VIPER

VOLUME 2, ISSUE 3 AN ARESCO PUBLICATION SEPTEMBER 1979, $2.00

TABLE OF CONTENTS

** A T T E N T I O N **

We have cancelled ARESCO's order for the VIP II, since production will be delayed for an indefinite time while VIP engineers work on another project higher in the company's priority list. As soon as we find out when the production models are available, we'll notify you. PLEASE DO NOT ORDER VIP II YET!!! Thanks, all - Terry

SUBSCRIPTION RATES, ADVERTISING RATES, & OTHER INFORMATION

THE VIPER IS PUBLISHED TEN TIMES PER YEAR BY ARESCO INC., AT 6303 GOLDEN HOOK, COLUMBIA MD 21044. MAILED TO SUBSCRIBERS ON THE 15TH DAY OF EACH MONTH EXCEPT JUNE AND DECEMBER. SINGLE COPY PRICE IS $2 AND SUBSCRIPTION PRICE IS $15 FOR ALL 10 ISSUES OF THE CURRENT VOLUME. SUBSCRIPTIONS DO NOT CARRY OVER FROM ONE VOLUME TO THE NEXT. SUBSCRIBERS WHO WISH TO ORDER LESS THAN THE FULL VOLUME SHOULD SEND $2 FOR EACH ISSUE DESIRED. RENEWALS ARE ACCEPTED DURING THE LAST TWO MONTHS OF THE CURRENT VOLUME YEAR, AND THE FIRST ISSUE OF EACH VOLUME IS PUBLISHED IN JULY. ENTIRE CONTENTS OF THE VIPER COPYRIGHT 1979 BY ARESCO INC.
APPLICATION TO MAIL AT SECOND CLASS POSTAGE RATES IS PENDING IN COLUMBIA MD 21045. POSTMASTER: SEND ALL ADDRESS CHANGES TO THE VIPER, BOX 1142, COLUMBIA MD 21044.
THE VIPER IS AN ARESCO INC. PUBLICATION. EDITOR: TERRY LAUDEREAU, CO-EDITOR: RICK SIMPSON. CORRESPONDING EDITOR IS TOM SWAN.
READERS ARE ENCOURAGED TO SUBMIT ARTICLES OF GENERAL INTEREST TO VIP OWNERS FOR PUBLICATION IN THE VIPER. MATERIAL SUBMITTED WILL BE CONSIDERED FREE OF COPYRIGHT RESTRICTIONS. THOSE SUBMISSIONS RECEIVED BY THE 1ST DAY OF A MONTH WILL BE CONSIDERED FOR INCLUSION IN THAT MONTH'S ISSUE.

SUBSCRIPTION RATES:
USA RESIDENTS: $15/10 ISSUES. NON USA RESIDENTS SHOULD INCLUDE AN ADDITIONAL $10 FOR AIR MAIL POSTAGE. IF SUCH POSTAGE IS NOT INCLUDED WITH THE ORDER, THE NEWSLETTER WILL BE SENT SECOND CLASS WITH THE REMAINDER OF THE MAILING. PURCHASE ORDERS WILL NOT BE ACCEPTED. IF SUBSCRIBER SPECIFIES UPS COD THE FIRST ISSUE WILL BE SENT COD WITH THE COLLECTIBLE AMOUNT TO BE $15 PLUS A $1 COD CHARGE PLUS SHIPPING COSTS. PERSONAL CHECKS, MASTER CHARGE, AND VISA ORDERS ARE ACCEPTED. CHECKS DRAWN ON FOREIGN BANKS SHOULD BE PAYABLE IN U.S. DOLLARS.

ADVERTISING RATES:
CLASSIFIED ADVERTISING: $10/3 LINES OF 60 CHARACTERS EACH.

PAGE RATES:

FULL PAGE	$85
HALF PAGE	$45
QUARTER PAGE	$25

PAYMENT MUST ACCOMPANY ADS. ADS SHOULD BE CAMERA READY POSITIVE ARTWORK. IF OUR PRINTER CHARGES US EXTRA FOR PREPARING YOUR AD COPY, WE WILL BILL YOU AT COST + 10%.

DEALERS ARE INVITED TO REQUEST THE ARESCO INC. DEALER INFORMATION PACKAGE BY WRITING IN CARE OF DEALER SERVICES.

Terry- I just got back from the 3rd Annual National Small Computer Show in New York and I thought I'd report on the RCA exhibit.

The big news is the introduction of the VIP II. While not scheduled for general marketting until "early next year" there were two units available for demo at the show. Imagine a VIP with 8K RAM, ASCII/numeric keyboard, color board, and simple sound board. Then add a full featured floating point BASIC and you have a VIP II.

The most impressive news though, is that RCA is expecting to market the VIP II for "about $400.00". As the breakdown below shows, this is a big price-performance increase over the VIP.

TYPE	DESCRIPTION	PRICE (8/6/79)
VP-711	VIP	249
VP-44	On board expansion (adds 2K RAM)	36
VP-570	Memory expansion (adds 4K RAM)	95
VP-611	ASCII/numeric keyboard	80
VP-620	Flat ribbon cable for VP611	20
VP-590	Color board	69
VP-595	Simple sound board	30
VP-575	Expansion board (to allow simultaneous use of VP-611, VP-590, and VP-595)	59
	TOTAL	$638.00

Note that this $638 price does not include the floating point BASIC, as it is just not available for the VIP! This BASIC contains 8 commands (including SAVE and LOAD), 50 statements, and 14 math functions. The 50 statements include two dimensional arrays and several "special" functions to ease the use of the sound and color features. All this plus the monitor are in 12K bytes of ROM.

If this weren't enough, there is also an onboard 9 pin "D" connector to ease interfacing joysticks and auxiliary keyboards.

The VIP II still supports both CHIP-8 and machine language. In fact, the new BASIC allows calls to machine language subroutines.

One of the RCA marketting representatives I spoke to at the show suggested that the floating point BASIC would probably be patched for use on existing VIPs shortly after introduction of the VIP II. He also felt that the price of the existing VIP machine would be dropped a bit to allow it to remain a competitive product. (The $150 difference between a VIP and VIP II could ruin the VIP market.)

Other news I picked up at the RCA exhibit is that your VIP hardware prices are not the latest. RCA published a new price list effective August 6, 1979. The major jump is in the ASCII keyboard.

I also spent some time playing with tiny BASIC (VP-700, still $39) and have a few reactions. First, it's a bit more versatile than earlier literature suggested in that it supports the basic 4 math functions, parentheses, and the greater/less than comparisons. In addition, there are 5 different diagnostic messages to help in debugging tine BASIC programs.

My two negative reactions to tiny BASIC are its speed (it was much slower in running a program I've used on TRS 80's, APPLE II's, and PETs), and its lack of a machine language subrouinte call feature. Another RCA rep told me that patches would be available to support machine language subroutines.

That wraps up my review of the show. Personally, I can't wait for the VIP II to hit the market. With its 8K memory, color, and sound it beats anything else now available on a price-performance ratio. I just hope there is still a market for used VIPs after the VIP II comes out. (If not, how about an article on a multi-processor/parallel processing system.) -Robert Kantor

---

Terry- Here are some things I'd like to see in VIPER:

A) hard information in VIP II or whatever it's going to be called. I saw a note on it in Electronics reporting on one of the electronics shows.

B) details of expansion interface for VIP. Fourty or 50 dollars is a lot of money for a couple of buffers, and I'll be many people would prefer to build their own if the circit details were worked out. (But maybe the RCA expansion interface includes a lot more than just buffers and latches? and so is worth the price.) Incidentally, I think the supersound board is worth the price.

Thanks for implementing the little box in VIPER saying whats available. Could I suggest also that you add a code indicating status: e.g.? (date) for projected availability, ! for "we've got oodles in stock, so please buy now to keep us solvent", # for "not in stock as we write this, but expected by the time you read this". -C.C.

Terry- Has anyone come up with a scheme to single step the VIP? Would appreciate any information on how to do this. It is a very helpful troubleshooting aide.

Volume #2 is a definite improvement. Keep up the good work. --J. L. Howard

I have an ELF II and not a VIP. I have found that with only minor modifications to CHIP-8 and the VIP operating system, I was able to run every program that I entered from the VIP manuals. However, as noted in the Lewis article from Issue 8, the ELF keyboard is static and not dynamic. I assembled a separate keyboard to be driven from the output port of the ELF Giant board. This has provided satisfactory operation in all aspects.

I have found and changed what I believe to be only one other area that requires a kludge. The VIP tape interface inverts the tape input signal in the operational amplifier circuit while the ELF does not. This requires three instructions in the VIP operating system to be changed to correctly test the EF-2 line. Otherwise if the VIP operating system is located at address 8000, there are very few changes required to match the ELF.

I did find it convenient to rewrite CHIP-8 from location 0000 to 0004 and the VIP operating system from location 8000 to 8027. The changes were required because the VIP hardware automatically starts to run at 8000 when the system is reset.

My modification to CHIP-8 puts a long branch to the VIP OS at location 0000-0002; changes the address of the interrupt program to match the available memory; and changes the 62 instruction for CHIP-8 (EX9E,EXAE) to a 67 instruction to use the ELF output port. Most of the register initialization is done in the first 27 hex bytes of the modified operating system.

The RCA tape level indicator circuit was so elegantly simple that I built it into the tape recorder next to the earphone jack. It works great for either the ELF or the VIP system.

Has anyone written a simple program to certify tapes for dropout?

Keep up the VIPER! It is one of the better magazines to which I subscribe. I would be willing to correspond with anyone who also has an ELF-VIP system. You may publish my name and address.

Leo F. Hood, 206 Cecil Ave, West Lawn, PA 19609

Leo: Thanks for taking the time to share your experiences with us. What would really please me is to see all the ELF and Super ELF people get together and agree on a standard set of mods so that the three most popular 1802 systems could all run the same software with the same results. This would mean that the VIPER could be a common reference for all three systems. That would result in more subscribers, more articles, and more software for everyone.

P.S. Has anyone come up with a scheme to run the RCA VIP expansion boards (Super Sound, etc.) on an ELF? If so, I would really appreciate an article on it.

Terry- A few months ago, we subscribed to the VIPER, hoping to find some software we could run on our ELF computer. We feel that this was a very wisely spent $15; thanks to the VIPER, we discovered a wonderful programming language called CHIP-8, which allowed us to run masterpieces such as "Animal Race". We must congratulate you on an excellent publication, and thank you for opening our eyes to the exciting world of CHIP-8.

If the voting is still on, we are both 100% in favor of the ELF II and the Studio II articles in the VIPER. The differences between thses computers and the VIP are, as we have found, very small, and we 1802 users should stick together.

The way we implemented CHIP-8 might be of interest to other readers. The only hardware modifications were the addition of a beeper and a simple circuit connected to the existing keyboard. The 74C922 encoder used for our keyboard has a positive-going data-ready signal at pin 12. This line simply has to be inverted and connected to a flag line (we used $\overline{EF3}$). See circuit below.

We use the top page of RAM for the character tables, the interrupt routine, the CHIP-8 patch (as described by Bobby Lewis) and a cassette routine (we have no operating system in ROM). The two pages below this are for the display and stack area. The CHIP-8 interpreter had to be modified to accommodate our keyboard. These modifications are listed below. The rest of the interpreter remains unchanged.

We plan to build a video board on the 1861 which would feature optional high resolution graphics and color under software control and we would appreciate it if some kind reader would send us a copy of the schematic for the VP-590 Color board. We would also like a schematic of the VIP tape input (we've had trouble with ours).

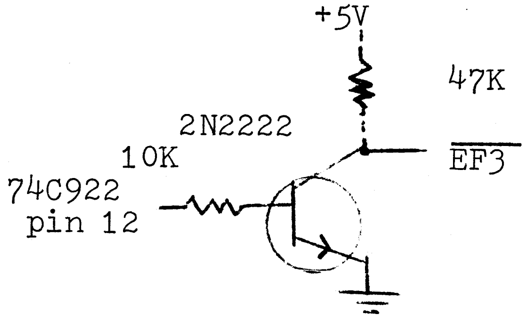
<u>CHIP-8 modifications</u>

```
010A E6       6 → X
  0B 3E 0B    wait for key pressed
  0D 36 0D    and released
  0F 6C       read keyboard latch
  10 FA 0F    save lowest digit
  12 56       store in VX
  13 D4       return
  14 00       filler

0199 45 F6    DF=lowest bit of 9E or A1 (0 or 1)
  9B 3E F6    go to 01F6 if no key pressed
  9D 22       decrement stack
  9E 6C       read key of pressed
  9F 06       VX   D
  A0 F3       XOR with stack
  A1 12       restore stack
  A2 30 F2    go to 01F2

01F2 FA 0F    save lowest 4 bits
  F4 32 F8    go to 01F8 if VX≠key
  F6 F8 FF    FF   D if VX≠key
  F8 7C 00    add DF to D
  FA 3086     go to 0186 to decide whetner to skip or not
```

Keyboard modification



--Gilles and Gilbert Detillieux
Ste-Anne, Manitoba
CANADA ROA 1RO

Terry- I wanted to let you know how satisfied I am with Tom Swan's book PIPs FOR VIPS. So far I've converted four of his most interesting programs over to PIPs for ELF II. The documentation Tom included really helped the conversion process. I've even interfaced my ASC II keyboard into his text editor program with no problem. The only real difficulty I had was with the Space Wars. A little modification to the interrupt routine was required to cure my severe jitter problem. One thing I did do was combine the chip-8 surround program with the two page space wars interpreter. Now I have twice the area in which to move. I'm looking foward to another year of VIPER. Keep the fine programs coming. -Niel Wiengand

Terry- I enjoyed reading Vol. 2 of VIPER, a good thing is getting better. I received PIPS and the documentation is damn good. With all the comments (improvements) to CHIP-8 interpreter, especially in the branching instructions, any possibility of coming out with an updated CHIP-8 including higher resolution? Keep up the good work. -Rick Wiack
P.S. We like the bowling game a lot.

Terry- My brother and I recently purchased "PIPs FOR VIPs" (and I agree it's a pip). Since we both run ELF IIs (mine from Netronics-his a home brew we've built). I've been going through the listings to make those changes necessary to run on our systems.

In so doing I'd like to call your attention to what appears to be a typo on page 49 in the Disassembly 7. Listing on line (MLC) 00BD the comment reads ";R3 = 0BC2 - Read Routine in PROM". I'm fairly sure the 0B should be 80. Only one typo so far - not bad!!

I'd also like to complement you not only on the prepublication offer but also on prepublication delivery! Quite a good show. Keep up the good work.

My brother and I have "VIP"ed our ELVES with Bobby R. Lewis' system and everything is working fine with one exception. The tape Write/Read routines. We can generate tapes but can't load them using the VIP OS. I haven't been able to determine if it's in the record or playback cycle. Do you know if anyone has had similar problems? So far it hasn't been a problem since I can load a VIP Format Tape using the Netronic Monitor and then correct the garbled data with a short program which was published in "IPSO FACTO".

So far I've only had "Surround" running since that was a straight Chip-8 program which did not require changes. It would have helped if the Chip-8 Program Editor had been listed in disassembled and comented form. Guess that'll be my first disassembly once I get Disassembler 7 running.

I've already voted for a continuation of article for other 1802 systems besides the VIP by subscribing for volume 2 but as a member of ACE "IPSO FACTO" does keep me supplied so I'm mostly interested in VIP articles. In any event keep up the good work. -Dave Hersker

CORRECTIONS

Rather than put in an entire page for this one correction, we decided to wse it as "filler" here. However, there is still a CORRECTIONS title in the index to help you find it in the future.

We inadvertently omitted a byte of data from Wayne Smith's listing in his article "Relative Branching in CHIP-8", which begins on page 2.01.11. The byte should be added at 01B3 (close to middle of page 2.01.12). The data at 01B3 should be F9.

LITTLE LOOPS

by Tom Swan

Perhaps the most unused instruction in the Chip-8 repertoire is the BMMM; go to OMMM plus the value in VO. The power of this instruction is being underestimated if its frequency of use is any indication, and I seriously hope that its unpopularity is not a result of its rather unfortunate name. Still, this month's column will focus on the BMMM, with the intention of spreading only knowledge, though I must assume full responsibility for the obvious risk I am taking of being accused of flinging something much more serious.

Why add something to a branch (go-to) address when it seems just as easy to simply branch to the right address in the first place? Good question and the answer is you are probably right for most simple game programs.

However, suppose you have eight or so routines of varying lengths each of which is to run based on a condition such as a key press or the result of a test for equality, etc. One way to handle the problem is detailed in flowchart 1. This takes only twice as many instructions as the tests to be conducted and fits well with the Chip-8 way of doing things as shown in listing 1.

But if the tests are not conducted in a neat row such as this, when it comes time to run the program which procedes to bullet from the sky into the side of the mountain with a firey crash that could very well re-open the San Andreas fault, you've a complicated debugging session ahead in order to untangle the wreckage.

The BMMM instruction avoids such a complication by allowing the use of a jump table to handle the flow of the program. Jump tables will usually be found in machine language programs where complexity is sure to gum up the works. Chip-8 give you the same capability. All the starting addresses are kept in a table for each of the routines. Then routines may be added, moved around or taken out by only manipulating the jump table, not the test which calls that particular routine.

Table 1 gives an example of a jump table which contains the low eight bits of the go-to addresses in flowchart 1 and listing 1. OXXX is the address of the table inself.

The calling section of the program is demonstrated in listing 2. The I pointer is first set to the address of the jump table at OXXX. The program waits for a key to be pressed and adds the value of that key to the I pointer with the F01E instruction. Please be sure you understand this before proceeding -- the I pointer will point downwards into the jump table at a point relative to the value of the key pressed.

At this point lies the address of the function to be selected by that particular key. This address is loaded into VO with the F065 instruction (listing 2) and the BMMM instruction selects the function. (BMMM should be in the form BMOO where M equals the page on which the functions reside. However BMOO looks even funnier in print so I hesitated using it.) As the lower part of the BMMM is "00", when the value of V0 is added to it, a jump to the correct function will be performed.

The beauty of this method is that only the jump table needs to be manipulated to change, for instance, Key C's function to do what Key 0 used to do. In order to disable any function, simply use a default entry which causes a jump to an instruction whose only purpose is to return control to the main program. (The "ZZ" entry in table 1) Never does the calling section need to be disturbed, and for that reason the program may be debugged by only enabling one function at a time. If the program fails (provided all else is correct) the bug has to exist with that function. The program itself runs independently of the jump table (as long as you were careful to program it that way, or have inserted "dummy" modules to take the place of those not yet written.)

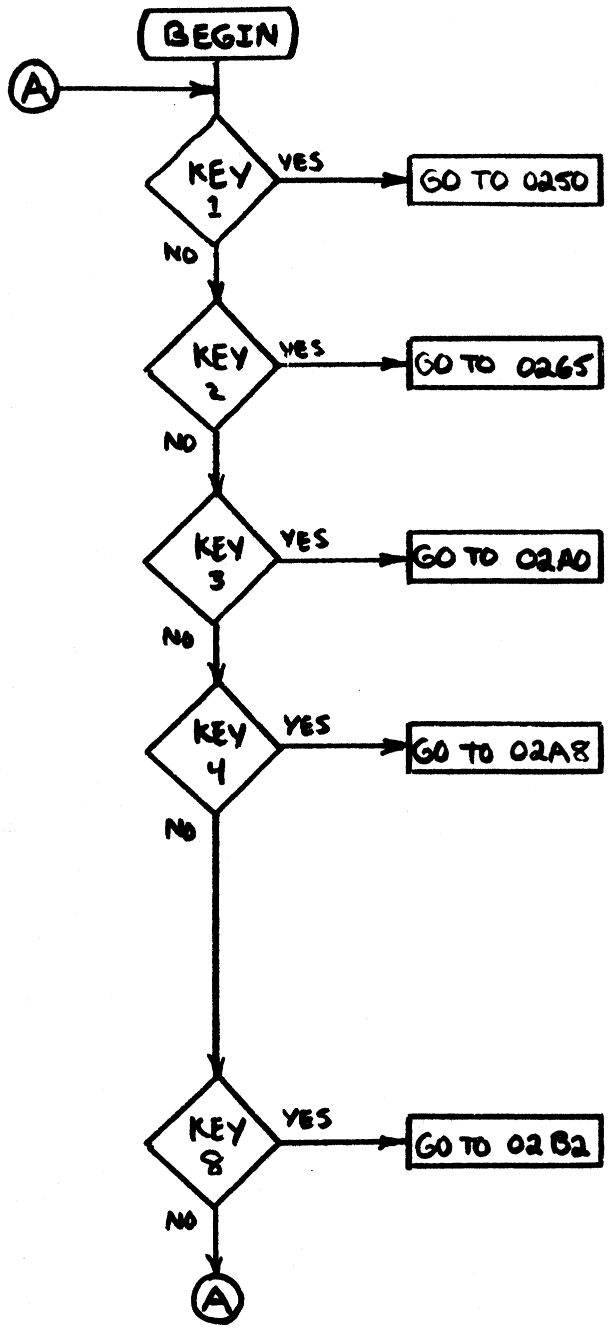
I hope you'll try using the BMMM -- with a straight face -- in your next program. It would be a shame to just let it sit there in your interpreter and go to -- no, I can't say it. Good luck with your programming and please feel free to write. (Stamped envelope appreciated--thanks.)

PROJECT: Using the BMMM instruction, write a sub-routine that sets I to the bit pattern of a particular figure depending on the value in V0. (Though you should be able to do this in other ways, the idea is to demonstrate the use of the BMMM.) Use this sub routine in your next game program to select patterns based on key presses of 2; 4; 6; <u>or</u> 8. Answer next month.

ALTERNATIVE PROJECT: Come up with a new name for the BMMM. Winner gets to treat the author to dinner (at the winner's expense) in the finest restaurant in Acapulco, near the author's apartment. Author agrees to furnish directions free of charge.

Editor's note: Don't go 'way! There's more of this article on the next page!

Flowchart 1:


BEGIN
A
KEY 1
YES
GO TO 0250
NO
KEY 2
YES
GO TO 0265
NO
KEY 3
YES
GO TO 02A0
NO
KEY 4
YES
GO TO 02A8
NO
KEY 8
YES
GO TO 02B2
NO
A

Listing 1:

Address	Instruction	Comment
OMMM	FOOA	VO=Key pressed
	4001	Skip if VO≠01 (Key 1)
	1250	Go to 0250
	4002	Skip if VO≠02
	1265	Go to 0265
	4003	Skip if VO≠03
	12A0	Go to 02A0
	.	.
	.	.
	.	.
	.	.
	.	.
	4008	Skip if VO≠08
	12B2	Go to 02B2
	1MMM	Go to OMMM for valid key press

Table 1: Jump Table

Address	Entry	Comment
OXXX	ZZ	First byte - default condition
	50	Address function #1
	65	Address function #2
	AO	Address function #3
	.	.
	.	.
	.	.
	.	.
	.	.
	B2	Address function #8
OXZZ	1MMM	Default - no operation

Listing 2:

Address	Instruction	Comment
0200	AXXX	Set "I" to address of jump table
	FOOA	VO=Key press
	F01E	I=I+VO (index the jump table)
	F065	VO=Byte addressed by I
	BMMM	Go to OMMM + VO

Answer to Last Month's LITTLE LOOPS

Relocatable loop x 02

```
0000  C0 05 00  LBR      ;Debug
  03  F8 00     LDI      ;Routine starts here
  05  A5        PLO R5   ;Initialize R5.0 for test (=00)
  06  00        BRKPT    ;View initial conditions with Debug program
  07  83        GLO R3   ;    See
  08  A4        PLO R4   ;    last
  09  24        DEC R4   ;    month's
  0A  93        GHI R3   ;    column
  0B  B4        PHI R4   ;    swap
  0C  D4        SEP R4   ;    registers R4 ↔ R3
  0D  15        INC R5   ;Add 01 to R5 -(the test function)
  0E  00        BRKPT    ;View first/second pass-results in RF.0)
  0F  D3        SEP R3   ;Causes second pass/NOP on second pass
0010  23        DEC R3   ;Stop (hit Key C to break from this loop)
```

1) Load the breakpoint and register display program (VIPER Oct. 1978) into ML 0500. (or other if you are using the 4K version)
2) Lines 0000-0002 perform a long branch to the debug program for start
3) On the first breakpoint at ML 0006, R5.0=00 -resume with Key B
4) The relocatable swap (described last month) switches R4 to the program counter and on the next breakpoint @ ML 000E, R5 will be equal to 01. Note the address in R3!
5) Resuming with Key B, R3 becomes the program counter, and on the next breakpoint (again at 000E) R5 will now be seen to equal 02 prooving that our relocatable program looped (and a backwards loop at that) through the INC R5 instruction at ML 000D <u>two</u> times without the use of branching.
6) If Key B is again hit, the program continues at ML 000F, this time with the SEP R3 acting as a NOP as R3 <u>is</u> the program counter once again. The final conditions can be viewed after ML 0010 by pressing Key C to break from the infinite loop.

Another interesting relocatable procedure I have played with since last month involves the ability to branch within a page <u>without</u> the use of the branch instruction (BR: 30) and <u>without</u> specifying an address. This then adds <u>relative</u> addressing capability to your programming. The following code demonstrates this, again using the Breakpoint program at 0500.

```
0000  C0 05 00  LBR     ;Debug
0003  00        IDL     ;Breakpoint
0004  E2        SEX     ;Dummy code-your program
0005  E2        SEX     ;    Goes here
0006  83        GLO R3  ;
0007  FF 04     SMI     ;Subtract 04 from R3.0
0009  00        IDL     ;Breakpoint
000A  A3        PLO R3  ;Relative jump
```

The code -- relocatable to any page, anywhere on the page will continue to loop back to ML 0003 every time it passes through ML 000A where R3, the program counter, is set back four addresses from where it was examined @ 0006. (When at 0006, the GLO R3 will place the value 07 into the D register as R3 already points to the next address while the GLO instruction is being executed.) Keep hitting Key B and watch what happens to R3, the program counter.

You must count the number of addresses forward or back you want to jump, then add or subtract that value to the program counter to effect the relative branch.

While of limited usefulness, the code is completely relocatable. If anyone knows of a way to write a relocatable, conditional branch, I would appreciate hearing of it.

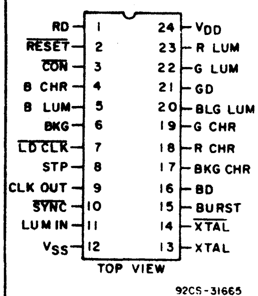
Tom Swan, P.O. Box 18987, San Antonio, TX, 78218

Does anyone know how to draw good VIP-oriented cartoons? We could have used one here!

1862 DATA SHEET
courtesy of Chris Barrish

CDP1862C Types

Preliminary Data



TERMINAL ASSIGNMENT

COS/MOS Color Generator Controller

Features:

- Static silicon-gate CMOS circuitry
- Interfaces directly with CDP1802 Microprocessor
- Interfaces directly with CDP1861 Video Dispaly Controller
- Programmable background color
- Programmable video (dot) color
- Single voltage supply (4 to 6.5 volts)
- Low quiescent and operating power
- On-chip crystal controlled oscillator
- NTSC compatible color and RGB compatible

The RCA-CDP1862C is a color generator controller designed for use in CDP1800-series microprocessor systems. It is intended for use with the RCA-CDP1861C video display controller and will interface directly with the CDP1802C/CDP1861C as shown in the system diagram below.

The CDP1862C utilizes many features of the CDP1802C and CDP1861C to simplify control and minimize the need for external components. The CDP1862C is NTSC color compatible. Red, blue and green luminance signals are also available for directly controlling the red, blue and green amplifiers of a video monitor. A 7.15909-MHz on-chip crystal-controlled oscillator or an external 7.15909-MHz clock is used to generate multiple phases of the 3.579545-MHz color burst frequency for NTSC-compatible color. The color burst is further divided by 2 to provide system timing for the CDP1802C and the CDP1861C. This frequency (1.789773 MHz) is available at CLK OUT. Two inputs, STP (Synchronous timing pulse) and $\overline{\text{SYNC}}$, are used to maintain system synchronization. The $\overline{\text{RESET}}$ input resets the CDP1862C and sets the background color to blue and the dot color to white

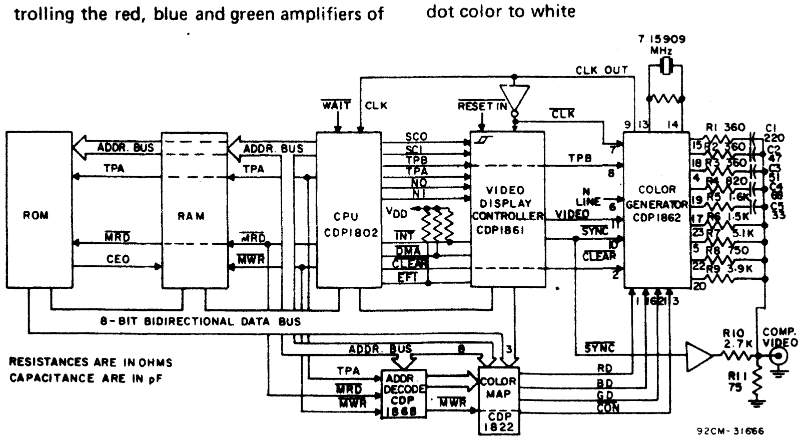


Fig. 1 – Typical CDP1802 microprocessor system using the CDP1862.

Background color: Four background colors are available. The colors are changed each time STP is pulsed high when BKG = high. The sequence is from blue to black to green to red and return to blue (see Fig. 2).

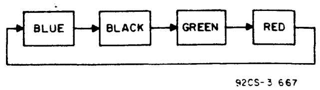


Fig. 2 – Background Color Sequencing.

Dot color: Color data (RD, BD, GD) is latched internally on the high-to-low transition of $\overline{LD\ CLK}$ when STP = high. Eight colors are available as shown in Table I. The color is overlayed onto the LUM IN data (video output from CDP1861C). Each color corresponds to eight horizontal bits of video information. Only the selected background color appears at the output if LUM IN = low. When used with the CDP1861C and set for the maximum resolution of 64X128, 1024 color blocks (8X128) are possible, and would require a 1K X 3 random-access memory storage area. This area would appear to be a write-only memory to the microprocessor because, in the programmed state, this area occupies an unique, unused 1K block of memory space. However, when it is read, this area responds to the same address space occupied by the CDP1861C refresh RAM. This is accomplished with proper decoding and requires the memory to have separate I/O lines.

TABLE I – Color Table

RD	BD	GD	COLOR
0	0	0	Black
0	0	1	Green
0	1	0	Blue
0	1	1	Cyan
1	0	0	Red
1	0	1	Yellow
1	1	0	Purple
1	1	1	White

The $\overline{CON}$ input enables the RD, BD and GD input latches. After a $\overline{RESET}$ condition, the dot color is set to white and any color change is inhibited until the $\overline{CON}$ input is pulsed low, which normally occurs when data is written into the color map. The $\overline{CON}$ input provides a means of inhibiting erroneous color data until the color map is properly loaded.

The color luminance (R LUM, B LUM, G LUM), color chrominance (R CHR, B CHR, G CHR), background luminance (BKG LUM), background chrominance (BKG CHR), color burst (BURST), and $\overline{SYNC}$ are combined by an external RC network to generate the composite video (see Fig. 1).

The BURST signal is normally high and oscillates at 1/2 the XTAL frequency from the low-to-high transition of $\overline{SYNC}$ until STP = high.

The CDP1862C types are supplied in 24-lead hermetic dual-in-line side-brazed ceramic packages (D suffix), and in 24-lead dual-in-line plastic packages (E suffix).

RECOMMENDED OPERATING CONDITIONS at T_A = 25°C, Except as Noted.

For maximum reliability, nominal operating conditions should be selected so that operation is always within the following ranges:

CHARACTERISTIC	V_{DD} (V)	Min.	Max.	UNITS
Supply-Voltage Range (For T_A = Full Package-Temperature Range)	–	4	6.5	V
Input Voltage Range	–	V_{SS}	V_{DD}	V
Input Signal Rise or Fall Time	5	–	5	μs
Clock Input Frequency, f_{CL}	5	7.15909		MHz

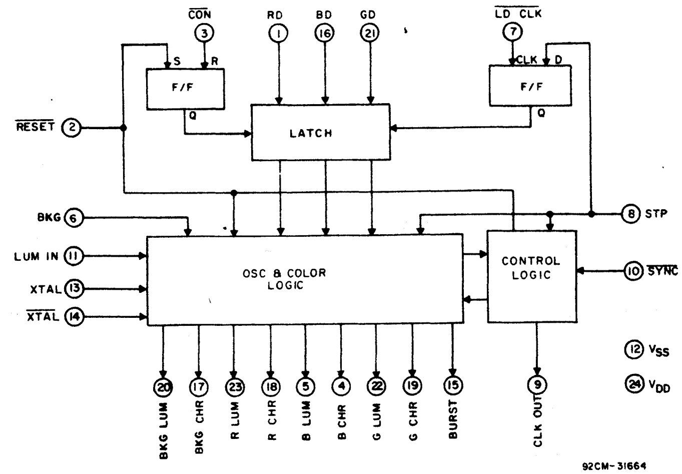


Fig. 3 – Functional block diagram.

SIGNAL DESCRIPTIONS

$\overline{RESET}$

A low level on this input initializes the internal counters, sets the background color to blue, and sets the dot color to white.

BKG

A high level on this input enables the background color to be changed when STP is pulsed high. This signal is normally connected to an I/O line of the CDP1802C microprocessor.

$\overline{LD\ CLK}$

An input signal used to latch the color data information. Color data (RD, BD, GD) is latched on the high-to-low transition of $\overline{LD\ CLK}$ when STP = high. This signal is normally connected to CLK OUT through an inverter.

STP

A high level on this input enables color data latching and sequences background color when BKG = high. This signal is normally connected to the TPB terminal of the CDP1802C microprocessor.

CLK OUT

An output signal, equal to the XTAL frequency divided by four, that provides the overall system synchronization. This signal is normally connected to the CLOCK terminal of the CDP1802C microprocessor. The inverse of this signal is normally connected to the $\overline{CLOCK}$ terminal of the CDP1861C and the $\overline{LD\ CLK}$ terminal of the CDP1862C.

$\overline{SYNC}$

An input signal used to provide horizontal line synchronization between the CDP1861C and the CDP1862C color signals. This signal is normally connected to the $\overline{SYNC}$ terminal of the CDP1861C.

LUM IN

The luminance video input, to which the color information is added. One color block corresponds to eight serial bits of data from this input. This input is normally connected to the VIDEO terminal of the CDP1861C.

V_{SS}

Negative supply voltage; ground.

XTAL, $\overline{XTAL}$

Terminal connections for an external crystal, in parallel with a resistance (10 megohms typ.) if the on-chip oscillator is utilized. Frequency trimming capacitors may be required at terminals 13 and 14. XTAL is the input for an externally generated single-phase clock.

BURST

The color reference output, which oscillates at the XTAL frequency divided by 2. This

2.03.15

signal provides approximately 11 cycles of 3.579545 MHz from the low-to-high transition of $\overline{SYNC}$ until STP = high. This signal is coupled through an external series RC circuit to the $\overline{SYNC}$ output of the CDP1861C.

RD, BD, GD

The red, blue, and green color data inputs. One of eight colors is latched on the high-to-low transition of $\overline{LD\ CLK}$ when STP = high, forming a color block of eight horizontal LUM IN data bits. Only the selected background color appears at the output if LUM IN = low. These inputs are normally connected to the DATA OUT terminals of the color map memory.

BKG LUM, R LUM, B LUM, G LUM

These output signals provide background and color luminance information. They are resistively added to the $\overline{SYNC}$ output of the CDP1861C.

BKG CHR, R CHR, B CHR, G CHR

These output signals provide background and color chrominance information. They are coupled through an external series RC circuit to the $\overline{SYNC}$ output of the CDP1861C. Each signal is phase-shifted from the BURST reference signal by the amount necessary for proper color operation.

$\overline{CON}$

The color data input latch enable signal. After a $\overline{RESET}$ condition, the internal RD, BD, and GD input latches are held in a reset state, providing a white color output. When $\overline{CON}$ is pulsed low, the reset state is removed and the latches are enabled, providing color output. This input is normally connected to the gated $\overline{MWR}$ signal from the CDP1802C.

V_{DD}

Positive supply voltage.

OPERATING AND HANDLING CONSIDERATIONS

1. **Handling**
   All inputs and outputs of RCA COS/MOS devices have a network for electrostatic protection during handling. Recommended handling practices for COS/MOS devices are described in ICAN-6525, "Guide to Better Handling and Operation of CMOS Integrated Circuits."
2. **Operating**

Operating Voltage

During operation near the maximum supply voltage limit, care should be taken to avoid or suppress power supply turn-on and turn-off transients, power supply ripple, or ground noise; any of these conditions must not cause $V_{DD}-V_{SS}$ to exceed the absolute maximum rating.

Input Signals

To prevent damage to the input protection circuit, input signals should never be greater than V_{DD} nor less than V_{SS}. Input currents must not exceed 10 mA even when the power supply is off.

Unused Inputs

A connection must be provided at every input terminal. All unused input terminals must be connected to either V_{DD} or V_{SS}, whichever is appropriate.

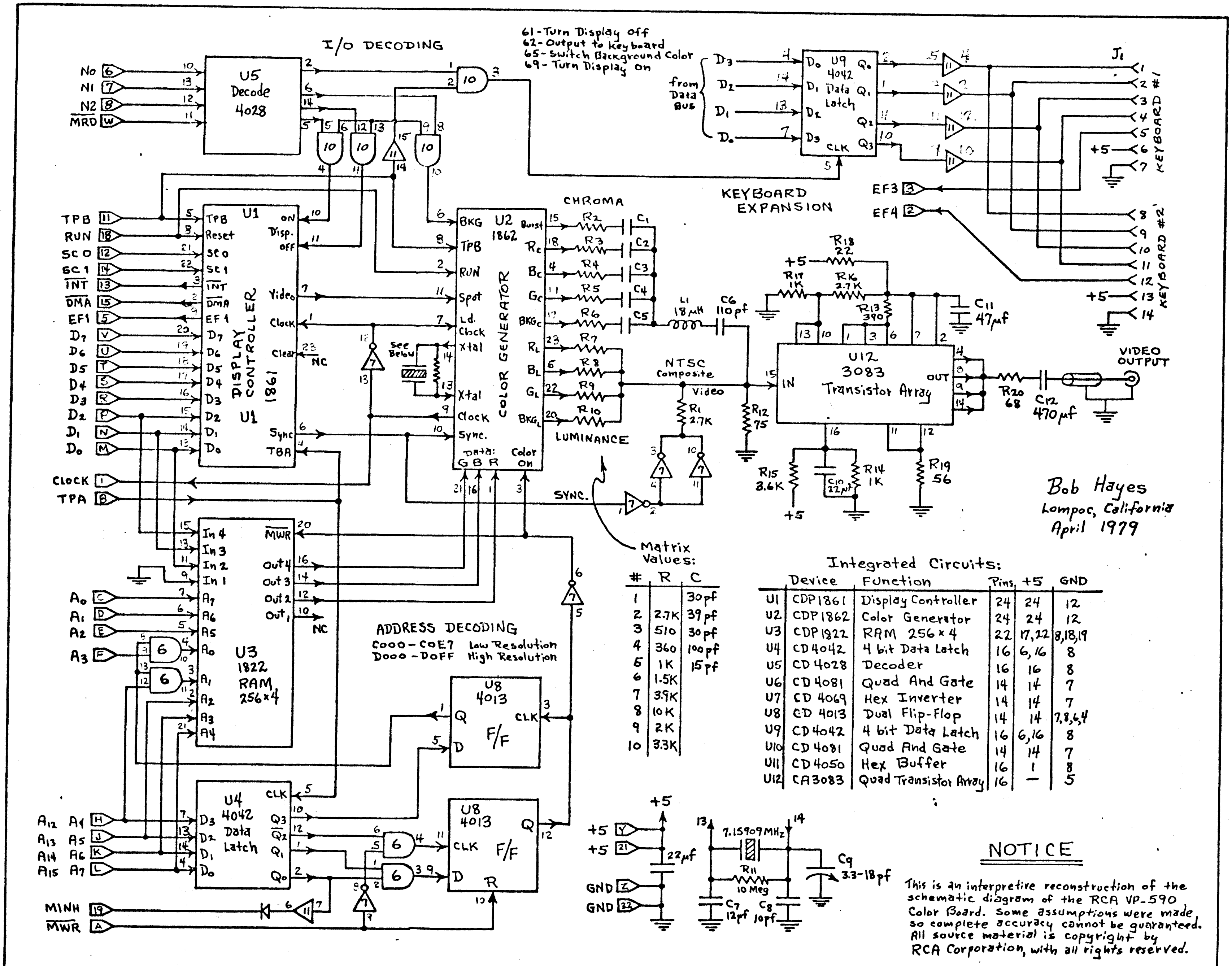
Output Short Circuits

Shorting of outputs to V_{DD} or V_{SS} may damage COS/MOS devices by exceeding the maximum device dissipation.

VP590 SCHEMATIC

Courtesy of Bob Hayes

Rick - the schematic diagram of the VP590 furnished by RCA is practically unreadable. The reproduction was very poor. So I reconstructed the schematic in a little different form that was somewhat easier for me to understand. I had to dig quite deep for the information, but I hope I have it all correct. Here are the copies of that effort. -Bob Hayes



Matrix Values:

#	R	C
1		30pf
2	2.7K	39pf
3	510	30pf
4	360	100pf
5	1K	15pf
6	1.5K	
7	3.9K	
8	10K	
9	2K	
10	3.3K	

Integrated Circuits:

	Device	Function	Pins	+5	GND
U1	CDP1861	Display Controller	24	24	12
U2	CDP1862	Color Generator	24	24	12
U3	CDP1822	RAM 256 x 4	22	17,22	8,18,19
U4	CD4042	4 bit Data Latch	16	6,16	8
U5	CD4028	Decoder	16	16	8
U6	CD4081	Quad And Gate	14	14	7
U7	CD4069	Hex Inverter	14	14	7
U8	CD4013	Dual Flip-Flop	14	14	7,8,6,4
U9	CD4042	4 bit Data Latch	16	6,16	8
U10	CD4081	Quad And Gate	14	14	7
U11	CD4050	Hex Buffer	16	1	8
U12	CA3083	Quad Transistor Array	16	—	5

NOTICE

This is an interpretive reconstruction of the schematic diagram of the RCA VP-590 Color Board. Some assumptions were made so complete accuracy cannot be guaranteed. All source material is copyright by RCA Corporation, with all rights reserved.

8-BIT MULTIPLY AND DIVIDE FOR THE COSMAC 18Ø2

by Wayne E. Smith, Jr.

The following machine language routine computes the 16-bit product of two 8-bit, unsigned numbers. It was adapted for the 18Ø2 from an article on pp. 57-58 of the June 1978 issue of Popular Electronics (an 8Ø8Ø version). The register assignments are:

```
Rj.Ø initially contains the 8-bit multiplier
Rx   contains the address of the multiplicand in memory
Rj   is returned with the 16 bit product
     Rj [          |multiplier]
        \_______ _______/
                result
Ri.Ø is used for the loop counter and is left as zero
     The DF Flag is modified.
```

```
        LDI 'ØØ'  } Zero out high order byte of result
        PHI Rj    }
        LDI 'Ø8'  } Set count to 8
        PLO Ri    }
        GLO Rj    } Shift multiplier to the right and save the
        SHR       } bit shifted out in DF
        PLO Rj    }
NXTBIT  GHI Rj      get MSByte of result
        BNF NO ADD  If DF was not set above, don't add
        ADD         add multiplicand (pointed to by RX)
NOADD   SHRC      } Shift the entire 16-bit result one bit to the
        PHI Rj    } right, shifting in (from the left) any carry
        GLO Rj    } created by the ADD and cleverly putting the
        SHRC      } next multiplier bit into DF.
        PLO Rj    }
        DEC Ri    } decrement the count and loop if not zero
        GLO Ri    }
        BNZ NXTBIT}
```

This routine, which uses a standard "shift-and-add" algorithm, only requires 22 Bytes and its execution requires 87 to 95 instructions with a run time of 791 to 864 microseconds (For a VIP instruction time of 9.Ø9 microseconds when the video is not on).

The companion divide routine computes an 8-bit quotient and an 8-bit remainder from an 8-bit divisor and an 8-bit dividend, again unsigned. The register assignments are:

```
Rj.Ø initially contains the 8-bit dividend
Rx   contains the address of the divisor in memory
Rj   is returned with the remainder and quotient
                          initially the dividend
         Rj  remainder quotient
                 .Ø        .1
Ri.Ø is used for the loop counter and is left as zero
     The DF flag is modified
```

```
        LDI '00'  } Zero out high order byte of result
        PHI Rj    }
        LDI '08'  } Set loop count to 8
        PLO Ri    }
        GLO Rj    } Shift dividend to the left and save the bit
        SHL       } shifted out (MSB) in DF
        PLO Rj    }
NXTBIT  GHI Rj      Get partial dividend shift DF bit into the
        SHLC        right end subtract the divisor (pointed to
        SM          by RX
        BDF NOADD   branch if no borrow
        ADD         add divisor back in
NOADD   PHI Rj      save parial dividend
        GLO Rj    } Shift DF bit into LSB of quotient and cleverly
        SHLC      } shift the MSB of the dividend into DF
        PLO Rj    }
        DEC Ri      Decrement the count and loop if not zero
        GLO Ri
        BNZ NXTBIT
```

The division routine uses 23 bytes and results in the execution of 95 to 1Ø3 instructions for a run time of 864 to 937 microseconds.

One way to integrate the multiply and divide routines into CHIP-8 is to replace the standard BMMM instruction with:

BXYN

For N = Ø	VX = (VX * VY) The least significant byte of the 16-bit product replaces VX. The most significant byte of the 16-bit product is stored in VF.
For N ≠ Ø	VX = (VX / VY) The quotient replaces VX and the remainder is stored in VF.

In CHIP-8, replace

Ø1A4	F8	LDI	'ØM'	page address for * and / routines
5	ØM			
6	BD	PHI	RD	
7	45	LDA	R5	
8	FA	ANI	'ØF'	isolate "N"
9	ØF			
A	3A	BNZ	'BØ'	
B	BØ			
C	F8	LSI	'ØØ'	address within page for *
D				
E	AD	PLO	RD	
Ø1AF	DD	SEP	RD	
Ø1BØ	F8	LDI	'21'	address within page for /
1				
2	AD	PLO	RD	RD is the PC for * and / routines
3	DD	SEP	RD	
ØMØØ	E6	SEX	R6	VX pointer (multiplicand)
1	Ø7	LDN	R7	Value of VY into R7.Ø (multiplier)
2	A7	PLO	R7	

```
   3    F8   LDI    'ØØ'
   4    ØØ
   5    B7   PHI    R7
   6    F8   LDI    'Ø8'
   7    Ø8
   8    AC   PLO    RC
   9    87   GLO    R7
   A    F6   SHR
   B    A7   PLO    R7
   C    97   GHI    R7     multiplication routine previously
   D    3B   BNF    '1Ø'   described with:
   E    1Ø                    Rj = R7
   F    F4   ADD              Ri = RC
ØM1Ø    76   SHRC             Rx = R6
  11    B7   PHI    R7
  12    87   GLO    R7
  13    76   SHRC
  14    A7   PLO    R7
  15    2C   DEC    RC
  16    8C   GLO    RC
  17    3A   BNZ    'ØC'
  18    ØC
  19    87   GLO    R7     VX = LSB of result
  1A    56   STR    R6
  1B    F8   LDI    'FF'
  1C    FF
  1D    A6   PLO    R6     VF = MSB of result
  1E    97   GHI    R7
  1F    56   STR    R6
ØM2Ø    D4   SEP    R4     Return to CHIP-8
  21    E7   SEX    R7     RX     VY = divisor
  22    Ø6   LDN    R6
                           VX = dividend
  23    AC   PLO    RC
  24    F8   LDI    'ØØ'
  25    ØØ
  26    BC   PHI    RC
  27    F8   LDI    'Ø8'
  28    Ø8
  29    AE   PLO    RE
  2A    8C   GLO    RC
  2B    FE   SHL
  2C    AC   PLO    RC
  2D    9C   GHI    RC
  2E    7E   SHLC
  2F    F7   SM
ØM30    33   BDF    '38'   division routine previously described
  31    38                 with:
  32    F4   ADD              Rj = RC
  33    BC   PHI    RC        Ri = RE
  34    8E   GLO    RE        Rx = R7
  35    FE   SHL
  36    3Ø   BR     '39'
  37    39
  38    BC   PHI    RC
  39    8C   GLO    RC
  3A    7E   SHLC
  3B    AC   PLO    RC
  3C    2E   DEC    RE
  3D    8E   GLO    RE
  3E    3A   BNZ    '2D'
  3F    2D
```

```
ØM4Ø   8C    GLO    RC  } VX = quotient
   1   56    STR    R6  }
   2   F8    LDI   'FF' }
   3   FF               }
   4   A6    PLO    R6  } VF = remainder
   5   9C    GHI    RC  }
   6   56    STR    R6  }
ØM47   D4    SEP    R4    Return to CHIP-8
```

Note: "M" into above address represents the page on which the routines are stored.

The following CHIP-8 program can be used to test the routines:

```
Ø2ØØ   65__   VX=__
   2   66__   VY=__
   4   B56N   VX= VX * VY or VX/VY
   6   8EFØ   Save VF
   8   632Ø   X-coord. for display
   A   222Ø   display lower byte (in decimal)
   C   63Ø6   X-coord.
   E   85EØ   value of VF (upper byte OR remainder)
Ø21Ø   222Ø   display byte (in decimal)
Ø212   1212   HALT
   :
Ø22Ø   A24Ø   I= 24Ø
   2   F553
   4   F265
   6   641Ø   Y= 1Ø
   8   FØ29
   A   D345   1st digit          Subroutine to
   C   73Ø6   incre X            display the
   E   F129                      decimal value
Ø23Ø   D345   2nd digit          of a byte
   2   73Ø6   incre X
   4   F229
   6   D345   3rd digit
Ø238   ØØEE   Return
```

A SHORT AND SIMPLE MEMORY LIST ROUTINE

by G. L. Cohen

This is a simple routine for displaying twenty bytes of memory on the screen at once. It makes following a program and debugging easier than the one-byte-at-a-time allowed by the operating system. The program is stored between 600 (hex) & 67E (hex) in the minimum COSMAC VIP system, and can be relocated to the next-to-the-last available page (below the RAM used by the operating system) in an expanded system. It is quite useful for those of us without a printer. When the program is run, the output on the screen looks like:

```
60  1A 3A 5B 12
64  D4 22 86 52
68  F8 F0 A7 0A
6C  57 87 F3 17
70  1A 3A 6B 12
```

The first column contains the least two significant digits of the address whose contents are shown in the second column. The next three columns contain the contents of the next three addresses. The listing shown is the CHIP-8 interpreter starting at 160 H.

To use the program, change the the first instruction in a CHIP-8 code at 200 H to GO TO 600. Then, key in the three digit hex address of the start of the desired listing, shown as x y z in the program below. Only the two LSD's of the addresses are displayed to save space on the screen. Hitting any key when the display is as above will result in a display of the next twenty bytes. To return to normal operati , use the operating system to change the first two program bytes to the correct entries for the beginning of the program at 200 H.

Two nonstandard CHIP-8 instructions described in VIPER 1(2),p2-6(8/78) are used in the routine. One, 8xyE,exists in the standard interpreter, but is not documented by RCA. It results in Vx containing Vy shifted one bit to the left. The other,FxF2, is an additional CHIP-8 instruction which sets I equal to the hex pattern of the MOST significant digit of Vx. The listing to implement the instruction is incorrect as shown in the reference. To us it, add the following at the end of the CHIP-8 interpreter:

```
01F2  F8  LDI
01F3  81
01F4  BA  PHI
01F5  06
01F6  F6  SHR
01F7  F6  SHR
01F8  F6  SHR
01F9  F6  SHR
01FA  30  BR
01FB  2F
```

```
0600  6404  V4=4; to be used as constant.
   2  60A0  V0=A0; Ax y z will be "poked into 636 H to point at address of list start.
   4  F50A  MSD (x) of three digit list start address keyed into V5.
   6  805A  V5+V0 into V0; V0 now contains Ax.
   8  F10A  second MSD (y) of start address keyed into V1.
   A  811E
   C  811E    shift the second MSD(y) to be the MSD of V1.
   E  811E
  10  811E
   2  F50A  key LSD of list start address(z) into V5.
   4  8154  V5+V1 into V1; now V0,V1 contains A x y z.
*  6  A636  Point at 636 H to put A x y z  into next.
   8  F155  V0,V1 into 636,637 H; list start address pointer has been poked into program.
   A  161C  Skip to next instruction; dummy byte for program modification.
   C  6B00  VB=0; initialize vertical display counter.
   E  8910  V1 into V9;
  20  8800  V0 into V8; Ax y z saved in V8,V9 for later reference.
```

```
   2  6600   V6=0; initialize count of lines shown in display.
   4  6A00   VA=0; initialize horizontal location counter.
   6  6DF0   VD=F0; VD will be incremented to point to V0,V1,V2 and V3 in turn.
   8  F1F2   set display pointer to hex pattern of MSD of V1(y).
   A  DAB5   show MSD of address.
   C  7A06   VA=VA+6; increment horozontal display counter to next digit position.
   E  F129   set display pointer to LSD of V1(z).
  30  DAB5   now both LSD's of address are on screen.
   2  7A08   VA=VA+8; increment horozontal display counter to first data position.
*  4  2666   gosub to poke memory content pointer(VD) into 63A H and 640 H.
   6  A000   location into which start address is poked in location 618 H.
   8  F365   copy four bytes to be displayed into V0,V1,V2,V3
   A  F0F2   point at MSD of byte to be dispalyed, stored in either V0(as shown) or V1,V2,V3
   C  DAB5   display MSD
   E  7A06   VA=VA+6; increment horizontal display pointer.
  40  F029   point at LSD of byte to be displayed.
   2  DAB5   dispaly LSD
   4  7A07   space 7 horozontally for display of next byte
   6  7D01   add one to Fn in VD, initially set to F0.
   8  4DF4   skip if VD ≠ F4; if not to end of 4 byte display on line yet.
*  A  1650   go to next line routine.
*  C  2666   gosub byte count placer routine.
*  E  163A   go back to get next byte for display.
  50  7601   New Line routine - add one to line count.
   2  4605   skip next if line count ≠ 5; if it = 5 we would be off screen.
*  4  1672   go to pause for next group of display bytes.
   6  7B06   B=B+6 for end of line check
   8  8944   add 4 to LSD of list address, note VF=1 if there was a carry.
   A  88F4   V8=V8+VF;  in case there was a carry, V8 needed to be incremented.
   C  8080   V8 into V0; A x into V0
   E  8190   V9 into V1; y z into V1, so that V0,V1 contains A x y z  for next address.
* 60  A636   point at byte pointer address.
   2  F155   poke new line start in place.
*  4  1624   go to display start.
   6  80D0   subroutine to put byte pointer; copy VD into V0
*  8  A63A   point at MSD display location pointer.
   A  F055   poke new byte pointer.
*  C  A640   point at LSD display location pointer.
   E  F055   poke new byte pointer.
0670  00EE return
   2  F50A   wait  for key to be hit(pause).
   4  00E0   clear screen.
   6  8944   add 4 to LSD of list address, VF = 1 if carry.
   8  88F4   add 1 to V8 if there was a carry.
   A  8080   put V8 into V0; A x in V0
   C  8190   put V9 into V1; A x y z in V0,V1.
*  E  1616   go to start.
```

* These statements refer to memory locations and need to be changed if this program starts other than at 0600.

SOFTWARE REVIEW

by Wayne E. Smith, Jr.

Cuddly Software (98 Thorndale Terrace, Rochester, NY, 14611, 716-328-8259) has released two software products which may be of interest to VIPER readers:

CSOS-31 (Cuddly Software Operating System) and
CSTP-24 (Cuddly Software Trace Program)

These products are well constructed, fully tested, and extensively documented. Cuddly Software is currently working an 18Ø2 Assembler for release later this year.

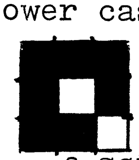
CSOS-31

CSOS-31 (Cuddly Software Operating System version 31) is a coordinated, expandable package of useful subroutines that can be accessed interactively thru the hex keyboard or called from user written machine language programs (using SCRT). CSOS-31's many capabilities include:

* moving 1 to 65,536 bytes from anywhere to anywhere in memory.
* keyboard (or subroutine) selectable screen resolution: 32 x 64, 64 x 64, or 128 x 64 dots.
* displaying ASCII characters anywhere on the screen or via a line oriented "DSPLYDRVR".
* A SCRNDRVR subroutine which is called from user machine language programs and provides a somewhat unique 2 page display (more about it later)
- Changing bytes in memory and adding or deleting bytes while expanding or contracting the affected page accordingly
- Erasing and displaying memory pages, including scrolling thru seccessive pages.
- Executing a program at a keyboard entered address.
- Input of bytes, nibbles, or ASCII characters form either the hex keyboard or a full ASCII keyboard.

Most of the capabilities are available from the hex keyboard thru the table-driven TASK: SELECT routine. The user can easily add new functions or modify existing ones.

SCRNDRVR is a remarkable full-ASCII display driver that scrolls and displays 1Ø lines of variable width (!!) characters allowing 1Ø to 32 characters per line. Numbers are fixed width to allow column alignment in tables. A cursor is provided that may be backspaced to make character corrections (on the last line only). The documentation gives a conplete description of how the character set may be redefined by the user. The character set that is provided is creative, but some of the characters are a little difficult to read, e.g., the lower case "e" is



An example of the use of SCRNDRVR is the following simple looping program that reads 2 hex keys representing an ASCII character (or control codes) and passes the character to SCRNDRVR for display:

```
Ø63E  D4     SEP  R4 }  call BYTERDR
      ØØ5C   'ØØ5C'  }
      D4     SEP  R4 }  call SCRMDRVR
      ØØBE   'ØØBE'  }
      3Ø3E   BR   '3E'  Loop
```

It acts somewhat like a "TV Typewriter".

To use SCRNDRVR, it is necessary to have at least 1Ø pages of memory. It may be excluded and less than 5 pages will be used by CSOS-31. Two pages are used for character display patterns. This may seem a little exorbitant for systems with small memories considering how often full upper and lower case ASCII is needed. On the other hand, SCRNDRVR is a very powerful routine and with it, CSOS-31 can be used to construct a text editor very easily!

Excellent and extensive documentation is provided with CSOS-31. It is a professionally written, fully tested system that is a great enhancement to the 18Ø2 and this reviewer hardily recommends it.

CSTP-24

CSTP-24 (Cuddly Software Trace Program version 24) is a remarkably sophisticated 18Ø2 simulator with a fancy screen display of all registers and status unformation (64 x 64 dots) that changes dynamically as it single-steps thru a program. Unfortunately, it requires an ASCII keyboard tied to an I/O part and occupies 1Ø pages of memory (the standard 2K VIP only has 8!)

The program being traced is loaded starting at location ØAØØ (which it believes to be ØØØØ). CSTP-24 itself is memory protected in that it is not possible for the traced program to store into the area in which CSTP-24 resides. Its author claims it has been "destruction tested".

The program execution being simulated may be interrupted from the keyboard wia UCP (User Command Processor) and any register or internal flag may be modified or the program flow may be altered.

Extensive, well written documentation is provided, including a complete description of all UCP instructions, formatted machine code listings, a storage map, and an index.

All in all, CSTP-24 is a professionally programmed package and should be a useful debugging tool if your system has an ASCII keyboard and enough memory.

new

vip II

Now everyone can enjoy the benefits of a personal computer

Introducing VIP II . . .

The fun computer
Video games and recreation unlimited: black jack, pinball, bowling, lunar landing, biorythms, a few of the dozens of games available for VIP II. But the choice is limited only by your imagination. With RCA's CHIP 8 graphic language you can create your own.

The creative computer
Features sound and color graphics. Compose or recreate music or sound effects. Express yourself — computer art, graphics and klaidescope effects in vivid color.

The educational computer
Math skills, spelling bees, language lessons — interactive programs you can develop for yourself, for your children or for a classroom.
Learn computer programming and the fundamentals of computers — even if you've never used a computer before.

The practical computer
Personal finance, home records, scientific calculations, mathematical problems, energy management, security system control and monitoring, process control, stock records — all achievable with the versatile VIP II.

The hobby computer
The ability of the VIP II to interact with the real world allows you to expand the scope of your current hobby — computer controlled model railroads, Ham applications including encoding and decoding, disco dancing lights, robotics.

The serious computer
Don't underestimate the power of this tool. The VIP II is a complete microcomputer system designed around the RCA CDP 1802 microprocessor — the processor chosen for space — the processor noted by one independent computer specialist as "the best microprocessor — bar none". And VIP II BASIC is the fastest and most extensive BASIC yet developed for the CDP 1802. The library of commands and statements even includes special commands for graphic color and sound control.
And VIP II is sophisticated enough for the professional engineer to develop machine language programs and prototype hardware.

Probably no modern invention has changed our lives as much as the computer. But until recently, few people had ever seen one — and only a handful understood them.

The Personal Computer "revolution" is changing all this — and RCA's VIP II makes it easy for you to join in.

The RCA VIP II can be used with your home TV or with a seperate video monitor, can store programs or data on standard audio cassettes, is easy to use yet challenging, fun yet practical — and above all is inexpensive enough for the average family to afford.

You can start learning immediately with the simple VIP II BASIC "English" programming language and can soon be developing and running programs such as those illustrated. The possibilities are nearly endless. And as you gain experience you can impliment programs in VIP II CHIP 8 or combine BASIC with machine language programming to speed execution, save memory space and handle more complicated routines.

Three Program Languages:
VIP II BASIC - The popular, easy to learn and use computer language. Commands include special VIP II-oriented commands for audio, video display and cassette storage or recall of programs.

VIP II CHIP 8 — Simple, easily learned VIP II oriented, hexidecimal interpretive language. Ideal for graphics. Reduces memory usage. Speeds program execution.

CDP1802 Microprocessor Machine Language. — Gives complete flexibility for the experienced programmer. Machine language subroutines can also be nested in CHIP 8 or BASIC programs.

Programmable Color and Sound
8 K RAM - expandable to 32 K
Cassette Interface
Full System Expansion Capability
ASCII Keyboard plus Hexidecimal Keypad
ASCII —58-keys including two user-definable. Full 128 character ASCII encoded.
Hexidecimal — 16 key layout for faster input of data and mathematics.

Flexible membrane switches, rugged, reliable and spillproof, with a finger positioning overlay provide excellent "feel" and the on-board tone generator, volume control and speaker provides aural feedback.

Specifications

CPU . . . . RCA CDP1802 Microprocessor

Memory

RAM: VP-2001 . . . . 4 K bytes
VP-2002 . . . . 8 K bytes
Both externally expandable to 32 K total.

ROM . . . . 12 K bytes Includes:

ROM Resident VIP II BASIC. provides full floating point capability plus special VIP oriented commands for audio cassette storage, tone generator and video display. VIP II BASIC display of 16 characters per line, 11 lines, with automatic line overflow and scrolling routine.

ROM Resident Monitor impliments memory write/examine, load to/record from audio cassette and key debounce.

Keyboard . . . . 74 light-touch keys. ROM resident monitor provides key debounce and activates tone generator for aural feedback. Includes two interdependent key layouts:

Typewriter: 58-key, ASCII encoded.
Includes two user definable keys. (30V, 0 1 A, 1 W DC max.)

Hexidecimal: 16-key. Includes system command keys.

Video

Format . . . . Video Display IC and ROM monitor bit map up to 1 K bytes of memory onto user-supplied video monitor (525 line, 60 Hz). Each bit in addressed memory forms a rectangular "dot" in the display field.

Color . . . . **Foreground:** One of eight program-defined colors (Purple, Red, Yellow, Green, Aqua, Blue, White, Black) within each of 1024 color zones.

Background: One of four program-defined full-screen colors (Light Red, Light Green, Light Blue, Black).

Operation: Each "dot" of the display when "On" displays the currently defined foreground color for its zone and when "Off" displays the currently-defined background color.

Interface . . . . RCA Phono Jack for 75 ohm video cable. 8-foot cable with matching plugs supplied. 1.0 Vp-p composite video into 75 ohms.

Audio . . . . One of 256 program-selectable tones ranging from approximately 107 Hz to 27.5 kHz. Includes over 4 complete octaves of musical notes. Program-definable tone duration. On-board audio amplifier and speaker. Miniature Phone Jack output connector.

Audio Cassette
Interface . . . . Miniature Phono Jack. 18-inch cable with matching plugs supplied. 100 byte/second data rate. Tone generator activated while tape is written. TAPE LED activated while tape is read.

Expansion Interface . . . . 25 pin, dual read-out (50-contact) card edge. Buffered
Provides data bus, address bus flags, N lines, audio, video, user keys and power.
9-pin "D" connector for interfacing Joysticks and auxiliary keyboards.

Power . . . . 5 V DC from supplied 120 V, 60 Hz receptacle-mounted power supply
Power plug and 10 ft. power cord supplied.

Documentation . . . . Instruction Manual including VIP II BASIC, VIP II CHIP 8 and CDP1802 machine language. Schematics. Board layout.

Size . . . . 16.5" length, 7" depth, 2" height.

Shipping Weight . . . . 5 pounds.

VIP II BASIC Vocabulary

Commands	Statements					Mathematics
CLOAD	ABS	FREQ	KEY	PT	TIMER	ATN
CSAVE	CLS	GCOL	LET	READ	TONE	COS
DLOAD	COLOR	GET	MEM	REM	TVOFF	EXP
DSAVE	DATA	GOKEY	NEXT	RESTORE	TVON	FLOATING PT +,-,x,/
LIST	DEFINT	GOSUB	OUT	RETURN	USR	INTEGER +,-,x,/
NEW	DIM	GOTO	PEEK	RND	WAIT	LOG
RUN	END	IF	PLOT	SCOL	@(Hex)	SIN
SAVE	FIXED	IN	POKE	SHOW	#(Hex)	SQR
	FMODE	INPUT	PRINT	STORE	:(Multiple Statement)	
	FOR	INT	PRINT AT	TIME		

RCA COSMAC VIP MARKETING
New Holland Avenue
Lancaster, PA 17604

VIP PRODUCTS FROM ARESCO

FLASH ** RCA has raised the prices on some of these items. We called and asked about the products we already have on order - some orders have been in since February - and we were assured we'd get the old prices on all items ordered before August 6th. The new price list didn't arrive until AFTER the printer had already finished with the August issue, so any orders received this month were returned with a copy of this page of the September VIPER.

	RCA PRODUCT		PRICE	AVAILABLE
	VP44	Four 2114 RAM ICs	36.	2-4 weeks
	VP45	Four 9131 RAM ICs	36.	"
	VP550	Supersound Board	49.	"
	VP560	EPROM Board	34.	"
	VP565	EPROM Programmer Board	99.	"
	VP570	Memory Expansion Board (4K)	95.	"
	VP575	Expansion Board (4 buffered, 1 unbuffered sockets for up to 5 accessory boards)	59.	"
	VP576	Expansion Board (I/O or Expansion port-for two accessory boards)	20.	4-6 weeks
	VP580	Keypad	20.	2-4 weeks
	VP585	Keypad Interface Board (not needed if VP590 Color Board is used)	15.	"
	VP590	Color Board	69.	"
	VP595	Simple Sound Board	30.	"
	VP600	ASCII Keyboard ** CANCELLED **		
*	VP601	ASCII keyboard	65.	October???
*	VP611	ASCII/hex keyboard	80.	October???
*	VP620	Keyboard Cable for VP601 & VP611	20.	October???
*	VP623	Keyboard Cable for VP602 & VP611 The VP620 connects at one end to the VIP (VP711). The VP623 cable has one end unterminated.	20.	October???
	VP700	BASIC ROM Board	39.	2-4 weeks
	VP710	Game Manual (code for 20 games)	11.	
	VP311	Instruction Manual	6.	
	VP320	CHIP-8 Manual	6.	
	MPM201B	1802 Manual	6.	
	CDP18S731	RAM/IO Expansion - four 9131 RAMs	69.	2-4 weeks
	CDP18S745	RAM/IO Expansion - four 2114 RAMs For the 18S022 VIP Kits	69.	"
	TC1210	9" video monitor	195.	"

ARESCO PRODUCT		
VIPER - Volume 1 plus index	15.	stock
VIPER - Volume 2	15.	As issued
PIPS FOR VIPS - Volume 1 plus cassette	19.95	stock
PIPS FOR VIPS - Volume 2 plus cassette This price good only until 10/14/79	14.95	October
PIPS FOR VIPS - Volume 3 plus cassette This price good only until 1/15/80	14.95	January

ORDER FORM * ORDER FORM * ORDER FORM * ORDER FORM * ORDER FORM

TO ORDER RCA PRODUCTS: Fill out the order form at the bottom of the page, indicating which RCA item you wish to order. Then handwrite a note to the effect that you are aware that delivery of RCA produced items may take more than 30 days and that we are authorized to hold your money until RCA can deliver to us. We will hold it - we won't cash your check or charge your credit card until the day we make shipment to you.

TO ORDER ARESCO PRODUCTS: Just fill out the form. All ARESCO products are available within two days of our receipt of your order, and we can ship immediately.

Please ship me the items listed below. I enclose full payment or authorize you to charge my MC/VISA credit card.

NAME____________________ MC/VISA #________________

STREET____________________ EXP. DATE ________________

CITY, STATE, ZIP____________________MC INTERBANK #______

CREDIT CARD ORDERS REQUIRE SIGNATURE____________________

VIP HARDWARE AND SOFTWARE PRICES

RCA PRODUCED PRODUCTS

VP44	Four 2114 RAMs - for on-board expansion..........	$ 36.00
VP45	Four 9131 RAMs - for on-board expansion..........	36.00
VP311	VIP Instruction Manual............................	6.00
VP320	VIP (CHIP-8) User Guide...........................	6.00
VP550	SuperSound Board..................................	49.00
VP560	EPROM Board.......................................	34.00
VP565	EPROM Programmer Board............................	99.00
VP570	Memory Expansion Board (4K static RAM)............	95.00
VP575	System Expansion..................................	59.00
VP580	Expansion Keypad..................................	15.00
VP585	Keypad Interface Board............................	10.00
VP590	Color Board.......................................	69.00
VP595	Simple Sound Board................................	24.00
VP600	ASCII Keyboard (available any time now)...........	49.00
VP700	Tiny BASIC ROM Board..............................	39.00
VP710	VIP Game Manual.........(20 games)................	10.00
VP711	VIP (assembled unit)..............................	249.00
TC1210	9" Video Monitor (black and white)...............	195.00
TC1212	12" Video Monitor (black and white)..............	325.00
TC1217	17" Video Monitor (black and white)..............	435.00
CDP18S731	Four 9131 RAMs, plus necessary components for I/O expansion ports. To be used only with VIP kits..	69.00
CDP18S745	Four 2114 RAMs, plus necessary components for I/O expansion ports. To be used only with VIP kits..	69.00
MPM201B	CDP 1802 User Manual..............................	6.00
1861 Data Sheet	....	1.50

Please note on your order for RCA products that you're aware that delivery may take more than 30 days. We'll ship as soon as RCA delivers the products to us.

ARESCO PRODUCTS

Hersh Editor (cassette only)................................	5.00
Stein Editor (cassette only)................................	5.00
Stein Editor (with documentation)...........................	15.00
Astle's LIFE (with documentation)...........................	10.00
Hottle's BASEBALL (cassette only)...........................	10.00
Tape of any program published in any issue of the VIPER....	5.00
ATV Research Microverter....................................	34.95
Studio II Information Kit A.................................	5.00
Studio II Conversion Kit PROM only..........................	15.00
Studio II Conversion Kit PCB only...........................	10.00
Studio II Conversion Kit D (assembled unit plus Kit A).....	50.00
Studio II Conversion Kit Backplane Board....................	20.00
Studio II Conversion Kit Backplane Board with 4 connectors.	36.00
PIPS FOR VIPS (Volume I)....................................	19.95
PIPS FOR VIPS (Volume II).....until 10/15/79...............	14.95

PLEASE give us a street address for UPS delivery. UPS cannot deliver to the post office! Also - we'll accept COD orders if you're willing to pay the $1.00 fee charged by UPS for collecting and the shipping fee (which may be up to $2.00 per delivery.)

ARESCO

P.O. Box 1142
Columbia, MD 21044
THE PAPER - VIPER - RAINBOW - SOURCE

Before ordering products from ARESCO, please read this notice!

We, like everyone else, must wait in line for RCA to deliver product to us. We have an advantage over the individual VIP owner, however, in that we can telephone a purchase order number and pay for the product when it arrives rather than by prepayment. We do not cash your check, nor charge your credit card, until the day we ship your merchandise to you. So you will have the use of those funds, rather than RCA or ARESCO, until then. Secondly, we do not, as a rule, stock any items listed as available from RCA. We don't have the capital to handle that. Therefore, you must make note on any order for RCA products that you are aware that shipment may take more than 30 days. It usually does. And, finally, we cannot accept purchase orders ourselves. We have found that purchase orders from even the most reputable firms do not get paid in time to meet our needs, and we don't have the capital to handle that, either.

To order any product listed on page 2.01.27, or any of its updates, simply fill in the blanks below. Please give a street address for UPS delivery within the USA, or send $4.00 for delivery by US mail to your post office box. For subscription orders, non USA people should send an extra $10 for the volume, if they want air mail. If not, it will go out second-class, like the USA subscriptions do.

Please ship the following items. I understand that payment in full must accompany my order, but that you will not charge my credit card or cash my check until the day you ship my products to me.

NAME____

ADDRESS______________________________________

CITY:____________________STATE:________ZIP:________

MC/VISA #____________________________EXP DATE:________

(If you're using Master Charge, we need the "other" four numbers on your card. These digits represent the Interbank Number. Visa cards don't have such a number.) INTERBANK #__________

ITEMS ORDERED:

- VIPER VOL. 1 (+ Index) - $15.00 ________
- VIPER VOL. 2 - $15.00 ________
- PIPS FOR VIPS(+tape) - $19.95 ________
- PIPS FOR VIPS (no tape)- $14.95 ________

ARESCO SOFTWARE:________________________________

RCA PRODUCTS:__________________________________

Credit card holder's signature:______________________

Total amount authorized or enclosed: $________________

* Initial here to indicate you understand that delivery of RCA products may take more than 30 days.________