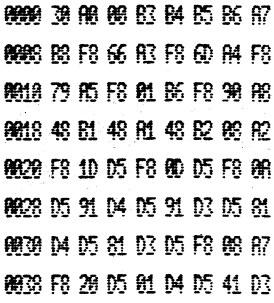
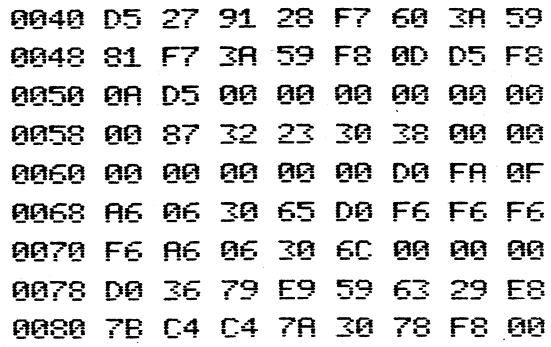
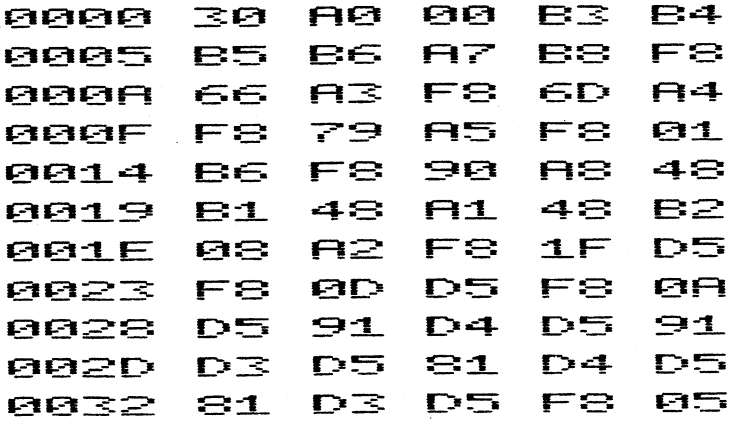
VIPER

VOLUME 1 NOVEMBER ISSUE 5







VIP-Printer Interfaces

$2.00 AN ARESCO PUBLICATION

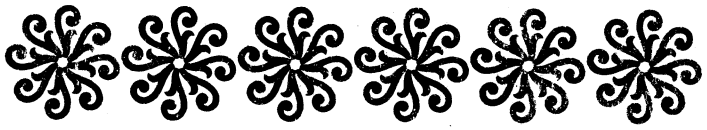
EDITORIAL

Hello again! Due to an attack of sloth on my part, as well as growing pains as ARESCO becomes a full-time occupation for me, this issue of the VIPER is a month late. Since we don't publish in December, the next issue should be on time.

To make up for our lateness, this issue features several innovations. We now have a printer on our VIP, so the poor quality listings you've seen in previous issues shouldn't occur again. Second, this issue introduces the VIPER Software Library, with an excellent package by Brian Astle as our first offering. And finally, we have decided to carry articles in the VIPER on another unique 1802 system. See the article "What Do You Think" for details.

Season's Greetings to all of you - have a merry and a happy!

Rick Simpson



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).

The VIPER is an Aresco Publication, edited by Rick Simpson. For information contact Editor, VIPER, P.O. Box 43, Audubon, PA 19407.

RAMBLING

Some of you must be wondering, "Just who is editing the VIPER?" since you've received letters from Rick Simpson (signed "editor") and you've all been in touch with me (as "editor"). Well, folks, at last the truth can be told!

I, Terry Laudereau, edited the very first issue of the VIPER, and I am editing this issue. Rick edited issues 2, 3, and 4. Rick has been managing the VIP product line for RCA since May, and he instigated the newsletter, encouraged RCA to respond to your comments regarding peripheral equipment and software you wanted, and in general helped make life easier for VIPpers while making VIP sales for RCA. The more familiar Rick became with the VIP, the more he wanted to do the VIPER - to write to all of you, to share your knowledge and experience, and to just dig in and find out all about the 1802 itself. He let our PET, our Apple, our TRS-80, the Z-80 system he built, and our Sorcerer sort of sit and gather dust while he went tooth and toenail after the VIP. He read every single one of your comments- the remarks you've made on your application for subscription forms; the letters you've sent to me, the notes I've made about telephoned questions. It didn't take long for him to get infected with your enthusiasm - and he gave notice at RCA: he plans to work full time with me at ARESCO, publishing newsletters, reviewing and evaluating software, digging up information to share with all of you. Many of you have written or called often enough so I feel I know you personally - and so does Rick. So now he's at RCA one or two days a week - long enough to be sure everything moves smoothly while the various option boards are going into production and delivery. He spends the rest of the week on VIP material here at ARESCO - doesn't even bother with KIM, which he deserted cruelly in favor of the VIP!

Meanwhile, I am working on newsletters for PET, Apple, and Sorcerer owners. When and if time permits (like now), I do a bit on the VIPER - but Rick claimed it as his own, and prefers to "do it himself". Since he understands hardware, machine language programming, and systems in general, I'm all for it - I'm what's called a computer generalist, and you people have simply outrun my knowledge! (Besides, I don't wanna lose my amateur standing!)

So there it is. Rick Simpson is really the editor of the VIPER. In fact, we just changed the masthead to show that. I am "staff". You'll be able to tell the difference between my work and Rick's by the difference in the tone - I ramble a lot, while Rick gets right to the nitty-gritty and doesn't waste words. So, after this one, last issue, the rest of the VIPER Volume 1 will be edited by Rick. Give him as much support as you've given me, and you'll have one helluva good newsletter!

Terry L. Laudereau

Terry L. Laudereau

A TEXT EDITOR FOR THE VIP -- Part Three

by Don Stein

To complete my text editor, I added tape input and output capability, under program control.

The first step was to decipher the VIP Operating System. I did this by brute force, and managed to locate the tape input and output routines on the ROM.

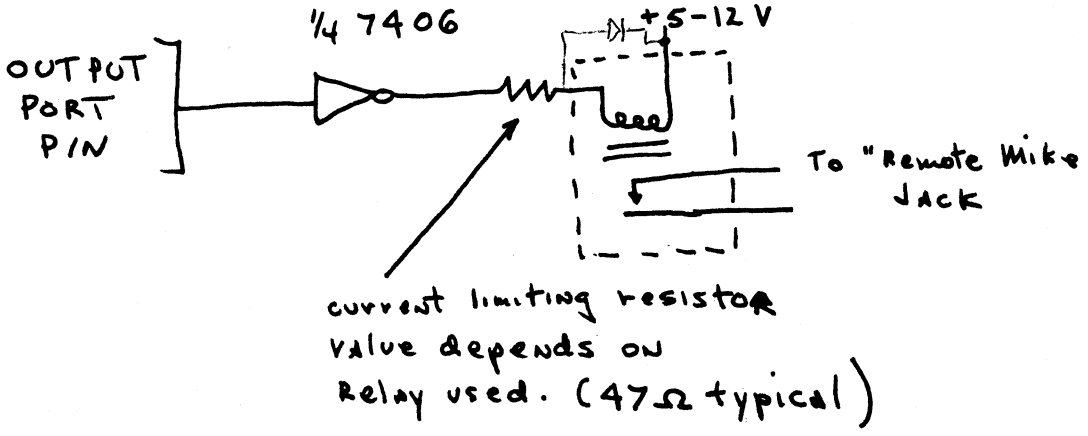
Contrary to the VIP Instruction Manual, the sync pulses are not four seconds long, but rather continue for about eight seconds. This is far too long for real-time tape I/O, and longer than is necessary to achieve proper sync. By trial and error, I found that the sync write interval could be shortened to two seconds (loop counter equals hex 10), with a sync read interval of ½ second (loop counter equals hex 04).

The text editor program, then, would have to save the registers (on the data stack) which were required by the VIP tape I/O routines; start the recorder motor; set up the correct values in the appropriate registers; call the VIP routines in the ROM; stop the recorder motor; and restore the original values in the registers.

Because of the necessity to modify the length of the sync intervals, it was necessary to copy part of the VIP tape routines in RAM, with the appropriate constants changed. Nonetheless, the tape routines require only 139 bytes, just over half a page.

Motor control is achieved by using the "remote mike" jacks on the cassette recorders. By closing the circuit, the motor would start; by opening the circuit, the motor would stop. Because of the relatively high current draw of cassette tape recorder motors (up to 250 ma), relays would be required.

The motor control circuit is shown in Figure 1.



A line of the output port would be fed to a 7406, which is a hex inverter/driver with an open-collector, high current-sink capability. The output of the 7406 would be fed through a current-limiting resistor to a relay coil, the other side of which is connected to the 5-volt supply. The normally-open contacts of the relay should be connected to the "remote mike" input of the tape recorder.

Up to eight tape recorders can be controlled in this manner. My text editor initially has been set up to control two recorders -- one for write, and one for read.

By the way -- the 7406 chip is readily available for less than 50¢; the relays are about $5 each.

One additional feature was built into the software. To avoid plugging and unplugging the "remote mike" wires, a subroutine was added to release the recorder motors from VIP control upon keyboard command. The "control T" key, followed by a digit, is used to release or engage any one or more of the connected tape units.

So that's my text editor. What next? How about an extended-BASIC compiler/editor? (I'm working on it).

73 to my fellow hams and computernicks.

The Am9131 and Am 91L31 memories are identical in every respect to their counter parts in the Am9131 and Am91L30 family, with the exception that the Memory Status output is not functional. Pin 10 on the AM9131/L31 products should not be used and should not be connected to any external circuit. Pin 10 is not used in the VIP circuitry, so 9131's and 9130's can be interchanged.

For those of you who have expressed an interest, here's a preview look at the games programs published in the new RCA Games Manual:

PINBALL	MESSAGE CENTER
BINGO	BLACKJACK
SLIDE	REVERSI
BOWLING	DEFLECTION
ANIMAL RACE	MOST DANGEROUS GAME
SUM FUN	SEQUENCE SHOOT
BIORHYTHM	PROGRAMMABLE SPACEFIGHTERS
LUNAR LANDER	BREAK-THROUGH

Here is the complete listing for Don Stein's text editor and operating system. Since the program takes over 1200 bytes, you may not wish to key it all in yourself. The VIPER is offering a special package consisting of a cassette tape of the program and complete set of Don's working documents. The document package consists of Xerox copies of Don's handwritten flowcharts, memory maps, and program listings, and is about sixty pages. We will also include a reprint of Don's entire article. Anyone seriously considering modifying the program for their own use will find these to be absolutely necessary.

The cassette is available from The VIPER for $5 and the documentation package is $12. Both are available together for $15. Note that the tapes will be hand-duplicated on our VIP using a Sankyo ST-40 recorder. The Sankyo tapes seem to read well on a variety of other recorders, but we make no promises.

Here are some other operating notes Don passed along to us:

1. To run the program, flip up the RUN switch and key in 0400 on the hex keypad. The screen should clear, leaving only a cursor. You can now proceed with input from your ASCII keyboard as described in the articles.

2. Tape drive #1 is controlled by output port bit 3. Drive #2 is controlled by bit 5. Tape drive #1 is the Read (input) deck, and #2 is the Write deck. If Tape #1 is to be used for both input and output, change byte 009E to a value of 08.

3. The control characters can be changed by modifying the table which starts at 04B2. Make sure the table concludes with the default entry 00 01 76.

Given the very limited availability of system software for the VIP, we are all indebted to Don for sharing his efforts with all VIP owners. If you have any questions, or just want to write to Don and let him know how useful you found his efforts, you can reach him at:

Don Stein
6012 Chatsworth Lane
Bethesda, MD 20014

```
      91 B2 BG
0000 [  ][  ][  ] AF 2F 2F 2F 2F
0008 8F B7 90 B1 B4 B5 B3 A7
0010 F8 4A A1 F8 6B A2 F8 58
0018 A4 F8 39 A5 F8 AB A6 F8
0020 23 A3 D3 E2 69 D4 00 40
0028 D4 00 40 93 BF F8 31 AF
0030 DF 60 72 A3 F0 B3 D3 00
0038 D3 16 46 A3 06 B3 30 38
0040 D4 00 68 D5 D4 00 00 D5
0048 42 70 C4 22 78 22 52 97
0050 B0 87 A0 28 E2 30 48 D3
0058 93 BF 83 AF 4F B3 4F A3
0060 9F 56 26 8F 56 26 30 57
0068 D4 00 7C D4 00 7C 60 72
0070 AF F0 FE FE FE FE 73 60
0078 8F F1 73 D5 2F 8F 73 73
0080 60 62 D3 D3 3E 7C 7B F8
0088 04 A8 88 3A 8A 36 8D 7A
0090 F8 04 A8 88 3A 93 8F FA
0098 0F 73 D5 94 BE F8 20 73
00A0 73 60 63 D4 00 B2 94 73
00A8 73 60 63 D5 84 BE F8 08
00B0 30 9F 96 BF 86 AF 60 72
00B8 AE 72 A6 F0 B6 9F 73 8F
00C0 73 97 73 93 B4 F8 E9 A4
00C8 F8 81 BC 73 61 9E 3A D9
00D0 F8 6F AC F8 10 B9 C0 80
00D8 97 F8 83 AC F8 04 B9 DC
00E0 33 DC 29 99 3A DF C0 80
00E8 CF 94 B3 F8 EF A3 D3 60
00F0 72 B7 72 A6 F0 B6 95 A7
00F8 B4 F8 58 A4 69 D5 00 00
0100 60 F0 FE 30 0B 73 D4 01
0108 80 60 F0 FE 33 12 FA 7F
0110 AD D5 F9 80 AD D5 9C 3A
0118 20 60 0A FA 0F F1 5A D5
0120 60 F0 F6 F6 F6 F6 73 60
0128 0A FA F0 F1 5A D5 F8 04
0130 AE 0D FA F0 73 D4 01 16
0138 8A FC 08 AA 9A 7C 00 BA
0140 4D FE FE FE FE 73 D4 01
0148 16 8A FC 08 AA 9A 7C 00
0150 BA 2E 8E 3A 31 D4 03 F6
0158 D5 9A 7F 00 BA D5 8C FC
0160 10 3B 64 D5 AC D4 01 F6
0168 D5 8C FF 60 33 6F D5 FC
0170 50 AC D4 01 F6 D5 1C 8C
0178 32 7E D4 01 F6 D5 2C D5
0180 F8 F0 73 8A FC 50 AA 9A
0188 7C 00 BA D4 01 16 8A FF
0190 50 AA 9A 7F 00 BA D5 94
0198 73 30 83 F8 B0 AF 94 5B
01A0 1B 2F 8F 3A 9E D4 03 AC
01A8 D4 03 C7 D5 D5 97 BA 87
01B0 AA 94 BC F8 50 AC D4 01
01B8 80 D5 94 BC BA 8C FA F0
01C0 FF 50 73 60 F6 F4 FE AA
01C8 9A 7E BA 8A FE AA 9A 7E
01D0 73 60 97 F4 BA 8C FA 0F
01D8 F6 73 60 3B DF 84 BC 8A
01E0 F1 AA D5 D4 01 5E 8C FA
01E8 F0 AC D4 01 F6 D5 F8 B0
01F0 73 D4 03 93 D5 D5 D4 01
01F8 97 D4 01 BA D4 01 80 D5
0200 00 00 00 00 44 44 44 04
0208 AA AA A0 00 AE AA AA EA
0210 4E 8E 2E 40 88 24 82 20
0218 4A AA 44 AA 44 44 40 00
0220 22 44 44 22 88 44 44 88
0228 0A A4 4A A0 44 4E E4 44
0230 00 00 44 48 00 0E E0 00
0238 00 00 00 44 02 24 48 80
0240 4E AA AA E4 C4 44 44 4E
0248 E2 22 E8 8E E2 2E 22 2E
0250 AA AA E2 22 E8 8E 22 2E
0258 88 8E AA AE E2 22 22 22
0260 EA AE AA AE EA AA E2 22
0268 04 40 00 44 44 00 04 48
0270 02 48 84 20 00 E0 0E 00
0278 08 42 24 80 EA 22 66 04
0280 4E 4E 4E 44 EE AA EE AA
0288 CA AC CA AC EE 88 88 EE
0290 CA AA AA AC E8 8E E8 8E
0298 E8 8E E8 88 E8 88 AA AE
02A0 AA AE EA AA E4 44 44 4E
02A8 22 22 AA AE AA CC CC AA
02B0 88 88 88 EE AE EE AA AA
02B8 AE EE EE EA EA AA AA AE
02C0 EA AA E8 88 EA AA AE EE
02C8 EA AE CC AA E8 8C 62 2E
02D0 EE 44 44 44 AA AA AA AE
02D8 AA AA AA 44 AA AA EE EA
```

```
02E0 AA E4 4E AA AA AE 44 44
02E8 E2 24 48 8E 4E 4E 4E 44
02F0 4E 4E 4E 44 4E 4E 4E 44
02F8 4E 4E 4E 44 EE EE EE EE
0300 1C 8C 3A 05 2C D5 93 93
0308 BF F8 0D AF DF F8 04 BE
0310 F8 B0 AE 1E 1E 4E 32 1B
0318 F3 3A 13 4E B3 0E A3 D3
0320 30 D9 60 8B F4 73 9B 7C
0328 00 73 D5 D4 03 20 60 72
0330 BF F0 AF 6B D3 D3 FE FE
0338 33 DF FE 33 DF D4 03 06
0340 D5 D4 01 97 D4 01 00 D4
0348 01 2E D4 03 00 D4 01 BA
0350 D4 01 80 D5 D4 01 9B D4
0358 01 AD D5 D4 01 97 F8 50
0360 AC AC 4B 73 D4 01 BA D4
0368 01 00 D4 01 2E 1C 8C 3A
0370 62 2C D4 03 AC D5 D5 8C
0378 FB 50 32 80 2C D4 01 F6
0380 D5 60 8B F7 AB 9B 7F 00
0388 BB FB 04 3A 92 F8 05 BB
0390 94 AB D5 60 8B F4 AB AF
0398 9B 7C 00 BB BF 8F FF 51
03A0 9F 7F 06 3B AB F8 06 BB
03A8 F8 50 AB D5 F8 B0 73 D4
03B0 03 81 D5 F8 10 73 D4 03
03B8 81 D4 04 57 D5 F8 10 73
03C0 D4 03 93 D4 04 57 D5 97
03C8 BA 87 AA F8 04 AF 94 5A
03D0 1A 8A 3A CE 2F 8F 3A CE
03D8 D5 8C FF 50 73 30 22 F0
03E0 5F 73 30 41 D4 01 97 D4
03E8 01 EE D4 04 57 D5 D4 01
03F0 97 D4 03 AC 30 EA 8A FF
03F8 40 AA 9A 7F 00 BA 2D D5
0400 94 AB F8 05 BB F8 02 BD
0408 D4 01 9B D4 01 AD 3E 0E
0410 D4 03 2B F8 30 A8 88 32
0418 1D 3E 0E 30 16 D4 03 2B
0420 F8 0B A8 30 16 D4 01 E3
0428 D4 03 20 60 72 B9 F0 A9
0430 F8 07 BF 94 AF F8 06 BE
0438 F8 F0 AE 2E 2F 0E 5F 89
0440 73 60 8E F3 3A 3B 99 73
0448 60 9E F3 3A 3B 2F 94 5F
```

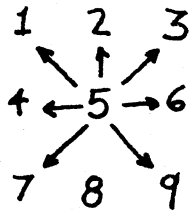
```
0450 89 73 60 8F F3 3A 4D 8C
0458 73 D4 03 5B 60 F0 AC D4
0460 01 F6 D5 8C FA F0 AC D4
0468 03 20 60 72 BE F0 AE FC
0470 10 AF 9E 7C 00 BF 4F 5E
0478 1E 9F FB 07 3A 76 94 5E
0480 1E 9E FB 07 3A 7E 30 57
0488 8C 73 F8 05 73 94 73 F8
0490 02 73 D5 D4 04 88 D4 00
0498 9B D4 04 5C D5 D4 04 88
04A0 D4 00 AC D4 04 59 D5 36
04A8 A7 73 D4 00 68 60 63 D5
04B0 00 00 03 03 54 13 01 76
04B8 08 01 AD 01 03 77 0A 01
04C0 5E 1A 01 5E 17 01 69 0D
04C8 01 E3 15 03 B3 04 03 BD
04D0 02 03 EE 10 03 E4 0C 04
04D8 25 18 04 63 09 04 9D 0F
04E0 04 93 14 04 A7 00 01 76
04E8 00 01 76 EE 40 00 00 00
04F0 48 24 48 24 40 00 00 00
04F8 4E EE 48 2E E0 E0 00 00
```

DRAW

Anders McCarthy
1359 W. Idaho Avenue
St. Paul, MN 55108

The display page used for this program is page three. This means that your picture may be saved on tape with the program and CHIP-8 interpreter by writing four pages on tape instead of three. It also means that the screen is not erased when you run the program. This is helpful when loading a picture from tape.

Originally, the drawing cursor is in the upper left-hand corner. To move it, pretend that it is on the "5" and press the key which corresponds to the direction to which you wish to move the cursor. For example, to move it down and to the right, you would press "9".



Originally, the drawing cursor will draw a white line wherever you move it. If you want to erase something, press "E". If you decide you want to draw some more, press "F". If you want to clear the whole screen, press "C". This also puts the drawing cursor in the upper left-hand corner. Finally, if you just want to pass over what you've drawn and not change anything, press "A".

This program has a built-in repeat, so if you hold down a key for more than about half a second, the key will start repeating itself. This is useful if you want to draw long lines, among other things.

If you wish to adjust the rate of repeat, the byte at location Ø26F may be changed. Similarly, the byte at Ø297 controls the delay before the repeating starts.

DRAW

```
0200  0270  DO MLS at 0270
0202  A26C  I=026C
0204  F365  V0 to V3 from M(I)
0206  A26D  I=026D
0208  2286  DO SBR at 0286
020A  340C  SKIP if V4=0C
020C  1212  GOTO 0212
020E  00E0  Erase screen
0210  1200  GOTO 0200
0212  3401  SKIP if V4=01
0214  121C  GOTO 021C
0216  71FF  V1=V1+FF
0218  72FF  V2=V2+FF
021A  125C  GOTO 025C
021C  4402  SKIP if V4≠02
021E  1218  GOTO 0218
0220  3403  SKIP if V4=03
0222  1228  GOTO 0228
0224  7101  V1=V1+01
0226  1218  GOTO 0218
0228  3404  SKIP if V4=04
022A  1230  GOTO 0230
022C  71FF  V1=V1+FF
022E  125C  GOTO 025C
0230  3406  SKIP if V4=06
0232  1238  GOTO 0238
0234  7101  V1=V1+01
0236  125C  GOTO 025C
0238  3407  SKIP if V4=07
023A  1240  GOTO 0240
023C  71FF  V1=V1+FF
023E  1244  GOTO 0244
0240  3408  SKIP if V4=08
0242  1248  GOTO 0248
```

DRAW

```
0244  7201  V2=V2+01
0246  125C  GOTO 025C
0248  3409  SKIP if V4=09
024A  1250  GOTO 0250
024C  7101  V1=V1+01
024E  1244  GOTO 0244
0250  440E  SKIP if V4≠0E
0252  6001  V0=01
0254  440F  SKIP if V4≠0F
0256  6000  V0=00
0258  440A  SKIP if V4≠0A
025A  60FF  V0=FF
025C  D121  SHOW 1 byte at V1, V2
025E  F315  TIMER=V3
0260  F807  V8=TIMER
0262  3800  SKIP if V8=00
0264  1260  GOTO 0260
0266  5F00  SKIP if VF=V0
0268  D121  SHOW 1 byte at V1, V2
026A  1208  GOTO 0208
026C  FF80
026E  0003
0270  01F8
0272  03BB
0274  E2D4
0276  640F  V4=0F
0278  E4A1  SKIP if V4≠KEY
027A  1282  GOTO 0282
027C  74FF  V4=V4+FF
027E  34FF  SKIP if V4=FF
0280  1278  GOTO 0278
0282  00EE  RETURN from SBR
0284  6B00  VB=00
0286  2276  DO SBR at 0276
0288  44FF  SKIP if V4≠FF
028A  1284  GOTO 0284
028C  4BFF  SKIP if VB≠FF
028E  1296  GOTO 0296
0290  4B00  SKIP if VB≠00
0292  6BFF  VB=FF
0294  00EE  RETURN from SBR
0296  6A20  VA=20
0298  6B01  VB=01
029A  FA15  TIMER=VA
29C   F807  V8=TIMER
029E  4800  SKIP if V8≠00
02A0  00EE  RETURN from SBR
02A2  2276  DO SBR at 0276
02A4  44FF  SKIP if V4≠FF
02A6  1284  GOTO 0284
02A8  129C  GOTO 029C
```

LIFE on VIP

Dear VIPER:

Your readers may be interested in a LIFE program I am working on. It started out as a 2K program but has expanded to 4K in my quest for additional features and speed. In its present state it produces about 5 generations per second for a typical population in its 64 x 32 universe. The range is from about 2 generations per second for an almost full universe up to about 25 generations per second for an almost empty one. Since its maximum speed is too fast for many purposes, the actual speed can be set by the player. Another useful feature is the wrap-around option. The player has the choice of selecting either no wrap-around, left to right wrap-around only, top to bottom wrap-around only, or full wrap-around. Patterns can be created and stored, and old patterns can be recalled and modified. Five pages are set aside for pattern storage. The control program is written in CHIP-8 and the Life calculations are done in a machine language subroutine which is almost 1K byte long. This subroutine also keeps track of the number of generations.

I liked the articles in the first few issues of VIPER which fill a real need for us users. I only wish that the quality of your duplicating could match the high standards set by the contents.

Sincerely,
Brian Astle

After receiving and reviewing a copy of Brian's program and his documentation, we decided that LIFE should become the first program in the VIPER Software Library. A tape of the LIFE program and seven pages of documentation, including a complete listing, are now available from The VIPER for $10 postpaid.

We would like to add other programs to the library, with the following ground rules:

1. The program should be of general interest to many VIPER readers.

2. It should be lengthy and complex enough that a reader could not easily create such a program on his own.

3. It must be well documented. We can provide the program listing.

4. If we accept the program for the Library, we will handle printing the documentation and copying the tape. A royalty of 20% of the retail price will be paid quarterly on each copy sold.

5. The author will be required to certify that he is the author of the program and has full right to sell the program.

For those of you who are not familiar with LIFE, here is Brian's Introduction:

The game of Life was invented by John Conway and was described by Martin Gardner in the October 1970 issue of Scientific American. It simulates to some degree the behavior of a living population of cells. The cells occupy a grid of squares so that each square either contains a cell or is empty. In the VIP program a live cell is represented by white and an empty square is represented by black.

The rules which govern the birth, life and death of cells are quite simple:

1. Every cell that touches two or three other cells survives into the next generation, otherwise it dies from either overcrowding or isolation.

2. Every empty square that touches three cells gives birth to a new cell on the next generation.

Note that cells can touch diagonally so that a cell can touch up to eight others.

These simple rules give rise to an endless and fascinating variety of changing patterns which can contain moving entities as well as complex oscillators. The present VIP program runs fast enough to give a visually stimulating display.

CORRECTION * CORRECTION * CORRECTION * CORRECTION * CORRECTION

The CHIP-8I code published in issue #3 is INCORRECT! Here is the correct code:

01A4 86 FA 01 3A AC E5 63 D4 E7 45 FA
01AF 01 3A F2 63 D4 3F F2 6B 3F F5 D4

Sorry about that! Thanks to all of you who wrote us about it.

Don't call a machine language subroutine from within a CHIP-8 subroutine. Return links are clobbered!.

Carmelo Cortez (212) 383-3265, wants to start a VIP Users group in the New York City area. Give him a call for details & info.

What Do You Think?

How would you like to be able to purchase an 1802-based microcomputer with the following specifications:

512 bytes of CMOS RAM
2K of ROM
20-key Keypad
1861-type video interface with RF modulator
power supply

all packaged in an attractive cabinet, complete with expansion connector for more memory or peripherals priced at LESS THAN $60. With some work it should be possible to install CHIP-8 and run most VIP applications on the system. Well, such a system is available today at your nearby Radio Shack store. It is called the RCA Studio II video game! The Studio II was originally sold by RCA for over $175, but for a variety of reasons, never caught on in the consumer market. Radio Shack bought the entire inventory, and is selling it for under $60. Included in the price are all cables and three cartridges containing a variety of games. The on-board ROM contains five games and a CHIP-8 - like interpreter. What the Studio II lacks to become a full microcomputer is:

1. A cassette interface for program storage.
2. A ROM or PROM monitor for controlling the machine.
3. Some kind of I/O interface for control applications.

Several VIPER readers have expressed an interest in articles on converting the Studio II into either a general purpose microcomputer or using it in dedicated applications. Since it's architecture is so similar to the VIP, I would be interested in publishing such articles. If you have a Studio II which you would like to convert, drop me a post card at the VIPER. If enough readers respond, we'll do it. If you have already worked on your Studio II or have information which would be useful in a conversion, please send it to the VIPER.

S T O P T H E P R E S S E S

We have just received a comprehensive package of information on converting the Studio II. The conversion includes a PROM monitor similar to the VIP, and requires no modification to the Studio II hardware. We will provide this information in the next few issues of the VIPER.

I suggest that you purchase a Studio II <u>immediately</u> if you are interested in such a project. Radio Shack stores expect to be sold out by Christmas, and no more units will be available.

Rick Simpson

Adding a Joystick to Your VIP

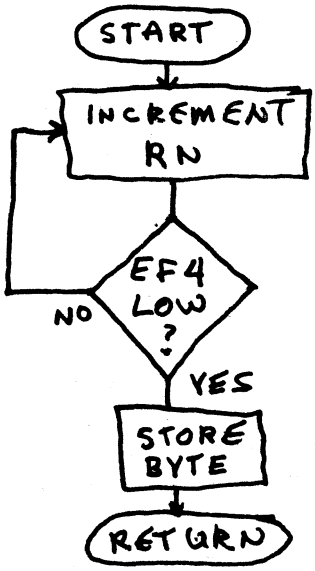
Note: The following application note was generated by John Rudy and Bob Thompson of the VIP Engineering Staff at RCA. Bob picked up the joystick at a computer show and added a switch to the top of the joystick (referred to in the schematic as the 'FIRE button'). You should note that the circuit is intended to interface to the external keypad connector on either the Color board or the Auxiliary Keypad Interface board. Both boards should be available from RCA by the time you read this. See #3 VIPER for ordering information.

VIP JOYSTICK

by
Bob Thompson and John Rudy

The A to D conversion for the joystick consists of triggering a monostable multivibrator with an output byte and incrementing a previously cleared register until the circuit times out and pulls EF4 low. The program loop monitors the EF4 line and stops incrementing when EF4 goes low. The count thus produced is directly proportional to the RC product and hence proportional to the position of the joystick.

The program stores the byte at a CHIP-8 variable address to be used as a display coordinate at a later time. A 62 output of 01 will trigger the horizontal circuit while the 62 output of 02 triggers the vertical. These bytes are latched by the keyboard interface and keep the respective circuits tied to the EF4 line via the transmission gates. An output of 04_{hex} will enable the "FIRE" button.



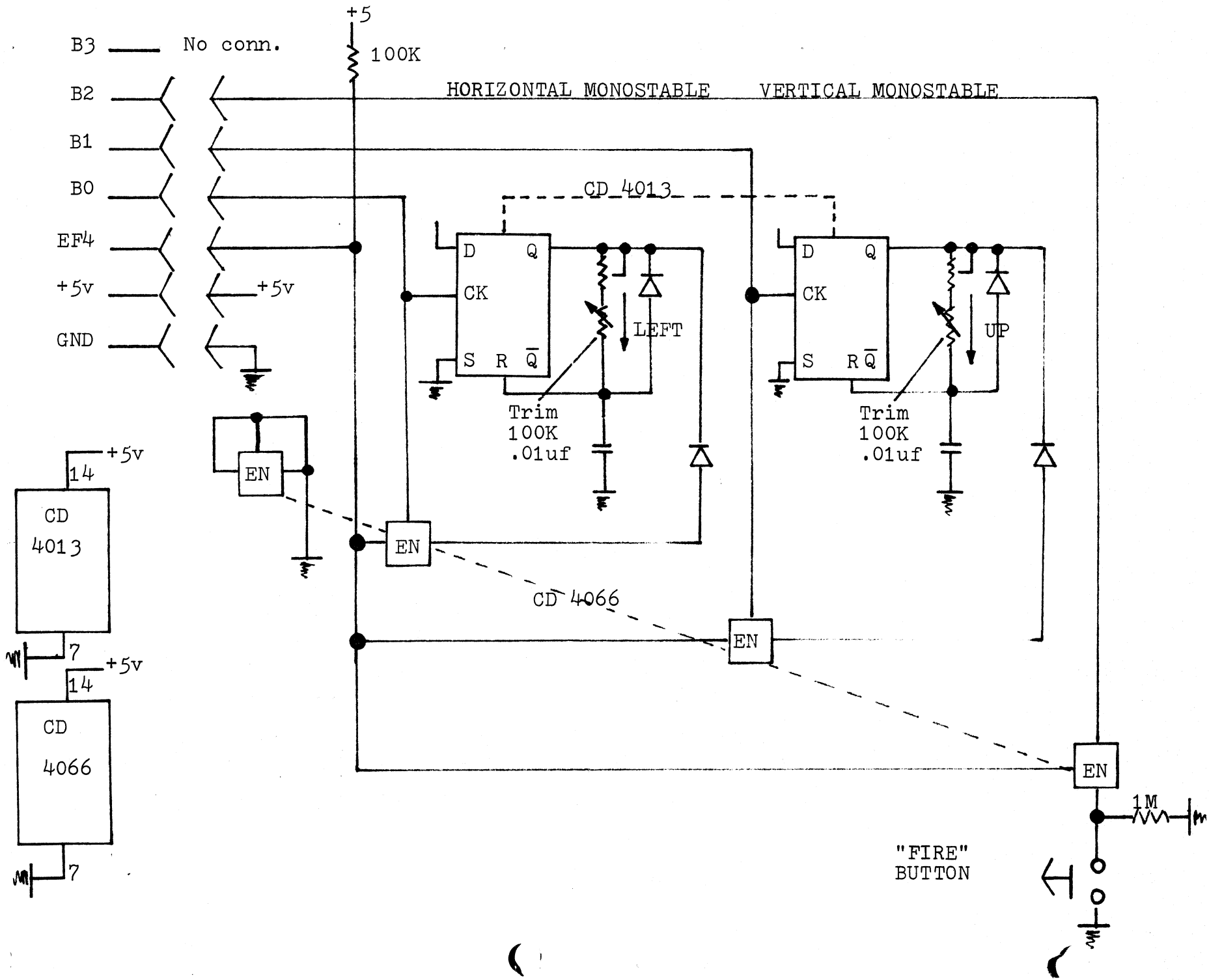
The onboard crystal of 3.52128 Mhz. is divided by 2 to provide the clock for the 1802. This yields a fetch plus execute time of 9.09 μS. per instruction. To completely increment RN to FF would require

approximately 256 passes thru loop X 2 instructions in loop X 9.09 uS/inst =4.65 mS. Since the VIP bit map is only 64 bits wide, RN need only be incremented to a maximum of 64 yielding the relation .7RC=4.65 mS/4 or 1.16 mS. A 100K joystick would be compatible with .015 uF capacitor for a time of 1.05 mS but with component tolerances taken into account you might find your pattern disappears off the right side of the screen and reappears on the left. A capacitor of .01 uF with a trim pot in series within the joystick allows for centering the range and for more exacting graphics. The low end error can be subtracted out in software. Since the vertical resolution in the bit map is only 32 a capacitor of .005 uF should be suitable in the vertical monostable circuit. The diodes between the monostables and the transmission gates provide protection should the programmer out 03_{hex} where the first circuit to time out will oppose the other.

The user is cautioned not to use EF3, which will conflict with the keyboard.

DEMO PROGRAM

```
0200  00E0  Erase screen
   2  0206  Do mach. sub @ 0206
   4  1238      Waits for TV interrupt to finish
   6  0096  Idle, 6Hl R6; Page # of CHIP-8 variables
   8  BCBD
   A  BEF8      Initialize    RC = OXFl CHIP-8 Vl
                              RD = OXF2  "   " V2
   C  FlAC                    RE = OXF3 Output Pointer
   E  ADAE
  10  1D1E
  12  1EF8  Initialize RF.0 to 00 for A to D conversion
  14  00AF
  16  SEEE  Store 00 @ M(R(E)) for
  18  622E  Output, clears keyboard latch
  1A  F801  Store 01 @ M(R(E)) & output
  1C  SE62  Triggers  x  monostable
  1E  2E1F  Loop, increment RF (A to D
  20  3F1F  counter), wait for EF4
  22  8F5C  Store x coordinate as CHIP-8 - Vl
  24  F800  Initialize A to D counter for
  26  AF5E  Another conversion and clear
  28  622E  keyboard latch
  2A  F802  Store 02 @ M(R(E)) & output
  2C  5E62  triggers y monostable
  2E  2E1F  Loop, increment RF (A to D
  30  3F2F  counter), wait for EF4
  32  8F5D  Store y coordinate as CHIP-8-V2
  34  D4XX  Return from machine code
  36  XXXX
  38  A240  Set I
  3A  D125  Display
  3C  D125  Erase
  3E  1200  BR to beginning
  40  2020
  42  F820       Bit pattern
  44  20XX         for display
```


B3
No conn.
B2
B1
B0
EF4
+5v
GND
+5
100K
HORIZONTAL MONOSTABLE
VERTICAL MONOSTABLE
CD 4013
D
Q
CK
S
R
Q
LEFT
UP
Trim
100K
.01uf
+5v
EN
CD 4066
14
CD
4013
7
4066
1M
"FIRE"
BUTTON

The Breakpoint and Register Display Program Revisited

We extend our apologies to the many VIPER readers who wrote or called us to complain or inquire about the register display program by William Barrett in VIPER #4. A combination of poor original copy and a very bad job by our printer (who also put on the wrong page for the back cover) led many subscribers to get nearly illegible copy. As you have probably noticed throughout this issue, we now have a printer on our VIP, and will be able to provide more legible listings from now on. All authors are asked to submit a tape of any program they send in for consideration, so that we can use the printer to provide all listings.

If anyone got a copy of The VIPER where the text of the article was unusable, drop up a post card and we will send you a better copy. The following is a listing of Mr. Barrett's code for a 2K system. With this as a guide, you should be able to make corrections for the 4K version as well.

```
0500 90 B3 F8 08 A3 E0 71 33
0508 91 BB F8 68 A1 F8 05 B1
0510 F8 FF A2 F8 04 B2 69 00
0518 3C 18 34 1A 70 23 C0 00
0520 03 E3 71 23 F8 2B A1 F8
0528 05 B1 D1 E2 60 F8 05 B3
0530 F8 38 A3 F8 AF 53 FE 72
0538 00 03 FB B4 32 4A 03 3B
0540 46 FC 10 53 30 37 FF 11
0548 30 43 72 A3 72 B3 60 60
0550 60 3C 51 34 53 69 3C 56
0558 34 58 60 72 A0 72 B0 60
0560 72 F6 7A 32 66 7B 72 70
0568 22 61 22 78 22 E2 73 F8
0570 00 39 75 F8 08 7E 73 73
0578 90 73 80 73 F8 D1 73 82
0580 FC 07 A0 92 7C 00 B0 40
0588 FA 0F 32 8A F9 40 73 FF
0590 20 52 D2 22 32 9D F8 0C
0598 52 62 22 3E 55 91 73 F8
05A0 68 73 82 FC 09 A0 92 7C
05A8 00 73 80 73 93 73 83 73
05B0 91 B3 7E F8 B9 A3 F8 94
05B8 53 00 73 03 33 C7 FC F0
05C0 53 FB A0 32 CB 30 B9 FF
05C8 EF 30 C0 A8 A3 F8 06 B3
05D0 D3 D3 9B BD F8 00 AA F8
05D8 FF AD ED 8A 73 8D 3A DB
05E0 5D E7 F8 C6 A5 F0 F6 F6
05E8 F6 F6 D5 F0 27 FA 0F D5
05F0 2E 8E 32 D1 8D FA 07 3A
05F8 E5 8D FC 28 AD 30 E5 00

0600 F8 06 B1 BC F8 C6 A1 F8
0608 07 BB F8 D2 A4 F8 05 B4
0610 F8 B8 AC F8 95 A6 F8 81
0618 B6 B5 BA 69 E3 70 23 82
0620 FC 24 A7 92 7C 00 B7 E2
0628 F8 24 A9 89 FF 14 3B 34
0630 F8 14 30 35 89 AE D4 D6
0638 FB 01 3A 44 89 FF 14 A9
0640 3B 1F 30 2B 02 32 77 FF
0648 0B 32 74 FF FF 32 77 FF
0650 05 3A 37 82 FC 24 A7 92
0658 7C 00 B7 07 FA 0F FE FC
0660 05 52 87 E2 F7 A7 97 7F
0668 00 B7 E7 72 AC F0 BC 2C
0670 9C 73 8C 73 C0 05 21 E2
0678 82 A7 92 B7 22 DC DC DC
0680 B9 DC DC DC A9 07 27 32
0688 A0 E2 09 73 F8 03 AE D4
0690 D6 FB 01 32 B1 12 19 17
0698 17 89 57 17 99 57 30 89
06A0 DC DC DC 59 F8 03 AE D4
06A8 12 12 12 12 D6 32 77 30
06B0 1F 12 12 12 12 30 1F D3
06B8 D6 FE FE FE FE AE D6 8E
06C0 F1 73 30 B7 42 70 22 78
06C8 22 52 C4 C4 C4 9B B0 F8
06D0 00 A0 80 E2 E2 20 A0 E2
06D8 20 A0 E2 20 A0 3C D2 7A
06E0 88 32 C4 7B 28 30 C4
```

DIRECT SWTP PR-40 PRINTER INTERFACE TO RCA'S COSMAC VIP

By Joe Weisbecker

The internal PR-40A changes shown in the diagram must be made for direct interfacing to the VIP 8-bit output port. Break the three connections shown, then add wiring for the gate (IC 16) and the buffer (IC 3). These are unused PR-40 gates. Now add the 2.7K resistor.

NOTES:

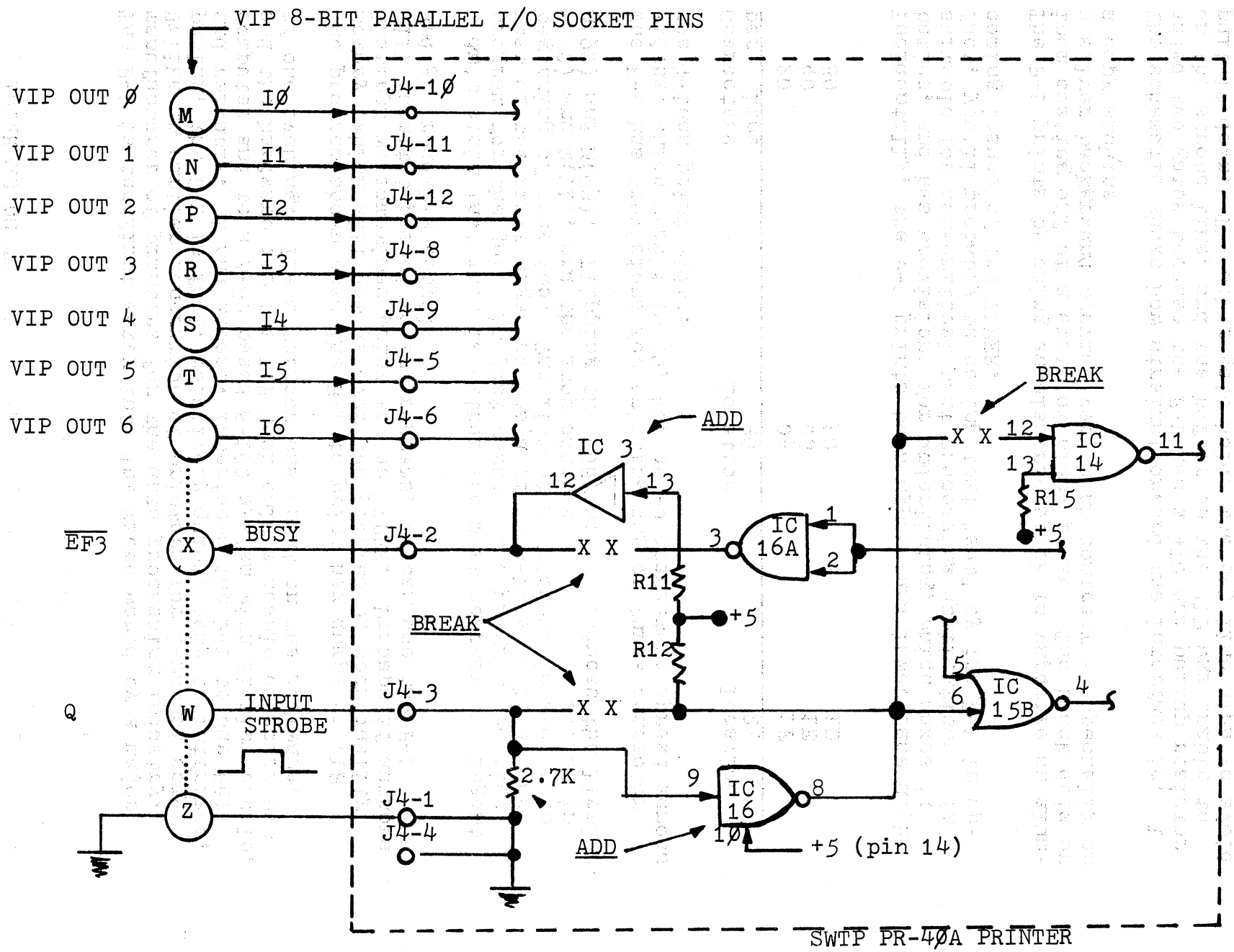
* When EF3=1, the printer is busy and can't accept data.
* Use an 1802 "63" instruction to set an ASCII byte in the output port.
* Set and reset Q to transfer output port byte to printer.
* Always leave ØØ in the output port when not printing. The printer will ignore this ØØ byte so that Q can be used for other purposes (keyboard tone, etc.).
* The line is printed when 4Ø characters or a carriage return is sent to the printer.
* Continuous printing can overheat the printer.

RCA COSMAC VIP TEST PROGRAM FOR THE PR-40 PRINTER

By Joe Weisbecker

This program continuously prints 40 character lines of a single character. The byte at M(ØØØ3) is the ASCII code for the character printed (ASCII 38= "8"). Do not run this program for more than a minute at a time, to avoid overheating the printer solenoids.

```
0000 3600    Wait for EF3=0 (printer not busy)
0002 6338    Set output port byte
0004 7B7A    Set and reset Q (printer strobe)
0006 3000    Repeat
```


VIP 8-BIT PARALLEL I/O SOCKET PINS
VIP OUT Ø
VIP OUT 1
VIP OUT 2
VIP OUT 3
VIP OUT 4
VIP OUT 5
VIP OUT 6
M
N
P
R
S
T
X
W
Z
IØ
I1
I2
I3
I4
I5
I6
J4-1Ø
J4-11
J4-12
J4-8
J4-9
J4-5
J4-6
J4-2
J4-3
J4-1
J4-4
EF3
BUSY
Q
INPUT STROBE
ADD
IC 3
12
13
X X
R11
R12
+5
BREAK
2.7K
IC 16A
IC 15B
IC 16
IC 14
R15
9
8
10
11
+5 (pin 14)
SWTP PR-4ØA PRINTER

INTERFACING THE CENTRONICS P-1 PRINTER TO THE VIP

By Rick Simpson

The P-1 is the first entry into the micro world by a major manufacturer of printers for the data processing industry. Unlike an earlier generation of printers for personal computing, the P-1 is fast, relatively quiet, easy to interface, and reliable. I had already interfaced the P-1 to our PET computer and to a Z-80 based S-100 system with little trouble, and the hardware for the VIP took only a couple of hours out of one evening. The afternoon was spent on the software.

The program shown uses the P-1 to print hex dumps of VIP memory (which will get us over the problem of poor quality listings we sometimes print in the VIPER. Perhaps if authors will submit tapes of their programs, we can use the P-1 and get readable copy?).

I don't claim that this is either the most compact or elegant program for the job - it just works. The program occupies most of the first page and has a sixteen-byte table at the beginning of page two. The list start address is stored in M(90) and M(91), and the stop address plus one is stored in M(92) and M(93). Flipping the RUN switch produces the listing.

The P-1 has provisions for printing any of three character widths, which result in 20, 40, or 80 characters spread across the 4-inch print width:

Characters per line	location 36	location 21
20	05	1F
40	08	1E
80	10	1D

Incidentally, the P-1 prints both upper- and lower-case alphbetics, and has provisions for underlining, which makes it suitable for non-critical text processing applications. In fact, you'll probably be seeing text "set" by the P-1 in upcoming issues of the VIPER.

The P-1 prints by using an electric spark to vaporize an aluminum coating off of a strip of black paper. The black dots on a silver background provide very high contrast - especially on a copy machine.

The schematic shows how straightforward the interface is. I added two gates because the P-1 has a 470 ohm pullup resistor on the strobe line, and the buffer on the VIP was unable to pull it down. The other gate holds the EF3 line high to avoid interference with keypad operation.

This program will work unmodified with a PR-40 printer which has been modified as shown in our other printer article for this month.

MAIN PROGRAM

```
0000 30 A0 00 B3 B4 B5 B6 A7
0008 B8 F8 66 A3 F8 6D A4 F8
0010 79 A5 F8 01 B6 F8 90 A8
0018 48 B1 48 A1 48 B2 08 A2
0020 F8 1E D5 F8 0D D5 F8 0A
0028 D5 91 D4 D5 91 D3 D5 81
0030 D4 D5 81 D3 D5 F8 08 A7
0038 F8 20 D5 01 D4 D5 41 D3
0040 D5 27 91 28 F7 60 3A 59
0048 81 F7 3A 59 F8 0D D5 F8
0050 0A D5 00 00 00 00 00 00
0058 00 87 32 23 30 38 00 00
0060 00 00 00 00 00 D0 FA 0F
0068 A6 06 30 65 D0 F6 F6 F6
0070 F6 A6 06 30 6C 00 00 00
0078 D0 36 79 E9 59 63 29 E8
0080 7B C4 C4 7A 30 78 F8 00
```

PATCH

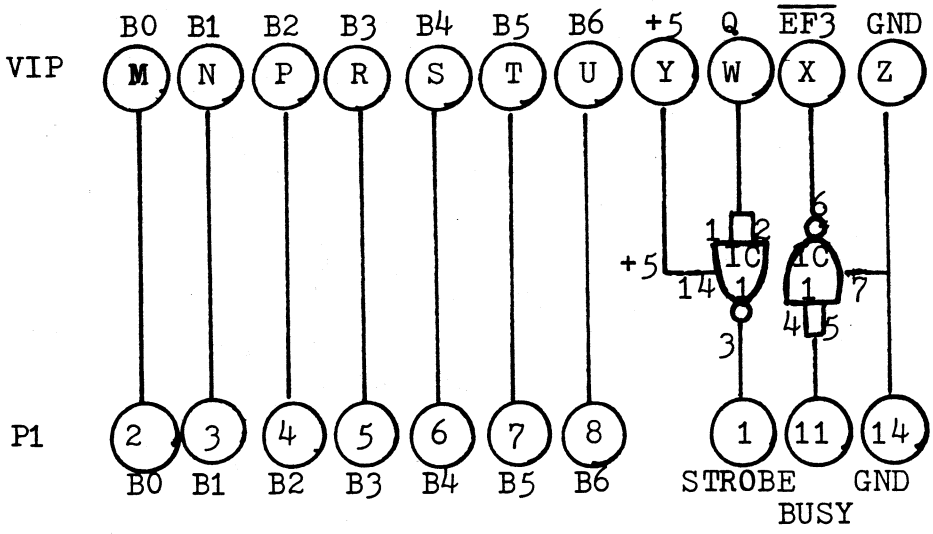
```
00A0 E8 F8 00 B9 F8 B0 A9 F8
00A8 00 30 03 EF F2 9F 2F 26
```

Hex/ASCII Conversion Table

```
0100 30 31 32 33 34 35 36 37
0108 38 39 41 42 43 44 45 46
```

CENTRONICS P-1 MICROPRINTER/RCA VIP INTERFACE

By Rick Simpson



IC 1 = 74LS00

64-BYTE CHECK PROGRAM FOR COSMAC VIP PROGRAMS

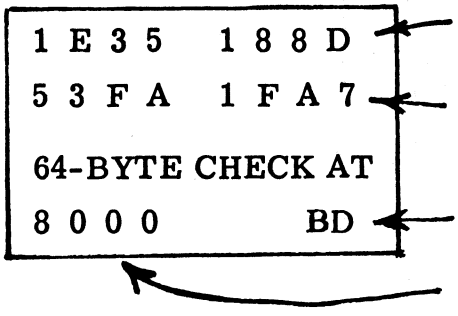
Programmer: John W. Wentworth

This program permits rapid detection and location of key-stroke errors in entering any pre-determined VIP program for which check data can be supplied to the user in advance. For greatest convenience, the program to be entered and checked should be presented to the user in the form of "blocks" of 64 bytes, arranged as 8 rows of 8 bytes each.

The 64-Byte Check Program may be loaded into any memory page (designated here as 0V) that is not occupied by the program under test and that is at least one page below the highest page of installed RAM in the VIP system; page 0V+1 is used as the display page for this program.

When initially loaded for purposes of making a cassette recording, the program may commence at Memory Location 0000--the resulting one-page recording may then be read into any other desired page. For checking VIP programs occupying no more than 5 memory pages, the use of page 05 is recommended for this Check Program; the check program can often be left in memory even when the main VIP program is running. (The Check Program will run well in page 06, but potentially confusing digits will then be seen at the top of the Operating System's display page; also, the operation of any CHIP 8 programs will "clobber" the Check Byte program if it is read into page 06, requiring re-entry from a cassette if any subsequent checking is needed.)

To use this 64-Byte Check Program, the first three bytes in the program under examination must first be changed temporarily to C0-0V-00 (where 0V is the variable page number selected for the Check Program). Upon going from RESET to RUN, the screen should display the words "64 BYTE CHECK AT". The start address for the 64-byte block of data to be checked should then be entered. The Check Program itself may be tested by entering the address 8000 to check the known contents of the Operating System at ROM locations 8000 through 803F. If the check program has been entered without errors, the screen display should appear as follows:



8 check digits corresponding to rows in the 64-byte data matrix

8 check digits corresponding to columns in the 64-byte data matrix

Overall check sum for the data matrix

Start address as entered by the user (for verification)

To check any other 64-byte section of the memory, it is necessary only to key in a new start address. (This feature makes it possible to use the program for a rapid check after a complete program has been entered, or to make a rapid examination of a previously-checked program that has become "clobbered" in some way.)

64-Byte Check Program for COSMAC VIP Programs

In general use of this program, failure of the check sum or the ROW and COLUMN check digits to match the values given in the program listing indicates the presence of errors in the program entry. The most probable types of errors will be indicated by both the check sum and the check digits. The check digits enable the error to be located rapidly without having to use the "Read Memory" mode of the Operating System to search through the entire matrix. For example, if the fifth digit in the top row and the second digit in the second row fail to match the given check digits, one may be sure that the error will be found at the intersection of the fifth row and the second column of the data matrix. It is possible, of course, that more than one error may exist in a given matrix. Certain compensating errors (one byte too high in numeric value and other too low by the same amount) may go undetected by the overall check sum, but such compensating errors will generally cause two pairs of row-and-column check digits to depart from the assigned values.

A mis-match between the check sum calculated by the Check Program and that given in the program listing not accompanied by any defective row-and-column check digits indicates the presence of a relatively improbable error involving both hexadecimal digits of a single byte; in this case, one must undertake the relatively laborious byte-by-byte search method, but at least the search is confined to a relatively small field. This program is not 100% foolproof, but the only errors that could go undetected are highly improbable compensating errors involving no less than 4 digits in 2 separate bytes.

The operating principle in this 64-Byte Check Program is really not mysterious. The "overall check sum" is literally the two least significant hexadecimal digits of the sum of all 64 bytes in the matrix; an error in any single digit will alter the numeric value of this sum. The row and column check digits are generated by a five-step process: (a) the 8 values in a given row or column are added together (ignoring carries, so that only a 2-digit sum is retained), (b) the MSD of the sum is shifted four binary places to the ight so that it lines up with the LSD, (c) the MSD is rotated (ring shifted) ne binary place to the left so that its numeric value is changed within the 0-F range, (d) the altered MSD is added to the LSD, and (e) the final result is masked to retain only a single hexadecimal digit. The objective of the ring-shifting operation is to permit row-and-column detection of frequently-encountered transposition errors within a single byte (as, for example, when a 17 was erroneously entered where 71 was intended.)

Listing for 64-Byte Checksum Program

```
0500 F8 81 B1 B5 B7 BC 90 B3
0508 B4 BA FC 01 B2 BB BD F8
0510 46 A1 F8 FF A2 F8 19 A3
0518 D3 F8 B3 A4 F8 BA A7 F8
0520 CF AA F8 95 AC F8 98 AD
0528 E2 F8 00 73 82 FB FF 3A
0530 29 D4 9C B4 BA F8 E1 A4
0538 69 D7 D7 D7 B6 BE D7 D7
0540 D7 A6 F8 40 AF 22 F8 00
0548 52 E6 02 F4 52 16 2F 8F
0550 3A 4A E2 22 9E B6 86 FF
0558 40 A6 D4 F8 40 AD 93 B4
0560 F8 BD A4 E6 9C BE F8 08
0568 BF F8 08 AF F8 00 52 02
0570 F4 52 9E 32 7A 86 FC 08
0578 A6 38 16 2F 8F 3A 6F D4
0580 9E 32 87 86 FF 3F A6 9F
0588 FF 01 BF 3A 69 9E 32 99
0590 86 FF 08 A6 F8 00 AD 30
0598 65 E2 F8 04 AD 0D F6 F6
05A0 F6 F6 5D 1D 8D FA 03 3A
05A8 9D 8D FB 68 32 32 8D FC
05B0 04 30 9C F8 C9 A5 D5 8D
05B8 FB A0 3A B3 D3 02 FA 70
05C0 F6 F6 F6 AF 02 FE 8F E2
05C8 74 E6 FA 0F D5 30 BC EA
05D0 8A EF A2 E2 0F 05 67 05
05D8 0F 57 52 72 22 22 70 40
05E0 60 40 70 75 45 47 45 75
05E8 77 44 64 44 77 48 50 60
05F0 50 58 77 52 72 52 52 00
05F8 01 00 00 00 00 00 00
```

Checksums: 0000=EA 0040=B5 0080=1F 00C0=CF

SELF-CHECKING VERSION OF CHIP-8 INTERPRETER LISTING
(64-BYTE CHECK PROGRAM PREVIOUSLY LOADED INTO MEMORY PAGE 05)

```
0000* C0 05 00 01 B2 B6 F8 CF      0100  00 00 00 00 00 45 A3 98
0008  A2 F8 81 B1 F8 46 A1 90      0108  56 D4 F8 81 BC F8 95 AC
0010  B4 F8 1B A4 F8 01 B5 F8      0110  22 DC 12 56 D4 06 B8 D4
0018  FC A5 D4 96 B7 E2 94 BC      0118  06 A8 D4 64 0A 01 E6 8A
0020  45 AF F6 F6 F6 F6 32 44      0120  F4 AA 3B 28 9A FC 01 BA
0028  F9 50 AC 8F FA 0F F9 F0      0128  D4 F8 81 BA 06 FA 0F AA
0030  A6 05 F6 F6 F6 F6 F9 F0      0130  0A AA D4 E6 06 BF 93 BE
0038  A7 4C B3 8C FC 0F AC 0C      0138  F8 1B AE 2A 1A F8 00 5A

64-BYTE CHECK   ROW 4133 A484     64-BYTE CHECK   ROW 1B5D C951
AT 0000  4E     COL 0722 169B     AT 0100  32     COL 06E1 4070

0040  A3 D3 30 1B 8F FA 0F B3      0140  0E F5 3B 4B 56 0A FC 01
0048  45 30 40 22 69 12 D4 00      0148  5A 30 40 4E F6 3B 3C 9F
0050  00 01 01 01 01 01 01 01      0150  56 2A 2A D4 00 22 86 52
0058  01 01 01 01 01 00 01 01      0158  F8 F0 A7 07 5A 87 F3 17
0060  00 7C 75 83 8B 95 B4 B7      0160  1A 3A 5B 12 D4 22 86 52
0068  BC 91 EB A4 D9 70 99 05      0168  F8 F0 A7 0A 57 87 F3 17
0070  06 FA 07 BE 06 FA 3F F6      0170  1A 3A 6B 12 D4 15 85 22
0078  F6 F6 22 52 07 FA 1F FE      0178  73 95 52 25 45 A5 86 FA

64-BYTE CHECK   ROW CA77 EC9C     64-BYTE CHECK   ROW 3862 02D6
AT 0040  7A     COL 62A4 7631     AT 0140  5D     COL FEB0 7BBF

0080  FE FE F1 AC 9B BC 45 FA      0180  0F B5 D4 45 E6 F3 3A 82
0088  0F AD A7 F8 D0 A6 93 AF      0188  15 15 D4 45 E6 F3 3A 88
0090  87 32 F3 27 4A BD 9E AE      0190  D4 45 07 30 8C 45 07 30
0098  8E 32 A4 9D F6 BD 8F 76      0198  84 E6 62 26 45 A3 36 88
00A0  AF 2E 30 98 9D 56 16 8F      01A0  D4 3E 88 D4 F8 F0 A7 E7
00A8  56 16 30 8E 00 EC F8 D0      01A8  45 F4 A5 86 FA 0F 3B B2
00B0  A6 93 A7 8D 32 D9 06 F2      01B0  FC 01 B5 D4 45 56 D4 45
00B8  2D 32 BE F8 01 A7 46 F3      01B8  E6 F4 56 D4 45 FA 0F 3A

64-BYTE CHECK   ROW 35A0 39E5     64-BYTE CHECK   ROW 092B 140D
AT 0080  A2     COL 9A35 9193     AT 0180  44     COL 5E1F BF45

00C0  5C 02 FB 07 32 D2 1C 06      01C0  C4 07 56 D4 AF 22 F8 D3
00C8  F2 32 CE F8 01 A7 06 F3      01C8  73 8F F9 F0 52 E6 07 D2
00D0  5C 2C 16 8C FC 08 AC 3B      01D0  56 F8 FF A6 F8 00 7E 56
00D8  B3 F8 FF A6 87 56 12 D4      01D8  D4 19 89 AE 93 BE 99 EE
00E0  9B BF F8 FF AF 93 5F 8F      01E0  F4 56 76 E6 F4 B9 56 45
00E8  32 DF 2F 30 E5 00 42 B5      01E8  F2 56 D4 45 AA 86 FA 0F
00F0  42 A5 D4 8D A7 87 32 AC      01F0  BA D4 00 00 00 00 00 00
00F8  2A 27 30 F5 00 00 00 00      01F8  00 00 00 00 00 E0 00 4B

64-BYTE CHECK   ROW 7C75 24E4     64-BYTE CHECK   ROW 4B6B BDFF
AT 00C0  D0     COL 9B9F 00A7     AT 01C0  89     COL 1B5B E229
```

* After Checking all entries, change first 3 bytes commencing at 0000 To 91-BB-FF. CHIP 8 is then ready for use.

VIP REGISTER DISPLAY PROGRAM

Program Objective: To display the contents of COSMAC registers R3 through RF in a single screen display as an aid in de-bugging machine-language VIP programs. (R0, R1 and R2 cannot be examined because their values are altered by the routine which stores the register contents.)

Initial Entry and Recording

(1) Using the TAPE READ routine of the Operating System, load the 64-Byte Check Program commencing at M. L. 0000.

(2) Using the MEMORY WRITE routine commencing at M. L. 0600, enter the Vip Register Display Program in accordance with the following program listing, using the 64-Byte Check Program to check each "block" of 64 bytes.

(3) Using the TAPE WRITE routine commencing at M. L. 0600, make a single-page cassette recording of the VIP Register Display Program. (This program may then be re-entered into any memory page other than the Operating System's display page not occupied by a program under test.)

```
0600  35 19 29 27 37 2B 2F 39
0608  31 25 33 10 21 15 1D 1F
0610  C0 A0 E0 A0 C0 C0 A0 A0
0618  A0 C0 40 40 40 E0 80 E0
0620  80 E0 80 80 80 E0 A0 E0
0628  20 E0 20 E0 80 E0 20 E0
0630  80 E0 A0 E0 A0 E0 A0 A0
0638  A0 E0 20 20 20 20 91 BB

64-BYTE CHECK    ROW 4B5C 8C54
AT 0600  FF      COL 7075 A572

0640  90 B2 B3 B4 B5 B6 B7 BA
0648  F8 81 B1 F8 46 A1 F8 FF
0650  A2 F8 55 A3 D3 F8 91 A4
0658  F8 D3 A5 A6 F8 E3 A7 F8
0660  B3 AB 4B 55 15 85 FB F0
0668  3A 62 82 AB EB F8 00 73
0670  8B 3A 6D 47 D4 8B FC D9
0678  AB 46 D4 8B FF E7 33 86

64-BYTE CHECK    ROW 6053 412C
AT 0640  E5      COL DC80 C539
```

```
0680  8B FC 07 AB 30 73 FB 08
0688  C6 69 23 8B FC 4A 30 83
0690  D3 B8 F6 F6 F6 F6 AA 0A
0698  AA 4A 55 15 85 FB F5 3A
06A0  99 55 F8 F0 A5 E5 98 FA
06A8  0F AA 0A AA 4A F6 F6 F6
06B0  F6 F1 5B 8B FC 08 AB 15
06B8  05 3A AC F8 F0 A5 30 90

64-BYTE CHECK    ROW A19D 1C4E
AT 0680  2D      COL F4C8 3C90
```

Instructions for Use

(1) It is assumed that a machine-language program to be de-bugged has already been loaded into the VIP, commencing at M. L. 0000 and occupying one or more memory pages. Use the TAPE READ routine to enter this Register Display Program into any unused page other than the one used by the VIP Operating System as its display page (Page 07 in the basic VIP system with 2K memory). In most instances, the use of page 06 is appropriate for this Register Display Program.

(2) Insert a temporary "stop" instruction in the program under test at the point desired for analysis. If the register serving as the program counter is known at this point, a single "Decrement PC" instruction may be used as the stop. (For example, if R3 is known to be the PC, then "23" is appropriate.) If there is uncertainty about the program counter (as, for example, when improper assignment of "P" is the very "bug" that needs detection), a temporary stop instruction can always take the form of a "30-MM" sequence, where MM is the address of the same cell where the "30" is located. A special stop instruction may not be needed if the fault under investigation happens to be one that simply causes the program to "hang up" at some unknown point--examination of the registers at this point can usually provide clues as to the reason for the hang up.

(3) Run the program under test to complete execution up to the point of the temporary stop or the unknown "hang up".

(4) Call the Operating System, enter the address 0000, select the MEMORY WRITE routine, and temporarily change the first three bytes of the program under test to C0-0V-3E (where 0V is the variable page number into which the VIP Register Display Program has been entered). The Operating System has automatically stored the register contents in locations B3 through CF in its display page, from whence they will be retrieved by this display program.

(5) Go to RESET, then RUN. Register contents will now be displayed on the screen in the following format:

R3		
R4	R8	RC
R5	R9	RD
R6	RA	RE
R7	RB	RF

(6) Before the program under test can be run again, it will be necessary to restore the first three bytes of the program and the bytes that may have been altered to introduce the "stop" condition. (Going from RESET to RUN a second time without restoring the original program will cause the display of "garbage" on the screen.)

ARESCO
P.O. Box 43
Audubon, PA 19407

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.

Name _____
(Please print or type)

Address _____

City _____ State _____ Zip _____

Cash, Check, Money Order Enclosed _____

MC/VISA/BAC Number _____ Exp. Date _____

MC Interbank No. _____

Required Credit Card Signature _____

☐ You may let other VIP owners in my area know I have a VIP, so they can contact me.

☐ I am interested in forming/joining (circle one) a VIP Users group

☐ I'd like to see articles in the VIPER about:

_____

_____

_____

_____

MAIL TO: VIPER; P.O. Box 43; Audubon, PA; 19407