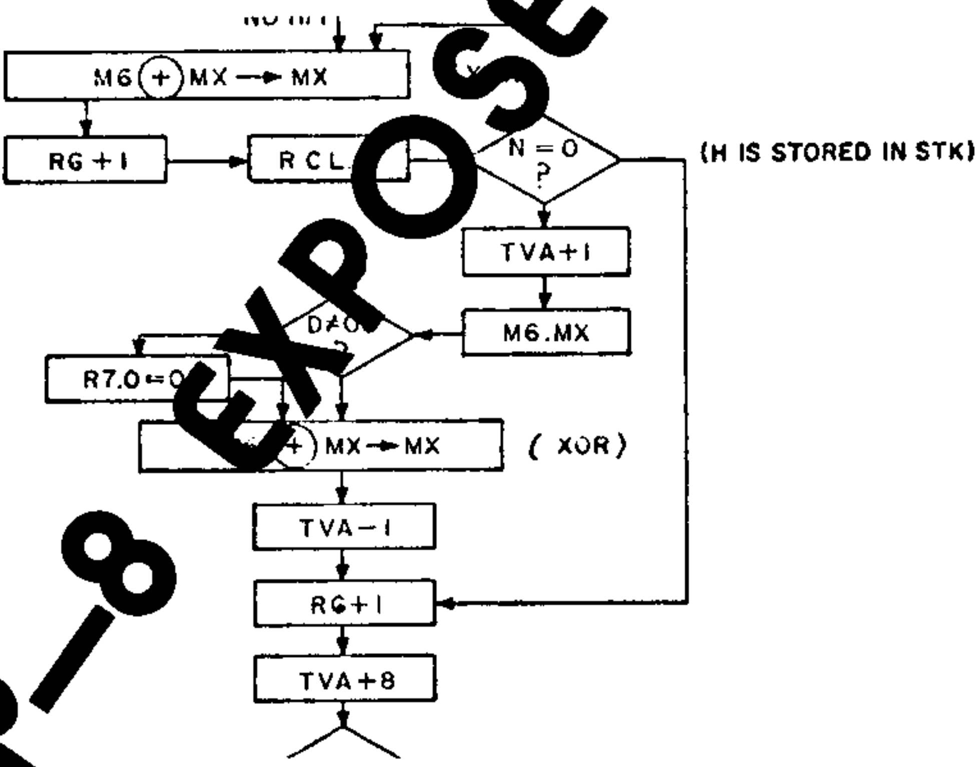
VIPER

VOLUME 1

AUGUST 1978

ISSUE 2



CHIP–8 EXPOSED!

PLUS **VIP TEXT EDITOR**

$2.00

AN ARESCO PUBLICATION

EDITORIAL

Editorials are not my strong point - and most of the VIPER issues will not have one. But I couldn't pass up the opportunity this month to tell you how much I appreciate the overwhelming response the VIPER has enjoyed from VIP owners (and prospective owners) all over the USA and Canada. In the first month alone, we've received more than twice as many subscription orders as we expected; articles, ideas, suggestions, and requests for specific information; even a few CHIP-8 programs.

I have shared your response with RCA's VIP product manager, Rick Simpson. He's as pleased and impressed as I am - as you can see in the New From RCA column in this issue, RCA has decided to support the VIP in a big way, and is turning out new VIP related products so fast it makes your head spin. We aren't supposed to know - or even guess - that there may be a VIP version of TINY BASIC in the works at RCA, so don't breathe a word to anyone about it - but I caught a peek at a memo which would suggest that someone at RCA is working very hard to get TINY BASIC up and running on the VIP by Christmas.

This issue contains the most-requested article (an indepth discussion of the CHIP-8 interpreter).
There are a few other goodies thrown in, as well. You'll see that this issue is not all prettily typeset, as issue #1 was - we couldn't take the chance of introducing errors into the manuscripts. In fact, from now on, most of the articles will be copies of the author's original work. Typists generally don't understand flowcharts, schematics, or code, and errors are remarkably easy to come by. One of the reasons this issue is two weeks late is a belated decision to forgo typesetting..... The next issue will be on time, since we already have most of the material in-house (thanks to all of you who wrote and shared your ideas and discoveries with us!)

Hope to see some of you at PC '78 in Philadelphia. Come by the RCA booth and see some of the marvelous new VIP related products
Until next month, then.

Terry

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).

Readers are encouraged to submit articles of general interest to VIP owners. Material submitted will be considered free of copyright restrictions and should be submitted by the 1st day of the month in which publication is desired. Non-profit organizations (i.e., computer clubs) may reprint any part of the VIPER without express permission, provided appropriate credit is given with the reprint. Any other persons or organization should contact the editor for permission to reprint VIPER material.

Advertising rates are as follows:

1/4 page	– $25.
1/2 page	– $45.
3/4 page	– $65.
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 Terry L. Laudereau. For information contact Editor, VIPER, P.O. Box 43, Audubon, PA 19407. (215)631-9052

The VIPER is not associated with RCA in any way, and RCA is not responsible for its contents.

Dear Terry,

Having for some time been fascinated by the 1802uP and by rather vague reports that it was designed to support compact interpreters, I ordered the VIP directly from RCA at PC '77 in Atlantic City. Before that time, I had breadboarded the "ELF" described in Popular Electronics.

Not being much interested in video games, my primary reason for purchasing the VIP was to learn numerical interpreter techniques; my second reason was because of the built in cassette I/O and video interface.

My video display is a 9" Hitachi black and white model PA-5 with the Pickles and Trout direct video entry conversion kit. This is a combination which I can heartily recommend to everyone. My cassette recorder is a low quality $29 model. At first, I had a great deal of difficulty with battery operation. An A.C. adapter solved those problems.

After writing a few simple CHIP-8 programs and implementing some of the games in the instruction manual, I analyzed the structure and operation of CHIP-8. In the process, I have produced a map of locations 0000-01FF and have flow charted some of the more complex subroutines (instructions) such as the DXYN instruction. I have also flow charted the ROM monitor program but much of it remains obscure to me. Although some might complain that this information should have been supplied with the VIP, I found the experience invaluable in learning machine (1802) language programming techniques. Also as a result of my analysis, I have found some possibilities in CHIP-8 which you may wish to communicate to your readers.

The 8XYN instruction (N=0,1,2,4,5) has four undocumented functions - 8XY3, 8XY6, 8XY7, and 8XYE. This is due to the fact that the 8XYN instruction operates by executing a single byte subroutine formed from the "N" digit. The description that follows applies to all values of N except N=0. In this case, the contents of VY is simply stored in VX.

The 8XYN subroutine begins execution with P=3, X=2, R5 pointing at the last byte of 8XYN, R6 pointing at VX, and R7 pointing at VY. If N is not 0, a hex "D3" is pushed on the stack -M(R2)- followed by a byte composed of the last byte of 8XYN orred with a hex "F0". X is then set to 6, the D register is loaded with the contents of VY and a SEP 2 -> P is executed. Thus, the single byte subroutine "FN" is executed, followed by a "D3" or SEP 3 -> P which returns control to the 8XYN subroutine. Following this, the contents of the D register is stored in VX and the state of DF (0 or 1) is put in VF.

Therefore, if N=3, 6, 7, or E, the functions of exclusive or, shift right, subtract, and shift left respectively are added. This is summarized in the following table:

INSTRUCTION	RESULT	"FN"	MNEMONIC
8XY0	VX <- VY		
8XY1	VX <- VX + VY (VF <- DF)	F1	OR
8XY2	VX <- VX * VY (VF <- DF)	F2	AND
* 8XY3	VX <- VX ⊕ VY (VF <- DF)	F3	XOR
8XY4	VX <- VX & VY (VF <- DF)	F4	ADD
8XY5	VX <- VX - VY (VF <- DF)	F5	SD
* 8XY6	VX <- (SHR)VY (VF <- DF)	F6	SHR
* 8XY7	VX <- VX - VY (VF <- DF)	F7	SM
* 8XYE	VX <- (SHL)VY (VF <- DF)	FE	SHL

N = 8 through D or F cannot be used because these values would result in the execution of an immediate instruction with uncertain (at best) results. A CHIP-8 program to demonstrate the 8XYN instruction follows:

```
01F2   F8 LDI 81->D
01F3   81
01F4   BA PHI D->RA.1
01F5   F6 SHR 0->D->DF
01F6   F6 SHR 0->D->DF
01F7   F6 SHR 0->D->DF
01F8   F6 SHR 0->D->DF
01F9   30 BR  BR 012F
01FA   2F

       ----

0200   6370 V3=70
0202   640F V4=0F
0204   6A00 VA=00
0206   6B00 VB=00
0208   A270 I='X='
020A   2244 DO S.R.
020C   8600 V6=V0
020E   6B06 VB=06
0210   A274 I='Y='
0212   2244 DO S.R.
0214   8700 V7=V0
0216   6B0C VB=0C
0218   A279 I='N='
021A   2244 DO S.R.
021C   8042 V0=V0*V4
021E   8031 V0=V0+V3
0220   A261 I=0261
0222   F055 MI=V0
0224   6F00 VF=00
0226   2260 DO S.R.
0228   80F0 V0=VF
022A   6B12 VB=12
022C   A27E I='F='
022E   DAB5 SHOW 5 @ A,B
0230   224E DO S.R.
```

```
0232   8060 V0=V6
0234   6B18 VB=18
0236   224E DO S.R.
0238   F20A DEBOUNCE
023A   E4A1 SKIP IF KEY≠V4
023C   11FC GOTO 01FC
023E   123A GOTO CHECK KEY
0240   0000
0242   0000
0244   DAB5 SHOW 5@ A,B
0246   F00A V0=KEY
0248   F10A V1=KEY
024A   0266 DO M.L.S.R.
024C   8011 V0=V0+V1
024E   6A09 VA=09
0250   F0F2 I=MSD V0
0252   DAB5 SHOW 5 @ A,B
0254   6A0F VA=0F
0256   F029 I=LSD V0
0258   DAB5 SHOW 5 @ A,B
025A   6A00 VA=00
025C   00EE
025E   0000
0260   8600 8XYN S.R.
0262   00EE
0264   0000
0266   F8 LDI F0->D
0267   F0
0268   A6 PLO D->R6.0
0269   06 LDN M(R6)->D
026A   FE SHL DF<-D<-0
026B   FE SHL DF<-D<-0
026C   FE SHL DF<-D<-0
026D   FE SHL DF<-D<-0
026E   56 STR D->M(R6)
026F   D4 SEP 2->P
```

```
0270  88     'X='
0271  53
0272  20
0273  53
0274  88     'Y='
0275  53
0276  20
0277  23
0278  20
0279  88     'N='
027A  CB
027B  A8
027C  96
027D  86
027E  F6     'F='
027F  83
0280  F0
0281  83
0282  80
```

Use of the program is simple - enter two digit values for X, Y, and N. These values and the resultant values of VF and VX are displayed. The first digit entered for N is ignored; the last digit of N determines the function performed - or, and, add, etc. Depressing key F restarts the program.

Note that a machine language subroutine was entered at location 01F2. This provides a new CHIP-8 instruction -FXF2- which sets I to the hex pattern of the most significant digit of VX. The instruction loads the contents of VX into D, shifts D right 4 times, then branches to the appropriate place in the FX29 subroutine. The space from 01F2 to 01FB is free for the addition of other "FX" type instructions which are found useful. For example, set timer equal VX and wait, shift VX left one digit position, and so on.

Another unused location begins at 00FC and ends at 0104. This space is suitable for often used machine language subroutines such as wait for timer equal zero. Or, by moving the two beginning bytes of the "FX" subroutine at locations 0105 and 0106 to locations 00FE and 00FF, another "FX" instruction -FX00- can be inserted at locations 0100 to 0106 in front of the FX07 instruction. A possible "FX" instruction subroutine which will fit here is 06FEFEFEFE56D4. This series of instructions will shift VX left four times or one digit position. However, if this is done, one other change must be made. The interpreter table at locations 0050 to 006F which contains the addresses of the CHIP-8 instruction subroutines must be changed to reflect the new entry point of the "FX" subroutine. Locations 005F and 006F contain 01 and 05 respectively which is the original starting address. If the bytes at 0105 and 0106 are moved to 00FE and 00FF, a 00 must be placed in 005F and an FE in 006F.

I have written a simple editor program which resides in the first two pages of RAM. It consists of a numerical interpreter in locations 0000-014F and the editor program, written the numerical language, in locations 0150-01FF. The functions of the editor allow me to display and alter any location. The display address can be rapidly or slowly incremented or decremented. There is also a copy function which will copy any range of locations to any location except 0000-01FF, of course.

I have also written an expanded CHIP-8 language which I call CHIP-8 1/2. It occupies 3 pages and although very similar, is totally incompatible with CHIP-8. I was able to add two new op codes by putting EXA1/9E into the "FX" series of instructions and by combining 5XY0 and 9XY0 into one op code. The two new functions are branch to MM if VX = 0 or VX ≠ 0 and take the form: NXMM. Another major change over CHIP-8 was the relocation of the "FX" instructions to page 2, allowing a full page of this instruction type. Also, the display instruction was expanded to include OR, AND, XOR, and test functions.

I have witten a LIFE program which occupies practically all of my VIP's 2K of memory. It consists of a large machine language subroutine supported by CHIP-8. The LIFE grid is a 64 X 32 cell array; a new generation is displayed every 2 1/4 seconds. Page 2 is occupied by a CHIP-8 program which allows the generation of a starting pattern, clearing the array, depositing predefined patterns, and starting and stopping the LIFE process. Page 3 is occupied by the LIFE subroutine. Page 4 is a lookup table which is used to find the population count of a cell. Pages 5 and 7 are the alternate generation display buffers. Page 6 is used to store predefined patterns. This program evolved from an all CHIP-8 program to the inclusion of larger and larger machine language subroutines as I sought to decrease the cycle time from ten minutes to the present 2 1/4 seconds. I don't believe that unrolling my current LIFE subroutine any more will bring substantial gain. Possibly there is a faster algorhithm which can be employed. However, I think that the only way to gain a significant increase in speed will be by a hardware change. That is, by the addition of a line buffer to reduce the overhead of repeated DMA requests for the same 8 bytes. Such a line buffer would have the added advantage of allowing the use of three cycle instructions.

In the future, I plan to design a line buffer which will take the form of a plug-in module containing the video interface chip, a line register, and miscellaneous logic. The plug in module will replace the video IC in its present location. At the same time, I may investigate the possibility of expanding the display size to 128 by 64 or some such size.

Another hardware change that I plan to implement is the addition of some sort of primitive disk-like random access device. It will probably be an endless tape loop - cassette or cartridge.

My software plans will be combined into a single operating system, a super CHIP-X, which will include numerical programming language with immediate execution of instructions entered from the keypad, editor, tape access with file management (if I can come up with a satisfactory random access device), and perhaps program relocation. The numerical instructions will probably be three or four bytes in length with one byte op codes. Of course, more than 2K RAM will be required for all this. I have ordered the memory expansion kit from RCA. Hopefully this will be enough.

I am employed by a large computer manufacturing company headquartered in Blue Bell, Pa. My background is primarily electronics, but my software experience is catching up with that.

Most of my adult employment has been in the educational/technical writing fields. I am more than willing to join/form a VIP user's group and to help anyone who wants help-with their VIP.

Please feel free to publish any or all parts of this letter.

Sincerely,

Peter K. Morrison

NEW FROM RCA

The VIP will be sporting vivid color this fall with the introduction of the VIP COLOR BOARD from RCA. You'll have program control of three background colors & eight foreground colors with CHIP-8C, the color-language addition to CHIP-8. Available late October. Priced under $80.00.

Convert the VIP single-tone output to 256 different frequencies with the new VIP TONE BOARD from RCA. With a single machine language subroutine added to either CHIP-8 or CHIP-8C, you'll be able to set the frequency and duration of the output tone. Speaker and jacks included. Available late '78; priced under $30.00.

Your VIP will be synthesizing two-part harmony with RCA's newest VIP product: the MUSIC BOARD. You'll have program control of frequency, duration, and amplitude envelope for each of two independent output channels, and an on-board potentiometer will control tempo. There will be a provision for sync output - for multi-track recording or slaving several VIPs for simultaneous play. The software, incidentally, will support the PAIA drum synthesizer which can be hooked on thru the output port. No speaker included. Under $50.00.

Add 4K of static RAM to your VIP by plugging in still another new VIP option. The MEMORY EXPANSION BOARD attaches through the expansion connector, and jumpers will address any of the first four 4K memory segments. Available by the end of the year, for under $100.00.

If you're a fan of two-player video games, this will please you! The new VIP EXPANSION KEYPAD is just what you've been waiting for. The 16-key keypad and cable connects to a socket on the color board or on its own (also new!) VIP KEYBOARD INTERFACE CARD. Instructions are included for use with either CHIP-8 or CHIP-8C. Available late October, each will be priced under $20.00.

At last you can program your own high-level language for the VIP with RCA's new EROM BOARD and the EROM PROGRAMMER. The board allows two Intel 2716 EROMs to be interfaced to the VIP and has provisi ns for placing EROMs anywhere in VIP memory space. It also allows re-allocation of on-board RAM in memory space. The programmer allows you to program the Intel 2716 EROM, and comes complete with software to program, copy, and verify EROM. All required EROM voltages are generated on board. Both should be available "soon". The EROM board is priced at under $50.00 and the Programmer will be less than $130.00
