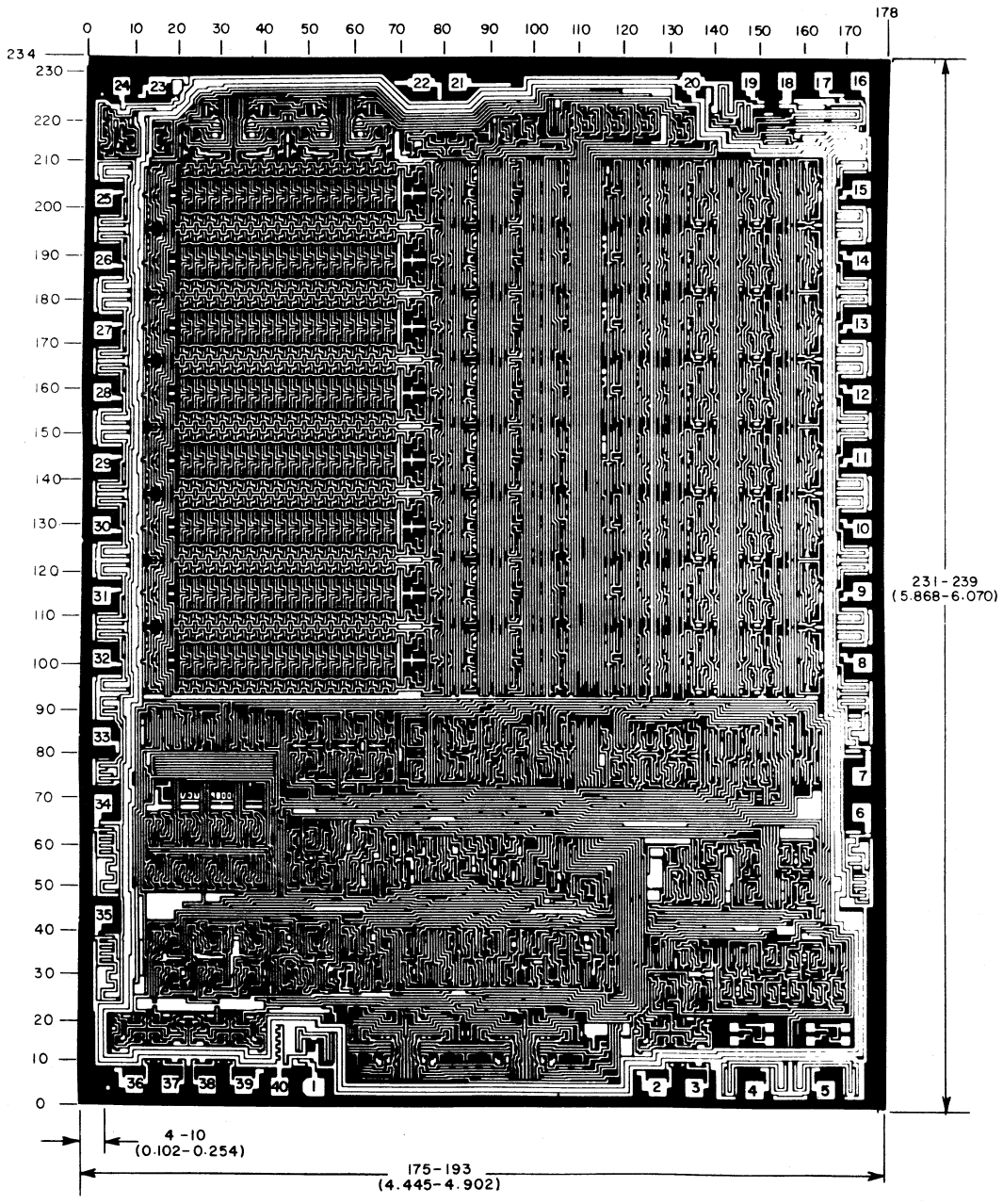
VIPER

VOLUME 1

MAY '79

ISSUE 10



THE 1802

$2.00

AN ARESCO PUBLICATION

EDITORIAL

We Made It! What a year! And, in this last editorial, there are so many things to tell you about that I hardly know where to start! First of all, thank you - for helping to make VIPER the best source of information around for VIP owners (Terry & I EDIT the VIPER - we don't write it!). Second, we're delighted to announce our very FIRST book for VIPpers - see page 12 for details. Third, the Index for Volume 1 is ready - send us a 15¢ stamp (not a SASE) and we'll mail it off to you. Fourth, if you have not already renewed your subscription, please do so - we need all the subscribers possible, in order to make Vol. 2 as interesting as Vol. 1 - or even better. Fifth, if you are "borrowing" a friend's copy of the VIPER, please subscribe instead - the more subscriptions we have, the better newsletter we can produce. And, last, but not at all least, we thank you again for your support and enthusiasm and contributions - and your patience with our mistakes. See you in July, with Issue #1 of Volume 2!

SUBSCRIPTION RATES, ADVERTISING RATES AND OTHER ESSENTIAL INFORMATION

The VIPER is published ten times per year and mailed to subscribers on the 15th day of each month except June and December. Single copy price is $2.00 per issue, subscription price is $15.00 per year (all ten issues of one volume). Dealer prices upon request. Outside of Continental U.S. and Canada, add $10.00 per subscription for postage ($1.00 for single copy).

Advertising rates are as follows:

1/4 page – 25.	3/4 page – 65.
1/2 page – 45.	full page – 85.

Less than 30% of the VIPER will be available for advertising. Please send camera ready copy in the exact page size of your ad on 8-1/2 x 11 white stock by the 1st day of the month in which you'd like the ad to appear. Photos should be glossy black & white in the exact size to be printed. Payment required with copy.

The VIPER is an Aresco Publication, edited by Rick Simpson. For information contact Editor, VIPER, **P.O. Box 1142 Columbia, MD 21044**

CHIP-8 MODS FOR ELF II - I TRIED IT! I LIKE IT!

by Neil Wiegand

I'm using an ELF II with a modified CHIP-8 interpreter that doesn't require the VIP OS. It works with the standard ELF II keypad. I really enjoy working with CHIP-8, and it was well worth my time to modify it for my system.

I have Tiny BASIC, but it runs so slow and takes up so much memory that it's really only good for demonstrating the ease with which a computer can be programmed. I'd like to share some of my experiences with CHIP-8.

The program modification was fairly straightforward. I duplicated any operating system functions required by CHIP-8 down on the 3rd page of memory, then modified the keyboard-related commands to input keyboard data and mask off the high four bits when the "I" button was depressed. Flipping the Run up with the I button depressed executes the CHIP-8 initialization routines and starts the CHIP-8 program execution. If I don't depress the I button, I go to the Giant Board monitor program. I've successfully run several of the programs out of the VIP Instruction Manual, the VIP Game Manual, and the VIPER. The biggest drawback has been shifting all programs up 256 bytes in memory to make room for my expanded CHIP-8. This isn't really a problem, though, unless the program isn't well documented (VIP Lunar Lander in the Game Manual is one of the worst I've seen!) or uses a large number of machine language subroutines or data tables.

I appreciate well documented programs, such as those that have appeared in the VIPER, because they are easier to modify to run on my system. The other pitfall I've experienced with CHIP-8 on my system has been concerned with the arrangement of the keyboard. A game such as VIP Card Matching required me to locate and modify the table describing the keyboard before it would run.

The concern expressed by Bobby Lewis in his article (about the keyboard not being dynamic) hasn't been a big drawback for me. It just requires me to keep a finger on the "I" button while I'm using the keyboard.

One addition I was able to make was an option to single-step through a program and output the low eight bits of CHIP-8 address to the ELF II hex display. This has been a great help to me in debugging programs!

Now: A request for a future article. I would like to see a list of literature applicable to the 1802. I'm sure RCA has a group of application notes, and there must be other people publishing software for the 1802. I've seen a reference to the COSMAC Source Book, but no details. I think we'd all find it helpful!

MODIFICATIONS TO THE HERSH EDITOR

by Norman Elliott

I've made a minor modification to Sam Hersh's Editor (see #4, Vol 1 of the VIPER) which may be of interest to other VIPpers. The mod consists of the addition of a new command, which allows the screen to be filled with a whole new page of addresses and instructions. This is useful mainly in checking newley keyed in programs for errors.

In this mode, five instructions are checked for correctness, then five new instructions (and their addresses) are placed on the screen for inspection. This instruction is accessed by pressing key "5".

To implement the modification, change location 027C from 1206 to 1348, then add the following code beginning at 0348:

ADDRESS	CODE	COMMENT
0348	3405	Skip if V4=5
034A	1206	Go to ST:
034C	02D2	Subroutine address +2
034E	02D2	
0350	02D2	
0352	02D2	
0354	02D2	
0256	1206	Go to ST:

SUBROUTINE HANDLING SCHEMES

by M L Dey

I have devised a small utility program that is now my standard sub-routine handling program. It uses R2 as the stack, and register E is the subroutine handling register (reserved). The program modifies one other register. Subroutines are called by loading the subroutine number or label (for example, "02"), and then doing a SEP DE. For example: F8 02 DE

Return from the subroutine is handled the same way. For instance, if 03 is the subroutine to be returned to, then use F8 03 DE.

The program counter is always R2, and the amount of nesting is limited only by the amount of memory allocated to stack R2. 42 bytes of memory are used to set up the handling routines, and the average calling sequence runs about 25 bytes (the return sequences are 22 bytes). Now for the question: Would you be interested in printing an article about this scheme?

M L - YES! We agree that this might be an excellent idea, well worth looking into. If you can type your article and code, it would help ensure that we make no transcription errors. Thanks! - Terry

LETTERS

Rick:

It would be nice if there were regularly a little box in VIPER which listed all the stuff currently available-including the RCA add-on boards. Next, I think the Studio II and ELF info in the VIPER is great. Maybe you could bring in the TEC/ACE people, too. My feeling is that the VIP OS and CHIP-8 are the winners; everyone else is finding ways to convert their 1802 systems to use them. From what I gather from the latest Ipso Facto, many novices are losing interest from sheer frustration - they got involved with a computer that didn't (until now) have the recources to support the newcomer adequeately. I think we should do everything possible to encourage standardization - or at least compatibility - between 1802 machines, since the 1802 is in a minority position. The novice of today might be the guy who contributes something significant tomorrow, and it saddens me to think of his losing interest. Frustration and disappointment can be had almost anywhere else for far less than $300! - Charlie Mc Carthy

Charlie - We agree that it's a loss to all of us when a novice loses interest in the 1802. We have no idea how many novices there are out there (we've had only one or two requests for beginner-level information). When we started the VIPER, we said we'd stick exclusively to VIP material...and already we're considering using stuff for the ELF and the Studio II on a regular basis. Judging from the "voting", most people seem to be eager to have non-VIP material, so long as it's 1802-oriented; a very few are absolutely opposed to <u>any</u> non-VIP stuff. When we published the ELF II article, it was with the idea of standardization: the new "conversions" could be considered "effectively" VIP people. As for the "VIP ADD_ONS" box, we like that idea, too, and will incorporate it in Volume 2 without fail. It may not appear in every issue, but it will appear frequently. - Rick

---

LETTERS

Rick -

Why not publish a list of articles you want most for the VIPER? You've published blurbs in the past...A/D converters, Supersound and Color board applications, and so on; but these must be only the top of the list. If you publish a complete list of things or topics or whatever that you might be interested in, then people like me might be able to pick up ideas that way. Is it worth a shot? - Phil Sumner

Phil - Sure is. Problem is that we can use items on almost any topic pertaining to the VIP: games applications, simple sound programs, reviews and comments on the RCA boards, "I did it this way" articles, utility applications such as assemblers, control applications, anecdotes, - just about anything at all. Our only reason for rejecting an article (so far) has been that the piece was poorly written (that is, incomprehensible) or that there have been 22 pages or so of handwritten code that Terry has to sit and type (and then blush over when she makes typos and doesn't catch them before we go to press). There isn't any way we can give you a "complete" list of topics...perhaps the list here will help. -Rick

---

LETTERS

Rick -

This is the turning point. I've spent two years playing with 1802 machine language and CHIP-8, and have yet to write two lines of code that will do anything. Additionally, CHIP-8 "instructions" keep bombing out their interpreter. That thing just HAS to be put in ROM! I don't see how anyone can write programs with that. But they do, obviously. And they don't all work, either.

I'm waiting for Tiny BASIC on ROM. If that proves to be successful, I'll sign up for Volume 2. Otherwise, I'll have one VIP with TB - for sale. 1802 code is like beating your head against a brick wall, and CHIP-8 is a chip off the old blockhead. 6502 code is no better. TRS-80 BASIC is okay, but too expensive right now. I'll keep on changing partners until I find one that works! Later - Maybe. D.L. Hartley

D.L. - **WOW!** You must really be frustrated! Have you tried reading the new VIP User's Guide? It might be helpful to you, since it was designed for someone who is having trouble with CHIP-8. It doesn't discuss 1802 code, unfortunately, but it covers CHIP-8 pretty well. You can get it from RCA or you can order it from us. RCA charges $5.00 for it; we charge $6.00 to help cover the postage costs. If you'd like to order, you have our address. - Terry

---

Terry -

Although I've enjoyed the fun and games in the VIPER, I would like to use the VIP for many of the "computer adaptable business" applications I have. Any articles along this line would be appreciated, especially if RCA gets the Tiny BASIC board out to us soon. The expanded video looks interesting, too; but I wonder if you might run an article on interfacing the VIP with a terminal so we can go full tilt without using the limited RCA video chip. Also, and article on memory expansion. - Ralph Egloff

Ralph - If you're talking about accounting type business applications, the VIP is a poor choice of computer; it's intended for graphic and control applications. If you're talking about control or monitoring applications, the VIP is super! Don't expect too much of Tiny BASIC, because it is just that - TINY. If we receive articles on the topic, we'll surely print them, but I don't expect that very many people have given much thought to hanging a $2000 terminal on a $300 VIP! RCA does provide memory expansion boards (and now, with the mini-expansion connector, you can have two boards on the VIP at once), and we're hoping for user reviews of the boards in Volume 2. - Terry

---

Terry - Keep the VIPER for the VIP! Stop with the ELF II and Studio II stuff! Leave that to the other 1802 newsletters that are out, and concentrate on our favorite 1802 machine, the VIP! - S D Wilson

S D - Thanks for your vote! Every vote counts, and we appreciate the time you took to let us know how you feel. And thanks for renewing your subscription. - Terry

SAVING AND RESTORING CHIP-8 VARIABLES

by John Bennett

The articles on CHIP-8 in the VIPER have been interesting, and have greatly improved my understanding of the CHIP-8 interpreter. One objection I have to CHIP-8 is the operation of the FX55/FX65 instructions. Unfortunately, the FX55/FX65 instructions always operate on V0 - VX. To circumvent this, I have added a minor modification to CHIP-8 - and now I have the FXF2 - FX55/FX65 instruction pair relationship.

The FXF2 instruction saves VX in RD.0, while the new FX55/FX65 instruction operates on the variable saved (VX) in RD.0. The following example will save the contents of VD-VE in locations 0400-0401:

```
A400    I=0400
FDF2    RD.0 = VD
FE55    M(I) = V(RD.0) : VX
```

This is a reduction of at least one instruction over the normal CHIP-8 requirement of first loading VD-VE into V0-V1, then saving V0-V1 in 0400-0401 (and destroying the contents of V0 and V1 in the process).

This is extremely simple to implement, and consists of the following modifications to the CHIP-8 interpreter:

ADDRESS	CODE	COMMENTS
0158	C4	NOP
	8D	R(N).0 - D
0168	C4	NOP
	8D	R(N).0 - D
01F2	86	R(N).0 - D
	AD	D - R(N).0
	D4	Return

Note that although page 36 of the VIP manual says that VC-VF are available, that's only true if they are not to be used to pass data between subroutines (Re: J. W. Wentworth's Analysis of VIP CHIP-8 Interpreter; VIPER, Volume 1, issue 2, pages 17-25). Many of the CHIP-8 routines utilize VC-VF as part of their code. For this reason, the FXF2 - FX55/FX65 instructions illustrated above should always be used in pairs, with no intervening code.

John included with his article several copies of the forms he uses when programming the VIP. We'll publish these forms as space permits. They include a CHIP-8 Coding Form, a Register/Variable Utilization Form, a CHIP-8 Display form, a Flowchart Worksheet, and a VIP (1802) coding sheet. John says that for the price of copying them and shipping them ($5.00 per hundred), he'll be glad to send good quality Xerox copies. Write to him at 3 Grafton St., Manchester, MD 21102 if you're interested in obtaining the forms.

FASCINATING KALEIDOSCOPE

Phil Sumner

In several ways, the Kaleidoscope game is a fascinating thing to watch. The ever-changing geometrical patterns can be very pretty, enjoyable at all times; the shifting patterns can also be very restful at the end of a tired day. There is also the element of anticipation -- what kind of pattern is going to develop next?

The prettiest or best pattern that I know of is the one given with the program listing in the VIP manual -- 44444442220. This pattern seems to run endlessly, providing a series of similar but different and still interesting patterns. The related pattern 22244444440 is very similar and interesting in itself -- it eventually erases itself completely and starts over, repeating endlessly. This makes it useful as a background display for birthday parties and other similar affairs; the pattern does not degrade with time.

Although not as pretty or nice, there are many other patterns that are also interesting for one reason or another; this is particularly true if diagonal spot motion and freeze motion capability are added (VIPER 9, April 1979). There are patterns that erase and repeat every few seconds (3220 and 2230), patterns that repeat at longer intervals (222224440), patterns that proceed to a certain point then "hang up" (454620), still other patterns that are just pretty or interesting to watch, and a vast number of patterns that are of interest only from the standpoint of avoiding them in the future.

The occasional pretty patterns, however, are worth the search effort. The best of the patterns that I have found are given below, together with appropriate comments. For completeness, all patterns that scored 5 or better on my arbitrary scale of 1 (yecchh!) to 10 (beautiful!) are included.

Rating	Pattern	Comment
10	22244444440	Very pretty; it continually seems to break down, then regroup to form a pretty pattern. Eventually erases and starts over, repeating endlessly.
10	44444442220	Very similar to the above, but does not repeat. Eventually gets into a "hang up" pattern, repeating a pattern progression.
9	51930	Very pretty lacework X pattern. Repeats fairly fast.
8	3380	Reverse diamond pattern that fills most of the screen. Repeats after about a minute.
8	53710	Pretty lacework X. Eventually repeats.
8	111330	Diamond pattern, sometimes subtle, mostly obvious.
8	2290	Square pattern. Eventually repeats.

8	22270	X pattern with ripples. Repeats slow.
8	3660	Sequence of diamond patterns. Gets prettier as it goes; eventually repeats.
7	53790	Lacework X pattern. Hangs up.
7	2270	Fancy X pattern. Eventually fills entire screen, then reverses to erase completely.
7	8830	Reverse-developing X pattern. Repeats fairly fast.
7	2210	Subtle diamond patterns; eventually repeats.
7	33580	Diamond pattern that develops with a hitch. Repeats fairly fast.
7	3340	Vertical diamond pattern that hangs up in a repetitive cycle.
7	1410	An X pattern that slowly but surely builds up until it fills the screen, then erases and repeats.
7	66610	Reverse image diamond that continually modifies but never really changes the basic pattern. Eventually repeats.
7	4490	Basic square that is continually modified but never really changed. Repeats.
7	888110	Diamond pattern that eventually develops into something pretty. Eventually repeats.
7	51730	Fancy X pattern. Repeats.
7	519330	Develops a succession of square "doily" patterns. Repeats.
6	7780	Vertical diamond that develops slow but does develop into a series of pretty patterns. Repeats slow.
6	2222770	A crazy-looking X pattern that repeats fairly fast.
6	4430	A lacework X pattern that fills the entire screen by edge migration, then erases and repeats.
6	3390	Diamond pattern that evolves through several iterations. Repeats.
6	480	Square pattern. Repeats fast.

6	59730	Slow-developing lacework X. Repeats slowly.
6	1140	Similar to 1410 above.
6	59170	Slow-developing square pattern. Repeats fast.
5	4444220	A full-screen X pattern that slowly fills in.
5	222224440	Proceeds through several regular patterns.
5	222244440	Interesting X pattern that "walks", alternately painting light and dark areas. Repeats slow.
5	454620	Constructs a lacework X that fills in the whole screen, at which point it hangs up.
5	2210	Slow-developing series of diamond patterns. Gets better as it goes. Repeats.
5	1440	Wide-screen X that erases rather quick.
5	7720	Diamond that erases quick.
5	7760	Vertical diamond with an interesting "hang up" pattern.
5	3390	Full screen diamond with many variations but no real changes. Repeats slow.
5	3360	Reverse-developing square pattern. Eventually hangs up in a pattern sequence.
5	4466222880	Paints a series of vertical bars. Interesting "inch worm" effect.
5	999770	Oddly developing and subtle diamond. Repeats.
5	22290	Vertical diamond with variations. Repeats slow.
5	2770	Full-screen diamond. Erases quick.
5	59760	Interesting variations on an X pattern. Repeats.
5	5470	Full-screen diamond that develops slowly to fill most of screen. Repeats.
5	5410	Full screen X that slowly expands to fill the entire screen. Repeats.

Although this listing shows quite a few interesting patterns, I know that all of them are not listed; there are many others there somewhere. Would you like to share any others that you know about?

Programs for the COSMAC ELF - Interpreters

PRODUCT REVIEW by Rick Simpson

It has often been said, and CHIP-8 is the best example of the matter, that the great strength of the 1802 architecture is the ease with which small interpretive languages can be implemented. We recently received a booklet which contains the ultimate example of this - an 1802 interpreter which resides in a little more than one half page of memory!

The booklet "Programs for the COSMAC ELF - Interpreters" was written by Paul C Moews, who has also authored two previous booklets of 1802 software. The booklet is 30 pages long.

Although written for the ELF, the interpreter could be modified for use on the VIP, especially if an LED hex display, similar to that used on the ELF, were added to your VIP. The interpreter contains 12 instructions, ten of which are drawn directly from CHIP-8. The major value for the VIP owner is that the code for the interpreter is very well commented so you can see just how a tiny interpreter is written. The only thing missing is a flowchart, similar to the one published in VIPER #2, Volume 1, for CHIP-8.

After explaining the tiny interpreter, Paul shows a complete implementation of CHIP-8 on the ELF. He then adds some material which will be of special interest to VIP owners: expansion of the CHIP-8 instruction set. He begins by expanding the 5XY0 instruction (Skip next instruction if VX=VY) to four instructions, so you'll be able to skip if VX is equal to, greater than, less than, or not equal to VY. He also expands the 9XY0 instruction to two instructions which perform 16-bit multiply and divide operations. In addition, he describes a modification to CHIP-8 which implements a new instruction to display any of the 64 ASCII characters, making message display much simpler. The entire routine fits in one extra page.

All in all, Paul has done an excellent job. His booklet is well written, interesting, and reasonably priced. It retails for $5.50 and is available from Paul Moews, 16 B Yale Road, Storrs CT 06268. I highly recommend it.

ATTENTION COMPUTERISTS!

Once again we (the public) are faced with the spectre of our government getting ready to pass some well-intentioned but badly designed legislation. This time, the law could affect every person who works around computers - even the home variety. The bill (S. 240; the Ribicoff Bill, US Senate) is designed to protect against a variety of abuses related to the use of computers. There are a number of widely publicized incidents of "computer abuse" in which people have abused their access to computer data banks and programs, to steal or alter confidential information. This bill is an attempt to prevent that. (continued on page 19)

PIPS For VIPS - A Pre-Publication Offer For VIPER Readers Only!

We recently received a package of programs from Tom Swan, who wrote the article titled "Long Branch In ML With Video On", in Issue #9. Over a hundred pages of documentation , including program descriptions, instructions, and listings. These seven programs, if published in THE VIPER, would have filled more than four full issues - so we decided to publish them all together as a book and offer them to you all at once instead. Since they're all Pretty Impressive Programs, the book is entitled "PIPs For VIPs". Most VIPER readers will want this book as soon as we can get it printed, so we've decided to make a special "pre-publication" offer: If you order "PIPs For VIPs" before July 15, 1979, the price is $14.95 (USA, plus $1.19 non-USA shipping). And you'll receive a tape containing all seven programs along with your copy of the book. The programs are:

1. EDITOR 21 - A text editor with 21 commands; featuring auto-scrolling, a display of 16 lines x 16 characters; an underline cursor; and built-in tape read/write commands. Six pages (over 1500 characters), both upper and lower case, can be edited on a 4K VIP; text input is directly through the VIP keypad using ASCII codes.

2. DISASSEMBLER 7 - A machine-language disassembler allowing modification of the disassembled code! Uses standard RCA mnemonics; up to three pages of machine language code can be worked on at once; and tape read/write routines are all built in. Seven commands available.

3. A CHIP-8 EDITOR - which replaces the usual code in CHIP-8 (pages 0 and 1), allowing you to enter and modify CHIP-8 instructions; page or scroll forward or backward; and relocate code for adding that extra instruction in the middle of the program.

4. CHARACTER DESIGNER - Allows you to design a character set of your choice for later display. Each character is assigned an ASCII code and can be displayed by calling its code from a CHIP-8 program. Up to 128 characters can be defined and stored in only 500 bytes; characters are displayed in standard two-page display format.

5. CHIP-8 MESSAGER - Allows you to use the character set defined by the Character Designer, to print text with your CHIP-8 programs: No more figuring out bit patterns!

6. SPACE WARS - An exhaustively documented "Shoot The Klingons" arcade-type game. The best we've seen of it's kind. Full documentation - make your own customized version!

7. SURROUND - Another superb arcade game; with full documentation so you can learn as well as enjoy.

When published in August, the price will be $19,95 including the tape (plus postage as mentioned above). Order before 7/15 and save $5.00! (No coupon is provided - so you won't even feel tempted to tear it out of your VIPER!)

ANOTHER HIRES GRAPHICS METHOD FOR THE VIP

by Charlie Mc Carthy

The VIP video display is limited to 64 dots horizontally. How to get more resolution? The classical method would be to have a separate video screen memory that VIP treats as a peripheral, together with separate video signal generator; the resolution that could be obtained would be limited only by one's budget for memory and skill at debugging high speed circuits. A second method was presented in VIPER recently which used a moderate amount of hardware to store two 8-byte sequences of spot information which was then pumped out to the video display as 128 bits of data; there was the great advantage of the software needed also being given. I would like to describe here a third method, which I have begun to implement. Its advantages are that it uses little hardware, and what it does use is not critical (I have neither 'scope nor logic probe, and my construction technique using wire wrap is not particularly neat; yet it works), and that it is readily adaptable to a 128 by 128 display. Its disadvantage is that the software to make it compatible with Chip-8 or other existing programs will probably be complicated--at any rate, an interesting exercise.

I think an apology is in order here. It is not nice, I feel, to present this circuit without the software needed to use it. My excuse is only that the circuit does work, and I would be very interested to see what sort of programming tricks others could develop to make use of it.

To obtain a 128 by 64 display, one needs four pages of memory. Assuming that we have a VIP with 4K of memory and that the screen memory is the top four pages, these would be addresses 0C00-0FFF. The dots that the 1861 video generator puts out are roughly 560 ns. long; the CLK signal available on VIP is alternately low and high, each level being held for 280 ns., half of the 560 ns. dot time. The idea is to use a program which generates a 64 by 64 dot picture, alternately from 0C00-0DFF and from 0E00-0FFF, but to display the dots from 0C00-0DFF only during the first 280 ns. of CLK, and the dots from 0E00-0FFF only during the last 280 ns. Effectively we get two half displays interlaced horizontally 30 times per second. This rate is fast enough so that there isn't trouble with flicker; indeed, it's the same sort of interlace that commercial TV signals use vertically.

The circuit that I have working is shown in figure 1. The first 4013, U2A, latches bit 1 of the high order byte of the memory address from which the 1861 fetches 8 bits of data to display; the Q output is 0 for pages 0C and 0D, and 1 for pages 0E and 0F. This determines

which of the 4070 exclusive-or gates acts as an inverter and which does not invert the CLK signal. For pages OC-OD, CLK is fed to U2B to latch the spot information for 560 ns. and this is fed to the video driver during CLK; for pages OE-OF, spot is latched at the beginning of CLK and fed to the video during the following CLK. The output is dark otherwise.

The 4049 buffers, 7409 open collector and gates, and transistor are what I need to drive my monitor (SW Tech) and are irrelevant to the signal generation except for the and gate U5A that combines the spot information with which-half-dot-do-we-want-now from U3B. I did need to remove the 1K and 10K resistors from the SPOT and SYNC terminals of the 1861 in order to get enough drive for this circuit.

What about U1 ? If this is not put in here, then the last 5 or 6 dots of each line get confused: the memory address that U2A latches is that of the instruction in memory following the DMA burst, so U1 allows the TPA signal to latch U2A only while DMA is low and the address involved is actually that of screen data. Thus there is a restriction on the location of the screen memory: the high order byte of the memory addresses containing data for a given scan line must be all the same. This is no restriction for VIP which uses whole pages for screen memory, and I think that it is not a problem with Pittman's Tiny Basic which appears to cross page boundaries during a scan, but only between lines.

The programming requirements, as I mentioned, may prove interesting. To show the pattern

a	b	c	d	e	...
p	q	r	s	t	...

in the upper left corner, you somehow have to get

```
0C00 = 1 1 0 ...      = a c e ...
0E00 = 0 1 ...        = b d ...
0C08 = 0 1 1 ...      = p r t ...
0E08 = 1 0 ...        = q s ...
```

Also, bit 1 of the high order byte of where the screen memory begins has to be complemented on each pass through the display routine.

Figure 2 is an alternate scheme of generating the left-half and right-half dots; I haven't actually wired this up yet (and probably won't for a while. I dislike building hardware, and only got around to building Fig. 1 because I wanted an excuse to put off doing the income tax.) With Fig. 2, probably the extra 4001 and 4070 gates left over could be used in lieu of the 4049 to drive the TTL gates.

So what next? Surely good software is required--something that requires as little modification as possible to existing Chip-8 and other programs. Or possibly using the same idea to generate a 256-dot wide display with the aid of the 3.58 MHz signal, showing one fourth of the display each scan; but I worry a little about how symmetrical the 3.58 MHz signal is, and I worry a little about flicker at this 15 Hz repetition rate. I think it would be nice, though, to be able to control the resolution somehow, perhaps through outputting a data byte with a certain N value which would say what phase the spots should be displayed during, and in the absence of other commands, would default to the standard 64 by 32 display.

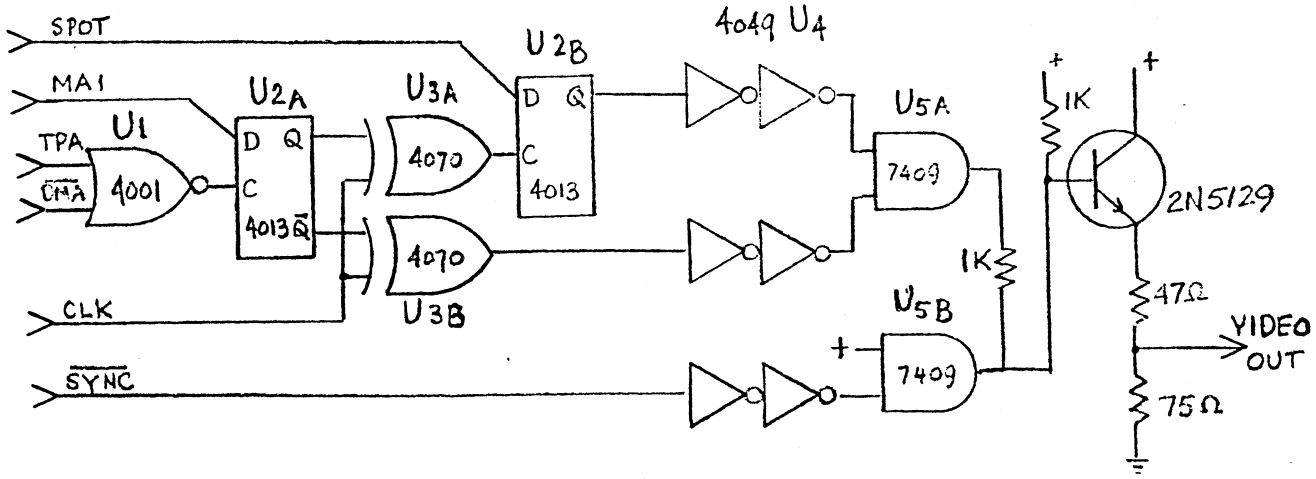


Fig. 1

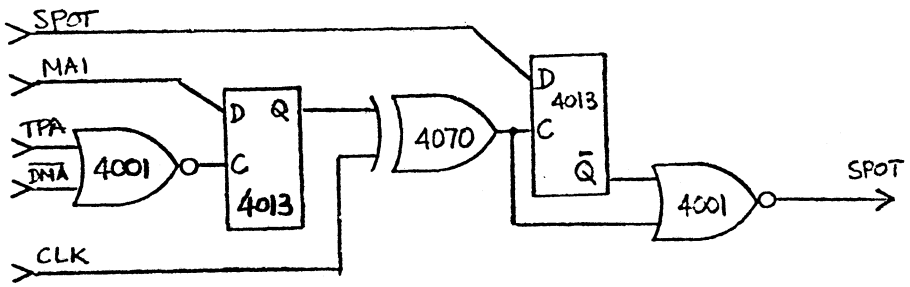


Fig. 2

BOOK REVIEW by Rick Simpson

TITLE: CMOS Databook TAB Books #984
AUTHOR: Bill Hunter PRICE: $6.95

As in any hobby, good books are the best investment you can make to increase your knowledge and pleasure in computing. As a confirmed "book junkie", I scan every catalog that comes my way, for new additions to my bookshelf. After a while, I begin to get a feel for what each publisher considers "marketable": Osborn only publishes "readable" material, Sams books have a practical slant I really appreciate. But TAB books vary from utter trash to real gems, and I never know what is waiting behind the four-color cover of a new TAB book. The first microprocessor book I ever saw was a TAB book, and it was awful! Nothing but a compendium of National Semiconductor data sheets, thinly rewritten; and the author chose to concentrate of four-bit processors (which have never been very popular with hobbyists). So, when I got the new CMOS Databook, I was a bit skeptical; especially since I couldn't see how anyone could improve on Don Lancaster's "CMOS Cookbook" (Sams #21398).

Well, Lancaster's book should still be your first CMOS hardware book, but the CMOS Databook is a worthy companion. Hunter's book neatly fills in all the technical information Lancaster ignored or glossed over. Lancaster provided a cookbook - a collection of tricks and tested circuits the user can paste together to form a CMOS "meal"; Hunter provides the rich technical background for the reader who wants to know how CMOS devices are fabricated and designed. He covers the electrical parameters in detail and gives valuable information on subjects such as noise margin considerations and proper power supply design. All of this is done with a minimum of technical jargon and virtually no mathematics. I suspect that this book would be especially valuable to an engineer who usually uses TTL and wants to start designing with CMOS.

My only real complaint with the book is that the text is constantly referencing illustrations which are two or three pages further ahead in the book, and it is a distraction to have to keep flipping pages to match up text and figures.

The book concludes with several practical design examples: a digital timer, an LCD-display multimeter, and a battery-powered 5MHz counter. The design considerations for each are well explained; you come to understand the design tradeoffs that are inherent in any piece of equipment. I think you'll find it a worthy addition to your library, and I recommend it.

We received a letter from "Joe" - we couldn't decide who Joe it was. But he offered an interesting suggestion: "How about asking for a list of good guys who are willing to help out other guys with software or hardware problems? And how about printing the names and addresses of the guys who write the articles, so people can write directly to them (with an SASE, of course) if they want more info?"

Okay! So who are the good guys out there? Authors who want their names and addresses and phone numbers printed should specifically say so - we can't print them without express permission!

NEW PRODUCT INFORMATION From Digital Service & Design

We received an interesting flyer from Digital Service and Design, Box 741, Newark, OH, 43055. They manufacture a series of PC cards for an 1802 system, based on the SS-50 bus structure. The SS-50 was pioneered by Southwest Technical Products, and is extensively used on 6800 based systems.

DS & D is offering a CPU card with 4K or 8K of EPROM and a built-in terminal interface (TTY or RS-232). This card can be connected to a motherboard with 5 memory and CPU slots and nine I/O card connectors. An I/O Network Card buffers the 1802 signals routed to the I/O card connectors. Memory is provided by a 16K RAM (2114's) Card or a 32K RAM/EPROM Card. I/O expansion cards include cards for serial interface, parallel interface, and cassette interface. An enclosure is also available.

A set of all the cards mentioned here is $125., but the cards are available individually for from $12. to $29. This price includes only the PC boards, the connectors, and the documentation. No components are provided. A pre-programmed 2708, containing a monitor is available for $30.

Since we have not seen this product, we recommend that you check with DS & S regarding delivery and performance before you order it. We did talk (via telephone) with the proprietor, and he's planning to send us a set of the documentation and a CPU board for further evaluation in a future issue. If it lives up to its promotion, it will be a good 1802 mainframe, with good support hardware. But, as is usual with the 1802, the main problem will be software.

Other than the monitor in EPROM, DS & S doesn't plan, at this time at any rate, to provide software for their cards.

Garrett Guske writes: "Has anyone developed a tiny assembler for the VIP? What exactly does the VIP tiny BASIC have in the way of commands? Three cheers for the ELF II articles and the Studio II articles and the programs such as Sam Hersh's Editor and Udo Pernisz's Relocation Program to make it easier to debug CHIP-8 and ML programs! Love the games! Keep up the good work!"

We haven't heard of a tiny assembler....has anyone else? We're expecting a review of the BASIC (but it doesn't have POKE or PEEK) ... The vote is 97 FOR continuing Studio II & ELF II articles and 38 AGAINST it; 26 have voted to "keep a separate section in each issue" for these two items IF we continue printing articles about them (but failed to say whether they were for or against it). Three people didn't much care, one way or another, so long as we keep the VIPER going. As for games, we have a couple really neat ones now in line for publication....watch this space!

RE: GRAPHICS LUNAR LANDER

by Udo Pernisz

I should like to mention some typing errors that appeared in the published version of the LANDER, in order of appearance:

a) Page 9, paragraph 3: Height 2095 should have been 4095
Fuel 447 should have been 448

b) page 12: top line of code is missing:

```
0254 F033     :MI=V)(3DD)
```

c) page 13: comment to code at address 02C4: (code is okay)

V7=V7+4 should have read V1=V1+4

d) page 15: code for addresses 0340 to 034F is missing:

```
0340 41 00 13 5E 82 85 80 F0
0348 80 E5 81 04 31 00 00 EE
```

e) page 16, comment to code at address 039A: (code is okay)

GOTO 0386 should be GOTO 0396

f) page 16, code at address 03D8 was printed as 6B81 VB=18
It should have been 6B18. Comment is okay.

g) page 17: eight bytes of data are missing, starting at 0490

```
0490 89 8D 00 D1 55 55 CA 00
```

I tried to submit reproducible copy for the program listing. Apparently you couldn't use it, but I'd like to know what was wrong with it. Would it be better if I submitted the program only on tape?

We do apologize again for the errors. The problem we have in getting reproducible copy from typewritten text is in the type of ribbon used to produce the manuscript. Most people use a fabric ribbon which puts ink on the paper; we can't get good copy, usually, from an "ink" ribbon. (We remember sadly the first few issues of the VIPER, when we were using the fabric ribbon...and got almost unreadable copy!) Nowadays, we use a "carbon film" ribbon, which places a bit of itself on the paper to make the impression. Unfortunately, it's a "one-time only" film; it can't be used repeatedly like a fabric ribbon can be.

In addition, when an error is made using the fabric ribbon, it shows in the reproduced text, even when it has been corrected in the manuscript. With the film ribbon and a bottle of "Liquid Paper" correction fluid, the errors and subsequent type-overs are almost invisible.

ATTENTION COMPUTERISTS! (continued from page 11)

Unfortunately, the bill authorizes prison terms of up to 15 years for "unauthorized" access to a computer system - such as playing Star Trek on the company machine. It also gives the government the right to seize the computer-based records of any company - without warning. The provisions of the bill even extend to calculators, word processing equipment, or anything vaguely related to computers. It is vague; it opens the possibility of a programmer being prosecuted for doing what he was told to do; it allows companies to "hide" information by simply putting it on a computer system.

If you are concerned about such legislation, we suggest you contact your US Senator and request more information.

John S James, 1090 Miller Avenue, Berkeley, CA 94708, is coordinating efforts to publicize the bill, and will provide a flyer with more information for a self-addressed, stamped, legal-sized envelope.

We at ARESCO urge you to inform yourself about the Ribicoff Bill (S. 240), and to contact your US Senator to let him know your opinion about it.

Don't let's lose our freedom from governmental red-tape, or offer ourselves up for prosecution for printing out Snoopy Calendars. This is not an issue about which we can remain both apathetic and unregulated, unlicensed, and untagged!

NEWS FROM RCA

RCA plans to be exhibiting at most of the major Personal Computing Shows this year; plans are firm for CES in Chicago (June) NCC in New York (June), Philadelphia and Boston in the fall. Terry & I will be in the booth at NCC, and we'll be able to show you all the newest expansion boards, the new keyboard, and VIP BASIC.

VIP Marketing tells us that all the VIP add-ons are in stock except the BASIC board and the ASCII keyboard. These two should be in full production sometime in June. New additions to the line include the VP-575 Expansion board, which we told you about (in the form of an RCA flyer, last month); we have a "final price": $59.00. Another new addition is the two-connector mini-expansion board (VP-576), for $20.00.

RCA now has a number of Manufacturer's Representatives in the field, visiting computer stores with a snazzy new VIP Demonstration Kit. We understand that this has already resulted in several new dealers for the VIP, so there's hope you'll have a dealer near you in the not-too-far-distant future!

See you at NCC!

Simple Music Program Part 1

Udo Pernisz

Several programs have been published that make the CDP 1802 produce musical tones using either the ELF or the VIP as a machine, see e.g.
(1) Edw.M. McCormick, in Dr. Dobb's JCCO, no.19, p.30
(2) Chris G. Smith, in Interface Age, vol.2 (1977), no.12, p.48
(3) Carmelo Cortez, in VIPER, vol.1 (1979), no.6, p.23.
All of these programs use the Q-line flip flop to generate a tone by switching Q on and off at a rate corresponding to the frequency of the tone, and for a time period equal to the duration of the note.

These programs are written in the 1802 machine language and are not readily incorporated into a CHIP-8 program, e.g. as a subroutine, since they use up most of the microprocessor's registers. This is different with the two music boards of RCA that can be used on the I/O port of the VIP.

The program that is described in the following is designed to drive the RCA VIP Simple Sound Board VP 595. The language does not use the CHIP-8X interpreter as is suggested for the VP 595 in the instruction manual but rather the CHIP-8I modification of the standard CHIP-8 interpreter which Rick Simpson published in VIPER, vol.1 (1978), no.3, p.4. This version provides I/O with the instruction B1X0 that sends the value of variable VX to the I/O port for output. If one wanted to use CHIP-8X, however, this instruction could be replaced by FXF8 in the program listing given. (Relocation of the program would also be necessary.)

The "Simple Music 0.0" is the first of a series of programs of increasing sophistication and facilities, and is meant to be a bare-bones demonstration program that lets you play a simple monophonic tune. Each musical note of this tune is encoded as a two-byte word and stored in memory from where the program takes one word after the other and converts it to sound.

The first byte of each word gives the pitch and will be referred to as the "tone", the second one gives the duration and is called the "note" in analogy to the form of notes rendered in conventional musical notation on five-line manuscript paper.

The encoding of the pitch is simply by the table in the manual for the VP 595 and allows, therefore, any tone to be played from the 5-octave frequency range that the music board covers.

In "Simple Music" 0.0 the duration is encoded as a two-hex-digit number that is a time in units of 1/60 sec. In view of the next-level program "Simple Music 1.1", and with respect to conventional notation the note values (i.e. durations of tones) will be referred to the duration of the quarter note by giving its M.M.(1/4) - value (Maelzel's metronome). This value defines the tempo of a piece of music by giving the number of quarter notes to be played per minute. From the M.M.

setting the value for the note byte is obtained as 3600/M.M. which can be used directly after conversion to a hexadecimal number.
A piece with M.M.(1/4) = 144 for example (allegro) corresponds to a note byte of 19 in hex.

The available range of tempi can be determined as follows: It is desireable, e.g. in order to play triplets that add up to an eighth note, to have as the shortest one a (1/64) note - this is assigned to the smallest note byte of 01 or 1/60 sec. On the upper end one would want to fit the duration of a whole measure into one byte - this is FF in hex. The range of the (1/4) note is then from hex-10 to hex-3F which corresponds to a range of M.M. values of approximately 60 to 240.

Tones are played with a short break in sounding at their end to mark the transition from one to the next tone (allowing two notes of the same pitch to be perceived as two different tones).

Words that contain a zero-duration note byte (2nd byte) will be ignored by the program. Tone bytes (1st one) of zero value which the VP 595 would interpret as some low pitch sound, cf. the manual, are used to play a pause of a length specified by the note byte.

Pressing key 0 at any time will terminate the tune and allows you to jump to other parts of the program. The listing given just halts in a loop.

The listing of "Simple Music 0.0" is given with a small tune created by Alan Fraser of the Conservatory of Brisbane, Australia.

Rather than encoding a piece of music and entering it word by word one can also play whatever is just in memory after power-up. This is theoretically quite interesting since it can be assumed that the distribution of bytes in memory is totally random. You will then hear a tune with the probability of the pitch falling into an interval of constant relative size, e.g. one half tone, being inversely proportional to the center frequency of this interval. This can be verified from the table in the manual. With other words, you will hear the famous 1/f - music, see also Scientific American, Apr.1978. There should be a pause every 255th tone on the average as Simple Music 0.0 plays through your RAMs.
Duration of notes, however, are equally probable over the whole range of the hex byte.

It is very convenient to enter and run the program plus music words via the Hersh Editor, VIPER vol.1 (1978), no.4, p.19, in conjunction with the Relocate Routine, VIPER vol1 (1979), no.9, p.19 - not to forget the anti-jitter corrections in VIPER vol.1 (1979), no.6, p.12 . This allows straighforward work with the two-byte tone-note words.

Simple Music 0.0

Machine:	COSMAC VIP with Simple Music Board VP 595
Language:	CHIP-8I
Program location:	0200 through 021D
Tone-note words:	0220 and up
Data structure:	- A musical note is encoded as a two-byte word. - The first byte is the tone byte and contains a number from 00 to hex-FF that is interpreted as a frequency according to the table in the VP 595 manual. 00 is played as a pause. - The second byte is the note byte and contains the duration of the tone in units of 1/60 sec with a range from 01 to hex-FF.
Features:	Ignores zero duration notes. Can be halted any time by pressing key 0.

```
ADDRESS CODE    COMMENT
0200    A220    I=0220          :Point to begin of tone-note word list
0202    F165    V0:V1=MI        :Read in word (byte pair)
0204    4100    SKIP;V1.NE.0    :Ignore word if duration byte = 0
0206    1202    GO 0202         :and fetch next word. Else
0208    B100    I/O = V0        :Send tone byte to output port
020A    F115    TIME = V1       :Time delay = note value to halt prog-
                                :ram for time of note duration
020C    71FF    V1=V1-1         :Reduce note by 1/60 sec to stop soun-
                                :ding the word just before duration
                                :time is up
020E    3000    SKP;V0=0        :If tone byte specifies a pause do not
0210    F118    TONE = V1       :Switch on the music board for V1/60 s.
0212    F107    V1 = TIME       :Use V1 to count down the delay time
0214    3100    SKIP;V1=0       :during which program halts. If
0216    1212    GO 0212         :not done count on in loop. Else
0218    E19E    SKIP;KEY=V1     :Check if key 0 (V1) is pressed
021A    1202    GO 0202         :If not go get next tone-note word
021C    121C    GO 121C         :Else stop (or go somewhere else)
021E    xxxx


0220    3416  340B  2E0B  290B  240B          Alan Fraser's Tune
022A    2216  2216  2216
0230    1E16  1E16  1E0B  190B                (1/4) note is hex-16
0238    222C  0016                            equivalent to M.M.=164
023C    6816  680B  5D0B  560B  4E0B
0246    4516  4516  4516
024C    4E16  5D16  4905  4521
0254    682C  0016
0258    00FF
```

NEW RCA PRICES EFFECTIVE 1/1/79

VP-44	Four 2114 RAM IC's	$ 36
VP-45	Four 9131 RAM IC's	$ 36
VP-550	Supersound Baord	$ 49
VP-560	EPROM Board	$ 34
VP-565	EPROM Programmer Board	$ 99
VP-570	Memory Expansion (4K RAM)	$ 95
VP-575	System Expansion Board	$ 59
VP-580	Expansion Keypad	$ 15
VP-585	Keypad Interface Board	$ 10
VP-590	Color Board	$ 69
VP-595	Simple Sound Board	$ 24
VP-600	ASCII Keyboard (avail Jul)	$ 49
VP-700	Tiny BASIC ROM Board	$ 39
VP-711	VIP (Assembled)	$249
TC1210	Nine Inch Monitor	$195
TC1212	Twelve Inch Monitor	$325
TC1217	Seventeen Inch Monitor	$435
VIP311	VIP Instruction Manual	$ 5
VIP320	VIP User's Guide	$ 5
VIP710	VIP Game Manual	$ 10
MPM201B	CDP1802 User Manual	$ 5
	1861 Data Sheet	$ 1.50

MISCELLANEOUS

Astle's LIFE.....$10
Hottle's BASEBALL.....$10
Stein's EDITOR.....$ 5
Stein's work notes....$10
(includes notes, flow charts, listings)
Hersh's EDITOR.....$ 5
Pernisz's LUNAR LANDER$ 5
Barrett's BREAKPOINT..$ 5

Studio II Kit A.....$ 5
Studio II Kit B.....$15
Studio II Kit C.....$30
Studio II Kit D.....$50
PC Board Only.....$10
PROM Only.....$15
Studio II Kit E.....$3.50

VIPER Back Issues...$2.00

YES! I'd like to subscribe to the VIPER and receive all ten issues of this year's volume! I enclose $15.00 in full payment.

Send your check for $15.00 (or use your VISA, MC, or BAC card) for a full year's subscription. Outside USA and Canada, add $10 for overseas airmail. Sorry — no C.O.D. or billing is available. Mail your order to THE VIPER, P.O. Box 1142, Columbia, MD 21044.

NAME _____

ADDRESS _____
(Please give street address, not postoffice box, for UPS shipping)

CITY, STATE, ZIP CODE _____

MC/VISA/BAC No. _____ BANK No. (MC ONLY) _____

EXPIRATION DATE _____ SIGNATURE _____
(Required with credit card orders)

☐ Vol. 1 (July 1978 - May 1979) ☐ Vol. 2 (July 1979 - May 1980)

THE VIPER * P.O. Box 1142 * Columbia, * MD * 21044 * (301) 730-5186

VIPER
P.O. Box 1142
Columbia, MD 21044

TO:

Place
Stamp
Here

VIPER
P.O. Box 1142
Columbia, MD 21044