

VIPER

February - March 1983

Volume 4, Number 6 Journal of the VIP Hobby Computer Assn.

The VIPER was founded by ARESCO, Inc. in June 1978

Contents

EDITORIAL	4.06.01
SOFTWARE	
Mini-Op for a Mini Assembler	4.06.02
by Tom Swan	
Double Array Modification	4.06.12
by Ron Applebach	
Advertisement	4.06.11
SOFTWARE	
PIN-8 Music by David Ruth	4.06.14
America The Beautiful	
READER I/O	4.05.15
VIPHCA INFO	4.05.18

Editorial

RENEWAL NOTICE

This is the final issue of Volume 4. To continue your membership in VIPHCA and receiving VIPER, you must send in your dues for 1983. It's still \$12 per year and includes six issues of VIPER. Please return the back page or a photocopy, just as last year. Don't forget to send the money! Last year, we had a couple of people who forgot to include a check with the renewal form. Those of you have already sent in your dues need not send in the form, so long as your address is unchanged. But I wouldn't mind hearing from you, either.

Also in this issue is a notice about the RCA Games book for the VIP, VP-710. You might like to order a copy at the same time as you renew and save a postage stamp. We got these as a gift from RCA, so the price you pay is to cover shipping and postage, plus put a little into VIPHCA's treasury.

Since the book originally sold for \$10, two bucks is a pretty good price! And for an additional \$3, you can have a cassette with all the programs.

I'm sure there are some of you who own computers other than the VIP or ELF. I'd be interested to know which machine(s) are the most popular. The prices of many of the "small" computers have come down to the point that they are hard to resist. The Timex-Sinclair TS-1000 no doubt leads the group in the lowest price sweepstakes, but the Commodore VIC-20 is available in many areas for \$150 or less. Atari and Texas Instruments have also been aggressively marketing their machines, and there have been persistent rumors that Radio Shack will again reduce the price of the TRS-80 Color Computer.

I'm looking forward to the fifth year of VIPER, and I hope you will continue to find our newsletter valuable. And, as always, I invite you to send in your programs or ideas you would like to share with other VIPHCA members. In the Reader I/O section, I've included a very interesting letter I got from Brian Hudson. He has some remarkable plans for the VIP; therefore, I felt that it was worthwhile to print his entire letter and the info sheet for his project. Any of you who would care to lend a hand with his project, please drop him a note.

73, Ray

Mini - Op for a Mini Assembler

by Tom Swan

Several of you have written requesting a VIP program that can be used to run the assembler and disassembler in my book, "Programmer's Guide to the 1802", (available from Hayden Publishing Co., Rochelle Park, NJ).

The hexadecimal listing included here is a bare-bones, "Mini" operating system for a VIP with 4k of memory. It is designed to simulate the features you would expect to find on a larger computer with a character display and keyboard, and can be used with or without the assembler, which of course is not listed here. A jump table, indexed by the keyboard's control character codes, makes adding new routines to the monitor program a snap.

HARDWARE STUFF

You must have an ASCII keyboard connected to the parallel input port of the VIP (the right slot looking from the front of the computer). Flag line EF4 is used to indicate that a key is being pressed. The keyboard's "ready" line, or output strobe line, should be connected to pads "L" (the EF4 line) and "K" (the VIP's parallel input "set" pin).

If your keyboard works with my Text Editor-21 program from Pips For Vips Vols 1 & 2, it should operate with the Mini-Op listed here. An RCA keyboard should also work, but you may have to cut data line #8, grounding this on the VIP end, to cancel the parity bit generated by the keyboard.

SOFTWARE STUFF

I apologize for the lack of a comprehensive listing, as I have usually tried to provide. Instead, I will cover the major points of interest and list the subroutine addresses. A full listing would take up too much room, and a lot of this same material has already been published in one place or another. After you have the program running, you can use the disassembler to list the entire operating system. It can also be used to make changes to itself, a feature I used heavily when designing the system.

IMPORTANT: Even though the listing indicates a starting address

of \$1000, you should enter the program starting at \$0000. (My print-dump program also runs at \$0000, and I was forced to relocate the Mini-Op to an external bank of memory at \$1000 for the purpose of producing the hexadecimal printout.)

First enter the listing printed here. Then, enter the assembler and disassembler programs starting at \$0420 through \$09FF from the Programmer's Guide. Save ten pages ("A") on tape from address \$0000. I have supplied the Viper with a tape of the listing as printed, but you will still have to enter the assembler from the book.

If you do not have the assembler entered, temporarily put a D5 (RETURN) instruction at locations \$0600 and \$0700, or the system is likely to crash when it tries to call the assembler.

USER STUFF

After the program is entered, flip the VIP to run. You should see a solid block cursor in the lower left corner of the screen. Try typing something. When you press the return key, you should receive an "ERROR" message (unless you do not have the assembler entered).

When you type something, it is entered into a character buffer located at \$0400 to \$041F. The characters in this buffer are then available for processing by the assembler, or by your own routines.

Typing a control code causes the system to call an address in page zero (\$0000-\$00FF) as indicated in a jump table located at \$00C0-\$00DF. The bytes in this table correspond with the decimal ASCII codes 0 to 31. Typing an "RETURN" character (ASCII 13 in decimal) for example, causes the system to first look in the jump table at 00CD (\$00C0 + decimal 13), get the \$8B byte located there, and then jump to address \$008B. If you examine that code, you will find a call to address \$009A followed by a branch back to \$002C. All routines should end with a branch back to \$002C where the monitor will again display a cursor and await the next input.

The assembler is called at \$009A with a call to \$0600. This section (\$009A-\$00A4) both calls the assembler and then redisplay the assembled line by calling a print routine at location \$0258.

Now, all that may seem complicated, but there really isn't that much happening. Most of the complicated parts of program are there to control the VIP's display, which doesn't come equipped for letters and symbols and the associated operations normally found on computer terminals. By the way, you may be interested to know that I used this program to write and test all of the routines in the "Mini

Library" at the end of the Programmer's Guide to the 1802.

Here is a list of the main points of interest, the register assignments, and some other useful information. Please let me know if you have any troubles or questions. I have sent the Viper a copy of the original listing (penciled in my somewhat legible scrawl), but I don't know by what means Ray can make this available. Perhaps there will be a note here explaining how you can obtain a copy.

Some people have expressed their surprise that I had developed an assembler program on a "little" RCA VIP computer. Most are equally surprised at my answer, "I didn't." The program, as are all of my programs, was hand assembled in pencil and well tested before anything was committed to silicon circuits. I find that the more planning I do away from the computer; the more reliable the results. I wonder if this will help those of you who have written expressing total confusion with assembly language. Try writing it down. Understand the logic of what you are trying to do before typing it into the computer. A little planning (even for a "little" computer) goes a long way.

OPERATING INSTRUCTIONS

To assemble:

- 1) Specify address -- e.g. \$0000 (You must type the \$, but you do not have to type leading zeros.)
- 2) Type in an 1802 mnemonic preceded by a space, and press RETURN.
 - a) Register types need "Rn" where n = 0 to F.
 - b) Hexadecimal arguments must be preceded with \$.
- 3) Examples:

```
$A00 LDI $FF
PHI R2
CALL $11D
RET
```
- 4) Notice in the above example that the address only needs to be specified the first time. It is automatically incremented correctly for subsequent instructions until you enter a new address. If assembly is proceeding correctly, you should be seeing a confirmation of what you type, expanded in the disassembler's style.
- 5) CAUTION: you are directly assembling machine code into memory. You can quite easily overwrite sections of the monitor program and blow everything up. Consult the memory organization list below for "not used" memory

areas. The block at \$0A00 to \$0AFF is available as a testing area for your own routines.

To Disassemble

- 1) Specify address as when assembling. (e.g. \$0200)
- 2) A space MUST follow the address.
- 3) Press ESCAPE key to disassemble 15 instructions.
- 4) Note: you may press ESCAPE repeatedly from then on to disassemble the next 15 instructions, and so on. This is quite handy for examining large sections of code. Anything in the computer's memory can be disassembled. I suggest you use this command to examine the subroutines listed below and try to figure out how they work.

General Monitor Commands (Do not require the assembler)

- 1) Call user routine
 - a) Specify address -- do NOT use a \$ sign
 - b) a single space must follow the address
 - c) Press CTRL-Y to execute the subroutine located at this address. This routine should end with a D5 (RETURN) byte to return control to the monitor.
 - d) Note: This operation can be used to call sections of the monitor to perform various operations. For example, to clear the display, call location \$11D. To do this, you would type:

11D ^Y

...where the ^Y is formed by pressing the CTRL and Y keys at the same time. (Don't forget the space after the address.) You may call any of the monitor subroutines individually in this way.

Register assignments

- R0 - DMA pointer
- R1 - Interrupt PC
- R2 - Stack Pointer
- R3 - Program Counter
- R4 - Call Routine PC
- R5 - Ret Routine PC
- R6 - Pointer to return & arguments

Mini - Op by Tom Swan

R7 - Display cursor
R8 - Addresses Char Count in Display Subroutines
R9, RA - not used
RB - Addresses input buffer
RC, RD, RE, RF - not used

Memory allocation

0000 - 00FF

0000 - 0022 -- Initialization
002C - 0070 -- Input (main monitor loop)
0071 - 007B -- Jump table control routine
007C - 0085 -- Backspace/Bell routines
0086 - 008A -- Part one execute user sub
008B - 008F -- C/R. Part one assemble call
0090 - 0099 -- Line feed
009A - 00A4 -- Part two assemble call
00A5 - 00AC -- Call disassembler
00AD - 00BD -- Part two execute user sub
00BE - 00BF -- not used
00C0 - 00DF -- Jump table
00E0 - 00E1 -- Default return for unwritten routines
00E2 - 00EE -- not used
00EF - 00FF -- 4-page video interrupt routine

0100 - 01FF

0100 - 010F -- Standard Call routine
0110 - 011C -- Standard Return routine
011D - 0132 -- Erase display
0133 - 0139 -- Input character
013A - 0143 -- Sound bell
0144 - 0155 -- Erase cursor
0156 - 0168 -- Backspace
0169 - 0180 -- List (disassemble) 15 lines
0181 - 01FF -- not used

0200 - 02FF

0200 - 0236 -- Display one character
0237 - 0257 -- Display one bit row (of a character)
0258 - 0275 -- Print string subroutine
0276 - 0298 -- Scroll display
0299 - 02A2 -- Erase to end of line
02A3 - 02FF -- not used

0300 - 0FFF

0300 - 03FF -- Character set bit patterns
0400 - 09FF -- (see Programmer's Guide, pg 121)
0A00 - 0AFF -- not used -- user program area
0B00 - 0BFF -- Stack
0C00 - 0FFF -- 4-page display refresh

Points of interest

ERASE : Erase display pages
(**\$011D**) Input : none
 Output: \$0C00 - \$0FFF cleared by
 fastest means known (so far!)
 Changes : RE, RF.0

ONECHAR: Display one character
(**\$0200**) Input : RE.1 = ASCII char in \$20-\$5F range
 : R7 = Display cursor address
 : R8 = Address of character count
 (i.e. line offset position)
 Output: Character displayed at cursor + count
 : Old char erased first at cursor
 : Count == count + 1
 Calls : BITROW
 Changes : RE, RF

BITROW : Display one bit row of a character
(**\$0237**) Input : R8 addresses character count
 : RE.1 holds bits to be displayed
 Output: One row bits @ cursor + count
 : R7 == R7 + 8 (cursor address)
 Changes : RE.1, R7

PRINT : Print string
(**\$0258**) Input : RB addresses string ending with
 either ASCII 13 (\$0D hex) or
 with a \$00 byte (null).
 Output: String printed
 : RB advanced to end of string
 Calls : SCROLL, ONECHAR
 Changes : R8, RB

SCROLL : Scroll display up one line
(**\$0276**) Input : none
 Output: Display scrolled
 : Char count set to 0
 Calls : ERSEOL

Changes : R8, RD.0, RE, RF

ERSEOL : Erase to end of line
(\$0299) Input : R8 addresses character count
 : R7 addresses screen (cursor)
 Output: Characters from count to end of line
 are cleared
 : Character count == 0 (normally, must
 be saved prior to calling ERSEOL)
 Calls : ONECHR
 Changes : RE.1

GETCHAR : Input a character from keyboard
(\$0133) Input : RB addresses input buffer
 Output : Waits for keypress via EF4
 : Character deposited at M(R(B))
 Changes : no registers (RB must be advanced
 by calling routine to input strings)

BELL : Sound bell
(\$013A) Input : none
 Output: Tone produced using VIP "Q" line
 Changes : RF

CRSOFF: Cursor off
(\$0144) Input : none
 Output: Cursor erased
 Calls : ONECHR

BKSPC : Backspace
(\$0156) Input : none
 Output: One backspace
 : RB == RB - 1 unless RB.0 = 0
 : If count = 0
 then count == \$0F
 else count == count - 1
 Calls : BELL
 Changes : RB

(Note: I had some trouble using this routine while preparing this article. I suspect a hardware problem on my system, but there could be a bug here.)

LIST : Disassemble listing
(\$0169) Input : RA addresses starting address
 Output: Disassembles 15 instructions
 : RA advanced 16 instructions
 Calls : DISASM (from Programmer's Guide)
 Changes : RB, RA

###

Mini-Op by Tom Swan

Program Code

```

1000 90 B1 B3 F8 0B B2 F8 FF
1008 A2 F8 F1 A1 F8 01 B4 A4
1010 B5 F8 11 A5 F8 0F B7 F8
1018 C0 A7 F8 1E A3 D3 D4 01
1020 1D E2 69 F8 02 BF F8 FF
1028 AF F8 00 5F F8 04 BB F8
1030 00 AB F8 02 B8 F8 FF A8
1038 F8 00 58 E2 08 73 F8 26
1040 BE D4 02 00 60 F0 58 D4
1048 01 33 0B FF 20 3B 71 0B
1050 FF 60 3B 59 FC 40 5B 30
1058 4A 4B BE D4 02 00 08 3A
1060 64 D4 02 76 8B FF 20 3B
1068 3B D4 01 3A D4 02 76 30
1070 2C D4 01 44 93 BF 0B FC
1078 C0 AF 0F A3 D4 01 56 30
1080 3B D4 01 3A 30 3B D4 00
1088 AD 30 2C D4 00 9A 30 2C
1090 08 73 D4 02 76 60 F0 58
1098 30 3B D4 06 00 F8 00 AB
10A0 58 D4 02 58 D5 D4 06 00
10A8 D4 01 69 30 2C F8 00 AB
10B0 D4 05 00 93 BF F8 B9 AF
10B8 DF 9D B3 8D A3 D3 00 00
10C0 E0 E0 E0 E0 E0 E0 E0 81
10C8 7C E0 90 E0 E0 8B E0 E0
10D0 E0 E0 E0 E0 E0 E0 E0 E0
10D8 E0 86 E0 A5 E0 E0 E0 E0
10E0 30 3B 00 00 00 00 00 00
10E8 00 00 00 00 00 00 00 72
10F0 70 C4 22 78 22 52 E2 E2
10F8 F8 0C B0 F8 00 A0 30 EF

```

```

1100 D3 E2 96 73 86 73 93 B6
1108 83 A6 46 B3 46 A3 30 00
1110 D3 96 B3 86 A3 E2 60 72
1118 A6 F0 B6 30 10 F8 0F BE
1120 F8 FF AE F8 04 AF EE F8
1128 00 73 8E 3A 27 2F 8F 3A
1130 27 5E D5 EB 3F 34 6B 37
1138 37 D5 7B F8 1F BF 2F 9F
1140 3A 3E 7A D5 F8 02 B8 F8
1148 FF A8 08 73 F8 20 BE D4
1150 02 00 60 F0 58 D5 8B 3A
1158 5D D4 01 3A D5 2B 08 32
1160 65 FF 01 58 D5 F8 0F 58
1168 D5 F8 0F 73 D4 07 00 F8
1170 10 AB F8 00 5B AB 58 D4
1178 02 58 60 F0 FF 01 3A 6B
1180 D5 00 00 00 00 00 00 00
1188 00 00 00 00 00 00 00 00
1190 00 00 00 00 00 00 00 00
1198 00 00 00 00 00 00 00 00
11A0 00 00 00 00 00 00 00 00
11A8 00 00 00 00 00 00 00 00
11B0 00 00 00 00 00 00 00 00
11B8 00 00 00 00 00 00 00 00
11C0 00 00 00 00 00 00 00 00
11C8 00 00 00 00 00 00 00 00
11D0 00 00 00 00 00 00 00 00
11D8 00 00 00 00 00 00 00 00
11E0 00 00 00 00 00 00 00 00
11E8 00 00 00 00 00 00 00 00
11F0 00 00 00 00 00 00 00 00
11F8 00 00 00 00 00 00 00 00

```

Reminder: Enter the code starting at address 0000.

This program was also supplied by Tom on a tape cassette. For those of you who would prefer to not have to enter the code by hand for the Mini-Op program, VIPHCA will have it available for \$2. Send you check to VIPHCA,

32 Ainsworth Avenue
East Brunswick, NJ 08816.

Mini-Op by Tom Swan

```

1200 9E FF 20 FE FE AF F8 03
1208 BF 87 FA F0 52 08 F6 F1
1210 A7 F8 04 AE 0F FA F0 BE
1218 D4 02 37 4F FE FE FE FE
1220 BE D4 02 37 2E 8E 3A 14
1228 08 FC 01 FA 0F 58 87 FF
1230 40 A7 97 7F 00 B7 D5 08
1238 F6 3B 47 9E F6 F6 F6 F6
1240 BE 07 FA F0 57 30 4B 07
1248 FA 0F 57 E7 9E F1 57 87
1250 FC 08 A7 97 7C 00 B7 D5
1258 93 B8 F8 FF A8 0B 3A 61
1260 D5 FB 0D 3A 69 D4 02 76
1268 D5 4B BE D4 02 00 08 3A
1270 5D D4 02 76 30 5D F8 0C
1278 BE BF F8 00 AE F8 40 AF
1280 F8 04 AD 4F 5E 1E 8F 3A
1288 83 2D 8D 3A 83 93 B8 F8
1290 FF A8 F8 00 58 D4 02 99
1298 D5 F8 20 BE D4 02 00 08
12A0 3A 99 D5 00 00 00 00 00
12A8 00 00 00 00 00 00 00 00
12B0 00 00 00 00 00 00 00 00
12B8 00 00 00 00 00 00 00 00
12C0 00 00 00 00 00 00 00 00
12C8 00 00 00 00 00 00 00 00
12D0 00 00 00 00 00 00 00 00
12D8 00 00 00 00 00 00 00 00
12E0 00 00 00 00 00 00 00 00
12E8 00 00 00 00 00 00 00 00
12F0 00 00 00 00 00 00 00 00
12F8 00 00 00 00 00 00 00 00

```

```

1300 00 00 00 00 22 20 20 00
1308 55 00 00 00 57 07 50 00
1310 27 43 70 00 66 43 23 00
1318 66 66 60 00 24 00 00 00
1320 12 22 10 00 42 22 40 00
1328 05 25 00 00 02 72 00 00
1330 00 02 40 00 00 30 00 00
1338 00 00 20 00 10 20 40 00
1340 65 55 30 00 26 22 70 00
1348 71 74 70 00 71 31 70 00
1350 45 71 10 00 74 73 70 00
1358 74 75 70 00 77 11 10 00
1360 75 75 70 00 75 71 70 00
1368 02 02 00 00 01 01 20 00
1370 12 42 10 00 07 07 00 00
1378 42 12 40 00 71 20 20 00
1380 07 54 70 00 77 57 50 00
1388 65 75 60 00 76 66 70 00
1390 73 33 70 00 76 76 70 00
1398 76 76 60 00 74 57 70 00
13A0 55 75 50 00 72 27 70 00
13A8 11 57 70 00 45 66 50 00
13B0 44 47 70 00 57 75 50 00
13B8 47 77 50 00 77 55 70 00
13C0 77 57 40 00 77 55 60 00
13C8 75 76 50 00 76 73 70 00
13D0 77 72 20 00 55 57 70 00
13D8 55 55 20 00 55 77 50 00
13E0 55 25 50 00 55 22 20 00
13E8 71 24 70 00 32 22 30 00
13F0 40 20 10 00 62 22 60 00
13F8 25 00 00 00 00 00 00 70

```

***** Announcement *****

RCA sent VIPHCA a large quantity of copies of the VP-710 Game Book. The games are intended for the VIP computer, but since they are all in CHIP-8, it should not be a problem to modify them for the other 1802 machines. There are a total of 16 games, and all be one run in 2K of memory. A few of the game titles are: Pinball, Bingo, Bowling, Programmable Spacefighters. This is the same book that originally sold for \$10, but VIPHCA will send you a copy for \$2, which includes shipping. A cassette of the programs will be available for \$3 additional if you order the book. Send your check to VIPHCA at the usual address.

Advertisement

NEW GAMES - ON TAPE

WORD SCRAMBLE - ASCII keyboard required

This game will scramble any word or phrase typed in. It will accept 13 characters per line and 5 lines. (2K memory)

THE RACE - Color

You control the speed and direction of your car. See how many laps you can make in a given time limit. Be careful, there are 6 other cars on the race track with you. (2K memory)

DOGFIGHT - Color - 2 player - extra keypad needed

Each player controls a fighter plane and tries to shoot the other plane down. (3K memory)

HUNT THE MONSTER - Color - 64 x 64 resolution

The screen is divided into 40 squares. The monster is hidden in one of the squares. You must locate and kill him before you run into him and he gets you. There is also a slime pit to be aware of. (3K memory)

LET'S PLAY THE PONIES - 1 to 6 players - 64 x 64 resolution

Each player checks the odds for the race, picks a horse and places a bet. After all bets are placed, the race is run. At the end of the race each player's score is figured and displayed. See who can win the most money! (4K memory)

3 GAME ASSORTMENT

1 - RECALL - test your memory. 7 digit numbers are displayed for one second. You then enter the number & score for every digit you get right. See how high you can score with 8 numbers.

2 - WORD HUNT - 4 rows of 5 letters are displayed. Players get 2 minutes each to write down all the words they can find. Any number of players.

3 - FOUR IN A ROW - There is an 8 X 8 grid. Two players take turns placing their markers in the column of their choice. First player to get 4 in a row across, up, down, or diagonally, wins. (3K memory)

Price: Any 3 for \$10.00 All 6 games for \$18.00

Send your orders to:

F.L Kramer 2464 W. Maple Grove Rd. Bloomington, IN. 47401

Double Array Modification

by Ron Applebach

I wrote this mod for use in a maze program. It can be used for computing, and for X-Y grid and matrix manipulation. It should be easy to use with modified CHIP languages and any of the CHIP-8 interpreters. What it does is set the "I" pointer so that you can pull (read) or push (write) data into an array. This mod is set up for "F by F" or one memory page arrays. To use this mod, you must select a CHIP-8 instruction you rarely use, like BMMM, and enter the address of this routine into the CHIP-8 subroutine table. Then, you specify X,Y, and the page you wish to use. By changing the page, you can have multiple double arrays, depending on how much memory is available in your system.

Changes to Chip-8: This creates a new instruction, BXYP. X and Y are the array pointers, and P is the array page.

```
005B 02 ;page address of routine
006B 20 ;byte address of routine
```

In this example, memory location 0220 now contains the new instruction's code, since there is not enough room to include it in the body of the CHIP-8 interpreter. You may decide that the code for the array instruction should be placed elsewhere. For example, you could make the first CHIP-8 instruction 1240, forcing a jump to memory location 0240, and place the array routine so that it will start at 0202. In that case, make 006B equal to 02 as well, and begin the body of your CHIP-8 program at 0240.

Here is the Double Array Routine:

```
0220 45 FA 0F BA 07 FA 0F FE FE FE FE AA 06 FA 0F 56
0230 E6 BA F4 AA D4 - - -
```

Demonstration program #1

```
0200 F10A ;ENTER X CO-ORD
0202 F20A ;ENTER Y CO-ORD
0204 B12F ;SET I TO X,Y ON PAGE F
0206 F00A ;ENTER VALUE TO STORE
0208 F055 ;STORE IT
020A 1200 ;DO IT AGAIN
```

This program allows any byte 0-F to be put on the screen. Remember, the display screen is 8 units across, so if you enter 8,0 the value will appear on the 2nd line down.

Demonstration Program #2

This program is a pattern generator. It stores a value in the entire array, then increments the value. Since the array page = F (which is also the display page), you can see the byte values on the screen.

0200	6001	0210	7201
0202	6200	0212	3210
0204	6100	0214	1204
0206	B12F	0216	7001
0208	F055	0218	1202
020A	7101	***END***	
020C	3110		
020E	1206		

In conclusion, this modification can be used for many different things, such as: locations in a maze, space invader screen locations, data handling routines, etc. I hope you find it as useful as I have!

Also, here is a machine language program for use with the color board. Press the "C" key and the pattern changes. It runs about 14,000 miles an hour. Hope you like it.

00	F8	0F	A3	B2	90	B3	A4	A2
08	F8	81	B1	F8	46	A1	D3	22
10	F8	0F	BB	B4	F8	D0*	BF	F8
18	00	A6	E2	69	84	22	52	86
20	F4	3B	24	15	A4	AF	12	85
28	54	5F	3E	1C	86	FC	01	A6
30	36	30	30	1C	-	-	-	-

*NOTE: The byte at 0015 is D0 for high resolution color; it is C0 for low resolution color.

A Pin-8 program for your VIP by David R. Ruth

AMERICA THE BEAUTIFUL

Step 1: Load the PIN-8 interpreter.

Step 2: Load the following:

```
0259  FF
02E0  0101 0101 0102 0202 0202 0303 0303 0300
0300  0105 090D 0F13 171B 1D21 2529 2B2F 3337
0310-037F 0000
0380  0105 090D 1014 181C 2024 282C 2F33 373B
0390-03FF 0000
0400  0069 8926 6669 8924 6466 6769 6B6D C9C9
0410  8926 6669 8924 6270 6F70 726B D069 9232
0420  706E 8E2D 6D6E 706D 6B69 CE6E 8E2B 6B6E
0430  8E29 6969 6B6E 6970 CE00 0000 0000 0000
0440-04FF 0000
0500  0066 8622 6266 8421 6161 6464 6767 A666
0510  6486 2262 6684 2426 6969 6968 6869 6867
0520  6989 2969 6687 2969 6969 6967 67A6 6769
0530  8727 6767 8626 6669 6B6E 6967 C600 0000
0540-05FF 0000
0600-06FE 0000
06FF  ED
```

Break Table:

```
0270  1201 E016 01E0 FE12 01E0 1601 E0FE 1201
0280  E016 01E0 FE12 01E0 1601 E0FF 0000 0000
```

Step 3: Store on tape 7 pages.

1/17/83

Brian H. Hudson
33½ Cerice Circle
Marietta, Ga. 30060

Raymond C. Sills
VIP Hobby Computer Assn.
32 Ainsworth Avenue
East Brunswick, N.J. 08816

Ray, Keep up the good work ! The VIPER is a lifeline between my VIP and the rest of the world, since RCA has abandoned its hobby computer market. I assembled my first VIP from one of the original kits as an educational toy, and had lost interest until I discovered the first volume of VIPER. My interest was renewed again by the excellent PIPS for VIPS programs by Tom Swan. Floating point BASIC from RCA, which I now have in EPROM, certainly extended the usefulness of the VIP. I understand that the VP-701 BASIC was to be included in the unreleased VIP II computer. RCA representatives say that in addition to limited interest in the hobby products the VIP II project was dropped to devote full effort to the new line of 1802 development systems and data terminals. I wonder if RCA would release any schematics or software from the VIP II project? I could incorporate some of their ideas into a single-plugboard Super VIP that I am designing. The VIP Superboard will contain many of the features of the add-on boards and the main VIP board on a single 4.5"X6.5" wire-wrap plugboard compatible with the VIP expansion buss. The board would plug into any slot of an undedicated slot expansion chassis, such as the 5-slot unit available from RCA for the Microboard products. The remaining slots would hold any of the VIP expansion boards except the VP590 Color Board. Features of the VIP Superboard are listed in an attachment. Are you aware of any VIPHCA members who would like to share notes with me on this hardware project?

On the subject of VIP software, I will miss Paul Piescik's column in VIPER, and I want to thank him for his dedication, since he wasn't writing for the money. Should I write Paul for copies of the released Cuddly Software programs, or can you provide some tape copies? I will reimburse you for the tapes and your time. Also, I wish that VIPHCA could publish Tom Swan's PIPS for VIPS Vol.IV for sale to the membership. Tom's contribution to the VIP software library has been tremendous, and I would look forward to any new programs from south of the border.

Please renew my membership to VIPHCA for another year. Enclosed is a check for \$12 with my thanks for making VIPER a continued success.

Good computing !

Brian H. Hudson

Brian H. Hudson
33½ Cerice Circle
Marietta, Ga. 30060

FEATURES OF THE VIP SUPERBOARD

- 1) 4K RAM fully decoded and relocatable. Deselected by MINH if addressed 0000-0FFF.
- 2) CDP1802A CPU, upgradable to CDP1805A. Clock frequency automatically switched from 1.7MHz to 3.4MHz by TVOFF command for faster program execution.
- 3) CDP1861/1862 Color video chip set with 1K color map RAM.
- 4) Fully-decoded I/O instructions with CDP1853. CDP1852 input port with keyboard connector. CDP1852 output port connected to CDP1863 frequency generator, but switchable to a connector.
- 5) 2K X 8 EPROM with Monitor, fully decoded at 8000-87FF.
- 6) Buffered Data, Address Busses, and Clock Signals.
- 7) Tape motor control for two cassette recorders.
- 8) Dual keypad/joystick interface for Atari® controllers.

READER I/O

Dear Ray:

Some day I will get a printer hooked up to my VIP. Meanwhile, I have to hope you can read this. Present project, nearly complete, is incorporation of the NETRONICS Video board into my VP-601 keyboard. Next project is expansion, using the Quest expansion board. Meanwhile, I think about memory. I wrote you about obtaining 5114 chips a while ago and you put my inquiry into VIPER. I got only one response from a very nice guy in Denver who said his local parts supplier got them regularly at around \$10 each! I have now awakened to the 6116. Two K for 5 bucks! I have modified my VIP #2 (second hand, experimental) replacing the four 9131s with one 6116-4. Works fine. So third project will be to change over completely.

My real purpose in writing again is to ask about the Cuddly Software programs that Paul Piescik has released. Can you line up some real nice guy who could put two or three copies each on a C30? I have a very strong preference for TDK tapes from by stereo experience, but I can copy anything. VIP format preferred, but I can also use the Friedman loader (VIPER 1.07) to copy up to 15 pages of ELF II. All available literature would be needed by me because I am not yet very good at understanding programs from code. I would be glad to pay for costs and time involved. If a project is undertaken to do each program separately, then CSPIOP, CSIO, and SCTP are the ones I haven't got.

Regarding your continued Directorship, I think you are going a wonderful job. I am just bery glad that you are willing to continue with VIPER and keep alive that wonderful, but little known chip called the 1802! Sincerely,

George E. Frater 1780 Maripose Dr. Las Cruces, NM 88001

(Ed. Reply)

Well, George, thanks for the vote of confidence! (Blush) Sounds like you have been quite busy with your VIP. The 6116 is a very attractive chip: 2K static RAM in a single unit, very low on power drain, and getting less expensive almost every day! I'm sure there are others who would like to know exactly how you converted your VIP to take the 6116. By the way, my VIP uses 9131s! And it was build from a kit.

What I would like to do about Paul Piescik's programs is make them the basis of a series of articles for the VIPER. I have most of them on hand here at VIPER HQ and I can get copies of the tapes made without too much trouble. I'm going to check it out with Paul, but I'm sure he won't mind if we use his material here in the VIPER. RS

VIPHCA INFO...

The VIPER, founded by ARESCO, Inc. in June 1978, is the Official Journal of the VIP Hobby Computer Association. Acknowledgement and appreciation is extended to ARESCO for permission to use the VIPER name. The Association is composed of people interested in the VIP and Computers using the 1802 microprocessor. The Association was founded by Raymond C. Sills and created by a constitution, with by-laws to govern the operation of the Association. Mr. Sills is serving as director of the Association, as well as editor and publisher of the VIPER.

VIP and COSMAC are registered trademarks of RCA Corporation. The VIP Hobby Computer Association is in no way associated with RCA, and RCA is not responsible for the contents of this newsletter. Please send all inquiries relating to the VIPER to VIPHCA, 32 Ainsworth Avenue, East Brunswick, NJ 08816.

The VIPER will be published six times per year and sent to all members in good standing. Issues of the VIPER will not carry over from one volume to another. Annual dues to the Association, which includes six issues of the VIPER, is \$12 per year. Membership in the VIP Hobby Computer Association is open to all people who desire to promote and enjoy the VIP and other 1802 based systems. Send a check for \$12 in U.S. funds payable to "VIP Hobby Computer Assn." c/o Raymond Sills, 32 Ainsworth Avenue, East Brunswick, NJ 08816. People outside the U.S., Canada and Mexico please send \$18, due to additional postage charges. The VIPER is normally sent via first class mail, and airmail to members outside North America.

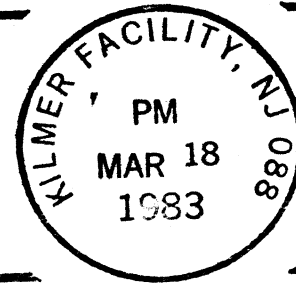
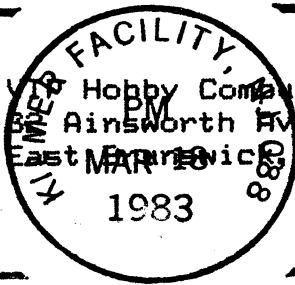
Contributions by members or interested people are welcome at any time. Material submitted by you is assumed to be free of copyright restrictions, and will be considered for publication in the VIPER. An honorarium payment is made to those whose material is published in VIPER to help cover the cost of a submission. Articles, letters, programs, etc., in camera-ready form on 21.5 x 28 cm (8.5 x 11 inch) paper will be given preferential consideration. Please send enough information about any program so that readers can operate the program properly. Fully documented programs are best, but memory dumps are okay if you provide enough information to run the program.

If you write to VIPER/VIPHCA, please indicate that it is okay to print your address in letters to the editor, if you want your address revealed to VIPER readers. Otherwise, we will not print your address in VIPER.

ADVERTISING RATES.....

1. Non-commercial classified ads from members: 5 cents per word, minimum of \$1.
2. Commercial ads and ads from non-members: 10 cents per word, minimum of \$2.
3. Display ads from camera ready copy: \$6/half page, \$10/full page.

Payment must accompany all ads. Rates subject to change.



RECEIVED THE CANADIAN DISTRICT
OF THE AMERICAN SOCIETY OF
FOR THE A.S.S. FOR THE A.S.S. DISTRICT
OF THE AMERICAN SOCIETY OF
FOR THE A.S.S. DISTRICT
OF THE AMERICAN SOCIETY OF

FIRST CLASS MAIL

Renewal of Membership

Please make any address correction and return this page or copy of it to VIPHCA headquarters along with your check for \$12. Outside the U.S., Canada, and Mexico send \$18 in U.S. funds, drawn on a correspondent bank.

Questionnaire

1. Do you do original programming on your 1802 machine?
☒ yes no sometimes
2. Is your programming done in CHIP-8?
☒ yes no
3. Is your programming done in machine code?
☒ yes no
4. Do you own another computer? (non 1802)
yes ☒ no

Which?.....