Mar/Apr 1981 Volume 1, Number 2 $2.50 (USA) £1.20 (UK)

The magazine for Sinclair ZX80 users

SYNC

- Programs:
  - Widget
  - Life
  - Tic Tac Toe
  - Make Music
- Boolean Operations
- Memory Display
- Reverse Video
- 6 Book Reviews
- Searle: What's Coming
- Speed Up Screen Displays

The magazine for Sinclair ZX80 users

SYNC

March/April 1981 Volume 1, Number 2

ARTICLES & INTERVIEWS

2 **Growth Plans and Pains** ..... *Ahl & Searle*
An interview with Nigel Searle

5 **Letters to the Editor** ..... *Readers*
Corrections and tips

6 **Interview With Alfred Milgram** ..... *Ahl & Milgram*
The president of Melbourne House speaks out

24 **Crash Cursor** ..... *Truman*
The saga continues

36 **The Home Computer Market** ..... *Searle*
Crystal ball gazing

40 **Puzzles & Problems** ..... *Townsend*

HARDWARE TIPS

8 **Micro Juice** ..... *Kanter*
Overcoming power line surges

10 **Using the MicroAce with Sinclair Accessories** .....
Changes you can make to the PC board

12 **Video Modifications for the ZX80** .....
Reverse video and direct video

PROGRAMMING TECHNIQUES

13 **A Display File in Machine Code** ..... *Logan*
Speed up your screen displays

16 **Memory Display** ..... *Cromley*
Look into the memory of your ZX80

18 **Truth in Programming** ..... *Lubar*
How to use Boolean operations

20 **Data Statement Simulation** ..... *Turner*
Your ZX80 can look like it has Data statements

PROGRAMS & SOFTWARE

22 **Widget** ..... *McGath*
Simulation of a manufacturing company

27 **Artillery** ..... *Whetstone*
Hit the target with your ZX80 cannon

28 **Game of Life** ..... *Ahl*
John Conway's classic game

32 **Tic Tac Toe** ..... *Hoffman*
Noughts and Crosses if you prefer

34 **Making Music With the ZX80** ..... *Forsen*
Second best thing to a music synthesizer

REVIEWS & EVALUATIONS

42 **ZX80 Programs** ..... *Stein*

42 **ZX80 Pocket Book** ..... *Stein*

43 **ZX80 Magic Book** ..... *Stein*

44 **Active Display and Breakout** ..... *Sutton*

45 **ZX80 Companion** ..... *Stein*

46 **Disassembled Basic** ..... *Lubar*

NEW PRODUCTS & RESOURCES

47 **Product Preview** ..... *Staff*

48 **Resources** ..... *Staff*

Staff

Publisher/Editor-in-Chief	**David H. Ahl**
Editorial Director	**George Blank**
Managing Editor	**Paul Grosjean**
Associate Editor	**David Lubar**
Secretary	**Elizabeth Magin**
Production Manager	**Laura MacKenzie**
Typesetter	**Jean Ann Vokoun**
Financial Coordinator	**William L. Baumann**
Bookkeeper	**Patricia Kennelly**
Customer Service	**Patricia Brown**
Order Processing	**Joan Swihart**
Circulation	**Suzanne Guppy**
	Frances Miskovich
	Dorothy Staples

Index to Advertisers

Advertiser	*Page*
Basic Computer Games	31
CAI Instruments	Cover 2
Computer Music Record	21
Creative Computing	Cover 3
Creative Computing Press	41
Creative Computing Back Issues	17
Lamo-Lem	17
Linsac	21
LJH Enterprises	21
K. MacDonald	23
Melbourne House	7
MicroAce	11
Outbound Engineering	23
Peripherals Plus	39
Phipps Associates	19
Rose Cassettes	23
Sinclair Research	3
SYNC	33, Cover 4
SYNC T-Shirt	15

Volume 1, Number 2

SYNC is published bi-monthly for $10.00 per year by Creative Computing, 39 E. Hanover Ave., Morris Plains, NJ 07950. Controlled Circulation pending at Allentown, PA.

Postmaster: Send address changes to SYNC, P.O. Box 789-M, Morristown, NJ 07960.

Subscriptions in USA: 6 issues $10; 12 issues $18; 18 issues $24. UK and foreign airmail subscriptions: 6 issues £10; 12 issues £18; 18 issues £25. Call (800) 631-8112 toll-free (in NJ, 201-540-0445) to begin your subscription.

Growth Plans and Pains
An Interview with Nigel Searle

Nigel Searle is Sinclair's man in Boston. He is responsible for operations in the United States and Canada of Sinclair Research Ltd.

Recently Mr. Searle visited Creative Computing and spoke with publisher David Ahl, about the future plans for the American operations of Sinclair.

N.S.: As a general plan I would like to see us explore one new market every six month period. I think that in four or five months just about anybody who is thinking about buying a personal computer (which is to say the reasonably technically oriented people, the same sort of people who have already bought computers) must be aware that there is an option of only spending a couple hundred dollars to get started. Market awareness is very high.

By the middle of 1981 I would like to achieve a similar level of awareness among people who make either purchasing decision or recommendations regarding computers to be used in schools. Simultaneously, we will focus our own mail-order efforts on the home educational aspects of computers in an attempt to reach beyond the technical audience. Any parent who is thinking of buying a computer for his children will know that there is a low-cost entry point, and will not have heard only of Apple and TRS-80.

It is surprising how many teachers there are who read only educational publications. Even though they may teach math or computing, they don't read the computer publications. At this point these teachers are totally unaware of the fact that there is such a thing as the ZX-80, and will remain ignorant unless we do something about that by addressing them through the appropriate channels.

My target for the first half of 1981 is to achieve a level of awareness in the educational market. We would like to find a single distributor who is capable of handling this market with relatively little support from us other than the product. We would share all the knowledge we have gained from our experience in other markets, but we are hoping to achieve a rate of growth that will be just about impossible to do on our own.

The negotiations that we are having at the moment don't look overly promising and I don't want to suggest that anything is going to happen soon. I believe that traditional publishing companies would make the best kind of partners for the educational market because of they are already selling to schools. They have the sales force to do the job. They have the ability to produce and publish the collateral material, software etc. And like most publishers they realize that their future lies not only in ink and paper. The problem is that their sales force is not necessarily experienced in selling hardware. But you must compromise somewhere.

D.A.: What arrangements do you have with Image Software?

N.S.: What we are selling there is our computer plus Image Computer Products' Computer Learning Lab as a single package. Our ads in the non-technical publications next year are going to go very strongly on this theme: selling the benefit of in-home education, rather than selling the product itself as we do now to the technical audience.

It does no good to tell someone who does not know what a computer is, that he needs a computer. You've got to tell him what it will do for him. Then he will understand. If you are advertising in *Popular Electronics*, the readers already know what a computer can do for them; they want to know what this particular product is. But for the wider, non-technical market you have to detail the benefits. I think that in a year most of our mail-order ads, for whatever market—technical or nontechnical—aren't going to look very much like ads for hardware. They will look like ads for software, because all the emphasis will be on applications.

D.A.: That's where we are now with our magazine positioning: it's applications and software. We think that eventually there is a much bigger market for that than for

Learn how to use a computer for $249. And keep the computer.

Introducing the Computer Learning Lab, with the Sinclair ZX80 personal computer.

The Sinclair ZX80 is a powerful, personal computer already in use in tens of thousands of homes and businesses.

Now, for just $249*, the Computer Learning Lab will teach you and your children how to use this complete, expandable computer right in your own home. And when you're done, you get to keep the computer!

The Computer Learning Lab is a self-paced course that teaches the ins and outs of computers. It cuts away jargon and mystique, taking you straight into BASIC—the most common, easy-to-use computer language. And the computer itself does the teaching.

100 LESSONS MAKE LEARNING EASY.

You just take the ZX80 out of the box, connect it to your TV and an ordinary cassette recorder (connectors are provided), and slip in the pre-programmed Learning Lab cassettes. There's nothing extra to purchase.

You'll be working with the computer your very first day!

The cassettes take you through 100 experiments that teach you how to solve problems with the ZX80.

You learn by doing. By actually working with the computer.

And the lessons are designed to be fun and involving. You create your own programs for games, code breaking, interest calculation, and other topics. Then you can apply the principles you've learned to more complex problems.

In fact, you'll be a master of the ZX80. To use in your business, for home budgeting, or just for fun.

We'll also send you a catalog full of ready-to-use programs—from Loan Amortization to Lunar Lander—available for as little as $6.95.

THE ADVANCED DESIGN OF THE ZX80.

The ZX80 is the world's first truly portable computer. It features a touch-sensitive keyboard and a 32-character by 24-line display.

And it performs like a much larger and more expensive computer. Single keystroke entries make typing programs fast and easy. An automatic error detection feature tells you if you make mistakes. And program editing helps you correct them. Yet the complete Computer Learning Lab, including computer, is still several hundred

The Computer Learning Lab is a family learning aid. Children 10 and above will quickly understand the principles of computing—and have fun learning.

dollars less expensive than any comparable computer alone.

The ZX80 is backed by a 30-day money-back guarantee and a 90-day limited warranty with a national service-by-mail facility. Extended service contracts for the ZX80 are available for a minimal charge.

The complete package includes the ZX80, a 128-page guide to computing, a workbook, six program cassettes, and two blank cassettes for storing your own programs.

To order call toll free: 800-543-3000. In Ohio call: 800-582-1364. Ask for operator #508. Phones open 24 hours a day, 7 days a week.

To: SINCLAIR RESEARCH LTD., ONE SINCLAIR PLAZA, NASHUA, NH 03061.

Please send me _____ Computer Learning Lab(s) with the ZX80 personal computer at $249 each (U.S. dollars).

Please send me _____ ZX80 computer(s) only, without the Learning Lab, at $199.95* each.

I enclose check/money order payable to Sinclair Research Ltd. for $_____ (Add $5.00 for shipping.)

*For Conn. deliveries, add sales tax.

Name _____

Address _____

City _____ State _____ Zip _____

Occupation _____ Age _____

Intended use of ZX80: _____

Have you ever used a computer? ☐ Yes ☐ No Do you own another personal computer? ☐ Yes ☐ No

Searle Interview, continued...

the hardware oriented magazine.

From a competitive standpoint, what do you think the impact will be of the new Commodore VIC: a four-color, 5K memory computer with lots of capability for only $300?

N.S.: It is obviously a much keener competitor than anything that exists at the present time. But we're not going to stand still. One of the amazing things to me is that we have sold as many computers as we have, when all we offer is Integer Basic and 1K bytes of RAM. Within a few months, I hope we will have the 16K RAM and the 8K Basic, which will greatly expand our market.

We feel that we are in a strong position to respond to competition. If Commodore were talking about a $200 price point for the VIC, then I would be more worried than I am. I think they will offer more competition than anyone else in the market at the moment. But I don't think we are going to have much difficulty in saying, "Here's our machine. Here's Commodore's. Look at the difference in price and look at the difference in functions, and about the only thing extra you get from Commodore is color."

D.A.: What about some of the newer Japanese entries? Are there any on the horizon that look like competition?

N.S.: The amazing thing to me, again, is that I don't see any signs from anyone other than Commodore that they are interested in the really low end of the business. I don't know why Commodore is the only company that has shown interest. Perhaps this is because it hasn't, in general, been doing very well here in the states. Therefore it has nothing to lose; it's not going to lose high-end sales by selling a low-end machine.

We have proved that a market exists. Our sales are sufficiently high that we have discussed an advertising strategy calling ourselves "The World's Number 1 In Personal Computers." This means that Sinclair has sold more personal computers than any other company—even when you include the models of all the other companies. Frankly, I am not in any hurry to make that claim, nor is Clive Sinclair. We kind of like the fact that nobody else yet seems convinced that a big market exists. The longer they remain unconvinced, the better I like it.

D.A.: You mentioned the 16K memory and 8K Basic. Are they two separate plug-ins?

N.S.: The 8K Basic is a single chip which takes the place of the 4K chip on the main board. You simply remove the cover, take out the 4K chip and plug-in the 8K chip. The 16K memory module plugs in to the connector on the back of the board. It is in a small case about 2 1/2" square.

D.A.: Do you have any projected prices on those two?

N.S.: The 16K RAM will be $99.95. The 8K Basic almost certainly will be $39.95.

D.A.: I know no one likes to talk about pricing, because you never know where things will end up, but if competition heats up substantially, would you meet it with price or with something else?

N.S.: I don't mind talking about it. We buy our components from the same people that everyone else does. If you are engaged in a price war, then you have to be very smart in your design, and you have to be smart in your buying. You've also got to be very careful that you don't build at yesterday's prices and sell at tomorrow's prices. I hope that we at Sinclair have learned something from being in the calculator business and that we will be better equipped to cope with that sort of downward spiral of prices if it turns out that way.

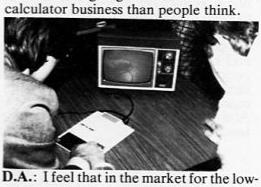
I am probably more pessimistic about that happening than are most people in the business. I don't think the personal computer, once it becomes a truly consumer-oriented item, is going to be very different from the calculator in terms of its marketing. Features will count for almost nothing except at the initial stage. All that anyone will care about is that they get the cheapest computer. Because they are dead scared that they will buy one for $50 and tomorrow a friend will buy one for $35 and they will feel like an idiot. That is all that motivated the drive in calculator prices down to $29.95 and then $19.95.

There eventually emerges a price that people are willing to pay and they want the best you can give them at that price. If they can get everything they want then they don't want to pay more than that critical price. I think we are well equipped to buy components to design the product using our experience with the calculator business. I think we are well equipped to survive that kind of price war and, hopefully, we are smart enough to look ahead to see what kind of product we will need afterward to compete with the Japanese—if indeed it is the Japanese who come along afterward as they did in the calculator business.

The other thing about the calculator business that may be true for the computer business is that there came a time at various stages in the calculator business where you couldn't sell a calculator unless it was of a particular type. I have said that people weren't feature conscious, but they were in a very unselective, media-directed way. There were articles about how to buy a calculator—they would say "do not buy a calculator with disposable batteries, it must have an A/C adaptr and recharger." That remained true for quite a long time until Rockwell brought out the first calculator that ran off a single nine-volt battery. It was far superior to a rechargable battery system.

But because people were told not to buy a throw away battery system because "they will run down in no time at all and will cost you a fortune" people still believed that to be true even when they had a single nine-volt battery unit available to them. Then there was a time you couldn't sell a calculator unless it had a percentage key, or a memory key on it. Despite the fact that of all the people who bought calculators with a memory key only 2.5% used them, no one would buy one that didn't have a percentage key; they read in *Consumer Reports* that you should have one.

It is dangerous to say that the computer business is going to be just like the calculator business. It will be different, and it is probably pointless to make comparisons with other products and other markets. It is, however, going to be more like the calculator business than people think.



D.A.: I feel that in the market for the low-end computer—not just the Sinclair, but the VIC, Mattell, APF etc.—a big mistake is being made in trying to initially sell the broad consumer market. The best market for these products is made up of people who already have a computer—those who have bought an Apple, Altair etc. — and don't want their kids playing with it all the time. Or they want a state-of-the-art unit for themselves.

N.S.: Absolutely. In a sampling of thd people who bought our computer from an ad in *Popular Electronics*, 38% already owned a personal computer. But, obviously, in the long run we do not want to sell exclusively to people who already own a personal computer, because then we can only sell as many units as all other manufacturers put together.

D.A.: My point is that I believe a computer is still a new enough device that someone buying it wants a recommendation from someone else who has one.

N.S.: Perhaps. But there may be another way. You can't convince a large part of the market until they have actually used the machine. Our consumer market advertising within the next couple of months will start to go on the attack. Rather than hide our free trial offer, money-back guarantee in the fine print, we will put it right up front. So much up front that it might well be the headline. We've been kicking around ideas at the advertising agency with lines such as "You can't try the Sinclair for 10 minutes in any store, but you can try it for 10 days in your own home at no cost." Just using the old fashioned mail-order ploy that says "Please, please, take one and try it. If you don't like it send it back. But please try it." We think that trying it will convince the customer to buy it. □

Letters...

Our Face is Red!

Dear Editor:

I recently received my first copy of Sync magazine and I was very pleased. However, I would like to point out a few things I found wrong.

1. Page 18—Nicomacus—Line 100 should read 70*a+21*b+15*c. As listed, this program correctly deduces numbers only from 7 to 104 or between 6 and 105.
2. Page 35—Multiplication Tables—Line 330 needs a semi-colon after A$ to run properly. Line 335 should have a comma instead of a period in the beginning of the print message.
3. Some of the listings require the use of spaces or changes in the print format in order for the messages to print out properly on the tv screen. Also, it would be a good idea to let your readers know if a listing will run in 1K memory. Some of your programs occupy so much memory, they will not run. For example; Page 36—Basic Accounting—As listed I could only enter up to line 2055 before I ran out of memory.

To be sure, the above items are only minor in nature, but I would appreciate hearing from you. All in all, I find the magazine to be an excellent one which I shall continue to subscribe to. Thank You.

John A. Sampson
College Point, NY

Four Tips for MicroAce Owners

Dear Editor:

These suggestions are based on my experience with the MicroAce.

1. Lacquer thinner and an old toothbrush are useful for cleaning solder flux from the PC board.

2. Changing R24 from 1K to 4.7K increases the sensitivity of the cassette input but still holds pin 2 of U11 close enough to ground for an adequate noise margin.

3. The crystal oscillator did not always start, especially when the computer was first plugged in. This problem was eliminated by a 100K resistor from pin 12 of U18 to ground (across C8). This is a leakage resistor which apparently draws off an accumulating charge on pin 12.

4. The keyboard input IC (U11) blew twice from a static charge while I was using the computer on a carpeted floor. This 74LS365 is replaceable by a 74LS367, which Radio Shack carries.

David A. Cromely

Channel 2 Interference Problem Cure

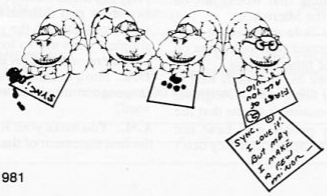
Dear Editor:

Enclosed is my check for a one year subscription to your SYNC magazine. Hope it is as good as your brochure states.

There are two items I'd like to relate about my experience with my ZX80. First, as you know, the computer is tuned to operate on Channel 2. Here in Atlanta, I live close to the station, and I could not display on this channel without a very objectionable amount of interference. I hoped there was some way to cure the problem, so I called the "technical rep" at Sinclair. His solution was-use the computer in the basement-find a room with metal all around-or move. Some help! Some old timers at work recommended I try inserting iron or brass in the tuning coil. By this time I had very carefully removed the cover from the non-adjustable tuner. Although the iron and brass did affect the frequency, it was not enough. A second idea was to very gently spread a few of the windings of the coil. Praise the Lord! It worked like a charm. I was able to to tune it to Chan. 3 with no problem at all. I closed up the unit and I've been on a good clear screen ever since. The second item concerned the cassette recorder. I set the controls, etc. just like the manual said. About 50% of the time I could not load from the cassette recorder to the computer. That is, until I tried the volume control setting at less than the maximum as recommended in the manual. In my case I can leave the setting on 7 and save and load with virtually no failures.

Well, thats it. Hope this info saves someone the frustrations I went through.

R.E. Henneberg



An Interview with Alfred Milgram

Alfred Milgram, president of Melbourne House and David Ahl chat about the ZX80, Sinclair Research, and two new books.

D.A.: Nigel Searle made an interesting comment about marketing kits by mail order. He said that kit builders in the U.K. generally knew what they were doing and there was very little trouble or returns, whereas in the U.S. it is entirely different. People buy a kit expecting it to be like a leathercraft or model car kit that any kid can assemble. Nigel told me of a person who called after buying a kit and asked, "What's a soldering iron?" In the U.S., MicroAce sells a version of the ZX80. While there's not a lot of soldering or assembly to do, people have to know or do have to handle the integrated circuits reasonably carefully. Static electricity is a real enemy, at least until the chip is installed. They also have to know that you can't use the same sort of soldering gun that you use for your plumbing repairs to solder a printed circuit board.

A.M.: How is MicroAce distributing their product here? I don't see them in the magazines, they are not visible.

D.A..: Strictly by mail order. They have been restricted by Sinclair from advertising in a certain group of publications that Sinclair has reserved for itself. Sinclair has more or less picked the top one or two magazines in each individual field and left the balance to MicroAce which is a policy, in my mind, to defeat MicroAce. What is the story on that? What actually happened between Sinclair and MicroAce?

A.M.: The people at CompShop have a history of copying. The guy behind MicroAce is the guy who runs CompShop in the U.K. At one stage they couldn't get enough supplies from Ohio Scientific, so they redesigned the OSI Superboard and got the ROM from Microsoft. Now they are paying the royalties to Microsoft, have redesigned the board and are selling it as the U.K. 101.

Then the Sinclair came out and they figured here was a golden opportunity to save on development costs and come out with effectively the same machine. They copied the ROM, byte for byte, moved the circuits about, cleaned it up a bit, and moved the RAM memory away from the heat sink. The case is black instead of white, and they were going to market it in the U.K. for £ 50 as opposed to £ 70, the price of the ZX80. Sinclair got upset and sued them for breach of copyright. In the case two items were put up for breach of copyright: stealing the ROM and the keyboard. The judge said in effect, "I can't read the ROM. Nobody can read the ROM. The ROM is not copyrightable. But the keyboard: that's the same keyboard, anyone can see that." They settled out of court.

D.A.: So Sinclair then licensed them to do a kit version?

A.M.: They allowed them, because at that stage they had gone quite a ways into manufacturing, and were already committed, with many thousand ROMs and printed circuit boards and all the rest of it. They couldn't afford to pull out of it at that stage. So Sinclair allowed them to market it as long as they didn't sell it in the U.K., only sold it in kit form, and complied with the restrictions on advertising.

D.A.: From what I gather there are people at Sinclair that would like to design peripherals and add-ons that specifically would not fit the MicroAce.

A.M.: One of the reasons they told me that they were not prepared to let the new ROM out, was that they did not want to be ripped-off in the same way. I don't think they can really avoid it in the long run. Because basically anyone who has a MicroAce can go and buy a new ROM from Sinclair and pop it in.

D.A.: Will it fit?

A.M.: Yes, because it is a straight copy. The only thing that MicroAce has done is to add another 1K on board. Apart from that it is exactly the same machine. It has the same routines which all run in the same places, the same operating system, the same number of chips, it is basically the same machine. I can't see how Sinclair can design anything that would not be compatible with the MicroAce.

D.A.: Tell me a little more about your plans. You mentioned two books, one on machine language programming.

A.M.: We've done exceptionally well in the U.K. with our first book, *30 Programs For the ZX-80*, because it seems that the market is mainly people who have not bought a computer before and they don't know what to do with the machine. We are finding that a lot of buyers are people who don't even know how to enter a program. They are are just following blind, letter for letter. But the cost of the machine, in our mind, made it so it wasn't worth marketing cassettes in the U.K. There are a lot of people who are marketing cassettes in the U.K. but they tend to have six or eight programs on one cassette.

There is a subgroup of users who are very interested in going further, and using the machine as a learning tool in developing their own programs. It is for those people that we are doing machine language programming and programming techniques. *Programming Techniques* is fairly basic. It uses the specific capabilities of the Sinclair. You could just buy a book on Basic programming, but the language isn't quite the same because you are working with a subset. So you must know how to cope with certain limitations—not having a step function in loop, for example. The biggest difference is the totally dynamic display screen where the display shrinks and expands as you use the machine. So you just can't poke in and out of the screen or easily move things about. Then there are the people who want to use machine language programming for its user subroutine capabilities. But again you need special advice beacause there is no RAM protected area.

There are ways of entering machine programs and yet being able to save them, because when you save a program on cassette you can save all the variables as well. So you can retain a machine languge program on tape and use it later.

D.A.: How do you save machine languge programs?

A.M.: In our *30 Programs* book, several programs utilize machine language routines. We show two different ways of loading those routines into the program. The two simplest ways are to load it into a REM statement or to load it into a variable.

D.A.: Into a Remark statement? A machine language routine? How can it then be used?

A.M.: You make your Remark statement the first statement of the program. So you

Interview, continued...

know its location. You poke the values into the Remark statement later on. Then, because you know absolutely its location, you can call up the subroutine. If you use a variable to poke it into, you have to remember that the computer is dynamic and every time you call a new variable it's likely to shift in position. So you have to peek into the location which tells you where the variable storage starts.

D.A.: We received one article, in which someone showed how different characters—graphic characters—could be put on the screen to create a moving graphics display. It emphasized that you had to peek to the beginning or look at the beginning of the graphics display in the same statement that you poke because of the dynamic allocation.

A.M.: That's true unless you have defined all of your variables first. However, you must remember that loop variables have a separate definition from your regular variables (because they are stored differently in the machine).

D.A.: Do you think that the 8K Basic will be any more standard?

A.M.: It is a step closer. It has step functions available, trigonometry and other scientific functions, decimal numbers, and string arrays. All of those things mean that the subset is that much closer to Microsoft Basic. What you don't have is the ability to define variables and integers, which I think is very stupid, because you are wasting a lot of memory when you are just using ordinary integer arithmetic.

D.A.: Where do you see Sinclair going in two or three years? Do you see prices coming down, etc?

A.M.: I think Clive Sinclair is going to move out of the computer market. I think that what he should be doing is moving into the small businessman's cheap computer. Because he has a reputation in the U.K. of being able to come out with a product which is technically brillant, but very cheap and very tinny. That's what he does, and he does it very well. If he came out with a computer for £300 which would be very basic—a 16K machine with a one-page word processor, that could handle invoicing for 30 accounts or whatever you can get into 16K, and a very cheap printer for £200, it would sell an immense number of small business machines. But I don't think he sees it this way. I think he is going to move out into other markets.

Sinclair is the first in the personal computer field who has come out with such a low price. But very soon there are going to be others who are going to offer better products for the same price. And he won't be able to compete any more. He has been able to do well because he was first. And he has shown the potential that exists both in terms of technology and in terms of marketing. Just as he wasn't able to keep up in electronic watches or pocket calculators, I don't think he will be able to keep up in the computer field.

D.A.: Word reaches us that the next big thing that Sinclair is working on is "flat screen television."

A.M.: He has been working on that for years and I think that he probably will do it.

D.A.: They claim that the flat screen television is being built and sets will be on the market by this December.

A.M.: I will be surprised if it is that early, because he has always had a problem with delivery. He will probably demonstrate a model by December.

D.A.: Is your machine language programming book going to take people by the hand assuming they don't know anything about programming at all.

A.M.: The people who are involved in the ZX-80 users club in the U.K.—and they know the Basic very well—still don't know how the user function works. They don't understand how you get a machine language program to run. If that is at all typical, and I think that it is, then the people who have now taught themselves some Basic want something different. They bought the computer because they think it is time to learn about computers, and they don't understand the way the machine language in the computer works. The ZX-80 machine language code is a very powerful one but it is also very complex. It is a very useful code; you can do a lot of things with it. Our book will try to put it within the grasp of every reader. □

Micro Juice

Elliott S. Kanter

Owners of the Sinclair ZX80 and Micro-Ace rapidly discover that interruptions of power can cause headaches, not to mention the necessity for re-keying programs. If you live in the "Sunshine" state. Florida, you also have to contend with "Florida-Flicker & Flash," better known as the local utility company. They achieved their reputation by the repeated switching between feeder lines, causing lights to flicker, creating all sorts of problems for computer users.

After re-keying several programs into my MicroAce, the Micro-Juice project was born. Most major computer installations use large and expensive line conditioning transformers, filters, and if the area or data is important enough, a device called a U.P.S. The uninterruptable power supply is illustrated in a simple block diagram (figure 1), and generally consists of a device to convert direct current from a storage battery to alternating current, with a switchover network to allow it to take over if commercial power fails. There are several variations on this basic circuit, but generally the U.P.S. allows operation for up to one hour, depending on current requirements, and the size of your budget. This protection doesn't come cheaply, because U.P.S. systems start at over two hundred dollars.

It makes little sense to spend two-hundred or more dollars for a ZX80 U.P.S., but Micro-Juice serves the purpose for about $10. Fortunately, the ZX80 and the Micro-Ace were well designed for this addition. Tests on my MicroAce showed that the unit drew approximately 330 mA in use, and the d.c. input from the power supply was about 14 Volts unloaded, with the on-board five volt regulator handling the transition to the five volts the ZX80 needs.

All we wanted to do was to insure that the computer and its memory would remain powered during a power flicker or loss. To do this we needed a source of d.c. somewhere between 5 and 15 volts, supplying 330 mA for about an hour. The transition would have to be so quick as to not disturb the memory. Another requirement was that the Micro-Juicer had to be easy to build and duplicate, using easily obtained parts, and represent as good a value as the computer itself. Considering all those requirements, I was both pleased and surprised to be able to come up a circuit meeting all these needs.

Examining the circuit in figure two, we see that Micro-Juice is somewhat similar to the block-diagramed U.P.S. system. Both sense a drop of commercial power and immediately apply an alternate power source. Micro-Juice, however is simpler and costs less, relying on 9 volt alkaline batteries, which provide power for over an hour.

The circuit is mounted in any convenient box, offering the builder the option of bypassing the two input jacks, and simply cutting the power cable from the wall-supply, feeding it directly into the cabinet. If you use this approach, make use of strain relief grommets and take careful note of polarity — the tip is positive, and the ring is negative.

After assembling the circuit and checking both the polarity of the connections for input and ouput power, and the orientation of the diodes, connect Micro-Juice to your ZX80. With the normal a.c. system in operation the cursor will be visible. When

Elliott S. Kanter, 1704 Raleigh Ave., Holly Hill, FL 32017.

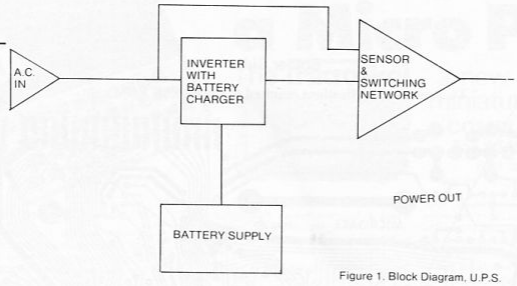


Figure 1. Block Diagram, U.P.S.

you pull the power supply from the wall outlet the 'K' remains.

Micro-Juice is not a portable d.c. supply capable of running the ZX80 indefinitely, but it is a fail safe device to guard your program from accidental power interruptions. Another convenience of this system is that all required parts are available at your local Radio Shack. The parts numbers are listed below. □

Parts List

274-251 — J1, J2 two conductor, open circuit 1/8 inch phone jacks.
274-286 — P1, P2 two conductor, 1/8 inch phone plugs.
276-1101 — D1, D2 diodes, at least 50PIV,1A.
23-553 — B1, B2 or equivilant 9 volt akaline batteries.
270-233 — Cabinet, 5 1/6 X 2 5/8 X 1 5/8 inches.
Solder
Assorted wire

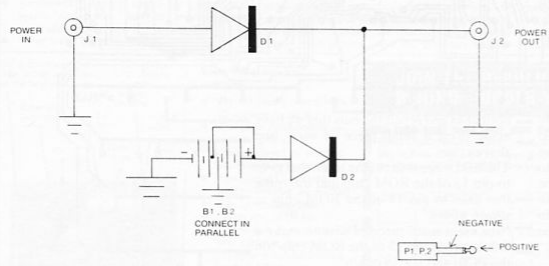
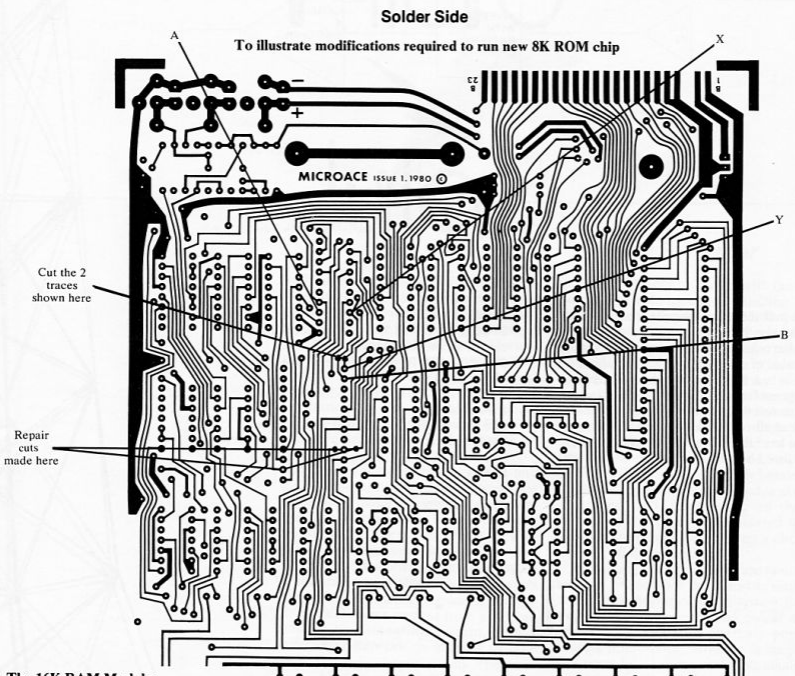


Figure 2 — Micro-Juice

Using the Microace with Sinclair Accessories

1—Repair cuts shown & remove links presently installed.
2—Cut tracks where shown.
3—Make link between A & B.
4—Make link between X & Y.

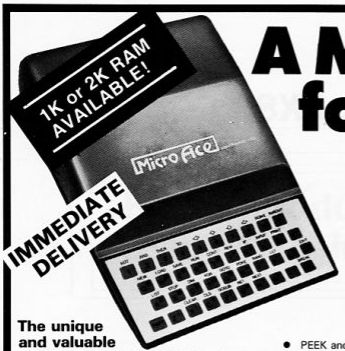


The 16K RAM Module:
The 16K RAM Module is completely compatible with the MicroAce 1K Kit; the unit can be simply plugged on the back. If the RAM Module is to be fitted to a 2K Kit then the extra 1K should be removed along with U17. Please be sure that the 2K2 resistor has been installed in place of R16. The Module may now be plugged onto the back of the machine.

The 8K Basic ROM:
The MicroAce kit comes with two track cuts on board, i.e. both sides of pin 21 of the ROM chip. You also need to make two links on the board.
Before installing the new ROM chip, remove both links and repair the cuts that we have made using a small piece of wire. (see above).
The next stage is to cut the track that goes to pin 13 of the ROM chip and the track that goes to pin 14 of the ROM chip as shown above
Now use a small piece of wire to make a link between pin 13 of the ROM chip (on the PCB) and pin 18 of U6.
Also make a link using another piece of wire between pin 14 of the ROM chip (on the PCB) and pin 3 of U6.
The new ROM chip may now be plugged into its socket and power applied to the board. □

Licenced by Sinclair Research Ltd.

A Microcomputer for everyone at a Micro Price



The MicroAce - a new generation of miniature computers

A COMPLETE COMPUTER for $149.00 for 1K Kit

Post and Packing FREE

(Add 6% Tax for Shipments inside California)

The unique and valuable components of the MicroAce

The MicroAce is not just another personal computer. Quite apart from its exceptionally low price, the MicroAce has two uniquely advanced components: the powerful BASIC interpreter, and the simple teach yourself BASIC manual.

The unique versatile BASIC interpreter offers remarkable programming advantages:

- **Unique 'one-touch' key word entry: the MicroAce eliminates a great deal of tiresome typing. Key words (RUN, PRINT, LIST, etc.) have their own single-key entry.**
- Unique syntax check. Only lines with correct syntax are accepted into programs. A cursor identifies errors immediately. This prevents entry of long and complicated programs with faults only discovered when you try to run them.
- Excellent string-handling capability — takes up to 26 string variables of any length. All strings can undergo all relational tests (e.g. comparison). The MicroAce also has string input — to request a line of text when necessary. Strings do not need to be dimensioned.
- Up to 26 single dimension arrays.
- FOR/NEXT loops nested up 26.
- Variable names of any length.
- BASIC language also handles full Boolean arithmetic, conditional expressions, etc.
- Exceptionally powerful edit facilities, allows modification of existing program lines.
- Randomise function, useful for games and secret codes, as well as more serious applications
- Timer under program control.
- PEEK and POKE enable entry of machine code instructions, USR causes jump to a user's machine language sub-routine.
- High-resolution graphics with 22 standard graphic symbols.
- All characters printable in reverse under program control.
- Lines of unlimited length.

'Excellent value' indeed!

For just $149.00 (excluding handling charge) you get everything you need to build a personal computer at home... PCB, with IC sockets for all ICs; case; leads for direct connection to a cassette recorder and television (black and white or color); everything!

Yet the MicroAce really is a complete, powerful, full-facility computer, matching or surpassing other personal computers at several times the price.

The MicroAce is programmed in BASIC, and you can use it to do quite literally anything, from playing chess to managing a business.

The MicroAce is pleasantly straightforward to assemble, using a fine-tipped soldering iron. It immediately proves what a good job you've done: connect it to your TV ... link it to the mains adaptor ... and you're ready to go.

Fewer chips, compact design, volume production-more power per Dollar!

The MicroAce owes its remarkable low price to its remarkable design: the whole system is packed on to fewer, newer, more powerful and advanced LSI chips. A single SUPER ROM, for instance, contains the BASIC interpreter, the character set, operating system, and monitor. And the MicroAce 1K byte RAM (expandable to 2K on board) is roughly equivalent to 4K bytes in a conventional computer — typically storing 100 lines of BASIC. (Key words occupy only a single byte.)

The display shows 32 characters by 24 lines.

And Benchmark tests show that the MicroAce is faster than all other personal computers.

No other personal computer offers this unique combination of high capability and low price.

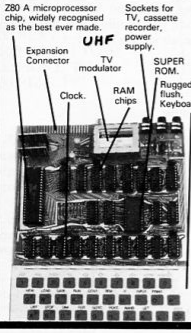
The MicroAce teach-yourself BASIC manual.

If the features of the BASIC interpreter mean little to you-don't worry. They're all explained in the specially-written book *free* with every kit! The book makes learning easy, exciting and enjoyable, and represents a complete course in BASIC programming-from first principles to complex programs. (Available separately-purchase price refunded if you buy a MicroAce later.)

A hardware manual is also included with every kit.

The MicroAce Kit: $149.00 with IK COMPLETE $169.00 with 2K

Demand for the MicroAce is very high: use the coupon to order today for the earliest possible delivery. All orders will be despatched in strict rotation. If you are unsuccessful in constructing your kit, we will repair it for a fee of $20.00, post and packing FREE. Of course, you may return your MicroAce as received within 14 days for a full refund. We want you to be satisfied beyond all doubt — and we have no doubt that you will be.



Your MicroAce kit contains...

- Printed circuit board, with IC sockets for all ICs.
- Complete components set, including all ICs-all manufactured by selected world-leading suppliers.
- New rugged keyboard, touch-sensitive, wipe-clean.
- Ready-moulded case.
- Leads and plugs for connection to domestic TV and cassette recorder. (Programs can be SAVEd and LOADed on to a portable cassette recorder.)
- Mains adaptor of 600 mA at 9VDC nominal unregulated.
- FREE course in BASIC programming and user manual.

JOIN THE REVOLUTION - DON'T GET LEFT BEHIND - ORDER YOUR MICRO ACE NOW!!

Send Check, Money Order or quote your Credit Card No. to:
MicroAce 1348 East Edinger, Santa Ana, California, Zip Code 92705.
or phone (714) 547 2526 quoting your Credit Card Number.

Quantity	Description	Unit Price	TOTAL
	MicroAce Kit 1K	$149.00	
	MicroAce Kit 2K	$169.00	
	Manual	$10.00	
	1K Upgrade Kit	$29.00	
Shipments inside California add 6% TAX		**TOTAL**	

☐ Amex.
☐ Diners
☐ Check
☐ Money Order
☐ Master Charge
☐ Visa

Card No. _____

Exp. Date _____

Name _____

Address _____

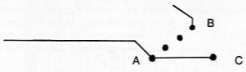
City _____ State _____ Zip _____

Video Modifications for the ZX80

REVERSE VIDEO

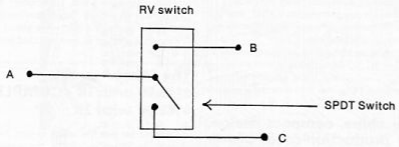
It is possible to directly invert the video signal that leaves the ZX80 and drives your TV. This is done by accessing pin 9/IC9, instead of pin 7/IC9, as the input (pin 4) to IC20. Below are the instructions for the modifications.

Remove the entire case from the ZX80. In the right center section of the back of the PCB you will see an etching that looks like this:



First, cut the trace between points A and C with a razor blade or exacto knife.

Now install the following circuit:

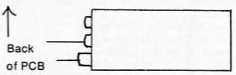


Drill an appropriately sized hole in the case. Mount the switch. Close up the case and power-up the system. You will notice that toggling the switch will cause the video to invert.

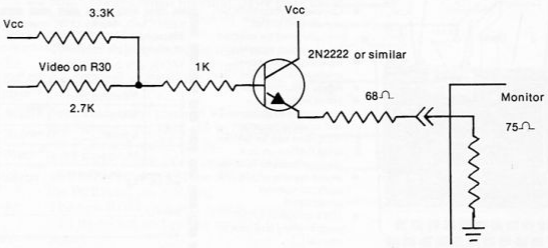
DIRECT VIDEO

It is possible to connect a video monitor directly to the ZX80. This is accomplished by passing the RF modulator and adding a small amount of buffer circuitry. This circuit also reverses the video signal which can be controlled by the RV switch shown earlier.

Open the ZX80. Look at the modulator. (See Below.)



Tapping the video lead on R30 (see Schematic), you should run it through the buffer circuit shown below:



How to Produce a Display File Using Machine Code.

How Is It Done ?

Dr. I.S. Logan

The ZX80 is supplied with a 4K ROM that contains the Basic interpreter. Therefore the average user will use the PRINT command of the Basic language to produce his pictures on the T.V. screen. However Basic is fairly slow.

For example the following Basic program draws a simple rectangular playing area, that could form part of many games, takes over 2 seconds to RUN.

```
10 FOR I=1 TO 32
20 PRINT "■";      (shift A)      } Top line
30 NEXT I
40 FOR J=1 TO 10
50 PRINT "■";
60 FOR I=1 TO 29
70 PRINT "■";      (space)        } The rows
80 NEXT I
90 PRINT "■"
100 NEXT J
110 FOR I=1 TO 32
120 PRINT "■";                    } Bottom line
130 NEXT I
```

Of course the above program could be rewritten using many little tricks and as a result could be quite fast, but it would probably use a lot of valuable memory.

As machine code routines are very fast it is obviously useful to be able to replace the slow parts of Basic programs with machine code routines.

This article goes through the steps involved in producing a Display File under machine code, and will show how the above Basic program can be replaced by a machine code routine called using a statement such as:

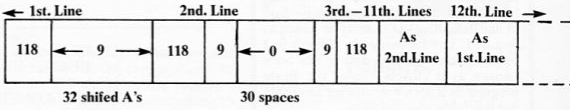
20 LET K=USR(16427)

However the structure of the normal display file must be discussed first.

The Display File: In the ZX80 the display file is the part of the RAM which holds the next picture to be shown. A complete picture is made up of 24 lines which can be from 0 to 32 characters in length. The display file has an initial delimiting "118", and another delimiting "118" to mark the end of each line.

When a Basic program is RUN the display file can initially be considered to be empty and as each PRINT statement is interpreted the Display File is filled with the required characters.

For example the display file produced by the Basic program above will have the following structure:



The part of the display file after the 12th. Line has been left empty in the diagram to show the place where further PRINT statements would add their characters. However if the Basic program is finished (or is stopped) then the end of program RUN routine will complete the display file by adding to the Display File the number of "118"s as specified in the *line counter* system variable 16421), and then adding the required E-LINE. The Display File will be complete as it holds the characters that will form 24 lines.

Once the above details have been understood, then it follows that the machine code routine below must result in a Display File of exactly the same structure.

An Outline Flow Diagram: There are three parts to any machine code routine constructing a Display File. These are shown in the diagram below, and each part will discussed in turn.

Dr. I.S. Logan, 24 Nurses Lane, Skellingthorpe, Lincoln LN6 OTT England.

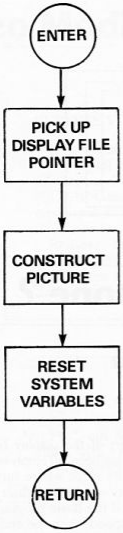


Chart 1

Top	LD B, #32	6,32	(32 characters/row)
Edge	LD (HL), #9	54,9	(the shifted A)
	INC HL	35	(move to next character)
	DJNZ, EDGE	16,251	(loop until row finished)
	LD A, #117	62,117	(the top row delimiter)
	INC A	60	(117 + 1 = 118)
	LD (HL), A	119	(enter delimiter)
	INC HL	35	(move to next character)
Rows	LD B, #10	6, 10	(there are 10 rows)
Line	PUSH BC	197	(save the row number)
	LD (HL), #9	54, 9	(the shifted A)
	INC HL	35	(move to next character)
	LD B, #30	6, 30	(30 spaces/row)
Space	LD (HL), #0	54, 0	(enter a 'space')
	INC HL	35	(move to next character)
	DJNZ, Space	16, 251	(loop for 30 spaces)
	LD (HL), #9	54, 9	(the shifted A)
	INC HL	35	(move to next character)
	LD A, #117	62, 117	(a row delimiter)
	INC A	60	(117 + 1 = 118)
	LD (HL), A	119	(enter delimiter)
	INC HL	35	(move to next character)
	POP BC	193	(collect row number)
	DJNZ, Line	16, 234	(loop for 10 rows)
B-Line	LD B, #32	6, 32	(32 characters/row)
Edge-2	LD (HL), #9	54, 9	(the shifted A)
	INC HL	35	(move to next character)
	DJNZ, Edge-2	16, 251	(loop until row finished)
	LD A, #117	62, 117	(a row delimiter)
	INC A	60	(117 + 1 = 118)
	LD (HL), A	119	(enter delimiter)
	INC HL	35	(move to next character)

Pick up Display File pointer: The Display file pointer is held as System Variables, 16396 & 16397, so this part the program is very straightforward. The HL register pair is loaded with the contents of locations 16396 & 16397, and then incremented to point to the location required for the starting character of the first line.

Construct picture: Following the example of the BASIC program that drew a simple rectangle, a machine code routine must be written to draw a similar rectangle. End of line delimiting "118's must be added as required. The following algorithm is just one solution of many, and it has been chosen as it closely copies the Basic program. It is not a very efficent algorithm but it is fairly easy to follow.

N.B. The decimal machine code instruction "118" has purposely been avoided, as the ZX80 will interpret it as an end of line delimiter if the machine code is stored in a REM statement. (see Chart 1)

Assembler language	**Decimal machine code**
LD HL, (D-FILE)	42, 12, 64
INC HL	35

Reset System Variables: There are three system variables that require to be reset after any picture.
The DF-EA and the DF-END need to be filled with the address of the first free location in the Display File, i.e. the current contents of the HL register pair, and the *line counter* at 16421 must be filled with the value required. (The *row counter* at 16420 may also need to be reset, but this is not required in the present example.)

System	LD (DF-EA), HL	34, 14, 64	
	LD (DF-END), HL	34, 16, 64	
	LD A, #11	62, 11	(rows unused)
	LD (LINE COUNTER), A	50; 37, 64	
	RETURN	201	

Entering The Machine Code Routine: The above machine code routine has 64 instructions and all these instructions must be loaded into the RAM before the routine can be called. There are many ways in which machine code routines can be stored in the ZX80 but the method used below prevents the routine from being overwritten. However never try to LIST the whole program. (well try it. After you have SAVED it)

Now enter the following program.

Machine code loader program:

```
10 REM   64 locations   (do not use spaces!)
20 LET CHECKSUM = 0
30 FOR I = 16427 TO 16490
40 INPUT A
50 LET CHECKSUM =  CHECKSUM + A
60 POKE I, A
70 PRINT PEEK (I),
80 NEXT I
90 PRINT "CHECKSUM = "; CHECKSUM
```

The following lines also need to be entered and afterwards deleted (line number and Newline) so as to scroll line 10 off the screen.

```
100 PRINT
101 PRINT
102 PRINT
103 PRINT
104 PRINT
105 PRINT
106 PRINT
107 PRINT
108 PRINT
109 PRINT
110 PRINT
```

11 Dummy Lines (or more if the above program was changed)

SAVEing the program at this stage is advisable.

Now the Loader Program is ready to be RUN, so enter RUN and NEWLINE.

Enter the decimalised machine code instructions carefully. The values are repeated below in the format that appears on the screen. A CHECKSUM is incorporated for convenience. If a code has been entered wrongly, the correct value can either be entered directly using a POKE command, or the whole of the Loader can be RUN again.

Screen Display:

42	12	64	35
6	32	54	9
35	16	251	62
117	60	119	35
6	10	197	54
9	35	6	30
54	0	35	16
251	54	9	35
62	117	60	119
35	193	16	234
6	32	54	9
35	16	251	62
117	60	119	35
34	14	64	34
16	64	62	11
50	37	64	201

CHECKSUM = 4033

Once the machine code has been entered correctly, SAVE the program again. Now comes the moment of truth! Enter a line containing USR(16427). e.g. LET K=USR(16427) and the rectangle should appear.

If the rectangle does not appear go back to one of the SAVEd versions and find the error.

The best form in which to save the routine is to delete lines 30-90, and to replace line 20 with:

20 REM LINE 10 IS OFF THE SCREEN, USE LET K=USR(16427) TO DRAW A RECTANGLE.

So now SAVE the final version.

A Demonstration Program

The following extract from a mixed BASIC/Machine code Life program for the ZX80 shows the above routine at work.

```
10 REM (prepared as above and held off
the screen)
20 LET A = USR (16427)
30 INPUT A
40 IF A   1 OR A   300 THEN GO TO 30
50 LET A=PEEK(16396)+PEEK(16397)
*256+34+A+((A-1)/30)*3
60 POKE A, -52*(PEEK(A)=0)
70 GO TO 30
```

The above program is used to enter characters into the required positions. The program also contains the facility of removing a character by overwriting the character with a "space".

I hope that the above article will prove useful to many readers, and I would certainly be interested in seeing any programs written as a direct result of reading this article. □

Are you in SYNC?

If not, you should be. We would like any programs, translations of existing programs, games or tips which you have to pass on to fellow Sinclair ZX-80 or Micro-Ace owners. Articles are much more lively if accompained by photos (black and white), diagrams, and illustrations. If you do not have an output printer, please type program listings and carefully check them against the listing on the screen. Sample runs should be included with programs rather than just a description of what the program does. Articles should be typed, double space. Your name and address, with phone number should be on first page; all other pages should be numbered. All submissions should include return postage. Payment ranges from $15 to $40 per printed page.

Please send all submissions to:

SYNC
39 E. Hanover Avenue
Morris Plains, New Jersey 07950

Memory Display

David A. Cromely

The *Memory Display Program* presents 16 Bytes of memory in binary, octal, decimal, and character. You can enter a one-to four-digit hexadecimal starting address, or hit NEWLINE, if you want to continue without a new entry. To end the display, enter Z.

Address E00 (hex) houses the generator portion of the ROM, the RAM is at address 4000 (hex), and the program begins at address 4028 (hex). Lines 60 to 90 convert the hexidecimal address to decimal; lines 110 to 140 call the subroutine to print the address in hex, and lines 180 to 210 trigger the subroutine to print the memory contents in binary, while lines 220 to 240 print in octal, and lines 250 to 270 in hexidecimal. Line 280 prints the contents in decimal with leading zeros, and the character. Lines 310 to 430 convert the decimal number in X to N digits using radix R, and then prints the digits. If the radix is 2 (binary) a black or white square is printed instead of 0 or 1.

David A. Cromely, 5136 Redmond Road, Cheyenne, WY 82001.

```
 10 LET A = - 16
 20 INPUT A$
 30 IF A$="Z" THEN LIST
 40 LET A = A + 16
 50 IF A$ = " " THEN GOTO 100
 60 LET A = 0
 70 LET A = A * 16 + CODE (A$) - 28
 80 LET A$ = TL$ (A$)
 90 IF NOT A$ = " " THEN GOTO 70
100 CLS
110 LET R = 16
120 LET N = 4
130 LET X = A
140 GOSUB 310
150 PRINT A
160 PRINT
170 FOR I = 0 TO 15
180 LET R = 2
190 LET N = 8
200 LET X = PEEK (A+1)
210 GOSUB 310
220 LET R = 8
230 LET N = 3
240 GOSUB 310
250 LET R = 16
260 LET N = 2
270 GOSUB 310
280 PRINT TL$ (STR$ (X + 1000));" "; CHR$ (X)
290 NEXT I
300 GOTO 20
310 LET Y = X
320 LET F = R ** (N - 1)
330 FOR J = 1 TO N
340 LET D = Y/F
350 LET Y = Y - D * F
360 LET F = F/R
370 IF NOT R = 2 THEN GOTO 400
380 IF D = 1 THEN LET D = 128
390 LET D = D - 28
400 PRINT CHR$ (D + 28);
410 NEXT J
420 PRINT " ";
430 RETURN
```

Worth A Fortune

Past issues of Creative Computing. What are they worth today? It varies. To a collector, Vol. 1, No. 1 is worth $7 or $8. To a scrap dealer, less than two cents.

But we're not selling old back issues. We're all out.

On the other hand, you know that much of the content of Creative Computing is timeless. The Depth Charge program in Vol. 1, No. 1 is just as challenging today as the day it was written. Walter Koetke's series of five articles on using computers in the classrom are as valid today as the day they first ppeared in print. And scores of people have written about obtaining reprints of Don Piele's classic problem-solving series.

Our Mistake

In our early growth years when we had 5,000 and then 10,000 subscribers we couldn't imagine we would ever need more than 1000 extra copies for back issue sales. That's about what we printed extra. However, by the time we were going into Volume 3, we found our stocks of Volume 1 issues virtually depleted.

Our Solution

So we selected the best material from Volume 1, edited it, put it together in book form and sold it for $8.95, about the same as the six individual issues. Nine months later, we did the same with Volume 2. Then a year and a half later we did it again with Volume 3.

Most other magazines in a high technology field like small computers find their contents are quickly out of date. However, because we've concentrated on applications and software, our content retains its value for a much longer time. Our subscribers know this and retain their copies of Creative Computing long after they've disposed of the more hardware-oriented magazines.

Now you can obtain the best material from the first three years of Creative Computing in book form and the next three years (minus four issues) in the original magazine form.

Our Offer

We have a unique special offer, so pay close attention to this paragraph. (Computer types ought to be able to understand this). If you order any one item below, you pay the full price. If you order any two items, take a 5% discount from the total; any three, take a 10% discount; any four, take a 15% discount, any five, take a 20% discount, and on all six take a whopping 25% discount from the total price.

Best of Creative Computing-Vol 1	$8.95
Best of Creative Computing-Vol 2	8.95
Best of Creative Computing-Vol 3	8.95
Volume 4 (Four issues)	6.00
Volume 5 (Ten issues)	15.00
Volume 6 (Twelve issues)	18.00

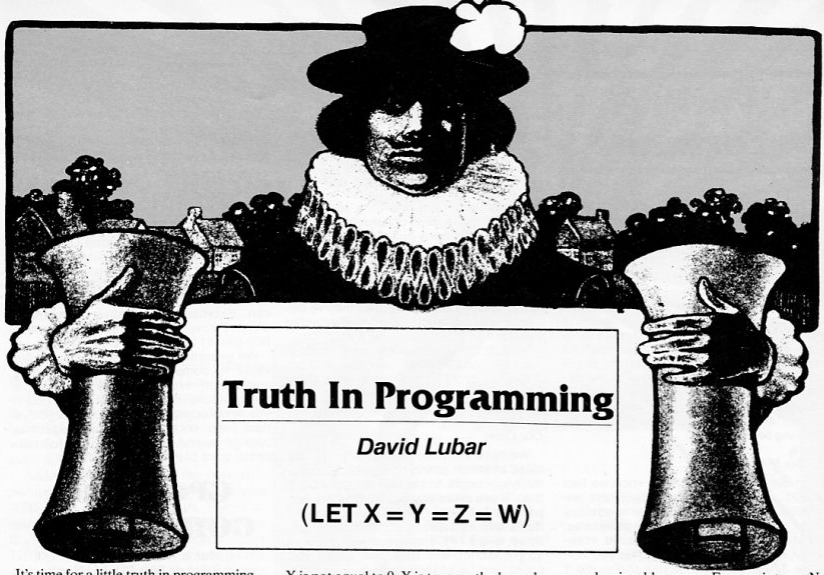
Less discount (5% for two items, 10% for three, 15% for four, 20% for five, 25% for all six) Shipping ($2.00 USA, $5.00 foreign)

This isn't really an order form. Cut it out or copy the items you want onto a sheet of paper. Send it with your name and address and payment or credit card number and expiration date (Visa, MasterCard or American Express) to Creative Computing, Morris Plains, NJ 07950. Or call toll-free to 800-631-8112 (in NJ, 201-540-0445).

We guarantee you'll never find a better value in computer applications reading matter. On average you're getting 128 pages of solid information for each $1.00. If you're not completely satisfied after you've read them, send the books or magazines back to us and we'll refund your full purchase price plus the return postage.

creative computing

Morris Plains, NJ 07950
Toll-free **800-631-8112**
(In NJ 201-540-0445)



Truth In Programming

David Lubar

(LET X = Y = Z = W)

It's time for a little truth in programming, time for a bit of logic. Why let your programs slave away with boring numbers when they can deal with such fascinating concepts as truth and falsity? The logical capabilities of the Sinclair represent one of the best ways to shorten and speed up programs. Many programers don't make full use of these Boolean operators, mainly because, while they are not difficult to use, they are different from other parts of Basic. Boolean logic uses operators such as "AND" and "OR." The major ways to use these operators will be discussed below.

Logical operators deal with the concepts "true" and "false". In the Sinclair, true is represented as -1, false is represented as 0. This is just a convention; some machines use 1 for true, some use 0 for true and 1 for false. Knowing how the Sinclair represents true and false, the next question is, what makes something true or false? In Basic, any expression or variable with a value of 0 is false, any other value is considered true.

There are many ways to use logical operators. Let's start with a simple example. Take the statement 10 IF X > 0 OR X < 0 THEN GO TO 50. This could be replaced with 10 IF NOT X = 0 THEN GO TO 50. But there is an even shorter way to perform the operation. We want to take the branch whenever X is not equal to 0. And, if X is not equal to zero, X is considered to be true. Using this, we can perform the function with the following line: 10 IF X THEN GO TO 50. The IF...THEN statement makes the branch only when the IF part is true. If X is not equal to 0, X is true, so the branch will be taken. Only when X is zero will the branch not be taken. Conversely, the statement 10 IF X = 0 THEN GO TO 50 can be replaced with 10 IF NOT X THEN GO TO 50. You can use this technique to check against numbers other than zero by changing the variable to an expression. For example, IF NOT X = 15 THEN GO TO 100 is equivalent to IF X - 15 THEN GO TO 100. When X is not equal to 15, the expression X - 15 will evaluate as true, and the branch will be taken. What the above example boils down to is this: the IF...THEN statement can be used with any logical operation.

Another important aspect of logical operators is their ability to provide a value that can be used to advantage in a program. Remember that true is represented as -1 and false is represented as 0. Take the following programming problem: a player has to guess whether a number is odd or even. The program must determine whether the guess is correct. The first step would be to get the guess. This could be done with:

```
10 PRINT "ODD OR EVEN?"
20 INPUT G$
```

Now, some way has to be used to compare the guess with the number. This brings up another problem: what is the simplest way to determine whether a number is odd or even? In integer arithmetic, an easy way is provided by the fact that division rounds off a number to the integer remainder. The result of dividing 7 by 2 would be 3. This gives a way to determine whether the number is odd or even. For any integer N, if N/2 = N - N/2, then the number is even. Try this with an odd number, then with an even one. N/2 will only be equal to N - N/2 when N is even. Now we could use lines such as 30 IF N/2 = N - N/2 AND G$ = "EVEN" THEN GO TO 100, where line 100 handles a correct guess. But the full comparison of guesses in this manner seems somewhat long and inelegant. Instead, we can take advantage of the values given to logical operations. It happens that the CODE for E is 42 and the CODE for O is 52. If we can find an expression that produces a value of 42 when the number is even, and 52 when the number is odd, we can compare this value with CODE (G$) to determine whether the guess is correct. This can be done using the fact that (N/2 = N - N/2) is true when N is even and false when N is odd. Since true gives a value of -1 and false a value of 0, the comparison can be done with the following line:

```
30 IF CODE (G$) = 52 (N/2 = N - N/2) *
      10 THEN GO TO 100
```

Since even numbers will cause the expression in the parentheses to be evaluated as -1, the result will be 52 -1* 10 thus matching the CODE for E. If the number is odd, the result will be 52 +0* 10, matching the CODE for O.

This concept bears repeating: airthmetical operators can be used to return logical values. All in all, a very valuable technique. It can be used to adjust scores in games or alter results depending on answers. Since false returns a value of 0, and multiplying any number by zero results in zero, when

you add a logical expression multiplied by a constant to a number, the number only changes if the expression is true.

The equal sign can also be used more than once in a line. This fact caused a bit of confusion when mentioned in the ad for *SYNC* magazine, and deserves a fuller explanation. LET X = Y = Z is a valid expression, but does not assign to both X and Y the value of Z. These variables must be defined first. (To take an analogous case, LET X = X 1 is valid, but will cause an error if X hasn't been defined.) What the expression does is compare Y to Z and give X the logical value of this comparison. If Y equals Z, X will be assigned the value of -1. This fact can be exploited when a program requires several IF...THEN statements that all use the same comparison. For example:

```
10 IF Y = Z THEN LET T = T + 1
20 IF Y = Z THEN PRINT "YOU ARE
       RIGHT"
```

Assume that several more uses of this comparison are made later. These can be replaced with:

```
10 LET X = Y = Z
20 IS X THEN LET T = T + 1
30  IF  X  THEN  PRINT  "YOU  ARE
       RIGHT"
```

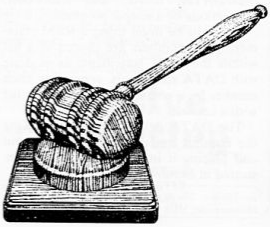
In the above example, no space was saved. But if the program makes repeated use of the comparison, this trick will save space since the expression IF X is shorter than the expression IF Y = Z. Also, the shorter version will execute more quickly since it has less to perform.

While arithmetical operators can be used to produce logical values, there are also certain functions that are used specifically for logical operations, namely AND, OR, and NOT. AND is considered true only when all expressions linked by it are true. OR is true if at least one of the expresions linked by it is true. So the statement IF X AND Y AND Z THEN GO TO 100 will take the branch only if all three variables have a value other than zero. If any one of the variables is zero, the expression will be false. Again, the value produced by this expression can be used in arithmetical operations. The statement LET A = (X AND Y AND Z) * 10 will give A the value -10 if X, Y, and Z are all true. If any of the three are false, A will have a value of 0.

These operators can be combined, producing lines such as IF NOT (A AND B) OR C THEN GO TO 100. While such expressions seem complex at first glance, they become simple when evaluated in sections. The first step would be to look at anything in parenthhses. A AND B by itself is simple. As stated before, this expression is true only if both A and B are true. NOT (A AND B) has the opposite value of (A AND B). So, if (A AND B) is false, NOT (A AND B) will be true. Now, the entire expression can be understood. It is true if either NOT(A AND B) is true or if C is true. To test your understanding of this, determine, before looking at the next paragraph, what logical values of the variables would make the expression false.

Since the major connective in the expression is an OR, the expression will be false only when both sides of the OR are false. So C has to be false. NOT (A AND B) must also be false. NOT (A AND B) is fale when (A and B) is true. (A and B) is only true when both A and B are true. So, when A is true, B is true, and C is false, the expression will be false. With any other values, it will be true.

Logical operations make an excellent addition to a programmer's bag of tricks. They can be used to shorten programs, speed up programs, and to perform things that would otherwise require great difficulty. □



DATA Statement Simulation

Richard Turner

One of the most noticeable shortcomings of the ZX80 Basic is that there are no DATA statements.

With numbers it is possible to simulate the DATA, READ and RESTORE instructions by a simple program.

This can be done using a REM statement. The numbers are placed in a list within the REM statement, as is done with DATA statements, separating each number by a comma, and ending the list with a comma.

The DATA is then read by PEEKing the number out of the REM statement and placing it in a variable, as demonstrated in *Demo 1*.

R. Turner, 396 James Reckitt Avenue, Hull, N. Humberside, HU8 OJA, England.

In this case the number is put into variable *C*. This may then be put into any other variable by using a LET statement such as:

LET A(2) = C

E stores the pointer to the next item of DATA and in this case the start of DATA is at memory location 16427.

LET E = 16427

this will act as the RESTORE statement.

The DATA can be placed anywhere in a program, but the memory location of the start of the data must be found.

The following program finds all the REM statements in a program and prints the memory location where the data will start. Type this in at the end of a program and then erase it after it is used. □

```
10  REM  12, 65, -2165, 9, 0, -1,    DATA STATEMENT
11  LET D = 0                        FLAG FOR + OR - NO.
15  LET C = 0                        NO. READ FROM DATA
16  LET E = 16427                    BEGINNING OF DATA
20  GOSUB 100                        EXTRACT FIRST NUMBER
22  IF D = 1 THEN LET C = -C         IS D SET? IF SO CHANGE SIGN
23  LET D = 0                        RESET FLAG
25  LET E = A + E + 1                POINTER TO NEXT NO. ON DATA
30  PRINT C                          NO. FROM DATA
40  LET C = 0
50  INPUT A$                         PRESS NEWLINE TO EXTRACT NEXT NUMBER
70  GOTO 20
80  STOP

100  FOR A = 0 TO 100
110  LET B = PEEK(E + A) - 28
120  IF B = 188 THEN RETURN           CHECK FOR COMMA
125  IF B = 90 THEN STOP              CHECK FOR END OF LINE
126  IF NOT B = 192 THEN GOTO 130     CHECK FOR MINUS SIGN
128  LET D = 1                        SET FLAG
129  GOTO 140
130  LET C = C * 10 + B
140  NEXT A
```

Do not use variable A, B, C, D, OR E elsewhere in the program.

Variables

A—Dummy in FOR/NEXT loops
B—Number read from REM statement should be PEEKing
C—Number read from DATA statement
D—Sign Flag + (0) - (1)
E—Pointer to next number on DATA list

```
2000  LET A = PEEK(16392) + PEEK(16393) * 256
2010  FOR B= 16424 TO A
2020  IF PEEK(B) = 254 THEN PRINT B + 1
2030  NEXT B
```

The use of this simulation means that one of the faults of the ZX80 Basic may be easily rectified.

A one-hour LP record of eight synthesizers may change your views about computer music forever

Binary Beatles

by David Ahl

Computer music. Who needs it? It's mostly boring beep, beep, beeps or wildly modern stuff. It's certainly nothing you'd want to listen to more than once. That's what I thought about computer music and most of my friends agreed.

In 1978 I entered Yankee Doodle Dandy into my Software Technology system just to be different. Dick Moberg heard of it and asked me to perform in the Philadelphia Computer Music Festival. I agreed expecting to be the only one with something out of the ordinary. I was wrong.

Computer Accompanist

Nine individuals and groups performed in the festival. There were the usual Bach pieces but even they were different. Gooitzen van der Wal performed the last movement of the 2nd Bach Suite in a unique way. He played the flute solo while using the computer as accompaniment.

Then Dorothy Siegel did the same thing, playing the clarinet solo part of Wanhal's Sonata in b flat. The audience went wild.

Hal Chamberlin played Bach's Tocatta and Fugue in d minor. But also with a difference. He used a large computer before hand to "compute" the waveform of every instrument playing every note. It took one hour of computation time for each two minutes of playback time. The result could hardly be distinguished from the organ in the Hapsburg Cathedral.

Don Schertz had a home brewed synthesizer truly mounted on a breadboard that allowed him to control 25 parameters of each note. It produced spectacular sounds in his arrangement of Red Wing.

Singing Computer

In 1962, D.H. Van Lenten at Bell Laboratories produced the first talking computer. Bell engineers taught it to recite the soliloquy from Hamlet. Then they went one step further and taught it to sing Daisy both alone and accompanied by another computer. This was also performed at the festival.

Yes, the Beatles were represented. Andrew Molda played Hey Jude on his COSMAC VIP system with a program called PIN-8 (Play it Now).

Superb Quality Recording

All these pieces and twelve others were recorded with broadcast quality equipment. Because of audience noise, eight were re-recorded later in a studio. We then took these tapes to Tru-Tone, a top recording studio and cut a lacquer master. It was a long session since the recording engineers insisted upon analyzing the sound from every source and setting up the equilization curves accordingly. It took over 12 hours to produce a one-hour lacquer master.

Finished recordings were then pressed on top-quality vinyl and inserted into liners and record jackets. These were then shrink wrapped in plastic for maximum protection. We guaranteee that every LP record is free from defects or we will replace it free of charge.

The extensive descriptions of each of the eight synthesizers and the festival would not all fit on the jacket so we've included an extra sheet with each record. This entire package is mailed in a protective corrugated package to insure that it reaches you in mint condition. The cost is a modest $6.00 postpaid in the U.S. and $7.00 foreign. Send order with payment or Visa, MasterCard or American Express number to Creative Computing, Morris Plains, NJ 07950.

This LP record contains one hour of eight computer music synthesizers that you'll listen to over and over again. Send in your order today at no risk whatsoever.

creative computing

Morris Plains, NJ 07950
Toll-free **800-631-8112**
(In NJ 201-540-0445)

KNOW YOUR ZX80

With LINSAC products for the Sinclair ZX80

NOW ONLY £7.95

THE ZX80 COMPANION
(Second Edition)
Maunder, Logan and Tooter

ISBN 0 907211 00 3. Price £7.95 incl. UK postage.

This best-selling manual on the Sinclair ZX80 covers ZX80 BASIC, hardware and programs and has a detailed explanation of the ZX80 Monitor, routines and entry points. A routine for generating moving displays is also included.

Ch.1 – Operating the ZX80.
Ch.2 – Theory of Computers
Ch.3 – ZX80 BASIC
Ch.4 – The ZX80 Monitor
Ch.5 – Construction and Hardware
Ch.6 – ZX80 Programs
App.1 – Comparison of Z-80 Opcodes and ZX80 Characters.
App.2 – The 8K ROM.

For a cassette of ten programs from the Companion add £4.95.

The ZX80 Monitor A complete assembly language listing of the ZX80's 4K Monitor, with annotations – Price £10.00 incl. UK postage.

Cassette Software

All LINSAC program packs are on single C12 cassettes with printed run instructions. All run on 1K ZX80's unless otherwise stated. Price £10.00 per pack incl. UK postage.

GAMES PACK 1 – Three Towers, Number Guessing, Mastermind, Sketcher, Hurkle, Nim, Symbol Simon.
GAMES PACK 2 – Nine Lives, The Maze Game, Plain Sailing, OXO, Chinese Puzzle, Tower of Hanoi, Battleships.
GAMES PACK 3 – (2K+) – Fruit Machine, Four-in-a-line, Zombies
EDUCATION PACK 1 – Maths Drill, Dot Recognition, Musical Notes, Spelling Quiz, Day Finder.
EDUCATION PACK 2 – Graph Plotter, Prime Factors, Number Bases, Bar Charts, Statistics.
EDUCATION PACK 3 – (2K+) – CAL Quiz Package with three sample data sets.
UTILITY PACK 1 – Memory Display, Hex Code Monitor, Renumber, Memory Search.

LINSAC 68 Barker Road, Linthorpe, Middlesbrough, Co. Cleveland TS5 5ES

NOW AVAILABLE

keyboard conversions

- **Standard Computer Keyboard**
- **Type programs in half the time**
- **Minimize errors**
- **Wired keyboard hooks up in minutes**

Plans for keyboard conversion with reverse video $10.00

Keyboard with complete parts and plans $65.00

Wired keyboard, complete with plans $85.00

Mail for information:

L.J.H. Enterprises
P.O. Box 6273, Orange, CA 92667

Widget

Gary McGath

Some of the many versions of *Widget* as illustrated by Timothy Truman.

Is it possible for a game to present an interesting challenge, yet fit in 1K of RAM? As evidence that the answer is "yes," I offer *Widget*, a relative of the games *Hammurabi* and *Lemonade Stand*, in which the player allocates his resources for the best effect. As the player, you are the head of the United Widget Company, trying to expand your business as rapidly as possible without going broke.

You start out with one plant and $90,000. A plant is capable of producing as many as 1000 widgets a month, which will sell for $10 each. The monthly fixed overhead is $4000, including the mortgage cost for your first plant. In addition, it costs $5 per widget to produce any widgets in excess of the first hundred. To sell any widgets, you have to advertise. Naturally, the return on advertising isn't entirely predictable, but you have to learn how much to spend in order to sell what you produce at a profit.

The key to making money is buying more plants. A plant costs $100,000 to buy and $2000 a month to maintain, and it lets you produce another thousand widgets a month. This means more fixed cost and more need for advertising; it also means economies of scale. You'll notice that as production increases, it takes more advertising dollars to sell each widget. On the other hand, if you keep a steady flow coming out of your plants, each month's sales will produce a carry-over effect into the next month.

With these costs in mind, you can start playing *Widget*. Each turn represents one month. At the start of the turn, the screen shows your current cash balance, number of plants, and inventory. If you have at least $100,000, you are asked, "Buy a plant?" To buy a plant, answer "Y" (just the one letter); anything else means "no."

Next (or first, if you had less than $100,000), the program asks for your ad budget. Enter the number of *thousands* of dollars that you want to spend on ads; for example, to spend $4000, enter "4." You will be asked again for input if you try to spend more than you have.

Finally you are asked how many widgets you want to produce. Enter the number of *hundreds* of widgets you want to produce, remembering that each plant can only produce 1000 widgets. Again, the program won't let you go until you've entered a legal number.

The advertising cost is then subtracted from your cash balance, sales are determined, revenue comes in, and then the other costs are subtracted. If you can't meet your costs, the message "You are bankrupt" appears, and the game is over. Otherwise, the program goes on to the next month.

Before starting, decide how many months you're going to play. Sixty turns (5 years) or 120 turns (10 years) make a good game. Keep score by calculating your company's net worth, which is its cash balance plus $100,000 per plant. There isn't any predetermined winning score, but the outcome of a fairly well-played game would be to have 5 plants and $50,000 or more at the end of 10 years. □

Gary McGath, 5 Ames Rd., RFD #3, Milford, NH 03055.

```
  5 RANDOMIZE
  6 LET Z = 1
 10 LET P = 1
 20 LET M = 990
 30 LET I = 0
 40 LET S = 10
100 CLS
110 PRINT "MONTH" ,Z
120 LET Z = Z + 1
300 PRINT "YOU HAVE" , "$" ;M; "00"
310 PRINT ,P;" PLANTS"
320 PRINT ,I;"00 WIDGETS"
330 PRINT "LAST MONTH YOU
          SOLD ";S; "00"
350 IF M < 1000 THEN GO TO 410
360 PRINT "BUY A PLANT?"
370 INPUT Y$
380 IF NOT Y$ = "Y" THEN GO TO 410
390 LET P = P + 1
400 LET M = M - 1000
410 PRINT "AD BUDGET IN 1000S ?"
420 INPUT B
430 IF B * 10 > M THEN GO TO 420
440 LET M = M - B * 10
450 PRINT "PRODUCTION IN 100S ?"
470 INPUT W
480 IF W > 10 * P THEN GO TO 470
490 LET I = I + W
500 LET S = S / (RND (3) + 3)
510 IF B < 6 THEN GO TO 550
520 LET S = S + (B - 6) * 5
530 LET B = 6
550 IF B < 2 THEN GO TO 600
560 LET S = S + (B - 2) * 7
570 LET B = 2
600 LET S = S + B * B
610 IF S > I THEN LET S = I
620 LET M = M + S * 10
630 LET I = I - S
640 LET M = M - 20 * P - 20
650 LET M = M - (W - 1) * 5
660 IF NOT M < 0 THEN GO TO 100
800 PRINT "YOU ARE BANKRUPT."
```

```
MONTH       1
YOU HAVE            $90000
          1 PLANTS
          000 WIDGETS
LAST MONTH YOU SOLD   1000
AD BUDGET IN 1000S?


MONTH     29
YOU HAVE            $101000
          1 PLANTS
          000 WIDGETS
LAST MONTH YOU SOLD  1300
BUY A PLANT?


MONTH     61
YOU HAVE            $71000
          2 PLANTS
          200 WIDGETS
LAST MONTH YOU SOLD  1900
AD BUDGET IN 1000S?


MONTH     85
YOU HAVE            $50000
          3 PLANTS
          000 WIDGETS
LAST MONTH YOU SOLD 3600
AD BUDGET IN 1000S?


MONTH    121
YOU HAVE            $53000
          5 PLANTS
          600 WIDGETS
LAST MONTH YOU SOLD 5300
AD BUDGET IN 10000S?
```

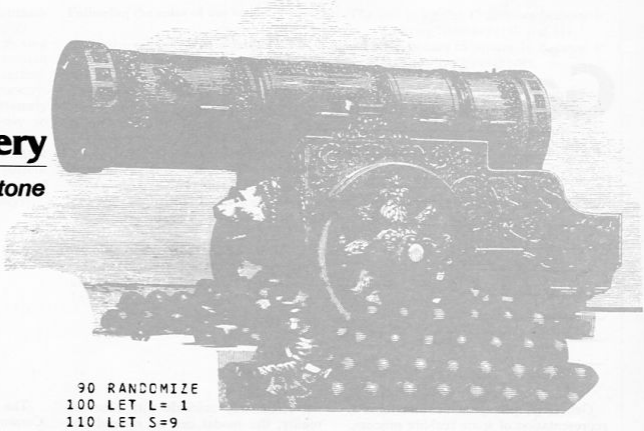

CRASH CURSOR
by TIMOTHY TRUMAN ©1981 A.R.
SOMEWHERE IN A GALAXY FAR, FAR AWAY FROM TOLEDO, OHIO...
STEP ON IT, CRASH!
INSIDE THE FLEEING CRAFT--CRASH CURSOR, SPACEHAWK, AND HIS FAITHFUL COMPUDROID, SYNK...!
I DON'T BELIEVE YOU! FIRST, YOU GET YOURSELF INTO A BREACH-OF-PROMISE SUIT WITH THAT FLOOZY ON PSORIASIS IV...
...NOW, YOU PASS A BAD CHECK ON PALEGRA I, AND RUN THE FIRST STOP-LIGHT WE HIT...!
DON'T WORRY, KID.. I'LL GET US OUT OF THIS...!
I WAS AFRAID YOU WERE GONNA SAY THAT..!
AVAST, FAITHFUL SYNK! WHAT'S THAT AHEAD?
AT THIS MOMENT, I'M ALMOST AFRAID TO VENTURE A READOUT...
"...BUT I WOULD SEEM TO BE A FREELY-FLOATING, PLANET-SHAPED BODY OF 'MUCHO PEDIS FUNGI'...! IN A WORD..."
"A BIG BALL OF ATHLETES FOOT FUNGUS!"
AHA! JUST WHAT WE NEED!
!?!
'OBSERVE! THE POLICE-SHIPS RUSH BRISKLY BY US! I'VE OUT-MANUEVERED THEM AGAIN, WITH MY PEERLESS PILOTING SKILL!
I'M SURE...


YES, ONCE AGAIN YOUR DAUNTLESS FRIEND AND MASTER HAS PROVEN HIMSELF THE TRUE EVASIVE-TACTICIAN THE GREAT UNIVERSE KNOWS HIM TO BE...!
YEAH. MARVELOUS. AHH-LET'S GO...!
IN ALL THE KNOWN UNIVERSE, FROM SIRIUS TO NEWARK, THERE IS NO ONE TO MATCH MY..
AHHH--CRASH--I HATE TO INTERRUPT YOUR IMPRESSIVE SELF-GRATIFICATION, BUT...
WE'RE BEING EATEN!
CHOMP!
HMMM...! WE SEEM TO BE IN ANOTHER FIX!
HOW VERY OBSERVANT...!
...NOW WHAT?
WHY, WE PURSUE THIS NEW ESCAPADE TO ITS OWN GROUND-- TRUE TO OUR PARTICULAR HEROIC FASHIONS!
YES, DEAR DROID! ADVENTURE IS DRAWN TO HEROES SUCH AS US, LIKE MOTHS TO FLAME, BEES TO HONEY, FLIES TO--
SHHH...! I HEAR SOMETHING!
WE SHALL FOLLOW THIS STRANGE, NEW, EVENTUALLY EXCITING PREDICAMENT TO ITS...
OH, CRASHY--
SYNK
2


NOT TO WORRY, MY LITTLE CHROMIUM FRIEND...!
I AM NEVER WITHOUT MY LETHAL*MARK-XII-OMNI-POWERED-ATOMIC-SONIC-DELTA-RADIATION-PISTOL (PATENT PENDING)!
WATCH, AS IT MAKES SHORT-WORK OF THESE SLIME-ENCRUSTED, LICHEN-EATING, ILL-TEMPERED, (AND RATHER FOUL-SMELLING) CAVERN-CRETINS!
CLICK!
DRAT! OUT OF PROTONS! ONLY MY FURIOUS, FURTIVE, FLAILING FIST CAN SAVE US NOW!
YEAH. RIGHT.
MASTER! WE HAVE THEM!
VERY GOOD LEROY! BRING THEM TO ME! NOW!
IT'S..
YES! 'TIS I ...! MASTER OF SPACE! CONTROLLER OF MEN! CREATOR OF DEMONS AND ICKY, WICKED THINGS! ...THE IMPECCABLY WELL-DRESSED...
...DING THE MERCIFULNESS-LESS!!!
DO WE EAT 'EM NOW, BOSS?
©1991 TIMOTHY TRUMAN
TO BE CONTINUED...!

Artillery

C.R. Whetstone



```
 90 RANDOMIZE
100 LET L= 1
110 LET S=9
120 LET A=RND(2030)
130 GO TO 290
140 DIM S(21)
150 FOR Q=1 TO 20
160 LET S(Q)=1
170 NEXT Q
180 LET S(A/100)=2
190 LET S(L/100)=3
200 PRINT X;
210 FOR Q=1 TO 20
220 IF S(Q)=1 THEN PRINT "-SHIFT T-";
230 IF S(Q)=2 THEN PRINT "T";
240 IF S(Q)=3 THEN PRINT "*";
250 NEXT Q
260 PRINT "M/R"
270 RETURN
300 PRINT "TARGET RANGE =";A;"00 YARDS"
310 PRINT
320 GO SUB 140
330 PRINT
350 PRINT "INPUT GUN ELEVATION",,"(0 TO 90 DEGREES)"
360 INPUT N
370 IF N<0 OR N>90 THEN GO TO 360
380 IF N>45 THEN LET N=90-N
390 CLS
500 LET L=(N**2)+5
510 IF ABS(A-L)<50 THEN GO TO 1000
520 LET S=S-1
530 IF S=0 THEN GO TO 2000
540 PRINT "RANGE =";A;"00 YARDS"
545 PRINT
550 GO SUB 150
560 PRINT "YOU WERE ";ABS(A-L);"00 YARDS";
570 IF A-L>0 THEN PRINT "SHORT"
580 IF A-L<0 THEN PRINT "LONG"
590 PRINT
600 PRINT "HIT N/L TO RELOAD"
610 INPUT Z$
615 CLS
620 IF NOT Z$="N" THEN GO TO 300
```

In Artillery the player fires a cannon at a stationary target. The screen displays the gun location, the target site, and the approximate range. To take aim the cannoneer selects the angle at which the cannon is raised, between 0 and 90 degrees. The display then indicates whether the shot was long, short, or on target.

With a maximum range of just over 200,000 yards—the display gives a relative range within approximately 10,000 yards.

C.R. Whetstone, 211 Clarendon Ave., Baltimore, MD 21208.

Game of Life

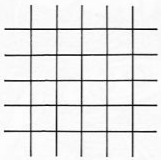
Generally, a mathematical model is a representation of some real-life process, expressed in mathematical form (such as a set of related equations) or in algorithmic form (such as a computer program). Usually the model is by necessity a simplification of the actual process, since real-life processes tend to be highly complex. One advantage of embodying the model as a computer program is that we can run the program and thus simulate the process being modelled. By varying certain features of the program, we can learn something about the relationships between the componenets and the overall structure of the process. In addition, if the output does not sufficiently coincide with observed reality, the model can be revised and improved.

It is also possible to model a purely abstract process. We don't often see this done. After all, if someone asked you to describe some abstract process, what would you say? However, many games start out as purely abstract processes. For example, tic-tac-toe or checkers are abstract from the point of view that they represent no real-life process. Occasionally, it turns out that an abstract process represents a real-life process either by accident or design. The following game is one such example which in some ways represents life itself.

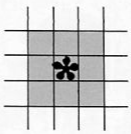
The game of *Life* was devised by John Conway, a mathematician at the University of Cambridge, and made popular by a series of articles written by Martin Gardner in recent issues of *Scientific American*. Ever since the first article appeared in October 1970, hundreds of mathematicians throughout the world have become fascinated with the model and have been exploring its properties.

The game consists of following the successive generations of a particular imaginary type of cellular life-form. The life processes of these cells are represented by the following mathematical model,

(1) **World** — Cells live on an infinite two-dimensional plane of squares (like an infinite checker-board, except that all squares are identical).



(2) **Neighborhood** — Each square has eight *neighbor* squares. In the diagram below, the *neighbor* squares for the square with the asterisk (*) have been colored in.



(3) **Survival** — A cell (always represented by a *) which is living in generatin n, will remain living in generation n+1 if and only if it has exactly two or three living neighbors in generation n.

(4) **Death** — However, in all other cases the cell dies. Specifically: If it has one or no neighbors it dies from isolation. If it has four, more than three neighbors, it dies from overpopulation.

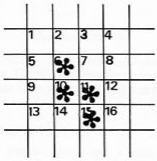
(5) **Birth** — If a square is empty during generation n, a living cell will be born into that square during generation n+1 if and only if that square had exactly three living neighbors during generation n.

The only trick is to remember that all survivals, deaths, and births occur simultaneously, and so the simplest way to keep the *bookkeeping* straight is to have two separate copies of the world — one for the old generation and one for the new one you are forming. For each square in the old world, decide what its state will be next time, and mark this down in the corresponding square in the new world.

which captures several properties common to all life-forms.

The game is played simply by picking some initial starting pattern and watching the development of some very interesting, and often beautiful patterns of symmetry. However, the player must be extremely careful because mistakes are easy to make.

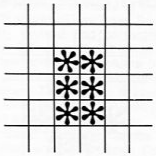
As an example, we will trace three generations of the following initial pattern (we have numbered some rows and columns for reference purposes only):



Following the rules of our model:

No births will occur in squares 1, 2, 3, 4, or 5 because none has three living neighbors.
The cell in square 6 will survive because it has two living neighbors (10 and 11).
A birth occurs in square 7 because there are three living neighbors (6, 10, and 11).
No birth occurs in squares 8 or 9.
The cell in square 10 survives because it has three living neighbors (6, 10, and 15).
The cell in square 11 survives also because it has three living neighbors (6, 10, and 15).
No birth occurs in squares 12 or 13.
A birth occurs in square 14 because there are three living neighbors (10, 11, and 15).
The cell in square 15 survives because it has two living neighbors (10 and 11).
No birth occurs in square 16 because it only has two living neighbors.

During this process, we have been filling in a picture of G1, and the end result is:



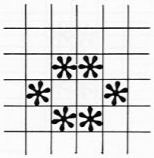
The following program for the game of *Life* is from The Melbourne House book, *Thirty Programs for The Sinclair ZX-80 1K*.

```
100 DIM A (63)
110 FOR I = 1 TO 63
120 LET A(I) = 128 * (RND (2) - 1)
130 NEXT I
200 FOR L = 0 TO 8
210 FOR J = 1 TO 7
220 LET I = J + 7 * L
230 PRINT CHR$ (A(I));
240 NEXT J
250 PRINT
260 NEXT L
270 PRINT "PRESS 1"
280 INPUT A
290 CLS
300 FOR I = 9 TO 55
310 LET X = 0
320 IF ((I + 1) AND 7) = 0 THEN GO TO 350
330 LET X = (A(I + 1) > 127)
340 IF ((I - 1) AND 7) = 0 THEN GO TO 360
350 LET X = X + (A(I - 1) > 127)
360 FOR J = 7 TO 9
370 FOR R = - 1 TO 1
380 IF R = 0 THEN GO TO 420
390 LET K = I + R * J
400 IF (K AND 7) = 0 THEN GO TO 420
410 LET X = X + (A(K) >127)
420 NEXT R
430 NEXT J
440 LET A(I) = A(I) - X
460 NEXT I
470 IF A(I) = 3 OR A(I) = 130 OR A(I) = 131 THEN GO TO 500
480 LET A(I) = 0
490 GO TO 510
500 LET A(I) = 128
510 NEXT I
520 GO TO 200
```

Life, continued...

Experiment 1

Using pencil and paper, carefully compute G2, the next generation for this same society of cells. If you do it correctly, you will find that G2 is the pattern which Conway calls the "beehive":

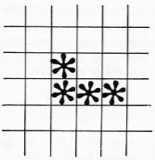


Experiment 2

Now compute G3. If you are again careful, you will discover that G3 is identical to G2. Why does Conway call the beehive a "still-life"? If you are not sure, think about G4, G5, G6 . . .

Experiment 3

Using pencil and paper, compute G0, G1, G2 and G3 for the initial pattern below:



If you do it correctly, G3 should look familiar to you.

Experiment 4

By now you've no doubt noticed that with pencil and paper, this game is an extremely slow process, and mistakes are all too common. If we ever hope to look at more than a few patterns, we're going to have to turn to the computer for help.

Write a computer program which simulates "Life" for **any** given initial pattern, and which has the following features:
(1) Allow for as large a world size as your particular computer facility will permit (obviously an infinite plane is not possible in a finite memory). You will probably want to use array structures with two subscripts (row and column).

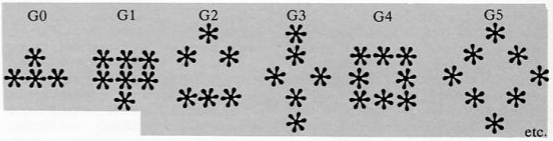
(2) Whatever world-size you are limited to, make sure your program doesn't try to alllow births outside your world, even though properly these *would* occur on an infinite plane.

(3) Make sure your algorithm allows all survivals, deaths, and births during a given generation to occur *simultaneously*, as discussed above.

(4) Allow the user to input the initial pattern in a convenient format, such as pairs of (row, column) coordinates.

(5) Make your program efficient and your output as close to the format of the pictures above as possible.

Once your algorithm is designed, and your program is written, debug your program by running it on the following initial GO pattern, and carefully check your output vs. the results below:



Warning !

Depending upon the world-size you are limited to, certain "large" patterns may grow differently than they would on an infinite plane.

If the society of cells above, however, fits inside your world-size, you will notice an interesting cyclic pattern beginning at G0, which Conway calls "traffic lights".

Experiment 5

When your program is thoroughly debugged and operational, or using the LIFE program with this article, the real fun comes in thinking up initial patterns and watching them grow. Interesting situations to watch for are:

(1) Other "still-life societies (like the "bee-hive")
(2) Other "cyclic" societies (like the "traffic lights")
(3) A society which lives for an extended period of time without dying, becoming still, or cycling

Experiment 6

Find copies of the October 1970 and/or February 1971 issues of *Scientific American* and read Gardner's articles on "Life." You may want to run your program on some of the societies he describes, such as: diagonal chains, the R pentomino, the Latin cross, the cheshire cat, and many others.

Experiment 7

Try to think up changes in the model (and your computer program) which will drastically alter the life patterns of the cells, i.e. by modifying the rules for birth or death or both. Based upon your experience so far, try to come up with sets of rules which will lead to more populous societies, or more sparse societies, or societies which are less symmetric than those of "Life", etc. The range of possibilities is very large.

Experiment 8

Make some major modifications in your computer program to make it more *general*, by allowing the user to specify the particular model he wants to investigate. For example, you might have your program begin by posing the following questions to the user:

How many neighbors for survival?

How many neighbors for birth?

Then, if the user answered 2, 3 for the first question and 3 for the second, your program would follow the rules of "Life." But if he gave other answers, the program would simulate for him some other model he wants to investigate.

Experiment 9

Is there any way you can streamline your program or the published one so that you can enlarge the size of the world it currently handles?

Experiment 10

How might you alter the general concept of "neighborhood" so that entirely different models could be tested? How would your computer program have to be changed in order to simulate these new models?

The story behind the two best selling computer games books in the world.

Computer Games

by David H. Ahl

Everybody likes games. Children like tic tac toe. Gamblers like blackjack. Trekkies like Star Trek. Almost everyone has a favorite game or two.

It Started in 1971

Ten years ago when I was at Digital Equipment Corp. (DEC), we wanted a painless way to show reluctant educators that computers weren't scary or difficult to use. Games and simulations seemed like a good method.

So I put out a call to all our customers to send us their best computer games. The response was overwhelming. I got 21 versions of blackjack, 15 of nim and 12 of battleship.

From this enormous outpouring I selected the 90 best games and added 11 that I had written myself for a total of 101. I edited these into a book called 101 Basic Computer Games which was published by DEC. It still is.

When I left DEC in 1974 I asked for the rights to print the book independently. They agreed as long as the name was changed.

Contents of *Basic Computer Games* (right) and *More Basic Computer Games* (below).

Artillery-3
Baccarat
Bible Quiz
Big 6
Binary
Blackbox
Bobstones
Bocce
Boga II
Bumbrun
Bridge-It
Camel
Chase
Chuck-A-Luck
Close Encounters
Column
Concentration
Condot
Convoy
Corral
Countdown
Cup
Dealer's Choice
Deepspace
Defuse
Dodgem
Doors
Drag
Dr. Z
Eliza
Father
Flip
Four In A Row
Geowar
Grand Prix
Guess-It
ICBM
Inkblot
Joust
Jumping Balls
Keno
L Game
Life Expectancy
Lissajous
Magic Square
Man-Eating Rabbit
Maneuvers
Mastermind
Masterbagels
Matpuzzle
Maze
Millionaire
Minotaur
Motorcycle Jump
Nomad
Not One
Obstacle
Octrix
Pasart
Pasart 2
Pinball
Rabbit Chase
Roadrace
Rotate
Safe
Scales
Schmoo
Seabattle
Seawar
Shoot
Smash
Strike 9
Tennis
Tickertape
TV Plot
Twonky
Two-to-Ten
UFO
Under & Over
Van Gam
Warfish
Word Search Puzzle
Wumpus 1
Wumpus 2

Introduction
The Basic Language
Conversion to Other Basics
Acey Ducey
Amazing
Animal
Awari
Bagels
Banner
Basketball
Batnum
Battle
Blackjack
Bombardment
Bombs Away
Bounce
Bowling
Boxing
Bug
Bullfight
Bullseye
Bunny
Buzzword
Calendar
Change
Checkers
Chemist
Chief
Chomp
Civil War
Combat
Craps
Cube
Depth Charge
Diamond
Dice
Digits
Even Wins
Flip Flop
Football
Fur Trader
Golf
Gomoko
Guess
Gunner
Hammurabi
Hangman
Hello
Hexapawn
Hi-Lo
High I-Q
Hockey
Horserace
Hurkle
Kinema
King
Letter
Life
Life For Two
Literature Quiz
Love
Lunar LEM Rocket
Master Mind
Math Dice
Mugwump
Name
Nicomachus
Nim
Number
One Check
Orbit
Pizza
Poetry
Poker
Queen
Reverse
Rock, Scissors, Paper
Roulette
Russian Roulette
Salvo
Sine Wave
Slalom
Slots
Splat
Stars
Stock Market
Super Star Trek
Synonym
Target
3-D Plot
3-D Tic-Tac-Toe
Tic Tac toe
Tower
Train
Trap
23 Matches
War
Weekday
Word

Converted to Microsoft Basic

The games in the original book were in many different dialects of Basic. So Steve North and I converted all the games to standard Microsoft Basic, expanded the descriptions and published the book under the new name Basic Computer Games.

Over the next three years, people sent in improved versions of many of the games along with scores of new ones. So in 1979, we totally revised and corrected Basic Computer Games and published a completely new companion volume of 84 additional games called More Basic Computer Games. This edition is available in both Microsoft Basic and TRS-80 Basic for owners of the TRS-80 computer.

Today Basic Computer Games is in its fifth printing and More Basic Computer Games is in its second. Combined sales are over one half million copies making them the best selling pair of books in recreational computing by a wide margin. There are many imitators, but all offer a fraction of the number of games and cost far more.

The games in these books include classic board games like checkers. They include challenging simulation games like Camel (get across the desert on your camel) and Super Star Trek. There are number games like Guess My Number, Stars and Battle of Numbers. You'll find gambling games like blackjack, keno, and poker. All told there are 185 different games in these two books.

Whether you're just getting started with computers or a proficient programmer, you'll find something of interest. You'll find 15-line games and 400-line games and everything in between.

The value offered by these books is outstanding. Every other publisher has raised the price of their books yet these sell for the same price as they did in 1974.

Moneyback Guarantee

Examine one or both of these books and key some games into your computer. If you're not completely satisfied we'll refund the full purchase price plus your return postage.

Basic Computer Games costs only $7.50 and More Basic Computer Games just $7.95 for either the Microsoft or TRS-80 edition (please specify your choice on your order). Both books together are $15. Send payment plus $2.00 shipping and handling to Creative Computing Press, Morris Plains, NJ 07950. Visa, MasterCard and American Express orders should include card number and expiration date. Charge card orders may also be called in toll-free to 800-631-8112 (in NJ 201-540-0445).

Order today to turn your computer into the best game player on the block.

creative computing

Morris Plains, NJ 07950
Toll-free **800-631-8112**
(In NJ 201-540-0445)

Tic Tac Toe or Noughts and Crosses

Jeffrey Hoffman

On a computer with a very limited amount of memory (1K) even a simple game like Tic Tac Toe might be a problem to fit. This version of the well known game is short and simple, anyone will be able to understand it. Lines 10 through 90 define the variables A$ to I$ (since there are no string arrays on the ZX80 with 4K ROM, one must use another way). Lines 120 to 160 create the familiar tic tac toe pattern using the "checkerboard" graphics symbol (located above the "A" key—you must use the "shift key"). In order to save memory, this program assumes that you know the rules of tic tac toe, and therefore allows you to enter any number you want, only actually using those between one and nine. However, you may not put an X where there is already an O (or visa versa), the computer will just take the number and go on to let your opponent go. Those people with larger memory machines can adapt this program to include a "play against the computer" mode. □

Jeffrey Hoffman, 332 Melbourne Rd., Great Neck, NY 11021.

```
10 LET A$="1"
20 LET B$="2"
30 LET C$="3"
40 LET D$="4"
50 LET E$="5"
60 LET F$="6"
70 LET G$="7"
80 LET H$="8"
90 LET I$="9"
100 FOR Q=1 TO 9
110 PRINT "    T I C  T A C  T O E"
120 PRINT A$;" ▒ ";B$;" ▒ ";C$
130 PRINT "▒▒▒▒▒▒▒▒▒"          (9 ▒ symbols)
140 PRINT D$;" ▒ ";E$;" ▒ ";F$
150 PRINT "▒▒▒▒▒▒▒▒▒"
160 PRINT G$;" ▒ ";H$;" ▒ ";I$
170 PRINT
180 IF Q=2*(Q/2) THEN GO TO 230
190 PRINT "X GOES"
200 INPUT C
210 LET P$="X"
220 GO TO 270
230 PRINT
240 PRINT "O GOES"
250 INPUT C
260 LET P$="O"
270 LET J$=STR$(C)
280 IF J$=A$ AND NOT A$=P$ THEN A$=P$
290 IF J$=B$ AND NOT B$=P$ THEN B$=P$
300 IF J$=C$ AND NOT C$=P$ THEN C$=P$
310 IF J$=D$ AND NOT D$=P$ THEN D$=P$
320 IF J$=E$ AND NOT E$=P$ THEN E$=P$
330 IF J$=F$ AND NOT F$=P$ THEN F$=P$
340 IF J$=G$ AND NOT G$=P$ THEN G$=P$
350 IF J$=H$ AND NOT H$=P$ THEN H$=P$
360 IF J$=I$ AND NOT I$=P$ THEN I$=P$
370 IF A$=B$ AND B$=C$ THEN 470
380 IF A$=D$ AND D$=G$ THEN 470
390 IF A$=E$ AND E$=I$ THEN 470
400 IF B$=E$ AND E$=H$ THEN 470
410 IF C$=F$ AND F$=I$ THEN 470
420 IF D$=E$ AND E$=F$ THEN 470
430 IF G$=E$ AND E$=C$ THEN 470
440 IF G$=H$ AND H$=I$ THEN 470
450 NEXT Q
460 GO TO 490
470 PRINT P$;"  WON"
480 GO TO 500
490 PRINT " T I E"
500 REM END
```

The Sinclair ZX80 is innovative and powerful. Now there's a magazine to help you get the most out of it.

Get in sync

SYNC magazine is different from other personal computing magazines. Not just different because it is about a unique computer, the Sinclair ZX80 (and kit version, the MicroAce). But different because of the creative and innovative philosophy of the editors.

A Fascinating Computer

The ZX80 doesn't have memory mapped video. Thus the screen goes blank when a key is pressed. To some reviewers this is a disadvantage. To our editors this is a challenge. One suggested that games could be written to take advantage of the screen blanking. For example, how about a game where characters and graphic symbols move around the screen while it is blanked? The object would be to crack the secret code governing the movements. Voila! A new game like Mastermind or Black Box uniquely for the ZX80.

We made some interesting discoveries soon after setting up the machine. For instance, the CHR$ function is not limited to a value between 0 and 255, but cycles repeatedly through the code. CHR$ (9) and CHR$ (265) will produce identical values. In other words, CHR$ operates in a MOD 256 fashion. We found that the "=" sign can be used several times on a single line, allowing the logical evaluation of variables. In the Sinclair, LET X=Y=Z=W is a valid expression.

Or consider the TL$ function which strips a string of its initial character. At first, we wondered what practical value it had. Then someone suggested it would be perfect for removing the dollar sign from numerical inputs.

Breakthroughs? Hardly. But indicative of the hints and kinds you'll find in every issue of **SYNC**. We intend to take the Sinclair to its limits and then push beyond, finding new tricks and tips, new applications, new ways to do what couldn't be done before. **SYNC** functions on many levels, with tutorials for the beginner and concepts that will keep the pros coming back for more. We'll show you how to duplicate commands available in other Basics. And, perhaps, how to do things that can't be done on other machines.

Many computer applications require that data be sorted. But did you realize there are over ten fundamentally different sorting algorithms? Many people settle for a simple bubble sort perhaps because it's described in so many programming manuals or because they've seen it in another program. However, sort routines such as heapsort or Shell-Metzner are over 100 times as fast as a bubble sort and may actually use less memory. Sure, 1K of memory isn't a lot to work with, but it can be stretched much further by using innovative, clever coding. You'll find this type of help in **SYNC**.

Lots of Games and Applications

Applications and software are the meat of **SYNC**. We recognize that along with useful, pragmatic applications, like financial analysis and graphing, you'll want games that are fun and challenging. In the charter issue of **SYNC** you'll find several games. Acey Ducey is a card game in which the dealer (the computer) deals two cards face up. You then have an option to bet depending upon whether you feel the next card dealt will have a value between the first two.

In Hurkle, another game in the charter issue, you have to find a happy little Hurkle who is hiding on a 10 X 10 grid. In response to your guesses, the Hurkle sends our a clue telling you in which direction to look next.

One of the most ancient forms of arithmetical puzzle is called a "boomerang." The oldest recorded example is that set down by Nicomachus in his *Arithmetica* around 100 A.D. You'll find a computer version of this puzzle in **SYNC**.

Hard-Hitting, Objective Evaluations

By selecting the ZX80 or MicroAce as your personal computer you've shown that you are an astute buyer looking for good performance, an innovative design and economical price. However, selecting software will not be easy. That's where **SYNC** comes in. **SYNC** evaluates software packages and other peripherals and doesn't just publish manufacturer descriptions. We put each package through its paces and give you an in-depth, objective report of its strengths and weaknesses.

SYNC is a Creative Computing publication. **Creative Computing** is the number 1 magazine of software and applications with nearly 100,000 circulation. The two most popular computer games books in the world, *Basic Computer Games* and *More Basic Computer Games* (combined sales over 500,000) are published by Creative Computing. Creative Computing Software manufactures over 150 software packages for six different personal computers.

Creative Computing, founded in 1974 by David Ahl, is a well-established firm committed to the future of personal computing. We expect the Sinclair ZX80 to be a highly successful computer and correspondingly, **SYNC** to be a respected and successful magazine.

Order SYNC Today

To order your subscription to SYNC, in the USA send $10 for one year (6 issues), $18 for two years (12 issues) or $24 for three years (18 issues). Send order and payment to the address below or call MasterCard, Visa or American Express orders to our toll-free number.

Subscriptions in the UK are mailed by air and cost £10 for one year, £18 for two years or £25 for three years. Send order and payment to the UK address below.

Canadian and other foreign surface subscriptions cost $15 per year or $27 for two years and should be sent to the USA address.

We guarantee your satisfaction or we will refund your entire subscription price.

Needless to say, we can't fill up all the pages without your help. So send in your programs, articles, hints and tips. Remember, illustrations and screen photos make a piece much more interesting. Send in your reviews of peripherals and software too—but be warned: reviews must be in-depth and objective. We want you to respect what you read on the pages of **SYNC** so be honest and forthright in the material you send us. Of course we pay for contributions—just don't expect to retire on it.

The exploration has begun. Join us.

The magazine for Sinclair ZX80 users

SYNC

39 East Hanover Avenue
Morris Plains, NJ 07950, USA
Toll free **800-631-8112**
(In NJ 201-540-0445)

27 Andrew Close, Stoke Golding
Nuneaton CV13 6EL, England

Making Music with the ZX80

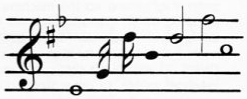
Richard Forsen

The Sinclair ZX80 comes with no built-in sound device, although someday someone will undoubtedly devise a gadget for this purpose. Until then we can use programs to control sounds using the weak AM signal coming from the microprocessor. This signal was strongest at the high end of the dial.

The basic theory behind the program is the operating system "keeps tabs" on the position of every FOR...NEXT loop in the program. For example, if you write a program similar to the one below, which repeats two identical loops indefinitely (or until you hit BREAK), you will hear a trill-like sound.

```
10 LET X = 50
20 FOR I = 1 TO X
30 NEXT I
40 FOR I = 1 TO X
50 NEXT I
60 GO TO 20
```

The first loop generates the higher pitch. It is interesting that a loop-within-a-loop structure will not produce the same result. The variable X represents the length of the note. Line 10 can be changed without affecting anything else but the duration of the tone. The value of 50 makes the tone last about one fifth of a second. A larger number will increase the length of time the note is played. Try changing line 10 to "10 INPUT X" and line 60 to '60 GOTO 20,' RUN the program, and input thes values: 1, 2, 5, 8, 15, 25, and 64, BREAKing after each entry. Then Add:



```
1  RANDOMISE
5  INPUT N
10 LET X = RND (N)
31 FOR I = 1 TO X
32 NEXT I
33 FOR I = 1 TO X
34 NEXT I
35 FOR I = 1 TO X
36 NEXT I
37 FOR I = 1 TO X
38 NEXT I
```

trying the same values.

Music Making, continued...

The program can be typed in quickly due to the excellent line editor built into the ZX80. Enter the first nineteen lines, then type 1000 FOR I 1 TO X and hit NEWLINE. Then press SHIFT and NEWLINE simultaneously. Line 1000 will appear below the listing. Next press SHIFT 5, RUBOUT, 3 and NEWLINE. You have now duplicated line 1000. Do this for everything, and you will save time.

After you've typed the program, RUN it. First it will ask "HOW MANY NOTES?". Here you must enter a number from 1 to 12. More than 12 notes will overload the memory, causing the program to terminate.

The computer will then ask the pitch and duration of each tone. When asked for the duration, type in the reciprocal of the actual duration. For example: to enter a quarter note, enter 4. One problem arises with dotted notes. A dotted eight note counts 3/16, and the reciprocal is 16/3. Since you can't enter this as a numerical value, you must divide it yourself. Instead of entering 16/3, enter 5.

To enter the pitch, type a number between 1 and 17. 1 represents the highest pitch, and 17 the lowest.

After you have entered the tune, the computer will play the notes in sequence and then repeat the score continuously. You can hear the composition by putting an AM radio on top of the computer, or vice versa. I found the best signal by putting the computer on top of my stereo.

The changes for two variations are listed below the main program. The first program, "Sound Effector," lets you create various arcade type sounds. This is the same as the original program, except for being able to adjust the speed of the oscillations, thereby creating interesting sound effects.

```
0001 DIM D (12)   (Duration for each note)
0002 DIM P (12)   (Pitch for each note)
0005 PRINT "HOW MANY NOTES?" (Up to 12)
0006 INPUT N                 (Number of notes = N)
0010 CLS                     (Self-explanatory)
0020 FOR I = 1 TO N
0025 PRINT "DURATION OF NOTE" (1 for whole note, 2 for 1/2 note, 4 for
    ;I;"?";                    quarter note, 8 for eighth note, etc...)
0030 INPUT M
0035 PRINT "(1/";M;")"
0040 LET D (I) = 576/M
0045 PRINT "PITCH?";     (1 is the highest pitch, 17 is the lowest pitch.)
0050 INPUT P(I)
0059 CLS
0060 NEXT I          (Ends loop for inputtting data)
0500 FOR F = 1 TO N      (Loop for playing tones)
0510 LET X = D (F)       (Sets duration)
0520 GO SUB 970 + P (F) * 30  (Plays actual note)
0530 NEXT F          (Ends loop)
0999 GO TO 500       (Repeats series of tones indefinitely)
1000 FOR I = 1 TO X  (First of a series of 17 FOR...NEXT sub-
1010 NEXT I              routines that produce 17 different tones.
1020 RETURN              They go from highest in pitch to lowest.
1030 FOR I = 1 TO X      No STOP statement is needed.)
1040 NEXT I
1050 RETURN
1060 FOR I = 1 TO X
1070 NEXT I
1080 RETURN
1090 FOR I = 1 TO X
1100 NEXT I
1110 RETURN
1120 FOR I = 1 TO X
1130 NEXT I
1140 RETURN
1150 FOR I = TO X
1160 NEXT I
1170 RETURN
1180 FOR I = 1 TO X
1190 NEXT I
1200 RETURN
1210 FOR I = 1 TO X
1220 NEXT I
1230 RETURN
1240 FOR I = 1 TO X
1250 NEXT I
1260 RETURN
1270 FOR I = 1 TO X
1280 NEXT I
1290 RETURN
1300 FOR I = 1 TO X
1310 NEXT I
1320 RETURN
1330 FOR I = 1 TO X
1340 NEXT I
1350 RETURN
1360 FOR I = 1 TO X
1370 NEXT I
1380 RETURN
1390 FOR I = 1 TO X
1400 NEXT I
1410 RETURN
1420 FOR I = 1 TO X
1430 NEXT I
1440 RETURN
1450 FOR I = 1 TO X
1460 NEXT I
1470 RETURN
1480 FOR I = 1 TO X
1490 NEXT I
1500 RETURN
```

Changes For Sound Effector

```
3 PRINT "SPEED";
4 INPUT SPEED
40 LET D (I) = SPEED/M
```

If you want, you can change line 510 to LET X=RND(D(F)) to make it sound more interesting.

The second program variation, "Compusition," has the computer do the composing. It makes up the pitch, a number between 1 and 17, and the duration is a random exponent of two.

Changes For Composition

```
30 LET M = 2 * * (RND (6) -1)
35 PRINT I;".) (1/";M;")";
50 LET P = RND (17)
55 PRINT P (I)
DELETE 25, 45 & 59
```

Line 30 picks a power of 2 between 0 (1) and 5 (32)
The six can be replaced by any number up to 10.

35 prints duration and which subroutine was used is printed in line 55.

Line 50 picks a pitch for P(I) from 1 to 17.

□

Rich Forsen, 9496 Weston Road, New Hartford, NY 13413.

The Home Computer Market, the ZX80 and the Future

Nigel Searle

Excerpts of a speech given to the Amateur Computer Group of New Jersey, December 11, 1980.

I would like to describe what Sinclair has been doing—what we are doing, and what we plan to do. I would also like to speculate about what Sinclair and other companies in the small computer business might be doing 10-15 years from now. We are still in a very young business, none of us has been around for more than a couple of years, and another 10-15 years is going to make perhaps more difference than any of us realizes.

Sinclair is a company which has been involved in the consumer electronics business (calculators, digital watchs, etc.). When we were designing calculators, particularly programmable and scientific calculators, we'd say, "wouldn't it be terrific if we could design a computer that wasn't much bigger than a calculator, but would be a real computer, programmable in a high level language." It seemed as though it would be a long time before we could do that when we thought of it in 1973-1974. Technological advances have made it possible much sooner than anyone expected.

We introduced the Sinclair ZX80 in Europe in February 1980, and in the U.S. in August 1980. We've sold a large number of computers. To say that we have been successful is an understatement. We are still growing very rapidly. We've been selling exclusively by mail order and primarily to technically-oriented people. We choose our markets by the type of magazines we advertise in and, although we are beginning to advertise in consumer magazines, most of our sales have come from technically-oriented people. We certainly don't intend that that will always be the case.

The Home Computer Market

We think that our success has proven the home computer market, which people talked about four and five years ago, and which they became diverted from because of the greater profits and the readier market in the small business area—that true consumer market (personal, home, etc. but definitely not small business)—does exist. The sales of our ZX80 have shown that. At the same time that we are not selling a small business computer, neither are we selling a home entertainment computer. The Sinclair ZX80 is not the greatest computer on which to play "Space Invaders." We do have such a software package coming out, but it doesn't have sound or color and its graphics are not high resolution. There are other disadvantages from the games and entertainment point of view which we will come to in a moment.

We are selling a serious computer for use in a particular application. Very often that application is education—the user's own education or his children's education. We are selling it to individuals who are paying with their own after-tax dollars and not with the tax deductible money of a business. So we have shown, I think, and certainly we have satisfied ourselves, that the personal computer market really does exist and there is no reason to believe that it won't go on growing. We are, of course, doing some of the things which we believe will help it to grow.

Perhaps the most amazing thing about our success so far—and we have sold literally tens of thousands of units, in excess of 50,000 units world wide—is that this unit has an integer-only Basic, has just 1K bytes of RAM, has virtually no application software, and has no off-the-shelf peripherals as of today. Obviously we expect the market to grow as we provide those things. The highest priorities on our own list, i.e., items that will be manufactured by Sinclair, are a 16K RAM module, which comes in a small case about 2.5" square and plugs in to the back of the main board through an edge connector.

Our next add-on option for the unit will be an 8K extended Basic, which will no longer be integer, but will be a fully floating-point Basic with multi-dimensional arrays, powerful string handling capabilities and a whole host of other features. I don't think it is an exaggeration to say that just as our 4K integer Basic is considerably more powerful than Radio Shack's Level I

Basic, so our 8K Basic will be considerably more powerful than Radio Shack's Level II Basic and, indeed, more powerful than Microsoft's 8K Basic. We are also working hard to produce exciting applications software.

Consider VisiCalc. It stands almost all alone as a software package, it is so good that people buy computers just to use VisiCalc and never use their computer for anything else. It may be that VisiCalc is the only software package that justifies the $1000-$2000 expenditure to buy an Apple, Atari, Commodore or whatever.

We envisage that there will be a lot of software packages that are to the ZX80 what VisiCalc has been to those larger computers. Our software packages will certainly be very powerful, but it doesn't take as much to persuade someone that it is worth going out and spending a couple of hundred dollars to be able to use a particular package. We have some software packages, which are scheduled to be released soon that I think are going to be as well known a year from now as VisiCalc is.

Not only has Sinclair satisfied itself that there exists a huge consumer market, but other people also have been persuaded that there is a large market. Consequently, Sinclair is not the only company that is working to support the basic product. In addition support provided by publications such as *SYNC* magazine, there are at least three independent companies that are far advanced in their plans to manufacture and market hardware and peripherals for the ZX80, including a general purpose interface which will include an RS-232 and also support disk drives. There are numerous companies developing software and printed materials. Image Computer Products is the semi-official software supplier in the U.S. for the ZX80. While most of the programs in their current catalog, because they are designed to run on the basic 1K machine, are certainly no great advance on anything you have seen before, Image is working hard on more advanced, complex programs which will be available as soon as the larger memory and extended Basic are available.

ZX80 Features

Some of the features of our machine that make it appealing to the first time user include the following:

Any line of Basic or command at the system level that you enter to the machine will appear at the bottom of the screen. When you enter a line into the program it will go to the top of the screen. The cursor originally appears as a "K," for "keyword," a Basic command word. As long as you only enter digits they will be entered and the cursor will move along and remain in the "K" mode. As long as the cursor is in the "K" mode you can enter at a single key stroke any of the Basic command words that are available. It will be entered and appear in its full format as a result of a single key stroke. So without the use of a shift key or anything else you can hit a single key and get "SPACE PRINT

This feature eliminates a great deal of typing and is extremely convenient when you get used to it. More importantly, the ease of input is mirrored by the simplicity and economy of storage, because the seven characters involved in "SPACE PRINT SPACE" are stored internally as a single byte. Offering a machine with only 1K bytes of user memory, we had to,be very mean in our use of memory and thus we have employed a number of tricks, or data compression techniques, to minimize the amount of memory the program storage takes.

The 1K bytes of user memory are dynamically allocated between program storage, working space and display. There is no separate video circuitry in the machine. As you get close to your memory limit with a program being stored, you have a decreasing amount of memory available for the storage of the information which is to be displayed to the screen. You then have to look at the results of your program in chunks. It will display as much as it can—usually a full screen, but sometimes 3/4 full—and then you have to continue execution to see the remaining results. Obviously with a 16K memory it is going to be relatively rare that you would run into that constraint. The boundaries in the memory are not fixed; there is no memory mapping.

If you enter a character such as + (plus) and say PRINT +, the machine recognizes that that is syntactically incorrect. That is not a meaningful statement in the Basic language. There is no way that a PRINT +can ever be continued in such a way as make it a statement, therefore the machine, indicates with the symbol "S" a syntax error. By its position it tells you where the error is located and, thus you must make a change at the point indicated before the line will be accepted into a program.

If you are a beginner learning to program, you won't have the frustration of entering a large program only to find at run time that you have put a comma at every place you wanted a semi-colon or that you misused some other feature of the language. You will be made aware of any syntax error in your program before you can complete that line of the program.

One of the disadvantages of the machine which makes it unsuited for games applications is that the microprocessor drives the display. Consequently, when it is computing it isn't displaying and when it is displaying it isn't computing. When you tell it to EXECUTE a program, it goes ahead and EXECUTEs the program and

when it is finished it displays the results. It tells you at the bottom where the execution terminated and what the termination condition was. Because we never allow a program to be entered with syntax errors in it, the termination errors are few.

After manufacturing was started, some of our workers decided to make the ZX80 compute and display at the same time. They were successful, and there will in some future version of the ZX80 be the facility to compute and display. Some people have already written machine code routines which enable the existing machine to display and compute. It is programmable in Z80 machine code but there is not an assembler available at the present time. There are PEEK and POKE commands and you must enter the machine code instructions one at a time with a POKE command. You can then call and execute them with a USR command.

Future Speculations

I would like to take the opportunity now to speculate. I am going to present some fantasy rather than fact, but I believe

that it is in the nature of most science fiction—fiction that will come true.

I think we are going to see the development of at least three different types of what have hitherto been known as personal computers, to serve the needs of three distinct markets. One of those markets will be the small business market; one will be the educational market; and the third (of which the ZX80 is perhaps the first) will be a market for truly personal consumer-oriented computers.

Within each of these markets, let's consider three elements of computer system design. In particular, what will be the primary means of input to a personal computer? What will be the primary method of data and program storage? What will be the primary means of output?

Future Types of Input

Each of the three markets—business, education and personal—have different needs in each of those areas. If one allows one's imagination to run wild it would seem that the most convenient method of input for a personal computer would be to communicate with it the way we communicate with anything else that we think is intelligent, the way we communicate with other human beings. That is, we should be able to speak to it and have it understand us in a fairly free format manner.

Obviously, a lot of people have thought about that and some people have done something about it. In fact, one company sells a voice input peripheral for less than $120 and is going make a version of it for the ZX80. At this stage, it's capability is limited to a small number of commands that you have to speak fairly consistently in order that it will understand. I have no doubt that speech input will be the primary method, if not in the next 10 years then in the decade following, for the personal/consumer computer.

When you come to the business computer, I think the keyboard is going to remain the dominant method of entry. Business has a huge pool of people who are used to using keyboards, who can use them very effectively, and will continue to do so. The typical business executive is not going to be interested in learning to use a keyboard any more than he or she has been interested in learning how to use a typewriter, a copying machine, or any other piece of equipment in his or her office. The business executive is not even going to be interested in learning how to talk to it. It will always be easier for him to buzz his or her secretary on the intercom and say "Do this or that on your computer or computer terminal." It is going to be easier for him or her to get something done than to do it for himself. Consequently, I think the primary method of input for small computers in business is going to be via a typewriter keyboard by exactly the same people who use keyboards now in business.

On the other hand, the educational market has, at a certain level, a requirement for input that isn't served by either voice or keyboard input. I'm thinking of the young child for whom I think the personal computer is going to be a large part of his or her life and education. Clearly that child is not going to be able to use a keyboard with facility, nor is the child readily going to be able to learn to speak to the computer in a way that the computer will understand. In fact, one reason that child will use the computer for education is that he or she cannot do those things with facility. I think the primary method of input for the home education market is going to be a touch sensitive screen.

So we have for the three main market areas three methods of input: speech for the consumer computer, keyboard for the business computer and touch video for the educational market. I don't want to suggest that these are the only methods of input that are going to exist or that they are the only ones that are going to exist in those markets, but I think they are the primary ones.

Memory and Data Storage

Just as we have found out that the disk is more convenient than cassettes, I think we will find that bubble memory or something of equivalent capacity, convenience, speed and low potential cost will be far superior to any medium such as disk, stringy floppy or cassette where you have one device that reads and writes the data and another medium on which you store the data. For both the consumer and business markets, I feel the primary method of data storage is going to be bubble memory or its equivalent if it is overtaken by some other technology with similar but improved characteristics.

In the educational market there may be a need for something that will look very much like a video disk. There seems to be a demand for a large quantity of data in excess of what can be stored within any affordable amount of bubble memory. The same programs which teach a child a given discipline in 1990 will probably be perfectly adequate in 1995. So there is a demand for a huge amount of data that does not change. I think that the video disk or something very similar to it is going to meet that need.

In the business and consumer markets there will be a heavy dependence upon remote data banks, with software programs and data being down loaded from a larger system at high speed over a telephone line. I foresee problems with being in constant communication with a remote data bank, but I feel the down-load mode is going to be extremely important. It also solves what otherwise will become a major problem of piracy of both software and data. It will be cheaper for a one-time download than to make a copy and have a means of storing that data or software.

Output in the Future

For the consumer computer I have no doubt that the primary form of output is going to be video. It has the huge advantage that most of us can and still will be able in 20-30 years time to read. We can take in a lot of data at once, far more than we can hear. We can see far more at a glance on a screen. The screen need not be large as long as the resolution is there, a screen of 2"diagonal is more than sufficient to present the information that most of us would require. We can look at a 2" screen with sufficient resolution and see what we want as readily as we can look at a 2"photograph or read that area on the page of a telephone book.

In the educational area video is not so useful. Although I have suggested a touch sensitive screen, probably more important to the educational market for the child is sound output. A child will respond to sound

better than to video. It will capture his attention when his attention might have wandered from the screen. It will convey information to the child which cannot be given visually because the child can understand the spoken word but can't read with great facility. Indeed the computer might be doing the job of teaching the child to read.

In the business market, I forsee primarily printed output. Again, there will be a screen on the small business computer or office terminal. But for the same reasons I cited before, when a business executive wants to see last week's or yesterday's or the last hour's sales figures, he or she will push a button and say, "Get me the sales figures." The screen and the keyboard will be outside the executive's office. Somebody will key in the information necessary to get the required data, and that data will come out on a sheet of paper, which can be handed to the boss. That is the way business executives deal with information—on paper. I think there will be a major problem getting them to deal with it on a screen.

That is not vastly different from the way in which computers are being used today in schools, although the system I am speaking about will be widely used in homes. However, while I say the system will not look that much different from the ones in use today, it will be vastly more powerful and less expensive.

It is in the personal area, in which I have suggested that the ZX80 may be the first computer, that I see the greatest change. I have suggested that the computer will have speech input and video output, for which a 2-3" diagonal screen will be quite sufficient. The memory will be of a bubble type that will be extremely compact and in which we will see the same type of advances that we have seen in semi-conductor memory over the last 15-20 years. The only other thing that will be needed is some means of connecting this computer to a telephone line so that you will be able to get data down-loaded from a large computer. Just as calculators took 10 years to get to their present size, so personal computers will be the size of the calculators of 10 years ago. And they will incredibly inexpensive.

it and integrate it into your life, you would no more go out of your house or indeed be anywhere in your house without your card, than you would without your clothes. About the only place you won't have it with you is in the shower.

I hesitate, but only momentarily, to go one step further and suggest that instead of looking 15 years ahead we double the period and look 30 years ahead. You may have the opportunity in 30-40 years time literally to insure that you will no go anywhere without your computer because it will be inside your body. Of course that will require that it be interfaced with your brain so that all you have to do is think that you would like to know the contents of page 73, volume 21, of The Encyclopedia Britannica, and no sooner will you have thought it than you will see it. That I believe is not impossible. I am not sure that it isn't dangerous. I know what is dangerous, and that is to think that it is not possible, if indeed it is going to happen. That is really dangerous. So whether we like it or not, whether we are scared by it or not, it is something that we need to think about. □

In the business market I see something that is going to be operated by the people who today operate typewriters. It will look exactly like a typewriter with the addition of a screen. It is going to look like one of today's word processors. It will have a keyboard to key in information and a very extensive solid state memory of its own. It will be connected remotely to larger computers and will have printed paper output. It will also have a screen for editing, word processing etc., but that will not be the primary method of output. Strangely enough, I don't see, in the one area that has already adopted small computers, a great deal of change.

In Summary

In education, I see a computer with a screen that will serve as an input device almost as much as an output device with a heavy dependence on sound output. It will probably have some voice input, depending on the age of the child and, in many instances something equivalent to a video disk that has the lessons—in David Ahl's words, the "courseware"—on the disk.

Ubiquitous and Essential

However, what will be expensive is the capability to utilize the power that will be put in your hands, and I think that will create a major social change. Not because there will one group of people who can afford the "credit card computer" and another group that can't: the question is who will choose to use it? Who will have the imagination and the basic education to use it? Maybe with the use of computers in education everybody will, but initially I think there will be a two groups of people that have been described by someone else in contrast to the haves and the have nots. These will be the "knows" and the "know nots." There will be some people who will know how to gain access to and manipulate for their own advantage just about any piece of information they can possibly wish to have. The "credit card computer" will revolutionize the way you do your job and the way you live your life. This card will replace all the others. It will be your electronic funds transfer card; it will be your personal ID; it will be everything. It will be so important to you that if you use

puzzles & problems

"I, Merlin, challenge the readers of SYNC to a battle of wits. Every issue I shall propound a series of problems for you to solve. They will vary in complexity and category and should prove an interesting diversion between articles. I also welcome contributions from our readers. If you have a favorite puzzle you feel will be of interest to us send it in. If I use it I shall send you a copy of one of my famous MERLIN'S PUZZLER books.

"And now to work! Our first puzzle is called . . . "

THE JUMPING JACK PUZZLE

This is a great old puzzle. Draw up a rough board like the one shown at the right. Place a checker in every square that has a letter in it. Your problem is, starting with the checker in the square marked "I", to remove all of the checkers from the board, save one, and have this last checker end up where you started in square one. You can jump one checker over another checker in any direction, sideways, up and down, or diagonally. Whenever you jump over a checker you must remove it from the board. However, as in checkers, the square beyond the checker you are jumping over must be empty. A continuous series of jumps, using one checker, will be considered to be one move. You must solve the puzzle in just *four moves*. On your mark, get set, start jumping!

	A		B	
C	D	E	F	G
	H	I	J	
K	L	M	N	O
	P		Q	

THE BARREL PUZZLE

Now, this puzzle should prove to be "barrels of fun" for all of you! Printed on the three barrels pictured below are the numbers "1," "6" and "3." Can you rearrange these barrels so that you have a 3 digit number that can be evenly divided by seven? The answer is quite ingenious.

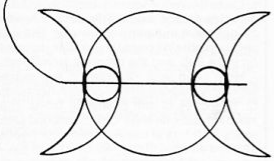
THE GOLDEN HELMET PROBLEM

The management at the Peabody Helmet Company discovered that one of the ten artisans that make their line of solid gold helmets has been cheating them. It has been determined that one of the men has been using one ounce less gold per helmet than required, and, he has been keeping this ounce of gold for his own use. Mr. Peabody, a man who is always in a hurry, has decided that he can determine who the thief is with just one weighing on the large company scales. One thing should be pointed out, each artisan stamps his initials inside each helmet that he crafts. Now, how could he find out, with just one weighing, who the guilty party is?

THE DOUBLE HORNED ZAT PUZZLE

Our last puzzle comes from the book "Merlin's Puzzler 2" and is called The Double Horned Zat Puzzle. (See the drawing at the left). Merlin claimed that this is a very powerful magical sign and is useful in warding off pests, the humors, and bill collectors. To generate the maximum amount of magical power you must draw the sign using one continuous line. Also, no part of the line may cross over any other part of the line.

Now, practice well, you never know when a good hex sign will be needed.



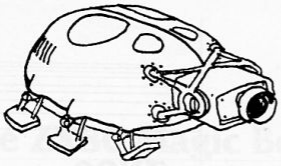
Answers on page 44.

I hope that you enjoyed Merlin's puzzles. Now don't forget, if you would like to try and stump our readers with one of your own puzzles, just send it along. If Merlin uses it he will send you a copy of one of his books that deal with puzzles, games and magic.

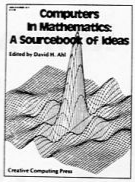
So long until next time . . .

Your editor,

Charles Barry Townsend



Have You Been Bitten By The Computer Bug?



Problems for Computer Solution

Here are 90 problems with a thorough discussion and references for each. Eleven types of problems are included, for example, arithmetic, algebra, geometry, number theory, probability and science. Even includes three classic unsolved problems and seven appendices. 104 pages softbound, $4.95 [9Z].

The teacher's edition contains solutions with complete listing in Basic, sample run and in-depth analysis explaining the algorithms and theory involved. 280 pp softbound, $9.95 [9Y].

Katie and the Computer

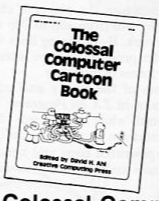
Fred D'Ignazio and Stan Gilliam. This is a delightful story told in words and full color drawings of Katie's adventures when she "falls" into a computer. In Katie's journey through the land of Cybernia she meets the Software Colonel, the Bytes, the Table Manager and even a ferocious Program Bug. Her journey parallels the path of a simple command through he stages of processing in a computer, thus explaining the fundamentals of computer operation to 4-10 year olds. Supplemental explanatory information is contained in the front and back end papers. 42 pp. hardbound $6.95. (12A)

Two Free Catalogs

Send for our big 20-page **Book Catalog** featuring a full line of Creative Computing Press and Book Service titles, back issues of Creative Computing Magazine, t-shirts, posters and games. A **Sensational Software Catalog** of over 400 outstanding microcomputer programs is also available. **Each** package is outlined in detail with accompanying screen photos and illustrations. Make the most of your computer resources with **Creative Computing!**

Computer Coin Games

Computer Coin Games by Joe Weisbecker aids newcomers to the field of computers by simplifying the concepts of computer circuitry through games which can be played with a few pennies and full sized playing boards in the book. Enhanced by outrageous cartoons, teachers, students and self-learners of all ages will enjoy this 96 page softbound book. [10R] $3.95.



The Colossal Computer Cartoon Book

Edited by David H. Ahl. The best collection of computer cartoons ever is now in its second printing. There are fifteen chapters of several hundred cartoons about robots, computer dating, computers in the office, and much more. Keep the book with your reference works. When needed, the right cartoon can say it all for you. Provides hours of fun and comic insight. 120 pp. 8½ x 11" softbound. (6G) $4.95.

Computers in Mathematics: A Sourcebook of Ideas

Here is a huge sourcebook of ideas for using computers in mathematics instruction. This large format book contains sections on computer literacy, problem solving techniques, art and graphing, simulations, computer assisted instruction, probability, functions, magic squares and programming styles.

One section presents over 250 problems, puzzles and programming ideas--more than is found in most "collection of problems" books.

Pragmatic, ready-to-use, classroom tested ideas are presented for everything from the most basic introduction to binary numbers to advanced techniques like multiple regression analysis and differential equations. Every item discussed has a complete explanation including flowcharts, programs and sample runs.

The book includes many activities that don't require a computer. And if you're considering expanding your computer facilities you'll find the section on how to select a computer complete with a microcomputer comparison chart invaluable.

Much of the material has appeared in **Creative Computing** but the back issues are no longer available. Hence this is your only source to this practical and valuable material. Edited by David H. Ahl,this mammoth 224-page softbound book costs only $15.95. (The individual issues, if they were available, would cost over $60.00). [12D]

To Order

Send your check for books plus $2.00 shipping and handling per order to Creative Computing, P.O. Box 789-M, Morristown, NJ 07960. NJ residents add 5% sales tax. Visa, Master Charge or American Express are also acceptable. For faster service, call in your bank card order toll free to

800-631-8112

(in NJ, call 201-540-0445)

creative computing

P.O. Box 789-M, Morristown, NJ 07960

Reviews

Jonathan A. Stein

ZX80 Programs

ZX80 Programs, Volume 1 edited by Chris Denning; published by Zipprint, 418 Poole Road, Parkstone, Poole, Dorset BH12 1DF, England. 82 pages, paperback. £5.25 in the U.K.; $13.00 in the United States.

Although intended for all ZX80 users, this book is aimed especially at people wih little prior computer experience. The book contains twenty programs, of which nine are "practical" programs for business and home, three are for educational purposes, and eight are games.

Stock List, one of the utility programs, is intended for keeping an inventory of your stocks. Although the program will only hold thirty entries, this problem can be bypassed by storing several copies on cassettes.

The next utility program, Wavelength/Frequency Conversion, converts wavelength measurements from kiloherz to meters, and vice versa. This function may be useful for people in The United Kingdom to convert radio station dial locations from meters to kiloherz, in which imported radios are calibrated.

The *VAT Calculator* program was intended to calculate the 15% sales tax found in The United Kingdom, although any tax rate may be entered. This program determines the amount of tax paid, when only the final total is known; it does not calculate the tax to be paid on the original purchase price.

The educational programs are *Basic Maths (Add/Subtract)* and *Basic Maths (Mulitiply/Divide)*. Both present drill and practice excercises, offering three levels of difficulty, as well as "emergency" features which provide the answer if the student cannot solve the problem. The two math programs share many of the same lines, making it possible to avoid much of the tedious process if the first program has already been saved.

Prime Number Tester, is an educational tool to allow to the user to determine whether or not a number is prime.

The game, *Brain Teaser*, is a pattern matching game, similar to the hand held electronic games, *Simon* or *Einstein*, in which the player must duplicate a pattern. In *Brain Teaser* the pattern is a series of twenty numbers, presented individually, with two levels of difficulty. When the game is concluded, either by an incorrect entry, or by successfully duplicating the series, the program posts the score out of the possible twenty correct turns.

The next program, *One Arm Bandit* simulates the operation of a slot machine. For each turn the player is charged 5p, and may win from 10p to 70p depending on his luck. By changing lines 90 through 120, the program may be altered to suit other forms of currency.

Some of the utility and educational programs in *ZX 80 Programs, Volume 1* are worth while and several of the game programs are quite good. Unfortunately, the typewritten copy and hand scrawled programs make it difficult to read. □

ZX80 Pocket Book

The ZX80 Pocket Book) by Trevor Toms; Phipps Associates, 3 Downs Avenue, Epsom, Surrey. KT18 5HQ, England. 109 pages, spiral bound. 4.95.

The ZX80 Pocket Book differs from other books about the ZX80 because it is aimed at the experienced programmer.

The book begins with a review of the Sinclair Basic, which unlike the sections in other ZX80 books, merely highlights this material from the original manual, instead of repeating it at length.

The section on programming is particularly valuable because it offers simple tips on how to become a better programmer, as well as suggestions on how to better use many ZX80 features. This section is not a how-to programming manual, but is a well presented chapter on improving programming technique.

The chapter, "Program Storage and Retrieval," offers useful instructions on how to safely save programs on cassette. The tips presented are both those of the author and Sinclair.

Section 5, "Data File Storage and Retrieval," discusses how to preserve both data and programs without using the DATA statment present in other Basics. The authors suggest saving important files in triplicate to ensure that the information will not be accidentally erased.

The ZX80 Pocket Book also contains thirteen programs, including a *Machine Code Conversion* and a *Share Valuation* program which keeps track of shares of stock and the price at which they were purchased. This latter program requires 3K of RAM.

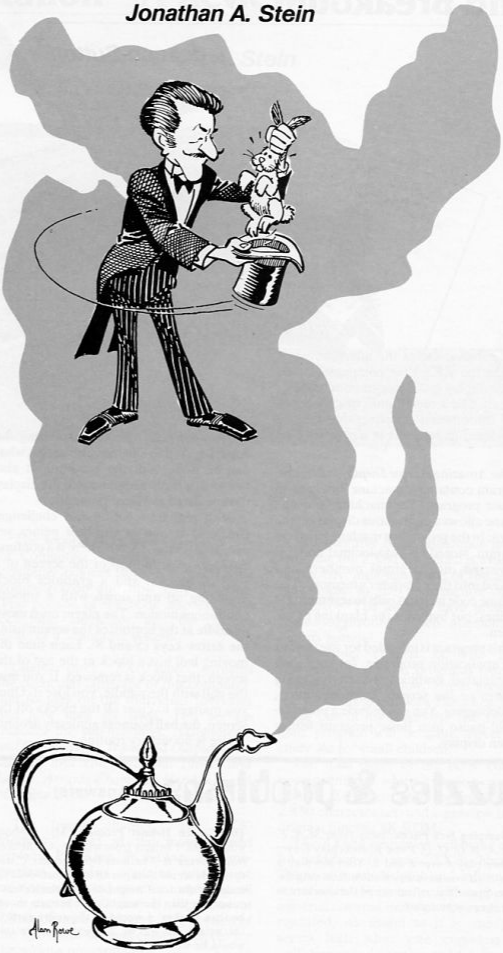
The final full section of the book presents all the ZX80 Basic commands and explains how they are used. Examples are offered, so one may easily see what these commands are intended to do.

The ZX80 Pocket Book ends with five appendices for the instruction set, the ZX80 Character set, graphics, error codes, and a command summary.

Phipps Associates has succeeded in giving the ZX80 community a useful, attractive volume at a moderate price. □

The ZX80 Magic Book—A Review

Jonathan A. Stein



The ZX-80 Magic Book is published by Timedata Limited, 57 Swallowdale, Basildon, Essex, England. 60 pages, softbound, £ 4.75.

This book offers both programs and instructions on how to make better use of this popular computer. The second half of the book contains material which helps one to understand how the ZX-80 works.

The text explains how to create and debug programs, how to convert from other Basics, and includes such information as instructions for improving the television picture, how to connect a video monitor in place of a television set, and other useful technical information.

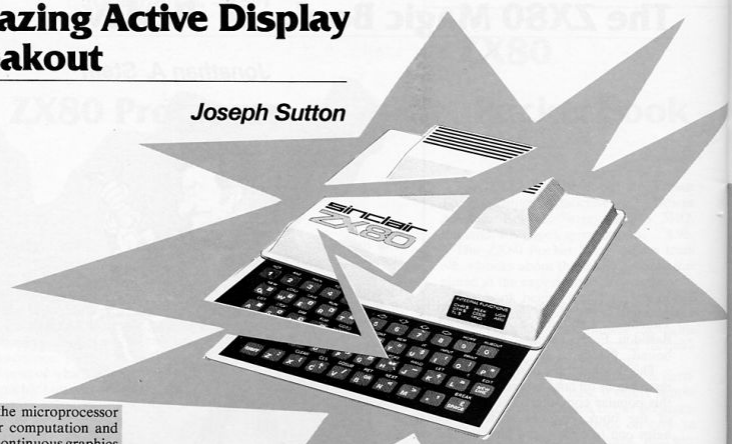
The twenty-one programs include *Russian Roulette*, which I hope no one takes too seriously; *Moon Lander*, a version of *Lunar Lander*, which is quite a difficult game; and a 2K version of *Hammurabi*, which is one of the best ZX-80 games I have played. This well known computer game was the subject of the SYNC Challenge last issue. As the ruler of ancient Sumeria, you must make all vital decisions on how to run the city, such as: how much grain is allocated for food and seed, how much land is traded for grain, and vice versa. On each turn you are told how much land and grain the city has and how the population fared from your previous decisions. If you calculate carefully, the population increases, and the area of the city grows. If not, people starve and the city loses land.

The other games are a mixed lot — some good, some not as good. A few non-game programs are also included such as *Sums Tester* and *More Sums*. *The ZX-80 Magic Book* should be a useful addition to most ZX-80 libraries, particularly for its technical sections. □

A Review

The Amazing Active Display and Breakout

Joseph Sutton



The shared use of the microprocessor chip in the ZX-80 for computation and screen display makes continuous graphics difficult. The screen flashes whenever you input information, an effect that may be considered undesirable in a game.

The Amazing Active Display is a hybrid program containing machine language in a Basic program. The machine language routine allows a continuous display on the screen. In the program, a machine language program, stored as hexadecimal code, is converted into decimal numbers and POKEd into the computer's memory. The machine code includes calls to several ROM routines, but bypasses the blanking of the screen.

This program is intended for use in your own application programs, for uses such as animated graphics, games requiring updates to the screen, instrumentation, and debugging. You can provide a programmable pause in a Basic program with a screen display.

Breakout is a game program using the Amazing Active Display, and shows what can be done with the program. It also comes as a Basic program with the display routine listed as Hex characters.

With real time action and challenge, Breakout is one of the best games yet available for the ZX-80. There is a continuous display at the top of the screen of a wall of blocks, and a graphics block bouncing up and down with a smooth continuous motion. The player must move a paddle at the bottom of the screen using the arrow keys (5 and 8). Each time the moving ball hits a block at the top of the screen, that block is removed. If you miss the ball with the paddle, you lose it. Once you manage to clear all the blocks off the screen, the ball bounces aimlessly around. There is no victory routine.

There are several weak points in the game. There is only one speed for the ball, no variations of the basic game, and no provisions for score keeping. The only way to stop the game is to unplug the computer. The Break key does not function. The instructions are quite minimal, and it may be difficult to understand the program well enough to use it in your own programs.

We like both programs, and enjoyed playing with them.

The Amazing Active Display ($20/£10)
Breakout ($20/£10)

K. MacDonald
26 Spiers Close
Knowle, Solihull West Midlands, B93 9ES
Great Britain

Joseph Sutton, 170 S. Hillside Ave., Succasunna, NJ 07876.

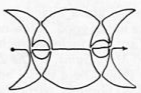
puzzles & problems (Answers)

The Jumping Jack Puzzle: Jump *I* over *M, N, F, D, C, A, B, G, O, Q, P* and *K*. Next play *L* over *H*. Then play *J* over *E* and *L*. Your last play is *I* over *J*. (If you have any trouble following the answer place "letters" on top of the checkers to make it easier to follow).

The Golden Helmet Problem: Mr. Peabody would take 1 helmet from worker *A*, 2 helmets from worker *B*, 3 helmets from worker *C*, and so on. He would then put all 55 helmets onto the scales. If the total weight of the helmets was 1 ounce less than the weight of 55 helmets should be than worker *A* would be the guilty party. If the weight was out by 2 ounces then the thief would be worker *B*, etc.

The Barrel Puzzle: Turn the six upside down. You now have a 9. The number we want is 931.

The Double Horned Zat Puzzle:



The ZX80 Companion—A Review

Jonathan A. Stein

The ZX80 Companion, by Bob Maunder, Terry Trotter, and Ian Logan; Linsac, 68 Barker Road, Middlebrough TS5 5ES, England, 128 pages, softbound, £10. Available in The United States through Image Computer Products, 615 Academy Drive, Northbrook, Illinois 60062, $19.95.

This book is designed to supplement the original ZX-80 manual. It includes tips on the operation, assembly, and programming of the ZX80 for maximum effectiveness. Although the original manual is useful, and gives a beginner clear, concise directions on the operation and programming of the computer, *The ZX80 Companion* continues beyond the offerings of that manual.

The book is divided into six chapters, and has two appendices. Chapter One, "Operating the ZX80," has five sections, explaining basic operating procedure for The ZX80. This section contains many of the tips contained within the authorized manual, although there are hints not found in the Sinclair booklet.

The second chapter, entitled "Theory of Computers," is especially useful for computer novices, since it explains what computers do and how they function. This portion of the book also examines computer languages, and the components of a computer system.

Chapter three, "ZX80 Basic," explains the particular language of this computer, and focuses on the individual functions and how they are used. This section includes the sub-headings: *Assignments and Calculations*, *Input and Output*, *Conditional and Unconditional Jumps and Loops*, *Arrays*, *Character Handling*, and *Machine Code*.

The fourth chapter, "The ZX80 Monitor," looks at the monitor, and explains PEEKing, POKEing, and subroutines. It also contains sub-routine memory maps and monitor maps, as well as a cursor table and a discussion of ZX80 machine language. This portion of the book will be particularly helpful for the serious programmer.

Chapter five offers assistance in constructing the ZX80 kit, which is available in the UK, although some of the advice may apply to the construction of the MicroAce. The authors have presented some suggestions which augment the factory manual.

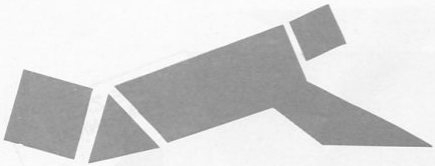
The final chapter of *The ZX80 Companion* contains programs from the Linsac tape collection. These programs are generally good, with *The Maze game* being one of my favorites. The *Spelling Quiz* program will be quite useful as a study aid for small children.

The final pages of the book contain two appendices, which include a comparison of Z80 instruction codes and the ZX80 character set, and a preview of the original Sinclair 8K ROM.

The ZX80 Companion will make a useful addition to most ZX80 libraries. There is much helpful and informative material in the volume, although some material covered in the owner's manual is repeated. As useful as it is, the price seems high when one considers the undistinguished printing and binding. □

Disassembled BASIC —A Review

David Lubar



Disassembled Basic, Linsac, 68 Barker Road, Middlebrough TS5 5ES, England. 48 pages, paperback.

Linsac has published a disassembled listing of the Sinclair Basic. What, you may ask, is a disassembly? Well, when machine language is placed in memory, either in RAM or ROM, it is stored as numbers. These numbers represent machine-language instructions, called "mneumonics" since they are abbreviated versions of the command. For example, an increment instruction is represent as INC. When a programmer enters a program using these or op codes, an assembler is used to turn the commands into numbers. A disassembler reverses the process, turning the numbers into lines of code so humans can understand it.

A few lines of disassembled code look like this:

```
07BE 2B      L07BE   DEC     HL              ;**INTERPRETER**
07BF 222640          LD      (L4026),HL      ;SAVE PROG ADDRESS
07C2 210000          LD      HL,L0000        ;ZERO  HL
07C5 00              NOP                     ;  AND 'NEW' CODE
07C6 221540          LD      (L4015),HL      ;ZERO POINTER
07C9 211940          LD      HL,L4019        ;POINT TO SYNTAX FLAGS
07CC CB6E            BIT     5,(HL)          ;TEST IMMEDIATE FLAG
07CE 2807            JR      Z,L07D7         ;JUMP IF IMMEDIATE
```

The first column contains the memory locations of the code. In this example, the code starts at hex address 7BE. The next column shows the hex value or values for the command. Commands require from one to three bytes, depending on how much information is required. For instance, decrementing a register requires only one byte. Loading a single register with a number would require two bytes, one for the command and one for the number to be loaded. Loading a register with a value from a memory location requires three bytes, one for the load command and two for the value of the location. The third column is for labels. In this case, the labels used are just the locations of the commands prefaced by an L (labels must start with a letter, though they can contain numbers). There are two uses of labels. First, when assembling the code, you can refer to the label instead of the location. Thus, if you have a segment of code that multiplies two numbers, you could label it MULT. When you wanted to jump to that code, you could say JP (for jump) MULT. When the program is assembled, the correct value for MULT will be inserted.

The other use for labels is to make the code easier to read. When you are scanning a disassembly and see MULT, you know you have found the multiplication routine. Unfortunately, the Linsac disassembly doesn't use this kind of label, making the listing harder to decipher. The label L07BE tells us that the line is at 7BE, but doesn't give us any useful information.

The fourth column contains the mneumonics, followed by any values used by the commands. The final column contains comments, making up in part for the lack of meaningful labels. The comments help, but they are vague in places.

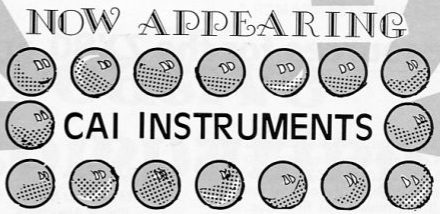
Stepping through the code, the first command decrements a pair of Z-80 registers called H and L. Next, the value in these registers is stored in memory. The HL registers are loaded again, this time with values from another area of memory. The NOP command stands for "no operation." Possibly, this byte was left open for future changes in the code, or was created by an earlier change in the code. The comment isn't clear. The next two commands again place the register values in memory and reload the registers. The BIT command tests a value stored in location 5 to see whether the command is from a program or from the immediate mode. If the command is immediate, the zero flag is set. The next line jumps to 7D7 if the zero condition is met. Otherwise, the program falls through to the next line (not listed here).

With a disassembled copy of Sinclair Basic in hand, an experienced programmer can work wonders. He can access any of the Basic functions through calls to the monitor, using the USR function. Even more powerful is the potential to use the Basic subroutines from machine language programs. There is no need to write your own PRINT subroutine when you can go through one that already exists in the monitor. In essence, the disassembly puts the full power of Basic at the programmer's disposal, even when he is writing programs in machine language.

The listing comes fully commented, with asterisks marking the function of each subroutine, such as *LIST*, and *PRINT CHARACTER*. As mentioned earlier, there are places where the comments could be clearer. The printout is a Xeroxed copy of a dot-matrix listing, but it is legible. The listing, including a table of referenced lines, is 48 pages long.

Besides being a good tool for programmers, this book could also be used by someone who is learning Z-80 code. By seeing how the commands are used, and what the results are, you can gain an understanding of this machine language.

Product Preview



This section is intended to preview the peripherals and accessories that will be available for the ZX80. Readers are encouraged to submit reviews of products mentioned here.

CAI Instruments of Midland, Michigan is developing a series of interfaces for the Sinclair ZX80. The company has announced a PC board which will plug into the ZX80 expansion port, in front of the 16K RAM module. This peripheral will work with a printer and includes an improved interface for storing programs on tape. The basic package includes the board, printer and tape recorder, and will sell for approximately $200. Other packages will be available with varying qualities of printers and recorders. One system, containing a printer, the board, and a high quality tape system will cost less than $350. CAI is also working on a full size plug-in keyboard for under $50. The company also has interfaces to work with floppy disks. CAI Instruments, 2559 Arbutus Court, Midland, MI 48640.

Innovision of Los Altos, California is devising an improved graphics device for the ZX80. The company will have a hand printed graphics system using memory mapped video. This peripheral will be available once the 8K ROM is released. Innovision, P.O. Box 1317, Los Altos, CA 94022

Voicetek of Goleta, California has been authorized by Sinclair Reasearch Limited to develop a speech recognition unit. The project has been delayed until the introduction of the ZX80 16K RAM. Voicetek, 6690 Abrego, Goleta, CA 93017.

Sinclair Research Limited now expects to have the 8K Basic ROM for the ZX80 available by March 1981. The new Basic allows the use of floating point arithmetic to nine digit accuracy, and contains log and trig functions with inverses. This ROM facilitates improved graphics, and contains 37 new functions including DRAW, DATA, ARCSIN, VAL, and SCROLL. The plug-in conversion includes a new keyboard template and a supplementary manual, and should sell for $40. Sinclair Research Ltd., 50 Staniford St., Boston, MA 02114.

Sinclair has also officially announced the long-awaited 16K RAM expansion for the ZX80 computer. The module has received FCC certification and is expected before the end of January 1981. The unit is housed in a plastic box the size of two cassette tapes, and plugs into the expansion port at the rear of the computer. Sinclair is now taking orders for the $99.99 RAM.□

Resources for the ZX80 and MicroAce

In the last issue SYNC presented a list of 24 companies and organizations with products related to the ZX80. Below there is a supplement to the original resource list. If you missed that first issue, send $2.50 or £1.25 to SYNC, and we will gladly send it.

We welcome entries from manufacturers and readers for the resources column. Please include the name of the item, a brief description, price, and complete data on how to obtain it. Send contributions to SYNC Resources, 39 East Hanover Avenue, Morris Plains, New Jersey 07950.

Software

Six programs for the ZX80 — £4.95 U.K.
Bramwell Enterprises
87 Anderson Crescent
Great Barr, Birmingham B43 7ST
England

Graphics for the 1K ZX80
C12 Cassette with four programs: *Symmetrical Patterns, Large Print, Draw a Picture, Plot a Picture*
Cassette and manual — £8.00
Graphics manual only — £3.00
Bridge Software
36 Fernwood
Marple Bridge
Stockport, Ches. SK6 5BE
England

Self-Instruction Courses for the ZX80
Cambridge Learning Ltd.
Unit SS
Rivermill Site
Free Post, St. Ives
Huntingdon.
Cambs. PE17 BR
England

Assorted games packages for the ZX80
4K — £5.95 and — £9.95
Database Consultancy
105 Fairholme Ave.
Gidea Park, Romford
Essex RM2 5UR
England

Assorted software for the ZX80 Kala — £2.75
Hewson Consultants
7 Grahme Close
Blewbury, Oxfordshire QX11 9QE
England

The ZX80 Pocket Book — £4.95 U.K.
Phipps Associates
3 Downs Avenue
Epsom, Surrey KT18 5HQ
England

ZX 80 Software:
Pointasketch, Nim, Mothership, Pharoah's Tomb, XOX, AddMax, SubMax, MultiMax, DivMax, (decimals-10 digit), Hotel (50 room bookings), Playmath, Biorhythms.
Send SASE for detailed list and free program.
Datalog
20 Aylen Rd.
Portsmouth, England PO3 5HB
England

ZX80 4K Programs:
Star Trek, Pontoon, Adventure - £5.95 (£2.45 each)
M. Bates
Dever Barn
Micheldever
Winchester, Hants
England

Cases for ZX80 Systems
Phoenix Management Services
Paragon House
46 Kent House Lane
Beckenham, Kent
England

ZX80 Software: *Moon Lander, Pontoon, Calendar, Maths test* — £2
P. Pickering
56 Lennox Road
Todmorden, Lancs. OL14QD
England

Software Cassette: *Random number guessing, Memory Test, Reaction Speed Game, Hangman* — £6.00

Educational Aid Cassette — £5.00
Rose Cassettes
P.O. Box 28
Solihull, West Midlands B91 3LU
England

Tutor programs for 1K and 2K ZX80
1K listing — £3.00 ($7.00)
2K listing — £5.00 ($12.00)
Wyncom
11 Furze Platt Road
Maidenhead Berks. SL6 7ND

Hardware

Interfaces and Printers for the ZX80. Packages from $200.00
CAI Instruments, Inc.
2559 Arbutus Court
Midland, MI 48640

Sinclair ZX80 sales in New Zealand
David Reid Electronics Ltd.
C.P.O. Box 2630
Auckland 1
New Zealand

Sinclair ZX80 sales in Canada
Future Distributors, Inc.
1189 Phillips Square
Montreal, Canada H3B 3C9

Gladstone Electronics
1736 Avenue Road
Toronto, Ontario M5M 3Y7

Full-size keyboard for MicroAce
Fully wired — $85.00
Kit — $65.00
Leonard Holmberg
P.O. Box 6273
Orange, CA 92667

RF shielding kit for MicroAce — $3.00
Outbound Engineering
P.O. Box 218
Chandler, TX 75758

Users Groups

Publishes club newsletter. Send self-addressed, stamped envelope for free copy.
National ZX80 Users Club
Unit 3
33 Woodthorpe Road
Ashford
Middlesex TW15 2RP
England

David Ahl, Founder and Publisher of Creative Computing

creative computing

"The beat covered by Creative Computing is one of the most important, explosive and fast-changing."—Alvin Toffler

You might think the term "creative computing" is a contradiction. How can something as precise and logical as electronic computing possibly be creative? We think it can be. Consider the way computers are being used to create special effects in movies—image generation, coloring and computer-driven cameras and props. Or an electronic "sketchpad" for your home computer that adds animation, coloring and shading at your direction. How about a computer simulation of an invasion of killer bees with you trying to find a way of keeping them under control?

Beyond Our Dreams

Computers are not creative per se. But the way in which they are used can be highly creative and imaginative. Five years ago when *Creative Computing* magazine first billed itself as "The number 1 magazine of computer applications and software," we had no idea how far that idea would take us. Today, these applications are becoming so broad, so all-encompassing that the computer field will soon include virtually everything!

In light of this generality, we take "application" to mean whatever can be done with computers, *ought* to be done with computers or *might* be done with computers. That is the meat of *Creative Computing*.

Alvin Toffler, author of *Future Shock* and *The Third Wave* says, "I read *Creative Computing* not only for information about how to make the most of my own equipment but to keep an eye on how the whole field is emerging.

Creative Computing, the company as well as the magazine, is uniquely lighthearted but also seriously interested in all aspects of computing. Ours is the magazine of software, graphics, games and simulations for beginners and relaxing professionals. We try to present the new and important ideas of the field in a way that a 14-year old or a Cobol programmer can understand them. Things like text editing, social simulations, control of household devices, animation and graphics, and communications networks.

Understandable Yet Challenging

As the premier magazine for beginners, it is our solemn responsibility to make what we publish comprehensible to the newcomer. That does not mean easy; our readers like to be challenged. It means providing the reader who has no preparation with every possible means to seize the subject matter and make it his own.

However, we don't want the experts in our audience to be bored. So we try to publish articles of interest to beginners and experts at the same time. Ideally, we would like every piece to have instructional or informative content—and some depth—even when communicated humorously or playfully. Thus, our favorite kind of piece is acessible to the beginner, theoretically non-trivial, interesting on more than one level, and perhaps even humorous.

David Gerrold of *Star Trek* fame says, "*Creative Computing* with its unpretentious, down-to-earth lucidity encourages the computer user to have fun. *Creative Computing* makes it possible for me to learn basic programming skills and use the computer better than any other source.

Hard-hitting Evaluations

At *Creative Computing* we obtain new computer systems, peripherals, and software as soon as they are announced. We put them through their paces in our Software Development Center and also in the environment for which they are intended—home, business, laboratory, or school.

Our evaluations are unbiased and accurate. We compared word processing printers and found two losers among highly promoted makes. Conversely, we found one computer had far more than its advertised capability. Of 16 educational packages, only seven offered solid learning value.

When we say unbiased reviews we mean it. More than once, our honesty has cost us an advertiser—temporarily. But we feel that our first obligation is to our readers and that editorial excellence and integrity are our highest goals.

Karl Zinn at the University of Michigan feels we are meeting these goals when he writes. "*Creative Computing* consistently provides value in articles, product reviews and systems comparisons . . . in a magazine that is fun to read."

Order Today

To order your subscription to *Creative Computing* send payment to the appropriate address below. Customers in the continental U.S. may call toll-free to charge a subscription to Visa, MasterCard or American Express.

		Canada and	
Term	*USA*	*Foreign Surface*	*Foreign Air*
1 year	$20	$29 or £12.50	$50 or £21
2 years	$37	$55 or £24.00	$97 or £41
3 years	$53	$80 or £34.50	$143 or £61

We guarantee your satisfaction or we will refund your entire subscription price.

Join over 80,000 subscribers like Ann Lewin, Director of the Capital Children's Museum who says, "I am very much impressed with *Creative Computing*. It is helping to demystify the computer. Its articles are helpful, humorous and humane. The world needs *Creative Computing*."

creative computing

Attn: Cindy
P.O. Box 789-M
Morristown, NJ 07960
Toll-free **800-631-8112**
(In NJ 201-540-0445)

27 Andrew Close, Stoke Golding
Nuneaton CV13 6EL, England

CAI INSTRUMENTS Introduces the 'WIDGET'® Series of Peripherals for the ZX80

The 'WIDGET'® Series of Peripherals open the door to real computing.

'WIDGET'® $49.95

The Widget interface board allows your ZX80 to communicate with the rest of the world through both a serial RS/232 data bank and a parallel TTL input and output ports.

'WIDGET'® $99.95 Widgitape

The Widgitape is a revolutionary tape storage system that provides the features of high priced floppy disks at a fraction of the cost.

'WIDGET'®$99.95 Widgiprint

The Widgiprinter is a 28 column thermo printer that provides a hard copy of results from your programs at a price you can afford.

Have you developed a unique option or application for your ZX80? We at CAI would be interested in discussing the possibility of marketing your idea for you. Contact us.

Remember this new "WIDGET" Series can open the door to real computing for your ZX80.
Send Check, Money Order, or Credit Card Company to:

CAI Instruments, Inc.
2559 Arbutus Court
Midland, MI 48640

Or order by phone (517) 835-6145 giving your credit card number.

Quantity	Description	Unit Price	Total
	WIDGET		
	WIDGIPRINT		
	WIDGITAPE		
	TOTAL		

Card No. Visa _____
Mastercharge _____
Exp. Date _____
Name _____
Address _____
City _____ State _____ ZIP _____
Michigan residents add 4% sales tax