels. Contact Mupac Corp, 646 Summer St, Brockton MA 02402. ■

Circle 575 on inquiry card.

80 Column Dot Matrix Printer Mechanism



Featuring a 100,000,000 character dot head, this 80 column Model 3110 dot matrix printer mechanism is being introduced by Epson America Inc, 23844 Hawthorne Blvd, Torrance CA 90505.

The Model 3110 prints 150 characters per second, with a 5 by 7 dot matrix character and 1/10 inch column spacing tailored for the small business and home computer markets.

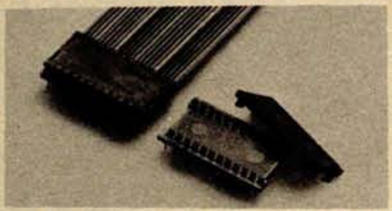
DC power used for the magnet, solenoid and detector allows OEMs to save manufacturing cost by purchasing the standard Epson mechanism for use with systems that will run on US or foreign line currents. The unit requires a 24 V, 30 to 42 V, and 5 V DC power supply plus case, control board and interface electronics.

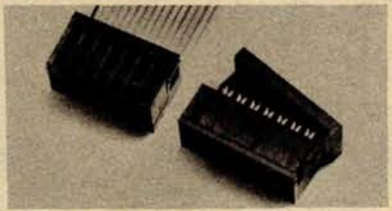
The Model 3110 is 3.74 inches (9.5 cm) high, 13.19 inches (33.5 cm) wide and 7.28 inches (18.5 cm) deep and the weight is 6.6 pounds (3 kg).

The Model 3110 sells for less than $250 in quantities of 500. A 40 column version of this mechanism, designated Model 512, sells for $155 in quantities of 500. ■

Circle 576 on inquiry card.

Planar Cable Connectors for Microcomputer Systems





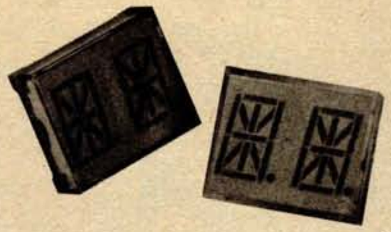
Two series of planar (ribbon) cable connectors have been announced by Spectra-Strip, 7100 Lampson Av, Garden Grove CA 92642. The first of the planar cable connectors are the 804 Series IDC DIP/socket connectors with fixed insulation-displacing contacts. They are available for mass termination to planar cables on .050 inch centers and accept 28 stranded and 28 and 30 solid AWG conductors. The IDC DIP/socket is a one-piece construction connector which features a fixed .013 inch IDC contact with a dual beam configuration. The 804 Series IDC/DIP socket connectors are available with 14, 16, 21 and 24 position versions.

The second of the planar cable connectors are the 805 Series IDC DIP/plug connectors which insert into integrated circuit sockets and provide planar cable interconnection to printed circuit boards. The male DIP/plugs provide preinstalled insulation-displacing contacts for easy mass termination to planar cables on .050 inch centers and offers assembly in seconds without prestripping the cable. The contacts accept 28 stranded and 28 and 30 solid AWG conductors.

The 804 Series connectors are priced in a quantity of 1000 pieces at $.09 per contact and the 805 Series connectors are priced in a quantity of 1000 pieces at $.06 per contact. ■

Circle 579 on inquiry card.

Dual End Stackable Alphanumeric LED Displays



The IEE-Hercules Models 1784/85R .54 inch dual, alphanumeric LED displays with common cathode and right hand decimal point have been announced by Industrial Electronic Engineers Inc, 7740 Lemona Av, Van Nuys CA 91405.

These models consist of two .54 inch high, red 14 segment characters combined in a compact package which can display alphabetic and numeric characters plus some symbols. The end stackable feature allows designers variable display lengths in accordance with their needs. Composed of gallium, arsenic and phosphorous (GaAsP) emitting material, these solid state displays have a typical 600 microcandle per segment luminous intensity at 20 mA at 1.6 V_F. The 18 horizontal double dual-in-line package pins on 0.1 inch spacing are set up for multiplex drive for maximum pinout economy.

Models 1784/85R install in integral, multidigit arrays with IEE-Atlas display mounting hardware. In 500 piece quantities, the price is $4.50 each. For additional information, request catalog HE-1. ■

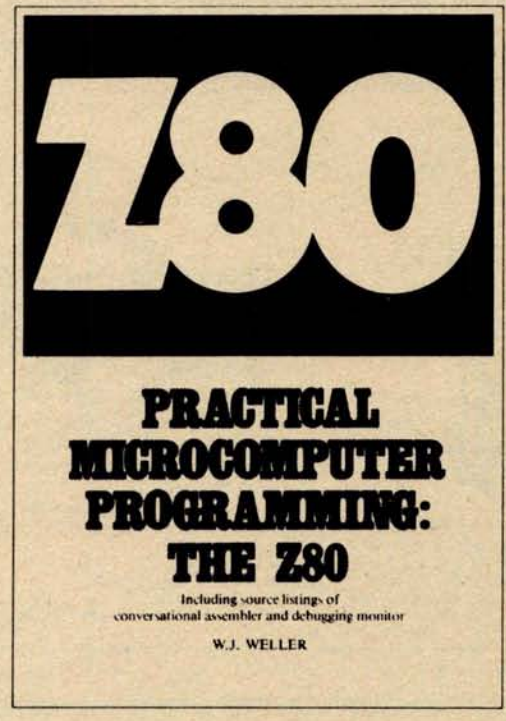
Circle 577 on inquiry card.

Motorola Offers 1024 by 4 Bit Static Programmable Memory

The MCM2114, a 1024 by 4 bit static programmable memory requires no clocks, no timing strobes, nor refreshing because of fully static operation. Data out and data in are of the same polarity and no address set up time is required. Four speed ranges are available: 200 ns, 250 ns, 300 ns and 450 ns, and two power versions, the MCM2114 at 550 mW and the MCM21L14, at 385 mW (maximum), both using a single 5 V supply with ± tolerance. Two industry standard 18 pin packages are available, plastic (P suffix) and lid-seal ceramic (L suffix). The MCM211P-45 (450 ns part in plastic) is priced at $12.25 in quantities of 100 to 999. Contact Motorola Inc, Integrated Circuit Division, Technical Communications, 3501 Ed Bluestein Blvd, Austin TX 78721. ■

Circle 578 on inquiry card.

EVERYTHING
YOU NEED TO LEARN TO PROGRAM THE Z80®
WITHOUT WASTING YOUR 8080 KNOWLEDGE



THE TEXTBOOK...

18 chapters of solid, accurate programming information, on such topics as:

- Single and multilength arithmetic
- Array and table handling and searching
- Number base conversion
- Floating point arithmetic
- Programmed input/output
- Decimal arithmetic *including* multiply and divide
- Stack pointer usage and subroutines
- Debugging techniques
- Interrupt modes and service
- Interrupt driven input/output

ONLY $29.95

THE ASSEMBLY PROGRAM...(easily worth $100.00 by itself)

The full **source listings** of a combined editor/assembler which supports the language used. This language uses 8080 mnemonics for 8080 compatible instructions and clear, logical extensions of the 8080 mnemonics for Z80 only instructions. The assembler is resident in less than 10K RAM and will function with any set of peripherals which transmit on a character by character basis, e.g., paper tape.

A FULL DEBUGGING MONITOR...

Not a simple ROM monitor. Contains facilities for breakpointing, modification of pseudo registers and much more.

AND BY THE WAY...

When the coupon in the book is filled out and returned, the object programs of the editor/assembler and debug are shipped

FREE

DIAL CHARGE CARD ORDERS TOLL FREE (800) 258-5477

BITS inc POB 428, 25 Route 101 West Peterborough, NH 03458

Practical Microcomputer Programming: The Z80
$29.95 plus $.75 postage and handling

☐ Check #: ____________ master charge VISA

☐ My card #: ____________

Expires: ____________

Name ____________

Address ____________

City ____________ State ____________ Zip ____________

Signature ____________

Circle 314 on inquiry card.

RONDURE COMPANY

2522 BUTLER ST.
DALLAS, TEXAS 75235
214-630-4621

the computer room

ASCII SELECTRIC

SPECIAL SALE
$875.00

Used Working and Clean

TESTED WITH NEW ASCII ELECTRONICS

Factory Reconditioned $1075.00

Printer Mechanism: Heavy duty input/output, Series 745.

Weight: 120 lbs. Dimensions: 29"Hx35"Wx33"D.

Print Speed: (14.8 characters per second)

Platen: 15" wide, pin feed or form feed device optional (132 print positions).

Parallel output only—15 characters per second accepts 7 bit ASCII parallel w/strobe & prints on Selectric. The unit still works as a typewriter in off-line mode.

ASCII Selectric with ASCII parallel electronics.
Immediate Delivery—Shipped from inventory.

DATEL SELECTRIC (IBM Selectric Mechanism)

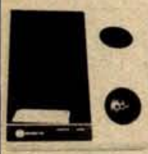
Specifications:
- Size: 21"Wx21"Dx8"H.
- Power Input 115 Volt Hz
- Inferface: RS232
- Weight: 54 lbs. (Shipping weight 65 lbs.)
- 15" Carriage
- 15 CPS
- Correspondence code
- Half Duplex
- 132 Print Positions, 10 Pitch

"As Is" Complete	$395.00
Working	$495.00
Reconditioned	$695.00

Used — $395

NOVATION DC3102A

Used Working



$150.00

RS232 Connection
300 Baud

TI 990/4

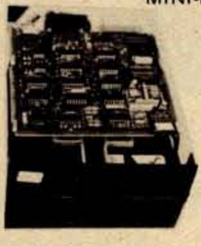
Single Board 16 Bit Micro Computer

NEW $250.00

USED MODEMS & COUPLERS

NOVATION TC102 (Acoustic) $45.00
AJ 233 (Acoustic) $25.00
CF 318 (Hard Wire) $25.00

SHUGART MINI-FLOPPY DRIVE



NEW PRICE
$325.00 ea.
Model SA-400

FLAT PACK ACOUSTICAL MODEM PICK-UP

Useable with most modem chips/kits
Used — $17.50 (w/prints)

ORDERING INFORMATION:
We ship the same day we receive a certified check or money order.
Texas residents add 5% sales tax. Please call if you have a question.
Write for our CATALOG of many parts, terminals, printers, etc.
All items subject to availability. Your money returned if we are out of stock.

SHIPPING INFORMATION:
Modems: $2.00 each; 2 for $4.00 UPS.
Large Items & Parts: Specify Freight or Air Freight Collect
Foreign Orders: Add appropriate freight or postage.
We now take Master Charge and Visa orders, Specify full number, bank number and expiration date.

$249.95
CLEANED & CHECKED OUT

$199.95
AS IS

IBM® Selectric-Based I/O Writers

Excellent Hobby Printers

- Series 72/731
- Heavy Duty
- 8½" Platen
- All Solenoids
- BCD Code

These terminals are from a large airline reservation system. They are heavy duty and were under continuous maintenance. The units have been in storage. We make every effort to ensure that all essential parts are included. Most work when plugged in. No warranties are given or implied.

Selectric Controller

The 3S-01 is a complete controller for the IBM Model 731 I/O typewriter for both input and output operations. With this controller the 731 becomes a versatile ASCII printer with the world famous Selectric quality and an alphanumeric ASCII-encoded keyboard with the wonderful Selectric feel. An eightbit parallel input/output port (bidirectional or separate) is all that is necessary to add the KING of the hardcopy terminals to your system. Serial RS-232C is also available for connection to a serial communications port or modem.

Power supply requirements are 5VDC at .75A and 48VDC at 1A for the basic parallel controller. Additional power needed for the serial unit is ± 12VDC.

PRICE $249.95
ASSEMBLED BOARD

Surplus power supply for above $30.00

Print only interface unit
Board and instructions only $59.95

Complete Terminal Unit

This unit consists of:

1. A cleaned, checkout, repainted used selectric. This unit has been converted for upper & lower case with new ball containing all BASIC characters.
2. Selectric controller unit allowing both input and output
3. Power supply (used)
4. Terminal table (new)
5. Assembled and tested. Ready to plug in and go.
6. ASC II to computer
7. Crated for shipping by motor freight (collect)

PRICE $775.00 FOB TULSA
DEALER INQUIRY INVITED

Have 10 HP 2671B card readers left at $299.95 each FOB Tulsa.

Cashier Check or Money Order. Personal check allow 3 weeks. Units shipped collect. Price Net FOB Tulsa.

SUPER SALE

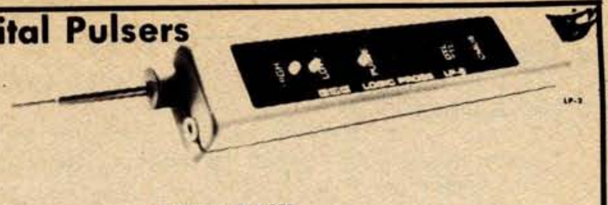
3 S Sales, Inc.
P. O. Box 45944
Tulsa, OK 74145
1-918-622-1058

Circle 356 on inquiry card.

CONTINENTAL SPECIALTIES CORPORATION

CSC

Logic Probes and Digital Pulsers



LOGIC PROBES

CSC logic probes are the ultimate tool for breadboard design and testing. These hand-held units provide an instant overview of circuit conditions. Simple to use: just clip power leads to circuit's power supply, set logic family switch to TTL/DTL or CMOS/HTL. Touch probe to test node. Trace logic levels and pulses through digital circuits. Even stretch and latch for easy pulse detection. Instant recognition of high, low or invalid levels, open circuits and nodes. Simple, dual-level detector LEDs tell it quickly, correctly. HI (Logic "1"); LO (Logic "0"). Also incorporates blinking pulse detector, e.g., HI and LO LEDs blink on or off, tracking "1" or "0" states at square wave frequencies up to 1.5 MHz. Pulse LED blinks on for ⅓ second during pulse transition. Choice of three models to meet individual requirements: budget, project and speed of logic circuits.

MODEL LP-1

Hand-held logic probe provides instant reading of logic levels for TTL, DTL, HTL or CMOS. **Input Impedance:** 100,000 ohms. **Minimum Detectable Pulse:** 50 ns. **Maximum Input Signal (Frequency):** 10 MHz. **Pulse Detector (LED):** High speed train or single event. **Pulse Memory:** Pulse or level transition detected and stored.

CSC Model LP-1 Logic Probe—Net Each **$44.95**

MODEL LP-2

Economy version of Model LP-1. Safer than a voltmeter. More accurate than a scope. **Input Impedance:** 300,000 ohms. **Minimum Detectable Pulse:** 300 ns. **Maximum Input Signal (Frequency):** 1.5 MHz. **Pulse Detector (LED):** High speed train or single event. **Pulse Memory:** None.

CSC Model LP-2 Logic Probe—Net Each **$24.95**

MODEL LP-3

High speed logic probe. Captures pulses as short as 10 ns. **Input Impedance:** 500,000 ohms. **Minimum Detectable Pulse:** 10 ns. **Maximum Input Signal (Frequency):** 50 MHz. **Pulse Detector (LED):** High speed train or single event. **Pulse Memory:** Pulse or level transition detected and stored.

CSC Model LP-3 Logic Probe—Net Each **$69.95**

DIGITAL PULSER

The ultimate in speed and ease of operation. Simply connect clip leads to positive and negative power, then touch DP-1's probe to a circuit node; automatic polarity sensor detects circuit's high or low condition. Depress the pushbutton and trigger an opposite polarity pulse into the circuit. Fast troubleshooting includes injecting signals at key points in TTL, DTL, CMOS or other popular circuits. Test with single pulse or 100 pulses per second via built-in dual control push-button; button selects single shot or continuous modes. LED indicator monitors operating modes by flashing once for single pulse or continuously for a pulse train. Completely automatic, pencil-size lab/field pulse generator for any family of digital circuits. **Output:** Tri-state. **Polarity:** Pulse-sensing auto-polarity. **Sync and Source:** 100 mA. **Pulse Train:** 100 pps. **LED Indicator:** Flashes for single pulse; stays lit for pulse train.

CSC Model DP-1 Digital Pulser—Net Each **$74.95**

Prototype Design Equipment

PROTO-BOARD® UNITS

All the speed and convenience of QT Sockets and Bus Strips in both kits and preassembled units. Assemble, test and modify circuits literally as fast as you can think.

PROTO-BOARD 6 KIT

Contains one preassembled QT-47S socket, two preassembled QT-47B bus strips, four 5-way binding posts, metal ground/base plate, non-marring feet and all hardware. Ten minute assembly time. **Size:** 6" l. x 4" w. x 1.4" h. **Weight:** 7 ozs.

CSC Model PB-6 Kit—Complete. List Price **$15.95**

PROTO-BOARD 100 KIT

Contains two preassembled QT-35S sockets, one preassembled QT-35B bus strip, two 5-way binding posts, non-metallic base plate, non-marring feet and all hardware. Ten minute assembly time. **Size:** 4.5" w. x 6" l. x 1.4" h. **Weight:** 7.5 ozs.

CSC Model PB-100 Kit—Complete. List Price **$19.95**

PROTO-BOARD 101

Fully assembled breadboard contains two QT-35S sockets and four QT-35B bus strips mounted on metal ground/base plate with non-marring feet. Excellent for audio and small digital projects. **Size:** 5.8" l. x 4.5" w. x 1.4" h. **Weight:** 9 ozs.

CSC Model PB-101 Breadboard—Complete. List Price ~~$29.95~~ $22.95

PROTO-BOARD 102

Fully assembled breadboard contains two QT-47S sockets, three QT-47B bus strips and one QT-35B bus strip on a metal ground/base plate with non-marring feet. Excellent for intermediate digital needs. **Size:** 7" l. x 4.5" w. x 1.4" h. **Weight:** 10 ozs.

CSC Model PB-102 Breadboard—Complete. List Price ~~$39.95~~ $26.95

PROTO-BOARD 103

Fully assembled breadboard contains three QT-59S sockets, four QT-59B and one QT-47B bus strips, four 5-way binding posts on a metal ground/base plate with non-marring feet. Build calculators, interfaces, networks, etc. **Size:** 9" l. x 6" w. x 1.4" h. **Weight:** 1¼ lbs.

CSC Model PB-103 Breadboard—Complete. List Price ~~$59.95~~ $44.95

PROTO-BOARD 104

Fully assembled breadboard contains four QT-59S sockets, seven QT-59B bus strips and four 5-way binding posts on a metal ground/base plate with non-marring feet. Build a CPU, encoder, complex display, etc. **Size:** 9.8" l. x 8" w. x 1.4" h. **Weight:** 1¾ lbs.

CSC Model PB-104 Breadboard—Complete. List Price ~~$79.95~~ $54.95

PROTO-BOARD 203

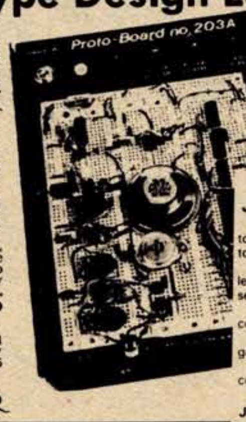
Fully assembled breadboard contains built-in, short-proof, fused, 5 VDC at 1 amp, regulated power supply, in addition to three QT-59S sockets, four QT-59B bus strips, one QT-47B bus strip and four 5-way binding posts. Capacity for most digital and many analog projects. **Size:** 9.75" l. x 6.6" w. x 3.25" h. **Weight:** 5 lbs.

CSC Model PB-203 Breadboard—Complete. List Price **$75.00**

PROTO-BOARD 203A

Provides all the features of Proto-Board 203 with additional +15 and −15 VDC at 0.5 amp power supplies with internally adjustable output voltages. **Size:** Same as PB-203. **Weight:** 5.5 lbs.

CSC Model PB-203A Breadboard—Complete. List Price .... ~~$149.00~~ $124.95



BREADBOARD JUMPER WIRE KIT

Each kit contains 350 wires cut to 14 different lengths from 0.1" to 5.0".

Each wire is stripped and the leads are bent 90° for easy insertion.

Wire length is classified by color coding.

All wire is solid tinned 22 gauge with PVC insulation.

The wires come packed in a convenient plastic box.

JK1 923351........ $10.00

LOGIC MONITOR 1

Trace signals through all types of digital circuits. Unit clips over any DIP IC up to 16 pins. Each of its 16 contacts connects to a single-bit level detector that drives a high-intensity, numbered LED readout activated when the applied voltage exceeds a fixed 2 V threhold. Logic "1" turns LED on; logic "0" keeps LED off. A power-seeking gate network automatically locates supply leads and feeds them to the LM-1's internal circuitry. Saves minutes, even hours in design, troubleshooting, debugging of equipment. **Voltage Threshold:** 2 V ± 0.2 V. **Input Impedance:** 100,000 ohms. **Input Voltage Range:** 4-15 V max. across any two or more inputs. **Current Drain:** 200 mA at 10 V. **Size:** 4" l. x 2" w. x 1.75" d. when open. **Weight:** 3 ozs.

CSC Model LM-1 Logic Monitor—Complete. List Price **$59.95**

LOGIC MONITOR 2

Provides greater versatility and precision in testing all types of digital circuits. The fully isolated power supply and selectable trigger threshold let you match the precise characteristics of the logic family under test, permitting more accurate measurements. The connector display unit clips over any DIP IC up to 16 pins. The power supply module contains the precision reference power supply and logic family selector switch. Operation is simple. Set the threshold switch to the proper logic family. Connect black clip lead to circuit ground. For CMOS circuitry, the red clip is connected to circuit positive or Vcc. The clip module is then clipped over the IC and the LED display instantly gives the logic states of all pins. **Logic Thresholds:** CMOS, 70% of circuit Vcc; HTL, 7.5 V; TTL, 2.4 V; DTL, 1.6 V; RTL, 1.2 V. **Maximum Visible Input Frequency:** 30 KHz at 50% duty cycle. **Size:** Test Clip, 4" l. x 2" w. x 1.75" d. when open; Power Module, 6" w. x 5.63" d. x 3" h. max. **Total Weight:** 20 ozs. **Power Required:** 117 VAC, 50/60 Hz, 10 W; also available for 220 VAC, 50/60 Hz at slightly higher price.

CSC Model LM-2 Logic Monitor—Complete. List Price **$129.95**

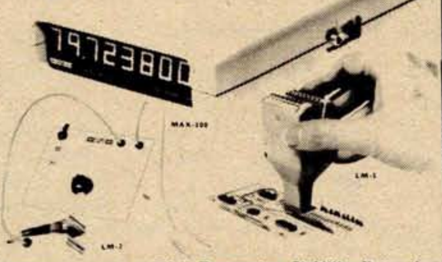
MAX-100 PORTABLE FREQUENCY COUNTER

MAX-100 is a portable, high precision frequency counter that sets new standards in performance and value. In a compact, portable case, it gives you continuous readings from 20 Hz to a guaranteed 100 MHz, with 8-digit accuracy. Fast readings with ⅙-sec. update and 1-sec. sampling rate. Precise readings, derived from a crystal-controlled time base with 3 ppm accuracy. High-sensitivity readings from signals as low as 30 mV, with diode overload protection up to 200 V peaks. Input signals over 100 MHz automatically flash the most significant digit, and to indicate low-battery condition and extend battery life, the entire display flashes at 1 kHz.

SPECIFICATIONS

Frequency Range: 20 Hz to 100 MHz guaranteed; 110 MHz typical. **Gate Time:** 1 sec. **Resolution:** 1 Hz. **Accuracy:** ± 1 count + time base error. **Input Impedance:** 1 megohm shunted by 56 pF. **Coupling:** AC. **Sine Wave Sensitivity:** 30 mV RMS at 50 MHz. **Internal Time Base Frequency:** 3.579545 MHz crystal oscillator.

Digital IC Testers



Stability: ±3 ppm at 25° C. **Temperature Stability:** Better than 0.2 ppm/°C, 0-50° C. Max. **Aging:** 10 ppm/year. **Display:** Eight 0.6" LED digits. **Lead-Zero Blanking:** Decimal point appears between 6th and 7th digit when input exceeds 1 MHz. **Overflow:** With signals over 99,999,999 Hz, most significant (left hand) digit flashes, allowing readings in excess of 100 MHz. **Display Update:** ⅙ sec. plus 1 sec. gate time. **Low Battery Indicator:** When battery supply falls below 6.6 VDC, all digits flash at 1 Hz. **Power Required:** Internal, 6 "AA" cells; external, 110 or 220 VAC charger/eliminator, auto cigarette lighter adapter or 7.2-10 VDC external supply. **Battery Charging:** 12-14 hrs. **Size:** 1.75" h. x 5.63" w. x 7.75" d. **Weight:** Less than 1.5 lbs. with batteries.

CSC Model MAX-100 Frequency Counter—Net Each **$134.95**

MAX-100 ACCESSORIES

Model 100-CLA	Mobile charger eliminator. Net Each	$ 3.95
Model 100-CA1	110 VAC charger eliminator. Net Ea.	9.95
Model 100-CA2	220 VAC charger eliminator. Net Ea.	9.95
Model 100-IPC	Input cable with clip leads. Net Each	5.95
Model 100-LLC	Low-loss tap-off	14.95
Model 100-MWA	Mini-whip antenna	3.95
Model 100-CC	Carrying case	9.95

NEW VOLKSMETERS!

With LCD Display— Excellent Readability in Direct Sunlight!

NLS

NEW

0.5% 3½ digits
LM-350 **$144.00***

1%
LM-300
Full three digits
$114.00*

FEATURES

- Measures DC volts, AC volts, ohms and current
- Automatic polarity, decimal and overload indication
- No zero adjustment and no full scale ohms adjust
- Large LCD display for easy reading without interpolation
- Size 1.9" H x 2.7" W x 4.0" D
- Parts and labor guaranteed for one year

SPECIFICATIONS

MODE	RANGE	ACCURACY LM-300	ACCURACY LM-350	RESOLUTION	INPUT RESISTANCE	TEST CURRENT
VOLTS DC*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
VOLTS AC*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
KILOHMS*	[illegible]	[illegible]	[illegible]	[illegible]		[illegible]
CURRENT	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	

Tilt Stand Option, add	$ 3.50
Leather Case	$ 20.00
3AA NiCad Batts. & Charger	$ 12.00

Standard AA-size batteries provide up to 20 hours of operation.
Rechargeable NiCad batteries and charger unit available as optional equipment.
*Batteries not included.

Purchase any of the LM series Meters and buy the LEATHER CASE for 1¢

PRIORITY ONE ELECTRONICS ©

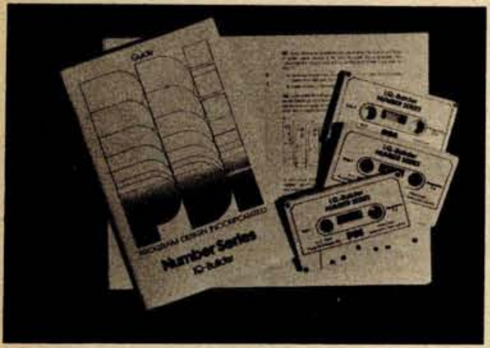
10031 A' Woodley Ave. Sepulveda CA. 91343

Terms: VISA, MC, BAC, check, Money Order, C.O.D., U.S. Funds Only. CA residents add 6% sales tax. Minimum order $10.00. Orders less than $75.00 include 10% shipping and handling; excess refunded. Just in case please include your phone no. **"Sorry, no over the counter sales"** GOOD THRU NOVEMBER 1978

Send for our latest brochure. **phone orders welcome (213) 893-8202** OEM and Institutional inquiries invited.

Circle 306 on inquiry card.

Educational Programs for PET, TRS-80 and Apple II



Educational programs for the PET, TRS-80 and Apple II computers are now available from Program Design Inc, 11 Idar Ct, Greenwich CT 06830. Each course comes with programs on cassette tapes, workbook or guide and other materials necessary for effective learning. Currently available courses include:

- IQ Builder: This series of three courses (analogies, number series and vocabulary builder) develops the skills needed to succeed on aptitude tests. Programs can be purchased and used separately or together. Analogies is priced at $9.50, number series is $9.50, and vocabulary builder is $12.50.
- Step by step: This is a course in the BASIC programming language. Structured lessons and guided practice sessions are presented on the computer and in the workbook. Step by step is priced at $29.95.
- Preschool IQ builder: This course helps 3 to 5 year olds develop intellectual skills essential for learning to read. This program is priced at $10.50.■

Circle 584 on inquiry card.

Microprocessor Cross Assembler Available for PDP-11 or LSI-II

Five microprocessor cross assemblers for use on DEC's PDP-11 minicomputers and LSI-11 microprocessors have been announced by Automated Logic Corp, 2675 Cumberland Pky, Suite 115, Atlanta GA 30339. The MicroSeries cross assemblers can be used for any of the following processors: Intel 4040, 8080, 8085, 8748, 8048, 8041, 8035 and 8021.

The MicroSeries runs in 12 K words of memory and enables a program to be developed using the PDP-11 with the RT-11 operation system. A companion program enables the output from the cross assembler to be shipped directly to burn the programmable read only memories.

The MicroSeries price is $250 and is distributed on floppy disks.■

Circle 585 on inquiry card.

New PDP-8 X8 Cross Assembler Series

Four new microprocessor cross assemblers have been added to Sierra Digital Systems' X8 cross assembler series for the Digital Equipment Corporation PDP-8 minicomputer. The X8 series cross assemblers now cover the Z-80, 1802, SC/MP and 8048 microprocessors in addition to the previous 6502, 6800, 8080, F8 and 2650 versions. By using an X8 series cross assembler, assembly language programs are converted into object code or put into programmable read only memory. The assemblers run in 8 K words of memory under the OS/8 operating system, and are written in PDP-8 assembly language. Pseudo-operations and runtime options provide for conditional assembly and listing control. Generated object code may be output in the microprocessor's standard loader format, or BNPF for read only memory programming with commercial stand alone programmers. Each cross assembler is priced at $400 and distributed in PDP-8 binary format on paper tape, DECtape, or DEC floppy diskette. Source files are also available for an additional $250. Contact Sierra Digital Systems, 13905 Rancheros Dr, Reno NV 89511.■

Circle 586 on inquiry card.

Disk Based Software Development Tools

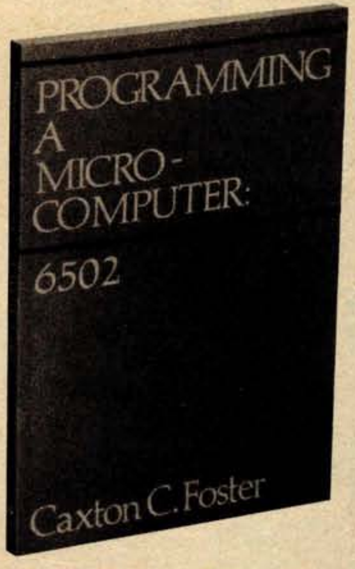
A full complement of Z-80/8080/8085 disk based software development tools oriented towards the CP/M operating system is now available from TSA Software, 5 N Salem Rd, Ridgefield CT 06877. System development tools include a relocatable linking macro-assembler with linking loader, cross-reference generator and full library of modules. Included with the assembler is a symbolic debugger allowing user defined symbols. Higher level language support is provided by interface with Micro-Soft FORTRAN.

For advanced systems, TSA/OS is an upward compatible CP/M-like operating system providing video screen control, automatic library search, an extended batch mode with turnkey system capability, as well as an advanced configuration scheme.

TSA Software has available a set of applications packages. The TSA Database System uses a mixture of assembly code and FORTRAN. The system uses table driven screen and record formats and has a minicompiler to optimize record search capability. The TSA Word Processor uses a normal terminal to provide natural text editing with advanced formatting features, including proportional printing. Full use of disk files is provided as well as file merging for mailing list and similar uses. Package prices are $100 and up.■

Circle 587 on inquiry card.

Learn to Program a Microcomputer in Machine Language



Programming a Microcomputer: 6502 by Caxton C Foster teaches you how to program a microcomputer in machine language. Although designed especially for the 6502 microprocessor used in the KIM-1, PET and Apple microcomputer systems, the basic principles covered apply to all computers, large or small. The 234 page book, which assumes no previous knowledge of computers, is published by Addison-Wesley Publishing Company Inc, Reading MA 01867.■

Circle 588 on inquiry card.

Foreign Language Vocabulary Programs Available in BASIC

Foreign Language Vocabulary is a bidirectional program for the instruction, practice and testing of language vocabulary skills. Languages offered include French, Spanish, Italian and German.

Program features include separate modes for vocabulary instruction, practice drills and testing, selectable by the user at any time during program operation. The user may also alternate language direction (English to French or French to English) to improve comprehension. The Educator option permits the creation of files for the storage of student identification, test responses and test scores, for use in a classroom situation.

Foreign Language Vocabulary is written in BASIC. Each volume includes an annotated program listing and program flowchart to assure ease of user loading and understanding. Single statement lines are used to avoid confusion and to permit ease of user modification.

Each volume of Foreign Language Vocabulary is priced at $5 with the Educator option costing an additional $3. All four volumes are $17.50 or $27.50 with the Educator option. For further information, contact Musgrove Engineering, 9547 Kindletree Dr, Houston TX 77040.■

Circle 589 on inquiry card.

SALE S·100 BUS EDGE CONNECTORS SALE

NLS MS-215 DUAL TRACE MINISCOPE $435.00

1¢ SALE

LM3A 3 dig 1% DC $134.00
LM3.5A 3½ dig .5% DC .... $ 158.50
LM40A 4 dig .1% DC....... $ 209.00
LM4A 4 dig .03% DC ...... $ 250.00

- Rechargeable batteries and charger included
- Measures DC Volts, AC Volts, Ohms and Current
- Automatic polarity, decimal and overload indication
- Rechargeable batteries and charger
- Measures DC Volts, AC Volts, Ohms and Current
- Automatic polarity, decimal and overload indication
- No zero adjustment and no full-scale ohms adjust
- Battery-operated — NiCad batteries; also AC line operation.
- Large LED display for easy reading without interpolation
- Size: 1.9"H x 2.7"W x 4"D
  Parts & labor guaranteed 1 year
  Tilt stand option .... $ 3.50
- Leather case $20.00

Purchase any of the LM series Meters and buy the LEATHER CASE for 1¢

$318. MS-15 MINISCOPE

With Rechargeable Batteries & Charger Unit

- 15 megahertz bandwidth.
- External and internal trigger.
- Time base — .1 microsec. to 0.5 Sec/div - 21 settings ±3%.
- Battery or line operation.
- Automatic & line sync modes.
- Power consumption < 15 watts.
- Vertical Gain — .01 to 50 V/div - 12 settings ± 3%.
- Viewing area 1.1" x 1.35".
- Case size 2.7"H x 6.4"W x 7.5"D, 3 pounds.
- Parts & Labor guaranteed 1 year
- 10 to 1, 10 meg probe .... $27.
- Leather carrying case .... $45.

PROBE 1¢

PROBE 1¢ with the purchase of SCOPE and the MENTION of this MAGAZINE

● MS-215 Dual Trace Version of MS-15 **$435.**

SALE S-100 BUS EDGE CONNECTORS SALE

S100-WWG 50/100 Cont. 125 ctrs. 3 LEVEL WIRE WRAP .025" sq. posts on .250 spaced rows. GOLD plated.

1-4	5-9	10-24
$4.00	$3.75	$3.50

S100-STG 50/100 Cont. 125 ctrs. DIP SOLDER TAIL on .250 spaced rows for VECTOR and MASI motherboards GOLD plated.

1-4	5-9	10-24
$4.00	$3.75	$3.50

RG81G 50/100 Cont. .125 ctrs DIP SOLDER TAIL on .140 spaced rows for ALTAIR motherboards. GOLD plated. $6.00

R681-3 50/100 Cont. .125 ctrs. PIERCED SOLDER EYELET tails. GOLD $7.35

Other Popular Edge Connectors

R644-G 22/44 Cont. .156 ctrs. PIERCED SOLDER EYELES tails. GOLD plated.

1-4	5-9	10-24
$3.00	$2.75	$2.50

R644-3 22/44 Cont. .156 ctrs WIRE WRAP tails. GOLD $4.71

ATTN: OEM'S and Dealers, many other connectors available call or quotation.

3 LEVEL GOLD WIRE WRAP SOCKETS

Sockets purchased in multiples of 50 per type may be combined for best price.

	1-24	25-49	50-99	100-249	250-999	1K-5K
8 pin*	.41	.38	.35	.31	.27	.23
14 pin*	.39	.38	.36	.32	.29	.27
16 pin*	.43	.42	.39	.35	.32	.30
18 pin	.63	.58	.54	.47	.42	.36
20 pin	.80	.75	.70	.63	.58	.53
22 pin*	.90	.85	.80	.70	.61	.57
24 pin	.90	.84	.78	.68	.63	.58
28 pin	1.10	1.00	.90	.84	.76	.71
40 pin	1.50	1.40	1.30	1.20	1.04	.89

All sockets are GOLD 3 level closed entry. * End and side stacable. 2 level, Solder Tail, Low Profile, Tin Sockets and Dip Plugs available. CALL FOR QUOTATION

LIQUID CRYSTAL DIGITAL CLOCK-CALENDAR

- For Auto, Home, Office
- Small in size (2x2⅛x¾)
- Push button for seconds release for date.
- Clocks mount anywhere with either 3M double-sided tape or VELCRO, included.
- 2 MODELS AVAILABLE: LCD-101, portable model runs on self-contained batteries for better than a year. LCD-102, runs on 12 Volt system and is back-lighted.
- LCD-101 or LCD-102 your choice .... $34.95 ea.
- Clear desk stand for $2.00

8803 MOTHER BOARD FOR S100 BUS MICRO-COMPUTERS — Vector

- Kit includes 12 tantalum capacitors for +5, +12, −12 buses and insulated mounting spacers.
- Wiring side shown. Component side bare epoxy glass with white markings for component locations.
- G10 epoxy glass board with 2 ounce copper, solder plated and .038 diameter holes for leads.
- Solder mask with solder windows on etched circuits to avoid accidental short circuits.
- Mounts 11 receptacles with 100 contacts (2 rows) on .125 centers with .250 row spacing. Vector part number R681-2; or mounts 10 receptacles plus interconnections to smaller mother board for expansion.
- Includes etched circuits and instructions for option of active, pull-up, or floating terminations.
- Large buses: +5V and GND (10 AMPS), ± 12V or 16V (7 AMPS). Current ratings are per MIL-STD-275 with 10°C rise.
- Fits in Vector-pak enclosures.
- Fits in IMSAI 8080 microcomputer as expander board.

Price: $29.50

Vector Plugboards

8800V

Universal Microcomputer/processor plugboard, use with S-100 bus. Complete with heat sink & hardware. 5.3" x 10" x 1/16".

1-4	5-9	10-24
$19.95	$17.95	$15.96

8801-1

Same as 8800V except plain; less power buses & heat sink.

1-4	5-9	10-24
$14.95	$13.46	$11.96

3682 9.6" x 4.5" $10.97

3682-2 6.5" x 4.5" $9.81

Hi-Density Dual-In-Line Plugboard for Wire Wrap with Power & Grd. Bus Epoxy Glass 1/16" 44 pin con. spaced .156

3677 9.6" x 4.5" **$10.90**

3677-2 6.5" x 4.5" **$9.74**

Gen. Purpose D.I.P. Boards with Bus Pattern for Solder or Wire Wrap. Epoxy Glass 1/16" 44 pin con. spaced .156

3662 6.5" x 4.5" **$7.65**

3662-2 9.6" x 4.5" **$11.45**

P pattern plugboards for IC's Epoxy Glass 1/16" 44 pin con. spaced .156

3690-12 CARD EXTENDER

Card Extender has 100 contacts-50 per side on .125 centers-Attached connector-is compatible with S-100 Bus Systems $25.00

3690 6.5" 22/44 pin .158 ctrs. Extenders $12.00

1/16 Vector BOARD

.042 dia holes on 0.1 spacing for IC's

PART NO.	SIZE	PRICE 1-9	10-19
Phenolic			
64P44XXXP	4.5x6.5"	$1.49	1.34
169P44XXXP	4.5x17"	$3.51	3.16
Epoxy Glass			
64P44	4.5x6.5"	$1.70	1.53
84P44	4.5x8.5"	$2.10	1.89
169P44	4.5x17"	$4.30	3.87
169P84	8.5x17"	$7.65	6.89

SLIT-N-WRAP

Wraps insulated wire on .025" square posts FOUR TIMES FASTER than regular manual wrap-post tools

P180 with two 100' spools of 28 ga. wire **$24.50**

P180-4T includes charger, wire **$75.00**

Vector

NO PRE-STRIPPING*
NO PRE-CUTTING*
SPOOL-FED WIRE*

*The spooled wire passes through the tool past a slitting edge next to the wrap post. A narrow longitudinal cut is made in the insulation where it presses the square post corner, the bared copper is indented by the sharp edge (7 turns = 28 contacts). Insulation is slit where wrapped but not between terminal posts when laid straight out of the tool.

"Daisy-chain" runs. Reliable and easy to use. Manual or power operation.

SLIT-N-WRAP WIRE NO. 28 GAGE INSULATED WIRE, 100' SPOOLS — $2[75] Package

W28-2-A-Pkg. 3. Green W28-2-C-Pkg. 3. Clear
W28-2-B-Pkg. 3. Red W28-2-D-Pkg. 3. Blue

2708 8K 450 ns EPROM

FACTORY PRIME

$ 9.00 EA.

25 + *Call For Price*

14 & 16 PIN GOLD 3 LEVEL WIRE WRAP SOCKETS

WOW

14 - G3 100 for $30.00

16-G3 100 for $30.00

50 of each for $32.00

Sockets are End & Side stackable, closed entry

PANAVISE

P.C. BOARD HOLDER 315 — PRICE: $16.95

315-S same as 315 but with 14" bar to accomodate "S100" boards. PRICE: $17.95

LOW-PROFILE BASE 305 — PRICE: $11.95

STANDARD BASE 300 — PRICE: $11.95

VACUUM BASE 380 — PRICE: $16.95

STANDARD VISE HEAD 303 — PRICE: $12.95

HORIZONTAL JAW VISE HEAD 304 — PRICE: $12.95

WIDE OPENING VISE HEAD 366 — PRICE: $12.95

PANAVISE TILTS, TURNS, AND ROTATES TO ANY POSITION. IT HOLDS YOUR WORK EXACTLY WHERE YOU WANT IT.

LEDU MG 10A List $72.

Perfectly balanced, fluorescent lighting with precision magnifier lens. For prof'l, techn'l & hobbist. Has die cast protective shade, inst. start 3 diopter lens, 42" reach.

$44.95 SPECIAL

NLS SC-5 / FM-7

SC-5 With Rechargeable Batteries & Charger Unit **$98**

CASE 1¢ EXTRA

FM-7 With Rechargeable Batteries & Charger Unit **$215**

Features include: • By using the new NLS SC-5 Prescaler, the range of the FM-7 Frequency Meter, which is 10 Hz to 60 MHz, may be extended to 512 MHz (the upper VHF & UHF frequency bands). • The FM-7 utilizes an LED readout, providing 7-digit resolution. • The FM-7 can be calibrated to an accuracy of 0.00001%. • The SC-5 is accurate to one part per million. • Each unit has 30 millivolts sensitivity, is battery powered and has a charger unit included. • Dimensions of each are 1.9" H x 2.7" W x 3.9" D. • The units may be obtained separately or as a "Frequency Duo." • Parts & Labor guaranteed 1 year.

Tilt stand option .... $ 3.50
Leather case $20.00

Price Breakthrough! $17.50 MA1003 CAR CLOCK

Bright Green Fluorescent Display Crystal Time Base Assembled, just add switches and 12 VDC.

SPECIAL

14CS2 100 for $14.00
16CS2 100 for $16.00
14 pin CS2 10 for $2.00
16 pin CS2 8 for $2.00

These low cost DIP sockets will accept both standard width plugs and chips.

For use with chips, the sockets offer a low profile height of only .125" above the board. These sockets are end stackable.

Vector WRAP POST

for .042 dia. holes (all boards on this page)
T-44 pkg. 100 ... $ 2.28
T-44 pkg. 1000 ...... $14.00
A-13 hand installing tool ...... $ 2.80

PRIORITY ONE ELECTRONICS ©

10031 A Woodley Ave. Sepulveda CA. 91343

Terms: VISA, MC, BAC, check, Money Order, C.O.D., U.S. Funds Only. CA residents add 6% sales tax. Minimum order $10.00. Orders less than $75.00 include 10% shipping and handling; excess refunded. Just in case . . . please include your phone no. **"Sorry, no over the counter sales"** GOOD THRU NOVEMBER 1978

Send for our latest brochure. phone orders welcome (213) 893-8202 OEM and Institutional inquiries invited.

24 PIN DIP PLUGS WITH COVERS

3 / $1.00
40 / $10.00

NLS MS-215 DUAL TRACE MINISCOPE $435.00

SALE S·100 BUS EDGE CONNECTORS SALE

Circle 306 on inquiry card.

CANADIANS

Announcing

HAMILTON LOGIC SYSTEMS

Specializing in logic devices, microprocessors, memorys, TTL, Cmos, etc.

Send for your catalogue

Box 7
STONEY CREEK
ONTARIO L8G 3X7

Circle 157 on inquiry card.

Dynamic RAM Breakthrough
16K Bits For $15.95

The best current prices and better than you can expect to beat for quite some time:

MC4116L-20 PRIME MOTOROLA DYNAMIC RAM (16 pin, 200ns access, 375ns cycle time, 70C, ceramic package, gold pins, fully guarranteed) $15.95 per chip. These plug directly into your TRS-80 or APPLE computer. Data sheet included with your order. For more than 100 units call for quantity pricing.

S-100 64K MEMORY BOARD KIT (for 16K to 64K of above chips) manuals, and all components except memory chips $125.

S-100 CONNECTORS, (used in excellent condition, some with wire wrap wire) 3 level wire wrap, Imsai Spacing, $2.95 ea, 10 for $27.00, 30 for $75.00.

CARD GUIDES Wierd (but very nice) with retaining clips $.30 a pair.

Minimum order $10.00. Mastercharge and Visa accepted. For questions and charge orders (no COD) call Phill Apley or David Lourie at (617) 242-3350.

Please add $1.00 for shipping and handling on all orders.

Remit to:

The Memory Coop
144 MIT Branch
Cambridge MA 02139

Due to the unpredictability of the electronics marketplace prices are subject to change without notice.

Circle 221 on inquiry card.

MICRO-VERTER

A SPECIAL COLOR MODULATOR FOR APPLE II USERS!!

UHF Version. Operates above channel 14. Eliminates worms!

Operates above the switching harmonics of the computer, thereby yielding a cleaner, worm-free picture. Tunable over a minimum of 4 channels. Interfaces directly with the Apple II as well as most other micros. Comes with video cable and RF output stub coupler. Two-toned cowl type decorator cabinet. Size: 5.5cm x 8.5cm x 11.5cm. Power: +5V. Current approx. 1 ma. Self-powered with 4 pencell batteries. Operating life in excess of 1000 hours or near shelf-life of batteries. Excellent stability. Precise frequency adjustment. No assembly required except for installation of batteries, not supplied. MODEL MVX-500.

AVAILABLE FROM YOUR LOCAL COMPUTER DEALER or direct from ATV RESEARCH. COST including shipping anywhere in USA and Canada — $35.00.

"PIXE-PLEXER" An IC type video-to-RF modulator includes FM sound sub-carrier, color subcarrier and separate R-Y and B-Y inputs. Designed around the LM-1889 chip. A designer's dream with full data sheets. Model PXP-4500. Kit form. $24.50 postpaid.

"PIXE-VERTER" The original computer video-to-RF interface module. Kit form: $8.50 Model PXV-2A

PHONE or WRITE TODAY. DIAL 402-987-3771.

13-B Broadway **ATV Research** Dakota City, Nebr. 68731

Circle 22 on inquiry card.

RADIO SHACK COMPUTER OWNERS

TRS-80

MONTHLY NEWSLETTER

BUSINESS • PERSONAL FINANCE
PRACTICAL APPLICATIONS
GAMBLING • GAMES
LATEST TRS-80 DEVELOPMENTS
SOFTWARE EXCHANGE • MARKET PLACE
QUESTIONS AND ANSWERS
PROGRAM PRINTOUTS
. . . AND MORE

MAJOR PROGRAMS PUBLISHED MONTHLY • INCOME TAX PROGRAM – LONG AND SHORT FORMS • INVENTORY CONTROL • EXTENSIVE MAILING LIST AND FILE PROGRAM • PAYROLL • STOCK SELECTION • PICKING WINNING HORSES • RENUMBER PROGRAM LINES • CHESS • CHECKERS • FINANCIAL APPLICATIONS PACKAGE • PERSONAL FINANCE PACKAGE • GRAPHICS • STATISTICS • MATHEMATICS • EDUCATION . . . AND MORE

$24 Per Year—Sample Issue $4
VISA Mastercharge

MATHEMATICAL APPLICATIONS SERVICE
Box 149 RB
NEW CITY, NY 10956
(914) 425-1535

(Send for our FREE Software Catalog)

Circle 217 on inquiry card.

O.S.I.

OHIO SCIENTIFIC ☆ COMPUTERS ☆

SALES & SERVICE

Business/Home/Hobby Hardware & Software

— SEE O.S.I. NOW! —

METRO D.C. AREA & MARYLAND
The Math Box, Inc.
(301) 277-6828

CENTRAL VIRGINIA AREA
H/B Computers, Inc.
(804) 295-1975

Circle 213 on inquiry card.

IT'S A GREAT BIG COMPUTER WORLD
But You Only Need

THE COMPUTER CORNER

- SOL – A New Dawn is Here!
- COMMODORE PET & KIM
- NORTHSTAR HORIZON
- IMSAI VDP-80
- Memories & I/O Boards
- Computer Book Service
- Magnetic Tapes & Disks
- Full Line of Magazines
- Brain Games & Puzzles
- Workshops & Club Information

Visit THE COMPUTER CORNER for all your computer needs. Stop in and browse – you'll like our personal service.

THE COMPUTER CORNER
White Plains Mall – Upper Level
200 Hamilton Avenue
White Plains, New York 10601
Tel: (914) WHY - DATA

Ample Parking
10-6 Daily & Saturday
10-9 Thursday

Circle 65 on inquiry card.

PRS

TECHNICAL WRITER needed to prepare ***SOFTWARE DOCUMENTATION.*** We also have an opening for an ***Administrative Assistant.*** Would prefer someone in the NYC area. Please apply immediately ***(Part-time free-lance position also available)*** to:

PRS The Program Of The Month Corporation
257 Central Park West
New York, NY 10024
212-787-1526

Circle 309 on inquiry card.

Attractive Durable Plastic

DISKETTE CASE

with each purchase of 10 Diskettes at regular price of $3.65

DEALER INQUIRIES WELCOME

V R DATA CORP.
777 Henderson Blvd.
Folcroft, PA 19032
Call Collect: 215-583-5101

Circle 387 on inquiry card.

WE SHIP FROM STOCK

TELETYPE MODEL 43

Only $985

with RS232. . . . . $1,085

INTERTUBE:
Smart CRT for intelligent shoppers.
Call or write for special low price.

HOLIDAY SALE

ON HAZELTINE TERMINALS

1500 Kit $885
1500 Assembled $995

To order: $10 shipping. 24 hr. shipping upon receipt of certified check or money order. Personal checks: allow 10 days. Credit cards: Add 4%. N.Y. residents add tax.

— We Also Export —

We have no reader inquiry number.
Please call or write.

OWENS ASSOCIATES
147 NORWOOD AVENUE
STATEN ISLAND, N.Y. 10304
Day, weekend, evening calls welcome.
(212) 448-6283 (212) 448-6298

wirewrap

Incredible! but True *Precut wire* *is Cheaper than cutting your own!*

Catalog available on request.

Fast • Reliable • Economical

- No more cutting & stripping by hand
- Good, clean, uniform strip
- Cheaper than using bulk wire

PRECUT WIRE	BULK WIRE
100 pcs of 3" at $.82 = **3 1/4¢/ft.**	50 ft. roll at $1.99 = **4¢/ft.**
100 pcs of 6" at $1.06 = **2¢/ft.**	100 ft. roll at $2.95 = **3¢/ft.**
Wire Kit #1 at $6.95 = **2 1/3¢/ft.**	

#30 Kynar stripped 1" on each end. Lengths are overall
Colors: Red, Blue, Green, Yellow, Black, Orange, White
Wire packaged in plastic bags. Add 25¢/length for tubes.

	100	500	1000	5000
2½ in.	.78	2.40	4.30/K	3.89/K
3 in.	.82	2.60	4.71/K	4.22/K
3½ in.	.86	2.80	5.12/K	4.55/K
4 in.	.90	3.00	5.52/K	4.88/K
4½ in.	.94	3.21	5.93/K	5.21/K
5 in.	.98	3.42	6.34/K	5.52/K
5½ in.	1.02	3.65	6.75/K	5.86/K
6 in.	1.06	3.85	7.16/K	6.19/K
6½ in.	1.15	4.05	7.57/K	6.52/K
7 in.	1.20	4.25	7.98/K	6.85/K
7½ in.	1.25	4.45	8.39/K	7.18/K
8 in.	1.29	4.65	8.80/K	7.53/K
8½ in.	1.32	4.85	9.21/K	7.84/K
9 in.	1.36	5.05	9.62/K	8.17/K
9½ in.	1.40	5.25	10.03/K	8.50/K
10 in.	1.45	5.51	10.44/K	8.83/K
Addl. in.	.10	.41	.82/K	.66/K

WIRE WRAP TOOLS

It's *Like* getting it for *$28.00*

HOBBY WRAP
Model BW 630 with

FREE WIRE KIT#1 ($6.95 Value) **$34.95**

Batteries & Charger	$11.00
WSU 30 Hand Wrap - Unwrap Strip Tool	6.25
WSU 30 M, for Modified Wrap	7.25
BT 30 Extra Bit	2.95

FREE PRECUT WIRE

2.2" x 6.5"
SK 10
$16.50
SOLDERLESS BREADBOARDS
INCLUDING
Over 100 pieces of precut wire in assorted lengths — FREE!

INTERCONNECT CABLES

Ribbon cable connectors for connecting boards to front panels, or board to board.

	SINGLE ENDED			DOUBLE ENDED		
	14 pin	16 pin	24 pin	14 pin	16 pin	24 pin
6"	1.24	1.34	2.05	2.24	2.45	3.37
12"	1.33	1.44	2.24	2.33	2.55	3.92
24"	1.52	1.65	2.63	2.52	2.76	4.31
48"	1.91	2.06	3.40	2.91	3.17	5.08

WIRE KITS

#1 $6.95 (2.2 ¢/ft.)

250	3"	100	4½"
250	3½"	100	5"
100	4"	100	6"

#2 $19.95 (1.9 ¢/ft.)

250	2½"	250	5"
500	3"	100	5½"
500	3½"	250	6"
500	4"	100	6½"
250	4½"	100	7"

1-250 ft. Roll

#3 $23.95 (1.7 ¢/ft.)

500	2½"	500	4½"
500	3"	500	5"
500	3½"	500	5½"
500	4"	500	6"

#4 $42.95 (1.5 ¢/ft.)

1000	2½"	1000	4½"
1000	3"	1000	5"
1000	3½"	1000	5½"
1000	4"	1000	6"

Choose One Color or Assortment

WIRE WRAP SOCKETS

	1-9	10-24	25-99	100-249	250-999	1K-5K
8 pin *	35	.33	.31	.29	.25	.23
14 pin *	.35	.33	.31	.29	.28	.27
16 pin *	.37	.35	.33	.31	.30	.29
18 pin *	.60	.55	.45	.43	.40	.37
20 pin	.84	.78	.71	.63	.59	.54
22 pin *	.90	.85	.82	.78	.70	.60
24 pin	.91	.84	.78	.68	.64	.59
25 pin strip	1.25	1.05	.95	.80	.70	.65
28 pin	.95	.89	.84	.80	.76	.74
40 pin	1.50	1.40	1.30	1.20	1.05	.90

Gold 3-Level Closed Entry Design
* End & Side Stackable All prices include gold
2-Level Sockets Available

EDGE CARD CONNECTOR SALE!

44 Pin	Solder Tail	$1.75	$15./10
100 Pin	Solder Tail	$3.50	$30./10
100 Pin	Wire Wrap	$3.50	$30./10

ALL connectors include Gold.
100 pin connectors are on IMSAI spacing.

PAGE DIGITAL ELECTRONICS

135 E. Chestnut St. #5 Monrovia, CA 91016 (213) 357-5005

ORDERING INFORMATION

- Orders under $25 and COD's, add $2
- All others, shipped Ppd in U.S. via UPS
- For Blue Label (Air) or 1st Class, add $1
- We accept Visa & Mastercharge
- Most orders shipped same day

Dealer Inquiries Invited

Circle 297 on inquiry card.

Fully Integrated Computing System in a Single Circuit

The VPD-40 is a fully integrated system featuring an 8085 processor, 32 K bytes or 64 K bytes of programmable memory, twin minifloppies, a video display, programmable keyboard, motherboard and serial and parallel IO ports in a flip top cabinet. Supporting software includes a disk operating system, text editor, extended and commercial BASIC, relocatable assembler, linkage editor, debugging program, floppy disk system diagnostic program and ANSI level two FORTRAN IV.

Expansion capability is available with the VDP-40 optional double density disk controller. Up to two minifloppy drives and four floppy drives can be supported. Since the VDP-40 can support two optional disk controllers, total disk expansion capacity approaches five megabytes.

The 24 line by 80 character video display features insert and delete, user-programmable character set, protected fields, inverse video and a 14 MHz bandwidth.

Text can be inserted or deleted by character or line. Protected fields help the user safeguard material being entered, or already entered, from being altered when in certain modes, such as insert or edit. Variable format allows creation of large characters for viewing at long distances.

The VDP-40 is priced under $4500 and is available from IMSAI Manufacturing Corp, 14860 Wicks Blvd, San Leandro CA 94577.■

Circle 590 on inquiry card.

Microprocessor Business Computing System

The Abacus 1 is a complete hardware and software package designed to handle basic accounting for small businesses. It includes a Z-80 processor, dual North Star floppy disk system, video display, keyboard and printer, plus software.

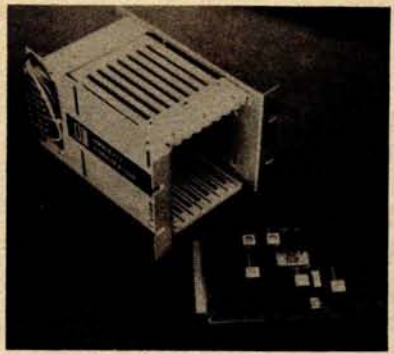
Functions performed by the Abacus 1 include general ledger accounting, accounts receivable, accounts payable, inventory, payroll, mailing lists, data entry, sorting and file management. A character oriented word processing system is available as an option.

The unit features an interactive, double entry bookkeeping system in which receivables decrease book inventory, payables increase book inventory, and general ledger accounts are updated automatically with extensive and valid accounting controls. 50 programs are included in the BASIC software package, with 120 pages of documentation.

Prices for the Abacus 1 start at $5995. For further information contact Computer Products of America, 633 W Katella Av, Orange CA 92667.■

Circle 592 on inquiry card.

Single Board Computer from Omnibyte



This new single board computer contains a processor, memory and IO on a single 4.5 by 6.5 inch (11.4 by 16.5 cm) card. The Model OB8001 also includes serial communications interface meeting both the 20 mA current loop and RS-232C standards. The microcomputer using the 6800 processor can be used to implement a wide variety of stand alone controllers. The processor module is available without chassis as a stand alone computing system with standard card edge connectors.

Included on the board are: the MC6800 processor, a 1 MHz crystal controlled clock, 1 K bytes of programmable memory, sockets for 2 K or 4 K of read only memory, serial interface with selectable data transmission rate, an MC6821 peripheral interface adapter (PIA) that provides two bytes of programmable binary IO along with four programmable control bits, fully buffered address, data and control lines for off board expansion, full decoding for eight pages of off board IO addressing and a separate 128 bytes of programmable memory to be used for scratchpad memory. This card is compatible with the Omnibyte family of memory and IO cards. For further information contact Omnibyte Corp, 2711B Curtiss St, Downers Grove IL 60515.■

Circle 591 on inquiry card.

Attention Readers, and Vendors. . .

Where Do New Product Items Come From?

The information printed in the new products pages of BYTE is obtained from "new product" or "press release" copy sent by the promoters of new products. If in our judgment the neat new whiz-bang gizmo or save the world software package is of interest to the personal computing experimenters and homebrewers who read BYTE, we print the information in some form. We openly solicit such information from manufacturers and suppliers to this marketplace. The information is printed more or less as a first in first out queue, subject to occasional priority modifications.

VISIBLE OR INFRA RED

USED FOR CHARACTER RECOGNITION FOR COMPUTERS WITH EXTERNAL CIRCUITS

MAY BE USED IN A VACUUM, UNDER WATER, HIGH ALTITUDE

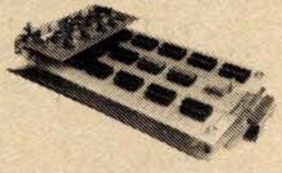
IN MAGNETIC ENVIRONMENT BECAUSE THERE IS NO HIGH VOLTAGE OR MAGNETIC DEFLECTION

MINATURE SOLID STATE

202 VIDEO CAMERA KIT

FEATURING A . . . 100 × 100 BIT SELF SCANNING CHARGED COUPLED DEVICE

THIS UNIQUE UPDATED CAMERA KIT FEATURES THE FAIRCHILD CCD 202C IMAGE SENSOR



ADVANTAGES

- **IN THE FUTURE WE WILL SUPPLY A COMPUTER VIDEO INTERFACE CARD**
- All clock voltages operate at 6V reguiring no adjustments
- Higher video output signal
- We supply the power board, so only a 5V 1 Amp power source is needed
- The circuitry has been simplified for easier assembly
- Two level TTL output is supplied for interfacing

FEATURES

- Sensitive to infra red as well as visible light
- May be used for IR surveillance with an IR light source
- Excellent for standard surveillance work, because of light weight and small size
- All components mounted on parallel 3¾" × 6½" single sided boards
- Total weight under 1 lb.

We supply all semiconductors, boards, data sheets, diagrams, resistors and capacitors, and 8MM lens. Sorry we do not supply the case, batteries and 5V supply.

$349.00 KIT

Add $75.00 to assemble and test

Add $2.00 Postage and Handling

UNIVERSAL 4Kx8 MEMORY BOARD KIT
$69.95
32-2102-1 fully buffered, 16 address lines, on board decoding for any 4 of 64 pages, standard 44 pin buss, may be used with F-8 & KIM

EXPANDABLE F8 CPU BOARD KIT
$99.00
featuring Fairbug PSU, 1K of static ram, RS 232 interface, documentation, 64 BYTE registers

C/MOS (DIODE CLAMPED)

4001 –	.18	4019 –	.37	4049 –	.35	74C73 –	.65
4002 –	.18	4020 –	.90	4050 –	.35	74C74 –	.45
4006 –	.95	4021 –	.90	4053 –	1.10	74C83 –	1.15
4007 –	.18	4022 –	.90	4055 –	1.25	74C86 –	.40
4009 –	.37	4023 –	.18	4066 –	.70	74C93 –	.75
4010 –	.37	4024 –	.75	4071 –	.18	74C151	1.40
4011 –	.18	4025 –	.18	4076 –	.97	74C160	1.05
4012 –	.18	4027 –	.37	4520	.70	74C161	1.05
4013 –	.29	4028 –	.80	74C00 –	.22	74C174	1.05
4014 –	.75	4029 –	.95	74C02 –	.22	74C175	1.05
4015 –	.75	4030 –	.33	74C04 –	.24	74C193	1.20
4016 –	.29	4035 –	.97	74C08 –	.22	74C901	.48
4017 –	.90	4042 –	.65	74C10 –	.27	74C902	.48
4018 –	.90	4046 –	1.35	74C42 –	.85	74C914	1.70

2708 8K EPROM (450 ns)	$ 9.25
2716 (TI)	$22.50
2522 STATIC SHIFT REG	$ 1.95
2513 CHARACTER GEN	$ 6.75
2518 HEX 32 BIT SR	$ 7.20
2102-1 (450ns)	$ .99
21LO2-1 (450ns)	$ 1.25
MM5270 4K X1 DYN	$ 3.45
MK 4008P	$ 1.95
2101-1 256 x 4 STATIC	$ 2.45
2111-1 256 x 4 STATIC	$ 3.45
2112-1 256 x 4 STATIC	$ 2.75
2114 4K STATIC RAM (450ns)	$ 8.50
4116	$18.95
5280/2107B 4K DYNAMIC RAM	$ 3.40
TMS 4050L	$ 3.95
5204 4K PROM	$ 4.95
82523	$ 1.95
825123	$ 3.25
AY-5-1013 UART	$ 5.25
TR1602B	$ 4.95
8703C TELEDYNE 8 BIT A/D CONVERT	$ 9.95
8080A	7.75
8224	$ 3.95
8228	$ 4.95
AMD 9214DC 512 x 8 ROM (450 NS)	$5.75

CRYSTALS $3.45 ea.
4.000 MHz
5.000 MHz
6.000 MHz
8.000 MHz
10.000 MHz

RIBBON CABLE FLAT (COLOR CODED) #30 WIRE
26 cond. - .50/per foot
40 cond. - .75/per foot
50 cond. - .90/per foot

CTS 206-8 eight position dip switch $1.60
CTS-206-4 four position dip switch $1.45
LIGHT ACTIVATED SCR's TO 18, 200V 1A . . . . $ .70

SILICON SOLAR CELLS
2¼" diameter .4V at 500 ma $4.00

LED READOUTS
FND 359 C.C. .4" $ .50
FCS 8024 4 digit C.C. 8" display $5.95
FND 503 C.C. .5" $ .85
FND 510 C.A. .5" $ .85
DL-704 .3" C.C. $ .85
DL-704 C.A. .3" $ .75
DL 747 C.A. .6" $1.75
FND 803 C.C. .8" $1.95
FND 810 C.A. .8" $1.95

PRINTED CIRCUIT BOARD
4-1/2" × 6-1/2" SINGLE SIDED EPOXY BOARD 1/16" thick, unetched
$.60 ea. . . . . 5/$2.60

7 WATT LD-65 LASER DIODE IR $8.95

2N 3820 P FET	$ .45
2N 5457 N FET	$ .45
2N2646 UJT	$ .45
ER 900 TRIGGER DIODES	4/$1.00
2N 6028 PROG. UJT	$ .65

MINIATURE MULTI-TURN TRIM POTS
100, 1K, 2K, 5K, 10K, 20K, 50K, 100K, 200K, 500K, 1Meg, 2Meg, $.75 each 3/$2.00

CHARGED COUPLE DEVICES
CCD 201C 100x100 Image Sensor $95.00
CCD 202C 100x100 Image Sensor $145.00

VERIPAX PC BOARD $4.00
This board is a 1/16" single sided paper epoxy board, 4½" × 6½" DRILLED and ETCHED which will hold up to 21 single 14 pin IC's or 8/16 or LSI DIP IC's with busses for power supply connector.

FP 100 PHOTO TRANS . . . . $ .50
RED, YELLOW, GREEN or AMBER LARGE LED's .2" . . . . . 6/$1.00
TIL-118 OPTO-ISOLATOR . . . . $ .75
MOLEX PINS . . . . 100/$1.00
1000/$8.00

1 WATT ZENERS: 3.3, 4.7, 5.6, 9.1, 10, 12, 15, 18, or 22V . . . . 6/$1.00
MC6860 MODEM CHIP $9.95
MCM 6571A 7 x 9 character gen . . . $10.75

Silicon Power Rectifiers

PRV	1A	3A	12A	50A	125A	240A
100	.06	.14	.30	.80	3.70	5.00
200	.07	.20	.35	1.15	4.25	6.50
400	.09	.25	.50	1.40	6.50	9.50
600	.11	.30	.70	1.80	8.50	12.50
800	.15	.35	.90	2.30	10.50	16.50
1000	.20	.45	1.10	2.75	12.50	20.00

SAD 1024-a REDICON 1024 stage analog "Bucket Brigade" shift register. $18.95

IN 4148 (IN914) . . . . 15/$1.00

RS232 CONNECTORS
DB 25P male $2.25
DB 25S female $2.95
HOODS $1.00

REGULATORS
309K . . . . . . $ .95
723 . . . . $ .50
LM 376 . . . . . $ .60
320T-12, 15 or 24V . . $1.25
340K-12,15 or 24V . . . . $ .95
340T-5, 6, 8, 12 15,18 or 24V $ .95
78 MG $1.35
79 MG $1.35

TRANSISTOR SPECIALS

2N6233-NPN SWITCHING POWER	$ 1.95
MRF-8004 a CB RF Transistor NPN	$ .75
2N3772 NPN Si TO 3	$ 1.00
2N1546 PNP GE TO-3	$ .75
2N4908 PNP Si TO 3	$ 1.00
2N6056 NPN Si TO 3 Darlington	$ 1.70
2N5086 PNP Si TO-92	4/$ 1.00
2N3137 NPN Si RF	$ .55
2N3919 NPN Si TO 3 RF	$ 1.50
2N1420 NPN Si TO 5	3/$ 1.00
2N3767 NPN Si TO-66	$ .70
2N2222 NPN Si TO-18	5/$ 1.00
2N3055 NPN Si TO-3	$ .50
2N3904 NPN Si TO 92	6/$ 1.00
2N3906 PNP Si TO 92	6/$ 1.00
2N5296 NPN Si TO-220	$ .50
2N6109 PNP Si TO-220	$ .55
2N3638 PNP Si TO 5	5/$ 1.00
MPSA 13 NPN Si	4/$ 1.00

TTL IC SERIES

		7445–	.65	74151–	.61
7400–	.13	7446–	.68	74153–	.61
7401–	.13	7447–	.58	74154–	.94
7402–	.13	7448–	.68	74155–	.58
7403–	.13	7450–	.15	74157–	.55
7404–	.15	7472–	.25	74161–	.55
7405–	.13	7473–	.28	74163–	.55
7406–	.16	7474–	.28	74164–	.85
7407–	.20	7475–	.45	74165–	.95
7408–	.18	7476–	.30	74170–	1.68
7409–	.18	7480–	.31	74173–	1.20
7410–	.13	7483–	.65	74174–	.95
7411–	.18	7485–	.87	74175–	.85
7412–	.13	7486–	.28	74176–	.75
7413–	.36	7489–	1.25	74177–	.75
7414–	.60	7490–	.42	74180–	.65
7416–	.22	7491–	.58	74181–	1.90
7417–	.25	7492–	.43	74190–	1.00
7420–	.13	7493–	.43	74191–	1.00
7425–	.25	7494–	.67	74192–	.79
7426–	.22	7495–	.65	74193–	.79
7427–	.19	7496–	.65	74194–	.80
7430–	.13	74107–	.28	74195–	.50
7432–	.22	74121–	.29	74196–	.86
7437–	.21	74122–	.38	74279–	.55
7438–	.21	74123–	.45	74367–	.50
7440–	.13	74125–	.40	75325–	1.50
7441–	.70	74126–	.40	75491–	.50
7442–	.37	74150–	.94	75492–	.50

DATA CASSETTES 1/2 HR $ .95

22/44 Pin Solder Tail .156" Conn. $1.95

MM 5387AA new clock chip which will directly drive LED's 12/24 hrs., 1 supply & alarm $5.95

NO. 30 WIRE WRAP WIRE SINGLE STRAND 100' $1.40

ALCO MINIATURE TOGGLE SWITCHES
MTA 106 SPDT . . . . $ .95
MTA 206 DPDT . . . . $ 1.70
MTA 206 P-DPDT CENTER OFF $ 1.85
MSD 206 P-DPDT CENTER OFF LEVER SWITCH . . . . . . $ 1.85

Full Wave Bridges

PRV	2A	6A	25A
100			1.30
200	.75	1.25	2.00
400	.95	1.50	3.00
600	1.20	1.75	4.00

DIP SOCKETS
8 PIN .17 24 PIN .35
14 PIN .20 28 PIN .40
16 PIN .22 40 PIN .60
18 PIN .25

SANKEN AUDIO POWER AMPS
Si 1010 G 10 WATTS . . . . $ 7.80
Si 1020 G 20 WATTS $15.70
Si 1050 G 50 WATTS $28.50

TANTULUM CAPACITORS

22UF 35V	5/$1.00	6.8UF 35V	4/$1.00
.47UF 35V	5/$1.00	10UF 10V	$ .25
.68UF 35V	5/$1.00	22UF 25V	$ .40
1UF 35V	5/$1.00	15UF 35V	3/$1.00
2.2 UF 20V	5/$1.00	30UF 6V	5/$1.00
3.3UF 20V	4/$1.00	47UF 20V	$ .35
4.7UF 15V	5/$1.00	68 UF 15V	$ .50
		100 UF 10V	$ .40

74LS SERIES

74LS00 –	.19	74LS132 –	.70
74LS02 –	.19	74LS138 –	.70
74LS03 –	.19	74LS139 –	.70
74LS04 –	.21	74LS151 –	.59
74LS05 –	.21	74LS153 –	.59
74LS08 –	.21	74LS155 –	.62
74LS09 –	.21	74LS156 –	.62
74LS10 –	.20	74LS157 –	.59
74LS11 –	.20	74LS160 –	.79
74LS13 –	.40	74LS161 –	.79
74LS14 –	.80	74LS162 –	.79
74LS15 –	.21	74LS163 –	.79
74LS20 –	.21	74LS168 –	.90
74LS21 –	.21	74LS169 –	.90
74LS22 –	.21	74LS170 –	1.40
74LS26 –	.29	74LS173 –	.85
74LS27 –	.25	74LS174 –	.72
74LS28 –	.26	74LS175 –	.75
74LS30 –	.21	74LS181 –	2.00
74LS32 –	.25	74LS190 –	.85
74LS37 –	.28	74LS191 –	.88
74LS38 –	.28	74LS192 –	.88
74LS40 –	.21	74LS193 –	.88
74LS42 –	.60	74LS195 –	.50
74LS47 –	.73	74LS196 –	.75
74LS51 –	.25	74LS197 –	.75
74LS54 –	.25	74LS221 –	1.00
74LS73 –	.36	74LS257 –	.68
74LS74 –	.34	74LS258 –	.68
74LS75 –	.50	74LS265 –	.25
74LS76 –	.35	74LS279 –	.50
74LS85 –	.75	74LS290 –	.80
74LS86 –	.35	74LS293 –	.58
74LS90 –	.50	74LS365 –	.49
74LS92 –	.60	74LS366 –	.49
74LS93 –	.50	74LS367 –	.49
74LS109 –	.25	74LS368 –	.49
74LS112 –	.35	74LS375 –	.60
74LS113 –	.35	74LS386 –	.35
74LS114 –	.35	74LS390 –	1.40
74LS125 –	.45	74LS670 –	2.00
74LS126 –	.45		

LINEAR CIRCUITS

LM 101 –	.75
LM 301/748	.25
LM307 –	.30
LM 308 –	.75
LM 311 –	.75
LM 318 –	1.20
LM 324 –	.70
LM 339 –	1.10
LM 358 –	.70
LM 370 –	1.15
LM 377 –	1.60
LM 380 –	.95
LM 381 –	1.25
LM 382 –	1.25
LM 387 –	1.25
LM 537 –	2.50
LM 553 –	2.50
LM 555 –	.39
LM 556 –	.85
NE540L –	2.25
560 –	2.00
565 –	.95
566 –	1.10
567 –	1.10
703 –	.90
733H –	.75
592H –	.95
709 –	.25
710 –	.35
741C or V –	.25
747 –	.50
LM 1310 –	2.50
1456 –	.95
1458 –	.50
3900 –	.40
8038CC –	3.90
791 –	1.95
LF356H –	1.20

TRIACS / SCR'S

PRV	TRIACS 1A	TRIACS 10A	TRIACS 25A	SCR'S 1.5A	SCR'S 6A	SCR'S 35A
100	.40	.70	1.30	.40	.50	1.20
200	.70	1.10	1.75	.60	.70	1.60
400	1.10	1.60	2.60	1.00	1.20	2.20
600	1.70	2.30	3.60		1.50	3.00

Terms: FOB Cambridge, Mass. Send Check or Money Order. Include Postage, Minimum Order $5.00, COD'S $20.00

Send 25¢ for our catalog featuring Transistors and Rectifiers
145 Hampshire St., Cambridge, Mass.

SOLID STATE SALES
P.O. BOX 74B
SOMERVILLE, MASS. 02143 TEL. (617) 547-7053

WE SHIP OVER 95% OF OUR ORDERS THE DAY WE RECEIVE THEM

Circle 340 on inquiry card.

What's New?

Micro Module Performs Four Functions

The Wintek Counter/Timer module is capable of performing as a frequency counter, event counter and free running timer, or doing period measurement. Applications include remote data logging automatic frequency adjustment, automatic crystal frequencies up to 30 MHz, periods from 2^{-15} to 2^{24} seconds, and elapsed time to 100 days, all with an accuracy of 0.001%. Automatic battery backup is an option. The module is available on a 44 pin 4½ by 6½ inch (11.43 by 16.51 cm) printed circuit board for $149. For further information contact Wintek Corp, 902 N 9th St, Lafayette IN 47904.■

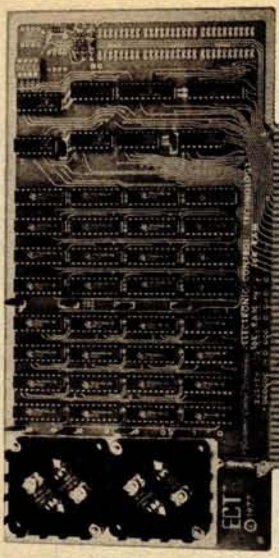
Circle 543 on inquiry card.

16 K Byte Fully Static Memory Board

This 16 K byte fully static S-100 memory board utilizes a 4 K byte fully static memory integrated circuit (TMS-4044). All signals to MOS devices are buffered by low power TTL to prevent damage by static electricity and to minimize capacitive loading on the bus. Low profile sockets are provided for all integrated circuits. 2 MHz operation is standard and 4 MHz is optional at a slightly higher price.

The 16 K byte fully static memory board is $350 in kit form and can be obtained by contacting Electronic Control Technology, 763 Ramsey Av, Hillside NJ 07205.■

Circle 544 on inquiry card.



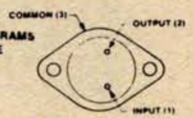
POWER OP AMP
250mA output current capability. Operates on as low as 3V. Input parameters are programmable for system optimizing. Electronic shut down allows output to float. Packaged in 8 pin mini-dip.
LM13080N.......................... $1.94
Specs and applications...... .60

16K DYNAMIC RAM
Replace your 4K dynamics for memory expansion. Super buy on this exact replacement for the 4116.
uPD416 (200nS)............... $20, 8/$144.00
uPD416 (300nS)............... $18, 8/$128.00

21L02-4 (450nS)
Special buy from major producer allows super pricing on these brand new prime memories!
MM2102AN-4L.............$1.44, 64/$73.60

6502 u Processor....................... $10.95

CONNECTION DIAGRAMS
TO-3 PACKAGE
(TOP VIEW)



10 AMP REGULATOR
JUST RELEASED
78P05 is a 3 terminal, 5V, 10 amp regulator as easy to use as the 78H05 and other TO-3 regulators. All the self protection features of the famous 78H05: thermal limit, current limit, low drop out voltage, in TO-3 steel package.
78P05SC......................$12.95
Specs........................... 60¢

TRI-TEK INC.
7808 North 27th Avenue
Phoenix, Arizona 85021

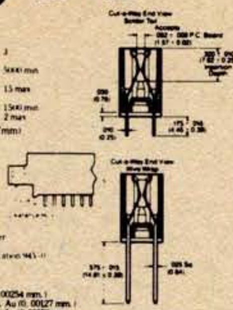
Charge card telephone orders ($20 min.) will be accepted 9-5:30 P.M. except weekends.
Telephone (602) 995-9352. No collect calls please.

GOLD .125" × .250"

CHARACTERISTICS

ELECTRICAL
Current Rating (amps) 3
Insulation Resistance (megohms) 5000 min.
Contact Resistance (milliohms) 15 max
Dielectric Withstanding Voltage (RMS) 1500 min.
Capacitance (picofarads) 2 max
MECHANICAL FOR .062" (1.57mm) BLADE
Insertion Force
12 oz./position max.
(3.36 N/position max.)
Withdrawal Force
2 oz./position min.
(0.56 N/position min.)

MATERIALS
Body Dielectric
Glass-filled thermoplastic polyester
Color black
Meets U. L. Flammability Classification 94S-0
Contacts
CA 725 copper-nickel-tin alloy
Gold inlay
Over min. .000100 in. nickel (0.00254 mm.)
Wire Wrap – min. .000050 in. Au (0.00127 mm.)
Solder Tail – min. .000075 in. Au (0.00190 mm.)
Wire Wrap posts half-hard copper alloy



EDGEBOARD CONNECTORS

Texas Instruments, world leader in metullurgical technology, is introducing its' new improved H43 connector and TRI-TEK is proud to offer it for the first time to our customers. The H43 represents the best value in the industry on this popular connector style.

Pin grid is designed to fit most of the S-100 bus machines such as Imsai, Vector, Cromemco. Will not fit Altair mother boards.

Heavy gold inlay gives you up to seven times the gold in the cirtical contact area at reduced cost. T.I. has the technology and TRI-TEK has T.I.
Solder tail H435121-50 $3.59 10/$32.00
Wire wrap H435111-50 $3.59 10/$32.00

Quantity pricing available

Z80 CPU
Famous single chip uprocessor with all the power of the 8080A plus 80 additional operations. Single +5V supply and single phase clock. Directly interfaces with standard static and dynamic RAMS.
Z80 CPU.............................$20
Z80A CPU........................... $28
Specs..................... $1

Z80CTC
Programmable four channel device that provices counting and timing functions for the Z80 CPU.
Z80CTC........................... $11.00
Z80ACTC.......................... $16.00
Specs.................... $1

Z80 PIO
Parallel interface controller is a two port TTL compatible interface with the CPU.
Z80PIO............................ $11.00
Z80APIO.......................... $16.00
Specs.........................$1

25 Pin Data Connectors
DBC-25P (male) $2.19, 10/19.80
DBC-25S (Female)............ $3.19, 10/30.50
DB-51226 ..Hood....................... $1.39
D20418.....Screwlock assembly........... $1.19

PRECISION VOLTAGE REFERENCE.
AD584 is a precision monolithic IC which has programmable outputs of 10V, 7.5V, 5V and 2.5V @ .3% maximum error! Full -55 to +150° operation.
AD584JH......(in 8 pin TO-5 can)...... $6.95
Spec sheets................... .60

- For premium shipping (first class, special handling, etc.) add extra. Excess will be refunded.
- Please give street address for UPS shipping when possible.
- C.O.D. NO parcel post C.O.D. UPS C.O.D. Add 85¢ to order.
- Any correspondence not connected with your order, please use seperate sheet and include SASE for reply.
- Claims for shortages or damaged material must be made within 10 days of receipt of package.
- Prices are subject to change without notice.
- Any refunds will be by check, not credit vouchers.
- If we should be temporarily out of stock on any item, it will be placed on back order. If we cannot ship in 30 days, you will be notified of the expected shipping date and furnished with a postage paid card with which to cancel your order if desired.
- Terms. Check, money order, credit card. Net 30 days to rated firms, schools and government agencies.
- Claims for defective material must be made within 60 days of receipt of parcel. Claim must include invoice number, date & defective parts.

Shipping and Handling
U.S. and CANADA ADD 5%
Other countries add 15%

Insurance (we're not responsible for uninsured parcels) .50

BECKIAN ENTERPRISES

All Prime Quality — New Parts Only
Satisfaction Guaranteed

EDGE CARD CONNECTORS: GOLD PLATED.
BODY: Non brittle, solvent resistant, high temp, G.E. Valox. The finest you can buy.
CONTACTS: Bifurcated Phos./Bronze; Gold/Nickel.

ALTAIR S-100: Cont./Ctrs. .125" Row Spacing, .140"

50/100 Dip Sold.	$3.95 ea.	5 pcs.	$3.75 ea.
50/100 Sold. Eye.	6.95 ea.	5 pcs.	6.50 ea.

IMSAI S-100: Cont./Ctrs. .125" Row Spacing, .250"

50/100 Dip Sold.	$4.20 ea.	5 pcs.	$3.95 ea.
50/100 W/Wrap 3	3.75 ea.	5 pcs.	3.50 ea.
IMSAI CARD GUIDES:	0.19 ea.	5 pcs.	0.16 ea.

CROMEMCO S-100: Cont./Ctrs. .125" Row Spacing, .250"

50/100 Dip Sold. (Or short W/Wrap)	$6.50 ea.	5 pcs.	$6.00 ea.

OTHER CONNECTORS AVAILABLE

.100" Contact Ctrs., .140" Row Spacing.

22/44 Dip Sold.	$2.30 ea.	5 pcs.	$2.10 ea.
25/50 Sold. Eye.	2.95 ea.	5 pcs.	2.75 ea.
40/80 Sold. Eye.	4.80 ea.	5 pcs.	4.50 ea.
43/86 Dip Sold.	4.90 ea.	5 pcs.	4.70 ea.
43/86 Sold. Eye.	4.90 ea.	5 pcs.	4.70 ea.

.156" Contact Ctrs., .140" Row Spacing.

6/– Sgle. Row (PET)	$1.00 ea.	5 pcs.	$0.90 ea.
22/44 Sold. Eye. (KIM)	1.90 ea.	5 pcs.	1.80 ea.
22/44 Dip Sold. (KIM)	1.90 ea.	5 pcs.	1.80 ea.
43/86 Dip Sold.	4.90 ea.	5 pcs.	4.70 ea.

.156" Contact Ctrs., .200" Row Spacing.

15/30 W/Wrap 3	$1.05 ea.	5 pcs.	$0.95 ea.
22/44 W/Wrap 3	2.30 ea.	5 pcs.	2.10 ea.
36/72 Sold. Eye.	3.45 ea.	5 pcs.	3.30 ea.
36/72 W/Wrap 3	3.85 ea.	5 pcs.	3.70 ea.
43/86 W/Wrap 3	5.50 ea.	5 pcs.	5.00 ea.

POLARIZING KEYS FOR ALL OF THE ABOVE:
Specify: IN Contact or BETWEEN Contact:
1 to 49 pcs. $0.10 ea. 50 pcs./Up $0.08 ea.

SPECIAL
12/24 Pin .156" Cont./Ctrs. .200" Row Spacing.
TIN PLATED CONTACTS.
IDEAL FOR PET INTERFACE & PARALLEL USER PORT.
$1.25 ea. 5 pcs. $1.10 ea.

SUBMINIATURE CONNECTORS: (DB 25 SERIES, RS 232.)

DB 25P Male Plug	$2.50 ea.	5 pcs.	$2.20 ea.
DB 25S Female Socket	3.60 ea.	5 pcs.	3.40 ea.
DB 51212-1 Grey Hood	1.20 ea.	5 pcs.	1.10 ea.
DB 51226-1A Black Hood	1.30 ea.	5 pcs.	1.20 ea.
D 20418-2 Hardware Set	0.75 ea.	5 pcs.	0.70 ea.

SAVE: BUY A SET: (1 DB25P, 1 DB25S, Any Hood.)
1 Set: $6.35 ea. 5 sets: $6.15 ea.
NOTE: For Hardware, (D20418-2) Add $.65/Set.

WHISPER FANS
Excellent for computer cabinet cooling. This is the most quiet fan you will find. Only measures 4 3/4" square by 1½" deep. U. L. Listed.
$21.00 ea. 5 pcs. $19.00 ea.

I. C. SOCKETS. GOLD. WIRE WRAP 3 TURN.
14 pin $0.36 ea.
16 pin 0.38 ea.

I. C. SOCKETS. Dip Solder. Tin.
14 pin $0.15 ea.
16 pin 0.17 ea.

2708 EPROMS PRIME
$14.00 ea.

8080 PRIME
$9.00 ea.

WRITE FOR LARGER QUANTITY DISCOUNTS. DEALER INQUIRIES ARE WELCOME.

WE ARE CONNECTOR (EDGE CARD) SPECIALISTS. IF YOU DO NOT SEE WHAT YOU NEED IN THIS ADVERTISEMENT, PLEASE WRITE US. WE WILL REPLY.

TERMS: Minimum Order $10.00: Add $1.25 for handling and shipping. All orders over $25.00 in USA and Canada: WE PAY THE SHIPPING.
NOTE: CA residents please add 6% sales tax.
NO C.O.D. SHIPMENTS OR ORDERS ACCEPTED.

MAIL ORDERS TO: ***Beckian Enterprises***
P.O. Box 3089
Simi Valley, CA 93063

PERIPHERALS

RS-232 Serial Interface for Anderson Jacobson 841 IO ASCII Terminal

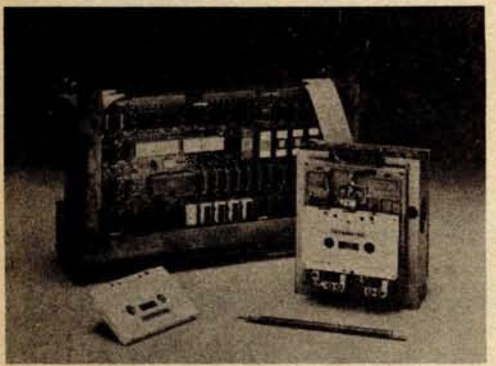
Anderson Jacobson has announced an RS-232 serial interface for their AJ841 IO terminal. This option includes such features as selectable data transmission rates of 110 to 1200 bps; an 896 character receive and print buffer and X-on/X-off control characters to and from the computer.

The AJ841 IO is also available with a parallel interface which can be connected to any S-100 bus microcomputer. Based on the IBM 745 heavy duty terminal mechanism and electronics by Anderson Jacobson, the IO can be used on line to a computer or off line as a typewriter. Both parallel and serial uints are available with a choice of EBCD or correspondence keyboards.

The AJ841 IO is priced at $995. For further information contact Anderson Jacobson Inc, 521 Charcot Av, San Jose CA 95131.■

Circle 618 on inquiry card.

SBC Compatible Cassette Interface



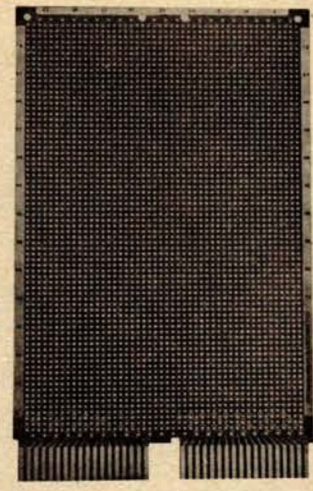
This new digital interface board, which is compatible with Intel's SBC 80 computer line, is being offered by Interdyne Company, 14761 Califa St, Van Nuys CA 91411. The interface board, Model 1B 4100, plugs directly into the SBC chassis and has IO connectors for two Interdyne IC 2500 series cassette drives. Driver software is provided in read only memory which plugs directly into the SBC 80 board. The driver software does the data block formatting, error detection, and reads from and writes to memory at 9600 bps.

Only this card and a cassette drive are needed to add storage capacity of ½ M bytes per cassette, a data transfer rate of 1 K bytes per second and fast memory access.

The IB 4100 board costs $250 including driver read only memory.■

Circle 619 on inquiry card.

Prototyping Board



This new general purpose prototyping circuit board permits construction of custom interface circuits for Heath H11 microcomputers and Digital Equipment Corp LSI-11, PDP-8 and PDP-11 minicomputers.

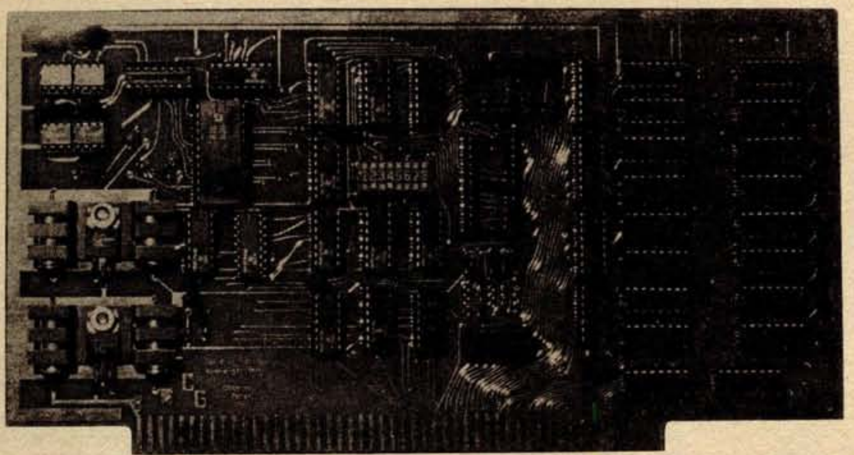
Form, size and connector compatible with the DEC Double Height, Extended-Length module, the Model 4607 Plugbord is 8.43 by 5.187 by 0.062 inches (21.41 by 13.17 by 0.06 cm). It has etched contacts spaced to fit the dual 36 pin connectors used in DEC and Heath computers. Contact terminations are labeled with DEC nomenclature.

To allow unrestricted component placement, the 4607 Plugbord is bare with an array of 0.042 inch (0.11 cm) diameter holes on 0.1 inch (0.25 cm) centers. Dual-in-line package sockets or discrete components may be placed anywhere on the board. Row and column markings are etched into edge strips to insure permanent marking.

In quantities of one to four, the 4607 Plugbord is priced at $15.95 each; $14.36 in quantities from five to nine; and $12.76 in quantities over 10. For further information, please contact Vector Electronic Company Inc, 12460 Gladstone Av, Sylmar CA 91342.■

Circle 620 on inquiry card.

High Speed Programmable Character Generator Board for Z-80



The Programmable Character Generator (PCG) is a high speed version of Objective Design's dense graphics add-on board for S-100 systems. This S-100 card can be used with the Processor Tech VDM or SOL, PolyMorphic Systems VTI, Solid State Music video board and other video boards using the Motorola family of 9 by 7 matrix generators. The card is made operational by removing the character generator read only memory from the video board and plugging it into a socket on the PCG, the running a 24 pin connector from the PCG to the empty CGR socket on the video board. With the PCG in place, the user can create individual characters, store them in on board programmable memory and access the characters directly from the keyboard. Each character can be created on a maximum 8 by 16 matrix with a resultant screen density as great as 512 by 256. Since characters can be stored in either the PCG programmable memory or regular system memory accessible from the bus, any number of characters can be displayed under software control with a maximum of 128 different characters (either ASCII or programmed) on screen at any one time.

The PCG is $165.95 as a kit and $215.95 assembled. For further information, contact Objective Design Inc, POB 20325, Tallahassee FL 32304.■

Circle 621 on inquiry card.

cybercom BOARDS

MB-1 MK-8 Computer RAM (not S-100), 4KX8, uses 2102 type RAMs, PCBD only $22.00

MB-3 1702A EROM Board, 4KX8, S-100 switchable address and wait cycles, kit less PROMS $58.00

MB-4 Basic 4KX8 ram, uses 2102 type rams S-100 buss. PC board $25.95

MB-6A Basic 8KX8 ram uses 2102 type rams, S-100 buss. KIT 450 NSEC ..... $125. PCBD $24.95

MB-7 16KX8, Static RAM uses µP410 Protection, fully buffered. KIT $299.95

MB-8A 2708 EROM Board, S-100, 8KX8 or 16KX8 kit without PROMS $75.00

MB-9 4KX8 RAM/PROM Board uses 2112 RAMS or 82S129 PROM kit without RAMs or PROMs $72.00

IO-2 S-100 8 bit parallel I/O port, ⅔ of boards is for kludging. Kit ..... $46.00 PCBD $25.95

IO-4 Two serial I/O ports with full handshaking 20/60 ma current loop: Two parallel I/O ports. Kit ..... $130. PCBD $25.95

VB-1B 64 x 16 video board, upper lower case Greek, composite and parallel video with software, S-100. Kit ..... $125.00 PCBD $25.95

Altair Compatible Mother Board, 11 x 11½ x ⅛". Board only ..... $40.00. With 15 connectors $94.95
Extended Board full size. Board only ..... $ 8.95
With connector $12.95

SP-1 Synthesizer Board S-100
PCBD ..... $39.95 KIT $135.95

82S23	$1.50	PRIME DEVICES	
82S123	1.50		
82S126	1.95	8080A	$9.95
82S129	1.95	8212	3.25
82S130	3.00	8214	6.50
82S131	3.00	8216	3.75
MMI6330	1.50	8224	3.49
4N26	.75	8228	6.50
4N27	.75	8251	9.95
4N28	.75	8255	9.95
LM323	2.95	21L14 (450 NSEC)	7.25
4116 (apple ram)	16.00	21L14 (250 NSEC)	7.99

WMC inc. WAMECO INC.

MEM-1 8KX8 fully buffered, S-100, uses 2102 type rams. PCBD $24.95
QM-12 MOTHER BOARD, 12 slot, terminated, S-100 board only $34.95
CPU-1 8080A Processor board S-100 with 8 level vector interrupt PCBD $25.95
RTC-1 Realtime clock board. Two independent interrupts. Software programmable. PCBD $25.95
EPM-1 1702A 4K Eprom card PCBD $25.95

EPM-2 2708/2716 16K/32K EPROM CARD PCBD $24.95

QM-9 MOTHER BOARD. Short Version of QM-12. 9 Slots PCBD $27.95

MEM-2 16K x 8 Fully Buffered 2114 Board PCBD $25.95

2102AL-2 Prime 250 NSEC			$1.70
2102AL-4 Prime 450 NSEC			$1.30
2708 Prime (National)			$8.95
2501B	$1.50	1488N	$1.50
2502B	1.50	1489N	1.25
2504	1.50	MC4044	2.25
2507V	1.50	8038	3.90
2510A	1.50	5320	5.95
2517V	1.50	5554	1.90
2518B	1.50	5555	2.50
2519B	1.50	5556	2.50
2521	1.50	5055	1.25
2522	1.50	5312	4.00
2525	1.50	MH0025	1.50
2527	1.50	MH0026	1.75
2532V	1.50	MH0028	1.90
2529	2.75	5262	.50
2533V	1.95	2101	3.50

MIKOS

419 Portofino Drive
San Carlos, California 94070
Please send for IC, Xistor and Computer parts list

KITS
MIKOS PARTS ASSORTMENT WITH WAMECO AND CYBERCOM PCBDS

NEW LOW PRICE on 16K BOARDS
MEM-2 with MIKOS #7 16K ram with L2114 450 NSEC $249.95
MEM-2 with MIKOS #13 16K ram with L2114 250 NSEC $279.95
MEM-1 with MIKOS #1 450 NSEC 8K RAM $125.00
CPU-1 with MIKOS #2 8080A CPU 89.95
MEM-1 with MIKOS #3 250 NSEC 8K RAM 155.00
QM-12 with MIKOS #4 13 slot mother board 79.95
RTC-1 with MIKOS #5 real time clock 60.95
VB-1B with MIKOS #6 video board less molex connectors 89.95
EMP-1 with MIKOS #10 4K 1702 less EPROMS 49.95
EPM-2 with MIKOS #11 16-32K EPROMS less EPROMS 49.95
QM-9 with MIKOS #12 9 slot mother board 67.95

MIKOS PARTS ASSORTMENTS ARE ALL FACTORY PRIME PARTS. KITS INCLUDE ALL PARTS LISTED AS REQUIRED FOR THE COMPLETE KIT LESS PARTS LISTED. ALL SOCKETS INCLUDED.

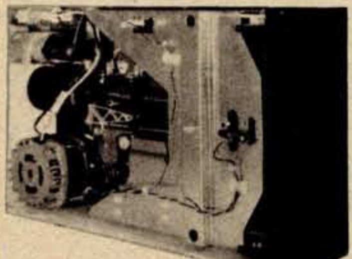
VISA or MASTERCHARGE. Send account number, interbank number, expiration date and sign your order. Approx. postage will be added. Check or money order with order will be sent post paid in U.S. If you are not a regular customer, please use charge, cashier's check or postal money order, otherwise there will be a two-week delay for checks to clear. Calif. residents add 6% tax. Money back 30 day guarantee. We cannot accept returned IC's that have been soldered to. Prices subject to change without notice. **$10 minimum order. $1.50 service charge on orders less than $10.00.**

Electrolabs — PO Box 6721 Stanford, CA. 94305

415·321·5601
TLX 345567

✱ TI Sockets 1 cent per pin. All low profile solder tail 8 pin – 40 pin.

The "Pro" fully encoded ASCII KEYBOARD by Cherry. Auto REPEAT feature, 5 special function keys. 300mA/5V. *(Shown as mounted in 'The Case', Below)* **$119.00, 3/99.00, 10+/89.00**



THE FANTASTIC!

MEMOREX FIVE-FIFTY

* *Hard and Soft Sectoring*
* *Single and Dual Density*
* *Double side configuration as a retrofit at any time.*
* *110/220V, 50/60Hz*
* *Pin for pin compatable with Shugart 800,801,850,851 (50 pin edge connector)*

1/$499, 2/475, 5/450, 10/425, 25/399, 100/375

Double Sided Retrofit **$300**

USED SYLVANIA 12" MONITORS
Working: $69.95
Cold Chassis, 25lbs.
With P39 (Green) Tube 109

The Dumb Terminal for Smart People

80X24 with full 128 char. ASCII UC+LC font with all control characters displayed. 300-19,200 baud RS232. 2nd font addressable from keyboard in you-program-it 2708 for APL, Graphics sets, etc. Plug in monitor I/O connector, 110VAC and you are ready. INCLUDES: 'The Case', Cherry Kbd. A used monitor, ESAT 200A, all options except vector addressable cursor and modem. Bulletproof design and construction. Normally $675.00 What you always wanted your ADM3 to be:
SYSTEM"A" $649.00 10/$599.00

NEW P4 (White) Tube 99

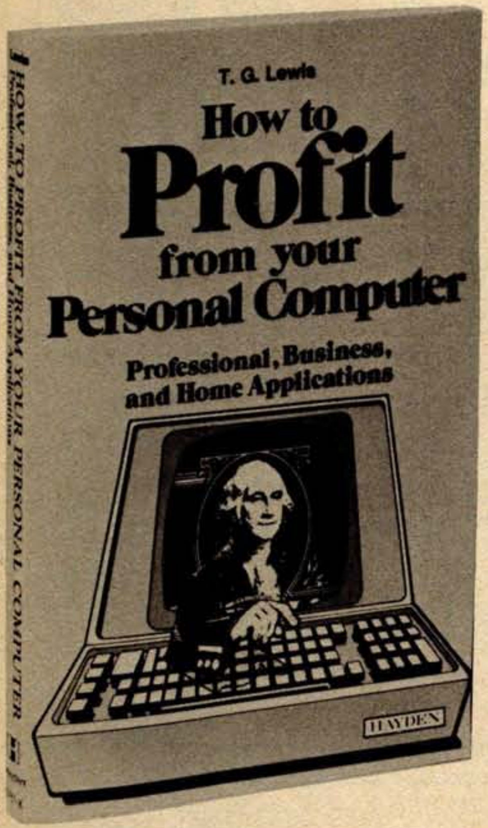
"The Case" Beautiful and sturdy anodized aluminum case in deep black designed to contain the ESAT 200A, and with a bezel cut out for the Cherry 'Pro' keyboard. (installed as shown above) Choose deep brown, light yellow, or crimson to accent or color code your installation. The only choice for hard-use institutional and educational applications. **$69.00, 10/ 59.00**

LATEST REVISION: ESAT 200B (BILINGUAL)
Stand Alone Single Board Communicating Terminal
80X24 Standard Format
Split Speed 110 to 19,200 Baud
Two Fonts, 128 Char. ea. in 2708 EPROM
Full Feature Cursor, Page Xmit., Scrolling
5V@3A, 8"X14" Board. Requires only a keyboard and almost any monitor for the best terminal for the money anywhere!
Supplied with U&LC ASCII in one font.
$329.00

✱ *NEW!* **32 K, S-100 Universal Static Store.** Accepts 2114 RAMs or 70 ns, 3625 PROMs paging up to 8 Mby. Board only with manual and paging software $69.00. 32 Kby RAM 450 ns $679.00, 250 ns $789.00. We have software application notes for multi-task multi-user applications utilizing paging feature.

Shipping and Handling: Surface: $0.40/lb. **Air:** $0;75/lb., 1.00 minimum
Cal. Tax: 6.5% **Insurance:** $0.50 per $100.00

Put a Personal Computer to Work for You!



The profitable uses of personal computers are detailed in this 191 page book published by Hayden Book Company Inc, 50 Essex St, Rochelle Park NJ 07662. Simply written, in an entertaining style, *How to Profit From Your Personal Computer* by T G Lewis, is of interest to the small business owner or professional who is now overwhelmed with paperwork and desires a simpler system. It describes the uses of personal computers in common business applications, such as accounting, handling payrolls, managing inventory and sorting mail lists. It will help you in selecting equipment, analyzing a problem for a business application, hobby or educational experiment and translating problem solutions into real computer systems.

It teaches how to configure a system to fit the needs of an application and to implement that system using programming techniques developed by the author. Programs in BASIC and blueprints of each program are included. The book uses terms, notations and techniques commonly used by programmers.

How to Profit From Your Personal Computer will give you a better understanding of the fundamentals of data processing including file structures, programming structures and some knowledge of computer hardware structures. It is priced at $7.95.■

Circle 560 on inquiry card.

Computer Selection and Operation Publication

Management Information Corporation's new publication *How Small Businesses Use Computers* is a 40 page collection of cases. In each study, the reasons that small businesses decide to implement computers, and the operation of their installations are investigated.

The report, according to the company, delves into the factors that influence a company to acquire a computer system, ranging from high labor costs or inability to use service bureaus to increased efficiency. The management of each company's system is stressed, including supervision, advanced training of operators and the maintenance program.

This complete set is being offered for $15. Contact MIC, 140 Barclay Center, Cherry Hill NJ 04034.■

Circle 561 on inquiry card.

A New Reference for Small Computer Users

The Computer Data Directory announces the first edition of its comprehensive catalog for small computer users. You can choose products and services from several hundred computer related firms.

Included in this directory are brand name manufacturers of systems, peripherals, and accessories. In software, companies handling languages, business applications, household control, games and custom programming services are listed. Information is included on where to find books, magazines, newsletters, home study courses, data banks, tools and repair services. Computer stores and clubs are indexed geographically. The directory is available at $4.98, postage included, from The Computer Data Directory, POB 598, Cleveland OH 44107.■

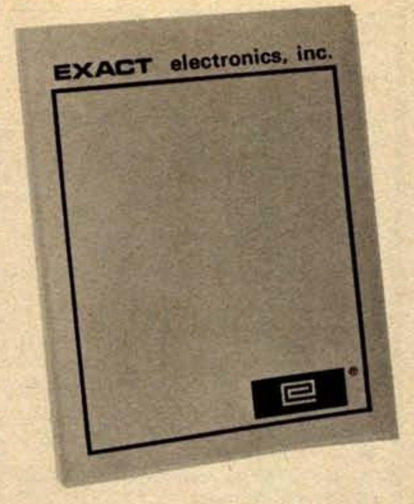
Circle 562 on inquiry card.

Component Catalog

Electronic components, hardware, wire, test equipment and tools are listed with prices in a newly released 88 page catalog (#518) from Mouser Electronics, 11511 Woodside Av, Lakeside CA 92040. This catalog contains over 8000 items including expanded lines of LEDs, semiconductors, switches and keyboards.■

Circle 563 on inquiry card.

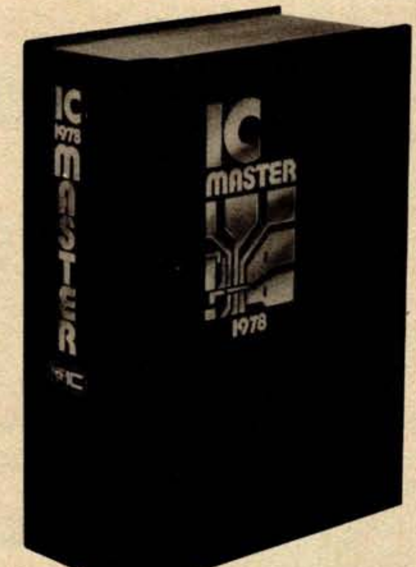
Exact Electronics Instrument Catalog



A new free 66 page 1978 instrument catalog has just been released by Exact Electronics Inc, POB 160, Hillsboro OR 97123. The catalog contains detailed specifications for each function generator, waveform generator and frequency synthesizer in their product line. A quick select short form and comparison chart plus a representative roster make it a desirable addition to any instrument catalog library.■

Circle 558 on inquiry card.

IC Master Lists Integrated Circuits



The *IC Master* lists over 25,000 standard integrated circuit products and over 45,000 alternates. The 2200 page *IC Master* offers a problem and solution approach to integrated circuit selection. The 1978 edition also includes a new section on microprocessor development systems. The price is $55, including quarterly updates. To order the *IC Master*, contact United Technical Publications, 1333 Lawrence Expy, Suite 420, Santa Clara CA 95051.■

Circle 559 on inquiry card.

Unclassified Ads

CONSULTANTS: Let us get together and run a full page color ad with individual addresses and divide the cost. Write Micro Logic Corp, POB 174, Hackensack NJ 07602.

FOR SALE: IBM 024 card punch, with full alphanumeric keyboard, in good condition, fully operational and with all documentation for maintenance, lube, parts and schematic. Can be used to keypunch cards, or readily modified for microprocessor card read and punch. A W Walker, 1914 Ridgewood Dr, San Diego CA 92139, (714) 479-9074.

ONE MICRO, ONE USER?: Why not support several users on your microprocessor with a multitasking operating system. This system is designed for the Motorola 6800, operates in conjunction with an interrupt timer device (for example, SwTPC MP-T) and maintains a time of day clock. Send $5 for complete documentation and source code to start timesharing on your system. Ed Joyce, 4603 Lyceum, San Antonio TX 78229.

FOR SALE: 2102 low power programmable memory, Fairchild 450, $1 each. 8 K, S-100, bare board for above ICs, $20. 7805, $0.75. Erasable programmable memory 2708 National 450 ns, $10. David Karpol, 18–20 Jordan Rd, Fairlawn NJ 07410, (201) 546-9268 days, 791-2044 evenings.

PET OWNERS' CLUB: Now, a great new idea in clubs! Join and you will receive *Program*, a cassette, each month. You LOAD – nothing to type – and it gives you articles, applications, and information right on your screen. What's more, the reverse side will contain three to five games (or more) and programs useful and fun for your sweet PET. Membership is $27 a year. If you wish to join, write to Richard Mansfield, POB 461, Philipsburg PA 16866.

FOR SALE: Processor Tech VDM-1, $150. MITS 88-PIO, $90. ASCII keyboard in walnut case, wired and set up for 88-PIO, $50. All the above in use and working. Webber Hall, 16967 Blanche Pl, Granada Hills CA 91344, (213) 363-2004.

FOR SALE: Eight Motorola 4 K programmable chips from TRS-80. Tested OK, but never used. Make offer by contacting Jerry Dawson, 710 King St, Medford OR 97501, (503) 779-7697.

FOR SALE: TRS-80 Basic I or II: Data Base Manager, $20; Inventory, $20, Stock and Bank Account $15; Lunar, Gunner, Slot and craps $13. Poi Pow, 96 Dothan St, Arlington MA 02174.

FOR SALE: MEK6800D2 System working with SwTPC video terminal III with 69 key Hall-effect keyboard. Includes 24 2102s, two extra 6810 programmable memories, huge 6800 applications manual, data manuals, software manual plus power supply kit. My cost $565. Will sell for $295 or best offer! C Jackson, 1715 Dogwood Dr, Rock Hill SC 29730, (803) 366-2309.

NEW UNCLASSIFIED POLICY

Readers who have equipment, software or other items to buy, sell or swap should send in a clearly typed notice to that effect. To be considered for publication, an advertisement must be clearly noncommercial, typed double spaced on plain white paper, contain 75 words or less, and include complete name and address information.

These notices are free of charge and will be printed one time only on a space available basis. Notices can be accepted from individuals or bona fide computer users clubs only. We can engage in no correspondence on these and your confirmation of placement is appearance in an issue of BYTE.

Please note that it may take three or four months for an ad to appear in the magazine. ■

FOR SALE: Apple II 16 K system only three months old and in perfect condition. This is a complete system. Includes high resolution graphics subroutines, radio frequency modulator, and a cassette recorder/player for input/output (IO). User manuals are also supplied. Entire system for $800. Dana E Ebersole, 441 Burchett # 110, Glendale CA 91203, (213) 244-0796.

FOR SALE: TRS-80 level 1 tapes and program lists. **Star Trek** (runs on 12 K), list $7, tape $10. Lunar Lander (runs on 4 K), list $3, tape $6. Biorythm (runs on 4 K), list $5, tape $9. J R Menzies, 7106 Colgate Dr, Alexandria VA 22307.

PET OWNERS: Here's a very realistic craps program which won't bore you or friends. It's highly interactive and features animated dice, pass or don't pass option, lower case literals, statistical analysis of game with a vertical bar graph and much more. For instance, dice land in random positions, and there's plenty of information regarding rules, size of bank roll and amount of bet. This is a 265 line program using over 5 K bytes. It is on tape, ready to load and go. Send only $7.50 to Richard Liebert, POB 268, Scarborough STA NY 10510.

FOR SALE: Heathkit Computer System – H8 computer, H8-1 memory board (4 K), H8-3 chip set (4 K), H8-5 serial input/output and cassette interface (one or two recorders), H8-2 parallel interface, ECP-3801 cassette recorder, H9 video terminal. Assembled and working, asking $1500. Thomas C Smith, 506 19th Ln, Vero Beach FL 32960, (305) 569-4622 after 6 PM.

FOR SALE: OSI 450 erasable read only memory blaster memory board (still bare), with documentation includes 6820 PIA and two 6834 erasable read only memories. All parts new and unused, $60 Ray Vogt, Box 240A, RD 1, Port Matilda PA 16870.

WANTED: SOL/Processor Technology keyboard, with numerical pad. Roger Wallace, 36 Las Vegas Rd, Orinda CA 94563, (415) 254-1248.

PROM PROGRAMMING: From Binary or Intel paper tape: 1702A ($4), 2708 ($8). From hexadecimal or octal listing: 1702A ($5), 2708 ($16). You supply the programmable memory. Quantity discounts on multiple copies. 48 hour turnaround. David Corbin 11704 Ibsen Dr, Rockville MD 20852, (301) 881-7571 after 6 PM.

FOR SALE: MITS 16 K static memory, disk controller board. Best Offer. F Peck, 19201 NE 18 Av, N Miami FL 33179, (305) 935-2613.

FOR SALE: Radio Shack TRS-80 Microprocessor with 16 K memory, Level 1 BASIC, monitor, cassette, payroll program. First $800 takes it and I pay the freight. Wilbur C Booth, POB 1151, Cordova AK 99574, (907) 424-3140.

FOR SALE: Tarbell Electronics cassette interface. All cables, in sockets, ready to go, $120. Morrows Wunderbuss mother board. Includes ten edge connectors, assembled, $120. Four SwTPC 4 K memory boards, in sockets and burned in, $100 each. Chip Kroll, 4136 Tennyson, Houston TX 77005, (713) 667-6318.

FOR SALE: 128 of each of the following items were recently purchased at a ham festival. Due to incompatibility with my home processor, they are available at my cost, to whomever can use them: AM 9016, 16 K dynamic programmable memory, $22.50; and AM 9050, 4 K dynamic programmable memory, $4.50. You can purchase all or part of 128 set. Documentation is available. R Shay, POB 413, Westmont IL 60559.

FOR SALE: F8 based chess game, with six levels of difficulty. Solves mate in two problems. $100 or best offer. Clete Jackson, 1715 Dogwood Dr, Rock Hill SC 29730, (803) 366-2309.

FOR SALE: Radio Shack TRS-80. Six months old with approximately $80 worth of software. I haven't enough time to learn to use it properly. $450. George Raniolo, 1212 Rt 25A, Stonybrook NY 11790, (516) 751-0101.

COSMAC ELF II: power supply, video modulator, 1802 users manual, all *Popular Electronics* articles. Built and working fine, $95 and shipped prepaid. Frank Dacey, 328 Barbara Dr, Pt Pleasant NJ (201) 899-4302.

FOR SALE: KIM-1 with 4K on board programmable memory neatly packaged in briefcase with custom power supply. Has breadboard area and convenient input/output connectors. Full documentation. Send SASE for photos. First cashiers check or money order for $200 takes it. R Kruse, 6221 Woodlow Dr, Wichita KS 67220.

FOR SALE: Biorythm program, FORTRAN source code, well-tested. Send $1 to cover copy and mailing costs to Jack Walsh, 4 Eaton St, Wakefield MA 01880.

FOR SALE: Assembled and tested ELF II with four 86 pin wire wrap edge card connectors, full documentation; giant board assembled and tested; kluge board for ELF II; 1.5 A power supply; radio frequency modulator and 1852 input/output port. Total value $250 new. Make an offer on whole or part. Also heavy duty Honeywell power supply 10 A at 5 V, 3 A at 16 V, 3 A at −16 V, and 4 A at 28 V. Alan Burnes, c/o ICSR, MIU, Fairfield IA 52556.

FOR SALE: File access software for North Star Disk. By calling routines within this package, a program is given multiple file input/output at the byte level. Entry points allow a file to be opened, closed, read, written, created or released. Requires less than 1 K in addition to disk operating system. Send $10 for commented source listing to Bob Burlingame, 3314 N Whitman, Tacoma WA 98406.

FOR SALE: HP-67 and Heath shortwave receiver. HP-67 plus all original accessories and more programs (physics solutions books, satellite tracking, antenna pointing program and more), second battery pack and security cradle. All for $350. Receiver: 0.5 to 31 MHz, AM/SSB/CW. Excellent sensitivity, selectivity, construction (June 1978) and excellent price: $70. Gary Altman, 1525 Linden Dr, Madison WI 53706, (608) 836-3400 or 262-3035.

FOR SALE: Assembled and tested MITS Altair 680b kit. All Chips socketed. May be interfaced to RS232. $250 plus shipping. John Rheinstein, 10 Gould Rd, Lexington MA 02173.

FOR SALE: TDL Z-80 processor (S-100 Bus Z-80 Card) assembled and tested. TDL Z16 memory board with 8 K memory installed also assembled and tested. TDL software: 2 K Zapple monitor, 9 K macro-assembler, 3 K text editor, 8 K BASIC. Software includes paper tape in relocatable format and all manuals. $500 for all the above. ZPU board only and software, $325. Charlie Shields, POB 809, Carrabelle FL 32322, (904) 697-2190.

TRS-80 SORT: in BASIC I, needs 4 K minimum Sorts recs in memory in 1 or 2 numeric fields in ascending or descending sequence. Output to tape or video. Can be subroutine. $10 on tape with documentation. Software, POB 6153, Syracuse NY 13217.

FOR SALE: SwTPC 6800 computer systems, 8 K memory, serial control interface, parallel interface, CT-64 terminal, SD Sales cassette interface, Pixie-Verter. All assembled, up and running, complete with documentation. Also, bare 4 K memory board, bare MP-A2 processory board. Rick Drapala, 2701 Av A, Yuma AZ 85364, (602) 726-6896 after 4 PM MST.

FOR SALE: TI-Programmer, with case and power supply. Almost new, under warrantee. In original carton, $30. John Burns, 25th floor, 350 Park Av, New York 10022.

CLUB PROGRAM EXCHANGE: Lincoln School Computer Club will exchange PET Computer programs. Send SASE for information and return postage with any cassette. Thanks. Lincoln Computer Club, Lincoln School, 750 E Yosemite Av, Manteca CA 95336.

FOR SALE: Sphere 6800-based microprocessor board including 4 K programmable memory, and 1 K read only memory with the PDS utility routines, monitor, editor, re-editor and assembler. Also an excellent power supply rated +12 V, −12 V, +5 V, −5 V and ground. Also a video display interface board which still has a few bugs. I haven't the technical knowhow to get the system running so I am selling. First check for $300 takes it, inquiries welcome. Nathan Engle, RR 2 POB 466A, New Albany IN 47150, (812) 923-8066.

FOR SALE: Univac type 0769 incremental printer 30 characters per second, 132 column, 63 characters. New unit still in Univac packing carton. With optional drive electronics (original cost $1800+). Also includes optional case, enclosure (hinged top), and stands. Does not include 6 V, −12 V, 24 V 1 A power supplies or system controller, but includes schematics and interface manual. Plus three cases of paper ($100+ alone). All for $575 plus freight. David Krivoshik, 18 Newcomb Pl, Elizabeth NJ 07202.

FOR SALE: Peripheral Vision Floppy disk drive, interface card, cable, documentation and diskette. In good working condition. Includes operating system on floppy. Everything needed to work. $500 or best offer. Will also trade for working TTY terminal or equivalent. Pierre duPont, Patterns, Rockland DE 19732.

FOR SALE: RCA Cosmac vip video computer with games and graphics. Professionally built and in perfect condition. Unit interfaces directly to video monitor. Also stores programs on cassette tape. Originally $275, will sell for $175 and ship anywhere free. Richard Parry, 38 W 255 Deerpath Rd, Batavia IL 60510, (312) 879-8987.

FOR SALE: 8080A system; Morrow's processor and cassette IO boards, VDM-1, 2–4 K programmable memory boards, ASCII keyboard and octal keypad, 20 A supply, attractive cabinets. Tiny Basic up and running with programs on tapes, three extra 100 pin connectors, Processor Technology BASIC 5 and TREK 80 tapes. Full documentation. Cost over $850 — will sell best offer over $600. M Chepko, 119 Belleville Ct, Thief River Fls MN 56701.

FOR SALE OR TRADE: Quantity of 66 American Microsystems AMS-7280 4 K by 1 dynamic programmable memory chips, prime. Equivalent to 2107B, TMS4060, etc. All 66 in carriers for $195 or will trade for 24 TMS4044 or exact equivalent. Watson R Gabriel Jr, 5936 Timberwood TRL, Kernersville NC 27284, (919) 993-3110.

FOR SALE: TRS-80 software. Four full-length simulations for 4 K level I machine: includes Pioneer, Survival, Capitalists and Space Voyage. All 4 for $10. Also, baseball, football, hockey, Lander, all with graphics. Seven in all for $8, or get all 11 for $15. For programs, an updated list or free-lance programming, write Michael T Flanagan, 130 Holly, Riveredge OH 44135.

WANTED: Used minicomputer system; TRS-80, Apple 2, PET, IMSAI, etc. Also hard or soft copy terminal. Mike Gleicher, 11 Leslie Ct, Springfield NJ 07081.

CHEAP MEMORY: Low power 4 K static full speed tested and reliable $75. G Mitchell, POB 35, Chula Vista CA 92012.

WANTED: X-Y Recorder. Offer combinations of the following in trade: SBE Keycom 1000 Scanning 40 channel CB, Pace 8015 and 8008 CBs, Dynaco PAT-4 stereo preamplifier, AR-XA turntable, Palomar TX-100 linearm amplifier, etc. Write Pete, POB 399, Sunnymead CA 92388.

BYTE NUMBER 1: I have one September 1975 BYTE (issue number 1). The magazine is still in its original wrapper and is in mint condition. First certified check for $50 takes it! Call to confirm that it hasn't been sold yet. Mitch Wolrich, 8 Bruce CIR, Randolph MA 02368, (617) 963-5578.

FOR SALE: Secret Word program for 4 K PET. Many hours of entertainment trying to guess one of 200 5 letter words. Can be played alone or with other participant. Features include selection of secret word by computer at random, scoring by number of tries, cumulative scoring, ability to list guess words and hits and misses, also ability to disclose secret word at any point in the guessing. Also other interesting features. On cassette, ready to load and run, $8.50. Henry E Nass, 68 East 56th St, New York NY 10022.

FOR SALE OR SWAP: Voltage Monitor for single phase computer power supply. Monitors under and over voltage, detects high speed transients — 2 ms spike response. Has clock, audible alarm, warning light and 32 day chart recorder. Warns of out of tolerance conditions. Airoyal model 11721. Will accept ham radio equipment in trade. Ed Hayden, 16 Decatur Av, Annapolis MD 21403, (301) 261-2679.

FOR SALE: Assembled H9 video terminal, $450. I went to hard copy so you can save. Includes shipping and manuals. Send cashier's check or money order to Steve Vickery POB 1548, Panama City, Fl 32401.

FOR SALE: Heath H8, H9, 16 K memory, assembled, up and running, tape recorder, all documentation. $1200. David Marcus, 430 Wolf Hill Rd, Dix Hills NY 11746, (516) 427-1926.

FREE PET COMPUTER PROGRAMS: Will trade original versions of Space War or Graphic Gomoku (9 by 9 tic-tac-toe) for your software creations of comparable size. Send programs on cassette for exchange, or send SASE for current list of programs. Please indicate serial number of PET used. Bennett Meyer, POB 575, White Plains NY 10602.

MOTOROLA D2 USERS: A new manual of experiments on 6800 Microcomputers containing 83 interesting assembly language problems. Step by step approach to learn to use the 6800 instructions, digital input/output, interrupts and many programming techniques. $5. Send check to K K Rao, Department of Physics, Western Michigan University, Kalamazoo MI 49008.

FOR SALE: Heath H-8, H-9, 16 K, cassette, and Benton Harbor BASIC. Up and running. $1250. Also ten new 8 inch disks, $25. A Thornburg RR 2, Thompsonville IL 62890, (618) 627-2166.

PET OWNERS: I have 18 game programs for the PET computer. I will trade one for one for other PET programs. Those wishing to trade should send their cassette with programs to Kurt Huebner, 1266 Valle Vista, Fullerton CA 92631.

FOR SALE: Apple II Software cassette: author title index program for books, records, tapes, super fast machine language, sort, 16 K blackjack, graphic, paddle input, sound, rules, autoplay by computer, $10 each includes documentation. Both for $15. George W Lee, 18803 S Christina Av, Cerritos CA 90701.

TRS-80 OWNERS: Original software including business system for Amway Distributors. Level I and II versions available. All will run in 4 K memory. Also have moving sign board machine language program. Write for information Allan E Sitter, 8 Driftwood CIR, Groton CT 06340.

FOR SALE: DECwriter II, EIA interface, caps lock, used two years moderately, on DEC service contract but never needed a call. $975 or best offer, you pay shipping. David Lewis, 1108 N Aurora St, Ithaca NY 14850, (607) 274-3107 weekdays, 273-9144 home.

STAMP COLLECTORS: TRS-80 owners who are also serious stamp collectors and are interested in exchanging ideas and software for philatelic applications are urged to contact me immediately. Collectors with other systems are also welcome to write or call if they are interested. Jeff Purser, 25 Newtown Rd, Danbury CT 06810, (203) 744-7631.

FOR SALE: Reference/maintenance manual, logic charts, and operator's manual for Itel Model 852 automatic typewriter and accompanying Selectric manual. $15. S K Otto, 150 Old Country Rd, Mineola NY 11501.

TRS-80 PROGRAMMING CONTEST: Win $500. Send SASE to TRS-80 Programming Contests-A, POB 621, Fenton MO 63026.

APPLE-II SOFTWARE: Available to swap. Send a list of your programs and SASE for my list (over 100 programs). Most in color graphics. Ed Avelar, 2850 Jennifer Dr, Castro Valley CA 94546, (415) 538-2431 any day, 6 PM to 11 PM.

DESPERATE: College student must sell at tremendous loss to pay for tuition: Tychon Assembler, IMSAI 8080 with 22 slots, etc, 8 K static memory with write protect 8 K static memory without write protect, key board with interface and enclosure, Northstar Disk drive microsystem, Polymorphic 64 character/line display with graphics (new parts — never tested), manuals, etc. Must sell: $1150. Will sell separately. John Fox, #105, 1110 W Stoughton, Urbana IL 61801 (217) 384-4474.

FOR SALE: Hewlett-Packard 97 programmable, printing calculator, $475, with box of thermal paper rolls, new magnetic cards, and all other standard accessories. Clean appearance and in excellent operating condition — a great buy! L S Reich, 3 Wessman Dr, W Orange NJ 07052, (201) 736-2843.

FOR SALE: HEATH H-8 system. 24 K memory, video terminal, GE cassette recorder, interface card and all cables. All completely assembled and working. Software includes Extended BASIC, editor, assembler, debugger, panel monitor (read only memory), plus games package and more. $1800 plus shipping. Walt Sully, 18 Palmer Rd, Kendall Park NJ 08824, (201) 490-2976.

FOR SALE: Hewlett-Packard 140A Scope, 1402-A Dual trace amplifier, 1421A time base and delay generator. One set full documentation. Excellent condition. Original cost over $2000. Will sell for around $600. J E Terry, 818 Briarwood Lake, St Louis MO 63367, (800) 325-4325, ext 225, 8 AM to 5 PM daily.

TRS-80 ASSEMBLER USERS: Z-80 disassembler available. Displays the contents of your memory as symbolic instructions with operands and ASCII code, where relevant. Zilog mnemonics used throughout. Also shows memory addresses and contents in hexadecimal. Requires Level-I BASIC with 16 K and assembler, or Level-II BASIC. (Specify which you have.) Execution tape and instructions, $20. For information only send SASE. Hubert Howe, 14 Lexington Rd, New City NY 10956.

TRS-80 USERS: I have business programs for all levels of TRS-80s. You describe your application and I will give you an estimated cost within 48 hours. Include SASE with your requirements and TRS-80 configuration. Also, I have game programs. Sandy Sigal, 6851 Mammoth Av, Van Nuys CA 91405.

FOR SALE: Three sets of 16 K memory for TRS-80 Radio Shack computer system. 100 percent good. $175 each. Larry Stanley, Rt 5, Box 131B, Circleville OH 43113, (614) 474-1741.

COLLEGE INSTRUCTOR: Needs standard payroll and general ledger programs in COBOL or BASIC; limited compensation. POB 355, Newtown PA 18940.

WANTED: 8008 software, Scelbi computer software manuals for the 8008, assembler program, editor program, monitor routines, SCEBAL high level language, bootstrap loader and Galaxie. L W Harrison, 4905 Hollyridge Dr, Raleigh NC 27612.

WANTED: Drawing and pin out data to interface microcomputer to Programatic Flexiwriter. Will pay for data. Bill Fujitsubo, 1506 Sandcastle Dr, Corona del Mar CA 92625.

STUDENT NEEDS HELP: I am a student in desparate need of both information and part sources for the following: image recognition and a 32 X 32 solid state MOS photodiode array, respectively. If there is a Cyclops camera for sale out there — contact me as I require one. Darcy Roberts, 660 Laurier Blvd, Brockville Ontario, CANADA K6V 5X8.

Reader Service

To get further information on the products advertised in BYTE, fill out the reader service card with your name and address. Then circle the appropriate numbers for the advertisers you select from the list. Add a 15 cent stamp to the card, then drop it in the mail. Not only do you gain information, but our advertisers are encouraged to use the marketplace provided by BYTE. This helps us bring you a bigger BYTE.

**Correspond directly with company.*

BOMB—BYTE's Ongoing Monitor Box

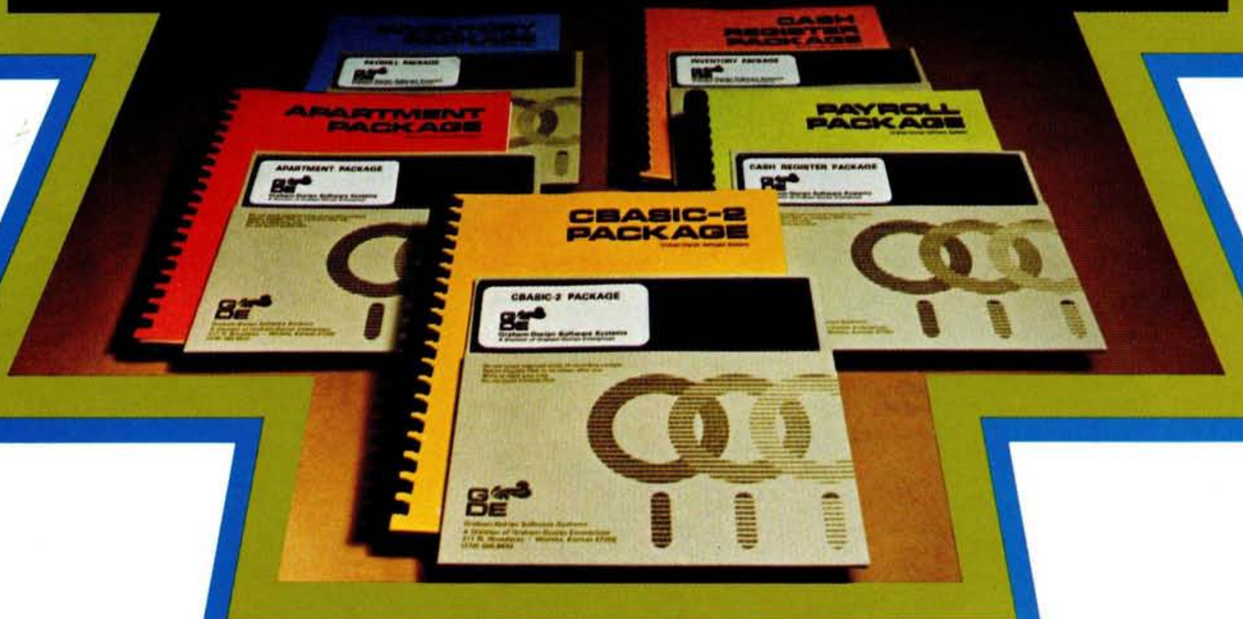
Article No.	ARTICLE	PAGE
1	The Spracklens: An Exchange Evaluator for Computer Chess	16
2	Ruckdeschel: Functional Approximations	34
3	Kasser: The Sky's the Limit	48
4	Horton: Distributed Network	62
5	Ciarcia: I've Got You in My Scanner!	76
6	Gerhold: Computer Assisted Instruction on a Microcomputer	90
7	Wilson: Defining a Language: PL/B	100
8	Hill: A Classroom Demonstration	112
9	Bruninga: A Multiuser Data Network	120
10	Christensen-Suess: Hobbyist Computerized Bulletin Board	150
11	Hubin: Heath Microprocessor Training System	158
12	Anderton: A Cassette Interface Switching Box for the TRS-80	160
13	Frey-Atkin: Creating a Chess Player, Part 2	162
14	Chung-Yuen: A "Tiny" Pascal Compiler, Part 3	182

Fingers Do Walking on August BOMB

Steve Ciarcia has won the $100 BOMB first prize again with his article "Let Your Fingers Do the Talking: Add a Noncontact Touch Scanner to Your Video Display," page 156. Second prize of $50 goes to Chip Weems for "Designing Structured Programs," page 143. The articles placed 1.9 and 1.4 standard deviations above the mean, respectively. In third place was "Pascal versus BASIC: An Exercise," page 168, followed by "Pascal: A Structurally Strong Language," page 78.

To cast votes for your favorite articles, use the convenient BOMB card on the opposite page.■

Programs you can use in your business... ready to go now!



Our programs will let you realize the full potential of your hardware.

We developed these programs because we needed them in our businesses, and, try as we might, could not purchase them. They're on-line now, working for us and others around the country.

As users ourselves we know the problems from your perspective — not just as a manufacturer of software. The bugs are out and they're ready now to go to work helping make your life easier, keeping you in better control of your business.

Our first four program packages are: • **Apartment Management • Cash Register • Inventory • Payroll**

Here's a typical program

To give you an idea of the thoroughness of these programs, here's a summary of what the inventory package does for you. Gives a detailed listing of items in inventory and itemizes all goods sold from inventory, including which sales person sold what, when it sold and for how much ... recaps on one sheet this same inventory activity information ... investigates and changes any information in inventory, on request ... prints list of items to be re-ordered ... provides profit analysis comparing sales personnel and/or various products. And it can be inter-connected with our cash register package as well, for total program management.

Each of our initial programs is conceived, proven and offered with this same exacting thoroughness and attention to detail.

We stay with you after the sale

We're in this for the long haul and our support program is dedicated to that objective. Registered program owners receive:

• Periodic newsletters which include users' ideas and information exchange, plus tips to owners on further increasing benefits of the package through updated operational flexibility.

• Availability of software technicians to provide immediate answers to questions, via phone or mail.

• Customer rewrites and adaptations available on request, at added cost.

CBASIC-2 free

It takes the world's most powerful commercial basic to run our programs and we deliver it to you free.

Each of our program packages contains a disk with CBASIC-2 Compiler, CBASIC-2 Run Command and your Graham-Dorian software programs in INT and BAS file form. You also receive User's Manuals and Hard Copy Source Listing. At a price which pays for itself!

CBASIC-2 was developed and written by Software Systems, the people who wrote CBASIC, and includes many powerful enhancements.* All systems are compatible with any Z-80 or 8080 CP/M™ system. They are deliverable in standard eight-inch disk — either double or single density — or mini-floppy disk.

Give us a call or fill out the Reader Service Card in this issue. We promise a response within 24 hours of receipt. That's the kind of information service we expect, and know you do too.

*CBASIC-2 may be purchased separately from Graham-Dorian Software Systems for $89.95.

Master Charge and Via cards accepted

GDE

Graham-Dorian Software Systems
A Division of Graham-Dorian Enterprises
211 N. Broadway / Wichita, Ks. 67202 / (316) 265-8633

Circle 153 on inquiry card.

"My 8 to 5 minifloppy™ now works nights and weekends."

"I own a fast-growing business and before I bought my computer system I put in a lot of late hours keeping up with my accounting and inventory control. Now the computer does my number crunching quickly, so I have time after hours to have some fun with the system. My son and I started out playing Star Trek on the system, and now we're learning to play chess.

"When I was shopping around for my system, the guys in the computer stores demonstrated all the unique features of the minifloppy. I've got to admit that at first I didn't really understand all the technical details. But now that I use the system every day, I really appreciate the minifloppy's fast random access and data transfer. I like the reliability, too.

"I'm glad I went with Shugart drives. Look, when you lay out your own money for a system, you want dependable performance and good value. Do what I did. Ask for the system with the minifloppy."

If it isn't Shugart, it isn't minifloppy.

Shugart Associates

435 Oakmead Parkway, Sunnyvale, California 94086

For a list of manufacturers featuring Shugart's minifloppy in their systems, circle reader response number.

Circle 312 on inquiry card.