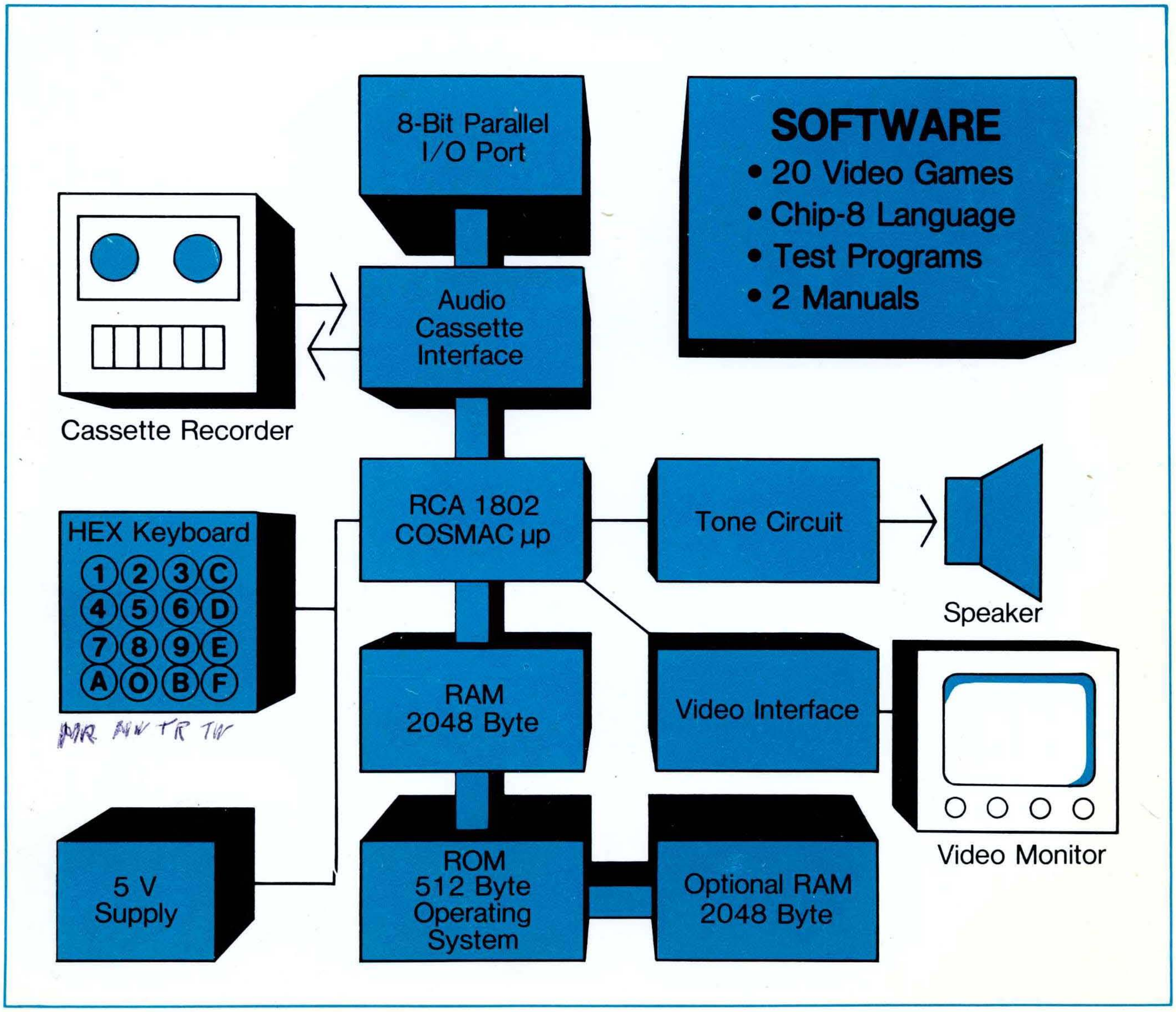
RCA Solid State

RCA COSMAC VIP CDP18S711 Instruction Manual



MR AW TR TW

VIP-311

Suggested Price $5.00

RCA COSMAC VIP CDP18S711 Instruction Manual

RCA Solid State Division, Somerville, N. J. 08876

Printed in USA/2-78

VIP-311

ACKNOWLEDGMENT

COSMAC VIP has been created by Joe Weisbecker of the RCA Laboratories, Princeton, N.J. so that everyone can have fun and useful personal computer experiences. The elegant and simple hardware system design and the powerful video output together with the customized CHIP-8 language interpreter constitute a fresh and promising approach to personal computers.

If questions arise regarding the VIP software or hardware, write to

VIP
RCA Solid State Division
Box 3200
Somerville, N.J. 08876

or telephone

Area code 201 526-6141

Contents

Contents (Continued)

I. Getting Started

COSMAC VIP (Video Interface Processor) CDP18S711 is a complete computer on a single printed-circuit card. It includes the following:

*RCA CDP1802 Microprocessor (91 instructions)
*2048-byte RAM
*Built-in hex keyboard (modern reliable touch-pad type)
*Graphic video display interface (standard video output)
*100-byte-per-second audio cassette interface
*Regulated power supply (wall-pack type)
*Crystal clock
*Sound circuits (for signal tones and games)
*512-byte ROM operating system
*Comprehensive documentation
*20 ready-to-use video game programs
*Unique CHIP-8 language (31 easy-to-use instructions)
*On-card RAM expansion up to 4096 bytes
*On-card parallel I/O port
*Connector for extensive external expansion capability

COSMAC VIP was designed for home hobby use. Just add an inexpensive video display and an audio cassette recorder for program storage. You don't need expensive, hidden extras such as power supply, computer terminal, external keyboard, or additional RAM. COSMAC VIP provides everything needed for years of creative computer fun for the whole family. With COSMAC VIP you're immediately ready to play video games, experiment with computer art or animation, write your own programs with a new language called CHIP-8, or get hands-on experience using machine language.

With COSMAC VIP you can easily create pictures on the display screen and move them around. This feature is invaluable for video games and not usually available with computers costing several times as much. The software you need to use your computer is provided free instead of at added cost or not at all. Simplified operation was a primary design goal so that you don't have to waste a lot of time learning and remembering complex operating procedures. COSMAC VIP uses state-of-the-art devices coupled with an efficient design. Full expansion capability allows you to inexpensively tailor COSMAC VIP to specific applications such as model railroad control, music synthesis, or color graphics. You will soon discover that COSMAC VIP provides a refreshingly new, lower-cost alternative to conventional computers which have been aimed more toward mathematics and business than fun.

What This Manual Covers

This manual serves several purposes. It lets you get started playing video games with minimum effort. Just set up your system as described in this section and learn how to use the operating system and cassette interface as described in the next section. You can immediately use all the video games in Appendix D without going any further.

If you want to learn to write your own programs, Section III describes an easy language to start with called CHIP-8. Most of the programs in Appendix D were designed using this language. CHIP-8 looks somewhat like machine language but is quicker to learn and easier to use than many of the more common high-level languages. It also requires much less RAM, which saves you a lot of money. CHIP-

8 includes a real time clock, random number generator, decimal conversion, and digit or graphic display capability. It only uses 512 bytes of RAM leaving over 1024 bytes for programs in a 2048-byte system. (You can get an additional 2048 bytes of RAM by plugging four more RAM chips into your card.)

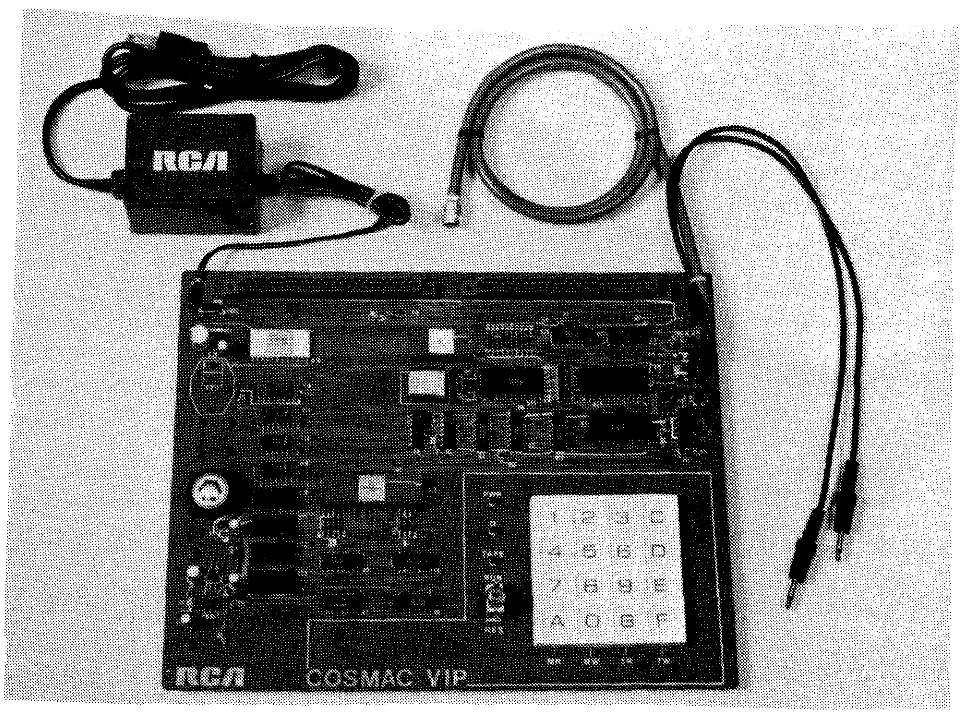
With the aid of the User Manual for the CDP1802 COSMAC Microprocessor, MPM-201, you can explore the fascinating world of machine language programming. You can even combine machine language programs with CHIP-8 programs or develop your own interpretive languages.

For hardware hackers, COSMAC VIP provides complete external interface capabilities. Some suggestions for inexpensive external devices and applications are listed in Section VI. Logic diagrams, data sheets, trouble-shooting hints, and test programs are provided so that you can explore the hardware in as much detail as you want.

This manual assumes that you are familiar with computer basics from reading one or more of the excellent magazines devoted to home computing. You should understand RAM, ROM, memory addressing, instructions, bytes, etc. The use of a scope will facilitate setting up the cassette system and identifying hardware problems in the rare case where they occur. Hex notation is used in this manual unless noted otherwise. (One byte equals two hex digits.)

The Power Supply

The output wires of the internally regulated power converter supplied with the COSMAC VIP CDP18S711 are connected to the +V DC and GND pads at the back left corner of the PC card. The power converter output is regulated +5 V DC at 600 mA. If you wish to add more RAM to your system, however, you may need a higher-current power supply. A 2048-byte system requires about 350 mA (600 mA worst case). A 4096-byte system should require average current of about 600 mA. If, however, your RAM chips require above average power, you may need to supply as much as 900 mA at 5 V DC, regulated. You can also use your own unregulated 8 to 10 V DC power supply by adding voltage regulator U28 (plus heatsink) to your COSMAC VIP card and cutting the printed circuit link called LK1. Never apply more than +5 V DC to the card unless the U28 regulator has been added and link LK1 cut.



Photograph of COSMAC VIP (Video Interface Processor) CDP18S711
The cables in the upper right are for the video display and for cassette operation. Cable on the upper left goes to the power converter.

What You See

You must now decide on the video display for your computer. The video pad at the back right corner of the COSMAC VIP card provides a video signal which you can connect directly to the high-impedance input of most standard video monitors. The horizontal sync frequency is 15,720 Hz and the vertical sync frequency is 60 Hz. One solution to your video display need is a commercial video monitor having a suitable input -- not rf or antenna input. Another option is your TV receiver used with a relatively inexpensive FCC-approved modulator. **Do not use a standard TV receiver with the VIP output connected to the VHF or UHF antenna terminals. Do not use transformerless TV receivers.**

Turning It On

After attaching a suitable video display, apply power. Make sure the RUN switch is in the down (or reset) position. Hold hex key C down while you flip the RUN switch up. You should hear a tone with key C pressed and the Q light should be on. When you release key C the tone and Q light should both go off. (The tone occurs whenever the Q light is on.) You should now see a random pattern of small square spots on the display. Push hex keys 8008 in sequence and you should see 8008 at the bottom left of the screen and 64 at the lower right. Adjust your display controls for the best picture (white spots on a black background). You can experiment with changing the values of R1, R2, and R4 on the COSMAC VIP card to improve picture quality although this step shouldn't be necessary. Certain modulators work better with an R4 of 1 kilohm instead of 200 ohms. If you don't get a video picture refer to Section VII for troubleshooting hints.

After completing the above set-up procedure, you are ready to enter and run programs on your COSMAC VIP. The COSMAC VIP operating system, explained in the next section, permits you to load programs into memory from the hex keyboard, verify them, and record them on cassettes for later reuse.

II. COSMAC VIP Operation

COSMAC VIP is operated with the RUN switch and hex keyboard. The PWR light shows that power is on. The Q light is activated by various programs. A tone is sounded whenever the Q light is on. The TAPE light glows when cassette input data is present. When using COSMAC VIP, always start with the RUN switch in the down (or reset) position. Flipping the RUN switch up initiates execution of machine language programs beginning at memory location 0000. If you have previously stored the CHIP-8 language interpreter program at locations 0000-01FF, execution of a program written in this language will begin at 0200. To manually terminate execution of any program, flip RUN down.

Using the Operating System

With COSMAC VIP you can load programs into memory from the hex keyboard or cassette recorder, record the contents of memory on cassettes, show the contents of memory bytes in hex form on the display, and examine the contents of CDP1802 microprocessor registers. These functions are performed with the aid of a special program called an operating system. This operating system is contained in a ROM so that it's ready to use as soon as power is turned on. It is located at memory locations 8000-81FF. A machine code listing and summary of this operating system is provided in Appendix B.

To use the operating system hold key C down on the hex keyboard when you flip RUN up. You will hear a tone. Release key C and you're ready to use the operating system.

After selecting the operating system you can do four different operations as shown in the following table:

KEY	OPERATION
0	MW (Memory Write)
A	MR (Memory Read)
F	TW (Tape Write)
B	TR (Tape Read)

For any of these operations you must first enter a memory address. Enter the 4 hex digits of any memory address using the hex keyboard (most significant digit first). You will see the address at the lower left of the screen and the byte contained in that address at the lower right. Remember that addresses and bytes are always entered and shown in hex form. Suppose you entered 0200. You will see 0200 at the bottom left of the screen and the byte stored at 0200 at the lower right.

Memory Write

If you want to change this byte, press the 0 key. Now press two digits of the new byte (most significant digit first) and it will be stored at 0200 replacing the original byte. You will see this change on the screen. If you enter another byte it will be shown and stored at the next higher address in sequence (0201 in this example). You can load any sequence of bytes directly from the hex keyboard in this manner. If you make a mistake, flip RUN down. With key C pressed, flip RUN back up. Enter the address at which you made the error. Press key 0 and resume entering your program.

Note the random bit pattern on the screen above the hex display. This pattern is the binary data contained in the last 256-byte page of the on-card RAM. If you have a 2048-byte RAM, you are seeing locations 0700-7FF on the screen. Bit 7 of the byte at 0700 is in the upper left corner. Try storing a sequence of eight AA bytes followed by eight 55 bytes starting at location 0700. Keep repeating this sequence to draw a checkerboard pattern on the screen. There are 32 rows of spots on the screen. Each row represents 8 memory bytes (64 bits). Locations 0700-0707 are shown in the top row, 0708-070F in the next row down. Draw a bit map on paper and you can construct pictures on the TV screen by entering the proper byte sequences. The byte pattern for displaying the word COSMAC is shown in Appendix A.

Memory Read

Suppose you wish to examine the contents of a memory location. Flip RUN up while pressing key C. Enter the address of the location you want to examine. Press key A for the Memory Read mode. You will see the memory address and the byte stored at that address on the screen. Press any hex key to step through memory and see the contents. Memory locations examined are left unchanged. If a program doesn't run properly you can use this mode to verify that it was stored correctly in memory.

You can now enter and run the short beeper program shown in Appendix A. Flip RUN up with key C pressed. Release key C and enter address 0000. Press key 0 to select the Memory Write mode. Now enter the beeper program one byte at a time using the hex keyboard. Flip RUN down to reset the computer. Flip RUN up to execute the beeper program you just loaded into locations 0000-000C. You can load and run any COSMAC VIP program in this manner. For most of the game programs you will first have to load the CHIP-8 interpreter (Appendix C) into locations 0000-01FF followed by the game program starting at location 0200.

Tape Write

Any program you load into memory will be lost when you turn off power. Unless it is safely stored, you will have to key it in by hand again the next time you want to use it. The cassette interface is provided so that after keying in a program you can then record it on an audio cassette; and when you want to use the program again, all you have to do is play it back into the memory from the cassette. This playback usually takes less than 30 seconds.

The COSMAC VIP cassette interface was designed to work with most standard audio cassette recorders. Panasonic models RQ-309DS, RQ-212D, and RQ-413S have yielded satisfactory results as has the Sony TC-150. In general, better quality recorders provide more reliable operation.

Your tape recorder must have an 8-ohm earphone or external speaker jack and a microphone input jack. Connect the cassette recorder to the COSMAC VIP tape-in tape-out pads on the right-hand side of the card as shown in the cassette attachment diagram in Appendix A.

After properly connecting your cassette recorder you can try recording and playing back a cassette using the operating system as described below. Follow the cassette recording guidelines provided in Appendix A for best results. If you run into trouble, use the cassette phase and data test procedures described in Appendix A for troubleshooting.

The memory is divided into 256-byte pages for recording. You can record 1 to 15 consecutive pages on tape. The low-order byte of your starting address should be 00. Select the operating system by holding key C down while flipping RUN up. Enter the 4-digit address of the first page to be recorded on tape. Press key F and you're ready to record. Rewind a blank cassette and place your cassette unit in the record mode. Wait about 10 seconds and tap the hex key that represents the number of pages you want to record on tape. The screen will go blank and you'll hear a tone while recording. When the specified number of pages has been recorded on the cassette, the tone will end and the last memory byte recorded on tape will be shown on the screen.

Tape Read

To load memory from a previously recorded cassette, first select the operating system (RUN and key C). Enter the memory address of the first page to be loaded (usually 0000). Press key B to select the Tape Read mode. Rewind and play the cassette. Immediately press the hex key representing the number of pages you want to load into memory from the cassette. The tape recorder tone control should be set to maximum high. The volume control should be set for a steadily glowing tape light when data is being read from the tape. The screen will go blank while the program is loaded from the tape into memory. It will show the last byte loaded into memory at the end of loading.

If the Q light and tone come on while a tape is being read, an error occurred. Flip RUN down, rewind the cassette, and try again. You may have to readjust the cassette volume control. Be sure that the cassette contains at least as many pages as you specify to be loaded. For most of the game programs, load the CHIP-8 interpreter program (Appendix C) into 0000-

01FF, then load the game program starting at 0200. Record a cassette from 0000 to the end of the game program. When you load this tape, starting at 0000, you will be ready to play the game.

Testing Your Cassette System

Test your cassette system by entering the beeper program at 0000 (Appendix A). Store 25 at 06FF. Now record 7 pages on a cassette starting at 0000. Load these 7 pages back into memory from the cassette starting at 0000. If no errors occur you should see "06FF 25" on the screen after loading is complete. Flip RUN down, then up, and the beeper program should be running.

After recording and checking a program cassette, you can break out the tabs at the top of the cassette to prevent accidental erasure. In the event you wish to record on a cassette after you have broken out the tabs, you can do so simply by pasting tape over the tab holes. You can record and keep your own cassette software library starting with the game programs in Appendix D. Cassette recording or playback should require 5 + 2.5N seconds. N is the number of pages recorded on tape. Recording or loading the entire 2048-byte RAM (8 pages) will require less than 30 seconds. The next section describes how you can design your own programs using a unique easy-to-learn programming language called CHIP-8.

III. CHIP-8 Language Programming

CHIP-8 is an easy-to-learn programming language that lets you write your own programs. To use the CHIP-8 language, you must first store the 512-byte CHIP-8 language program at memory locations 0000 to 01FF. The CHIP-8 language program is shown in Appendix C in hex form so you can enter it directly in memory using the hex keyboard. You can then record it on a memory cassette for future use. Each CHIP-8 instruction is a two-byte (4-hex-digit) code. There are 31, easy-to-use CHIP-8 instructions as shown in Table I.

When using CHIP-8 instructions your program must always begin at location 0200. There are 16 one-byte variables labeled 0-F. VX or VY refers to the value of one of these variables. A 63FF instruction sets variable 3 to the value FF (V3=FF). I is a memory pointer that can be used to specify any location in RAM. An A232 instruction would set I=0232. I would then address memory location 0232.

Branch Instructions

There are several types of jump or branch instructions in the CHIP-8 language. Instruction 1242 would cause an unconditional branch to the instruction at memory location 0242. Instruction BMMM lets you index the branch address by adding the value of variable 0 to it before branching. Eight conditional skip instructions let you test the values of the 16 one-byte variables or determine if a specific hex key is being pressed. This latter capability is useful in video game programs. (Only the least significant hex digit of VX is used to specify the key.)

A 2570 instruction would branch to a subroutine starting at location 0570. 00EE at the end of this subroutine will return program execution to the instruction following the 2570. The subroutine itself could use another 2MMM instruction to branch to (or call) another subroutine. This technique is known as subroutine nesting. Note that all subroutines called (or branched to) by 2MMM instructions must end with 00EE. **Ignoring this rule will cause hard-to-find program bugs.**

How to Change and Use the Variables

The CXKK instruction sets a random byte value into VX. This random byte would have any bits matching 0 bit positions in KK set to 0. For example, a C407 instruction would set V4 equal to a random byte value between 00 and 07.

A timer (or real-time clock) can be set to any value between 00 and FF by a FX15 instruction. This timer is automatically decremented by one, 60 times per second until it reaches 00. Setting it to FF would require about 4 seconds for it to reach 00. This timer can be examined with a FX07 instruction. A FX18 instruction causes a tone to be sounded for the time specified by the value of VX. A value of FF would result in a 4-second tone. The minimum time that the speaker will respond to is that corresponding to the variable value 02.

A FX33 instruction converts the value of VX to decimal form. Suppose I=0422 and V9=A7. A F933 instruction would cause the following bytes to be stored in memory:

```
0422  01
0423  06
0424  07
```

Since A7 in hex equals 167 in decimal, we see that the

Table I – CHIP-8 Instructions

Instruction	Operation
1MMM	Go to 0MMM
BMMM	Go to 0MMM + V0
2MMM	Do subroutine at 0MMM (must end with 00EE)
00EE	Return from subroutine
3XKK	Skip next instruction if VX = KK
4XKK	Skip next instruction if VX ≠ KK
5XY0	Skip next instruction if VX = VY
9XY0	Skip next instruction if VX ≠ VY
EX9E	Skip next instruction if VX = Hex key (LSD)
EXA1	Skip next instruction if VX ≠ Hex key (LSD)
6XKK	Let VX = KK
CXKK	Let VX = Random Byte (KK = Mask)
7XKK	Let VX = VX + KK
8XY0	Let VX = VY
8XY1	Let VX = VX/VY (VF changed)
8XY2	Let VX = VX & VY (VF changed)
8XY4	Let VX = VX + VY (VF = 00 if VX + VY ⩽ FF, VF = 01 if VX + VY > FF)
8XY5	Let VX = VX – VY (VF = 00 if VX < VY, VF = 01 if VX ⩾ VY)
FX07	Let VX = current timer value
FX0A	Let VX = hex key digit (waits for any key pressed)
FX15	Set timer = VX (01 = 1/60 second)
FX18	Set tone duration = VX (01 = 1/60 second)
AMMM	Let I = 0MMM
FX1E	Let I = I + VX
FX29	Let I = 5-byte display pattern for LSD of VX
FX33	Let MI = 3-decimal digit equivalent of VX (I unchanged)
FX55	Let MI = V0 : VX (I = I + X + 1)
FX65	Let V0 : VX = MI (I = I + X + 1)
00E0	Erase display (all 0's)
DXYN	Show n-byte MI pattern at VX-VY coordinates. I unchanged. MI pattern is combined with existing display via EXCLUSIVE-OR function. VF = 01 if a 1 in MI pattern matches 1 in existing display.
0MMM	Do machine language subroutine at 0MMM (subroutine must end with D4 byte)

three RAM bytes addressed by I contain the decimal equivalent of the value of V9.

If I=0327, a F355 instruction will cause the values of V0, V1, V2, and V3 to be stored at memory locations 0327, 0328, 0329, and 032A. If I=0410, a F265 instruction would set V0, V1, and V2 to the values of the bytes stored at RAM locations 0410, 0411, and 0412. FX55 and FX65 let you store the values of variables in RAM and set the values of variables to RAM bytes. A sequence of variables (V0 to VX) is always transferred to or from RAM. If X=0, only V0 is transferred.

The 8XY1, 8XY2, and 8XY4, and 8XY5 instructions perform logic and binary arithmetic operations on two 1-byte variables. VF is used for overflow in the arithmetic operations.

Using the Display Instructions

An 00E0 instruction erases the screen to all 0's. When the CHIP-8 language is used, 256 bytes of RAM are displayed on the screen as an array of spots 64 wide by 32 high. A white spot represents a 1 bit in RAM, while a dark (or off) spot represents a 0 bit in RAM. Each spot position on the screen can be located by a pair of coordinates as shown in Fig. 1.

The VX byte value specifies the number of horizontal spot positions from the upper left corner of the display. The VY byte value specifies the number of vertical spot positions from the upper left corner of the display.

The DXYN instruction is used to show a pattern of spots on the screen. Suppose we wanted to form the

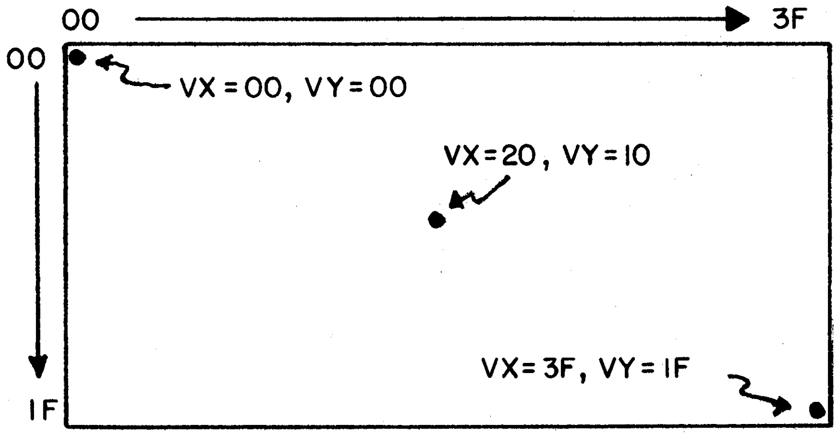


Fig. 1 — Display screen coordinate structure.

pattern for the digit "8" on the screen. First we make up a pattern of bits to form "8" as shown in Fig. 2.

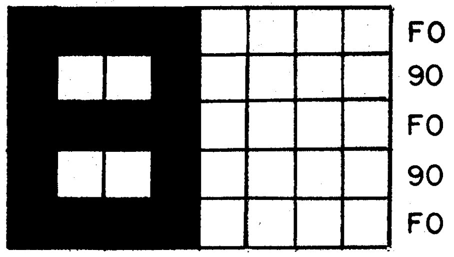


Fig. 2 — Pattern of bits forming digit 8.

In this example we made the "8" pattern five spots high by four spots wide. Patterns to be shown on the screen using the DXYN instruction must always be one byte wide and no more than fifteen bytes high. (Several small patterns can be combined to form larger ones on the screen when required). To the right of the "8" pattern in Fig. 2 are the equivalent byte values in hex form. We could now store this pattern as a list of five bytes at RAM location 020A as follows:

```
020A  F0
020B  90
020C  F0
020D  90
020E  F0
```

Suppose we now want to show this pattern in the upper left corner of the screen. We'll assign V1=VX and V2=VY. Now we let V1=V2=00 and set I=020A. If we now do a D125 instruction, the "8" pattern will be shown on the screen in the upper left corner.

You can write a program to show the "8" pattern on the screen as follows:

```
0200 A20A I=020A
0202 6100 V1=00
0204 6200 V2=00
0206 D125 SHOW 5MI@V1V2
0208 1208 GO 0208
020A F090
020C F090
020E F000
```

The first column of this program shows the memory locations at which the instruction bytes in the second column are stored. The third column indicates the function performed by each instruction in shorthand form. Only the bytes in the second column are actually stored in memory.

With the CHIP-8 interpreter stored at 0000-01FF, you can load the above program in memory and run it. Set V1 and V2 to different values to relocate the "8" pattern on the screen. The VX-VY coordinates always specify the screen position of the upper left-hand bit of your pattern. This bit can be either 0 or 1. The last digit of the DXYN instruction specifies the height of your patterns or the number of bytes in your pattern list.

When a pattern is displayed, it is compared with any pattern already on the screen. If a 1 bit in your pattern matches a 1 bit already on the screen, then a 0 bit will be shown at this spot position and VF will be set to a value of 01. You can test VF following a DXYN instruction to determine if your pattern touched any part of a previously displayed pattern. This feature permits programming video games which require knowing if one moving pattern touches or hits another pattern.

Because trying to display two 1 spots at the same position on the screen results in a 0 spot, you can use the DXYN instruction to erase a previously displayed pattern by displaying it a second time in the same position. (The entire screen can be erased with a single 00E0 instruction.) The following program shows the "8" pattern, shows it again to erase it, and then changes VX and VY coordinates to create a moving pattern:

```
0200 A210 I=0210
0202 6100 V1=00
0204 6200 V2=00
0206 D125 SHOW 5MI@V1V2
0208 D125 SHOW 5MI@V1V2
020A 7101 V1+01
020C 7201 V2+01
020E 1206 GO 0206
0210 F090
0212 F090
0214 F000
```

The "8" pattern byte list was moved to 0210 to make room for the other instructions. Try changing the values that V1 and V2 are incremented by for different movement speeds and angles. A delay could be inserted between the two DXYN instructions for slower motion.

The FX29 instruction sets I to the RAM address of a five-byte pattern representing the least significant hex digit of VX. If VX=07, then I would be set to the address of a "7" pattern which could then be shown on the screen with a DXYN instruction. N should always be 5 for these built-in hex-digit patterns. Appendix C shows the format for these standard hex patterns. The following program illustrates the use of the FX29 and FX33 instructions:

```
0200 6300 V3=00
0202 A300 I=0300
0204 F333 MI=V3(3DD)
0206 F265 V0:V2=MI
0208 6400 V4=00
020A 6500 V5=00
020C F029 I=V0(LSDP)
020E D455 SHOW 5MI@V4V5
0210 7405 V4+05
0212 F129 I=V1(LSDP)
0214 D455 SHOW 5MI@V4V5
0216 7405 V4+05
0218 F229 I=V2(LSDP)
021A D455 SHOW 5MI@V4V5
021C 6603 V6=03
021E F618 TONE=V6
0220 6620 V6=20
0222 F615 TIME=V6
0224 F607 V6=TIME
0226 3600 SKIP;V6 EQ 00
0228 1224 GO 0224
022A 7301 V3+01
022C 00E0 ERASE
022E 1202 GO 0202
```

This program continuously increments V3, converts it to decimal form, and displays it on the screen.

The FX0A instruction waits for a hex key to be pressed, VX is then set to the value of the pressed key, and program execution continues when the key is released. (If key 3 is pressed, VX=03). A tone is heard while the key is pressed. This instruction is used to wait for keyboard input.

Applying CHIP-8

You should now be able to write some simple CHIP-8 programs of your own. Here are some things to try:

1. Wait for a key to be pressed and show it on the display in decimal form.

2. Show an 8-bit by 8-bit square on the screen and make it move left or right when keys 4 or 6 are held down.

3. Show an 8-bit square on the screen. Make it move randomly around the screen.

4. Show a single bit and make it move randomly around the screen leaving a trail.

5. Program a simple number game. Show 100 (decimal) on the screen. Take turns with another player. On each turn you can subtract 1-9 from the number by pressing key 19. The first player to reach 000 wins. The game is more interesting if you are only allowed to press a key which is horizontally or vertically adjacent to the last key pressed.

If you are unsure of the operation of any CHIP-8 instruction, just write a short program using it. This step should clear up any questions regarding its operation. In your CHIP-8 programs be careful not to write into memory locations 0000-01FF or you will lose the CHIP-8 interpreter and will have to reload it. You can insert stopping points in your program for debugging purposes. Suppose you want to stop and examine variables when your program reaches the instruction at 0260. Just write a 1260 instruction at location 0260. Flip RUN down and use operating system mode A to examine variables V0-VF. The memory map in Appendix C shows where you can find them.

After the above practice you are ready to design more sophisticated CHIP-8 programs. **Always prepare a flowchart before actually writing a program.** The last 352 bytes of on-card RAM are used for variables and display refresh. In a 2048-byte RAM system you can use locations 0200-069F for your programs. This area is enough for 592 CHIP-8 instructions (1184 bytes). In a 4096-byte RAM system you can use locations 0200-0E8F. This area is equal to 1608-CHIP-8 instructions (3216 bytes).

Some Program Ideas

Here are a few ideas for programs to write using the CHIP-8 language:

1. INTOXICATION TESTER - Display a six-digit random number on the screen for several seconds. You must remember this number and enter it from the keyboard within ten seconds after the screen goes blank to prove that you're sober and score.

2. NUMBER BASE QUIZ - Display numbers in binary or octal on the screen. You must enter their decimal equivalent to score points.

3. DICE - Push any key to simulate rolling dice displayed on the screen.

4. PUPPETS - Show large face on the screen. Let small children move mouth and roll eyes by pushing keys.

5. BUSY BOX - Let small children push keys to make different object appear on the screen, move, and make sounds.

6. SHUFFLEBOARD - Simulate shuffleboard-type games on the screen.

7. COMPUTER ART - Design new programs to generate pleasing geometric moving patterns on the screen.

8. INVISIBLE MAZE - Try to move a spot through an invisible maze. Tones indicate when you bump into a wall.

9. LUNAR LANDING - Program a graphic lunar landing game.

10. COLLIDE - Try to maneuver a spot from one edge of the screen to the other without hitting randomly moving obstacles.

11. CAPTURE - Try to chase and catch randomly moving spots within a specified time limit.

12. LEARNING EXPERIENCES - Program graphic hand and eye coordination exercises for young children or those with learning disabilities.

13. NUMBER RECOGNITION - Show groups of objects or spots on the screen. Young child must press key representing number of objects shown to score.

14. WALL BALL - Program a wall-ball-type paddle game for one player.

15. FOOTBALL - Each player enters his play via the hex keyboard and the computer moves the ball on the screen.

16. BLACKJACK - Play "21" against the computer dealer.

17. HOLIDAY DISPLAYS - Design custom, animated displays for birthdays, Halloween, Christmas, etc.

18. METRIC CONVERSION - Help children learn metric by showing lengths on screen in inches and requiring centimeter equivalent to be entered to score.

19. TURING MACHINE - Simulate a simplifed Turing machine on the screen.

20. TIMER - Use the computer to time chess games, etc.

21. HEXAPAWN - Program Hexapawn so that the computer learns to play a perfect game.

22. NIM - Program Nim with groups of spots shown on the screen.

23. BLOCK PUZZLES - You can simulate a variety of sliding block-type puzzles on the screen.

24. BOMBS AWAY - Show a moving ship at the bottom of the screen. Try to hit the ship by releasing bombs from a moving plane at the top of the screen.

25. PROGRAMMED SPOT - Introduce children to programming concepts by letting them preprogram the movements of a spot or object on the screen.

The next section will discuss machine language programming. You can even combine machine language subroutines with CHIP-8 programs if desired.

IV. Machine Language Programming

VIP Machine Coding

For a complete description of machine language instructions, refer to the **User Manual for the CDP1802 COSMAC Microprocessor MPM-201A.** Your COSMAC VIP computer incorporates the following special machine-language input and output instructions:

CODE	OPERATION
69	Turn display on (Bus → MX,D)
6B	Input port byte → MX,D (Optional)
61	Turn display off (MX → Bus,RX+1)
62	MX(LSD) → Hex keyboard latch, RX+1
63	MX → Output port, RX+1 (Optional)
64	MX → Bus, RX+1

One 64 instruction is always executed by the Operating System. It can also be used in expanded systems if desired. Instructions 65, 66, 67, 6A, 6C, 6D, 6E, and 6F are also available for use in expanded systems.

The External Flag lines are used as follows:

FLAG	USE
EF1	Generated by the video interface (CDP1861)
EF2	Serial data from cassette player
EF3	Hex key pressed signal
EF4	Not used in basic system

EF4 can be used for system expansion. EF3 can also be used in expanded systems if no key will be depressed at the same time that an external device is using EF3. EF1 can only be used by an external device when the display is turned off. EF2 should not be used in expanded systems.

The latched Q line output performs several functions in the COSMAC VIP system. When set, it holds the Q light on and generates a continuous speaker tone. The Q line is also used for serial output data to a cassette recorder. You can use the Q output line as a control signal in an expanded system if you avoid conflicts with its normal functions.

You can store a machine language program starting at location 0000. It will be executed when you flip the RUN switch up. Initially P=0, X=0, R0=0000, Q=0, and R1=0XFF, where 0X= last page of on-card RAM. (0X=07 in 2048-byte RAM system). The operating system uses the last 84 bytes of on-card RAM. You should avoid using these last 84 RAM bytes when writing machine language programs. With a 2048-byte RAM, locations 07AC-07FF would be reserved for use by the operating system. Note that R1 initially contains the address of the last on-card RAM byte. Your machine language program can use R1 to determine the amount of RAM in your system when required.

Putting Machine Coding and CHIP-8 Language Together

The operating system and the CHIP-8 language interpreter use a video display format that is 64 bits wide by 32 bits high. This 256-byte display can easily be modified by writing your own video refresh interrupt routine as explained in the CDP1861 data sheet provided in Appendix G. Display formats up to 64 bits wide by 128 bits high are possible with no hardware modification. Th 4096-bit picture program in Appendix D uses a machine language refresh interrupt routine that provides a format 64 bits wide by 64 bits high.

The CHIP-8 language described in the previous section, permits machine language subroutines to be called with a 0MMM instruction. A D4 machine language instruction at the end of the machine language subroutine returns control to the CHIP-8 instruction following the 0MMM instruction. In Appendix C, the CDP1802 register use for the CHIP-8 language is provided. R5 is used as the CHIP-8 program counter. When you call a machine language subroutine with a 0MMM instruction, R5 will be addressing the CHIP-8 instruction following the 0MMM. The machine language subroutine could retrieve the next two CHIP-8 program bytes as parameters by addressing with R5 and incrementing it by 2 before returning control to the CHIP-8 program with a D4 instruction. RC, RD, RE, and RF are available for use in machine language subroutines. RA is the CHIP-8 memory pointer (I). Changing the high-order byte of RB will cause any desired RAM page to be displayed. R3 is the machine language subroutine program counter.

CHIP-8 uses the operating system refresh interrupt routine contained in ROM for display. You can use this ROM interrupt routine for 256-byte display in your own machine language programs. First initialize R1 to 8146 and R2 as a stack pointer before turning on the video interface with a 69 instruction. Set the desired display page into RB.1. This interrupt routine uses R0 as the display refresh pointer and modifies RB.0. R8.1 and R8.0 are decremented by 1 during each interrupt unless they are equal to 00. Interrupts occur 60 times per second when the video interface is turned on. This rate is controlled by a crystal clock so that R8.0 and R8.1 can be used as real-time clocks when needed.

While the video interface is turned on, you should not use any of the 3-machine-cycle CDP1802 instructions (except those used for sync in the refresh interrupt routine itself). If you are not using the video interface, then you can use the CDP1802 3-cycle instructions in your machine language programs. When you initiate a machine language program at 0000 by flipping RUN up, the video interface will be off. You must turn it on with a 69 instruction to use the COSMAC VIP graphic display capability.

Machine Language Programming Summed Up

In summary, COSMAC VIP provides you with an easy-to-use language called CHIP-8. You can insert machine language subroutines in CHIP-8 programs for greater flexibility or expanded I/O capability. You can write complete machine language programs to fully utilize CDP1802 capabilities. The operating system facilitates debugging machine language programs by permitting you to examine general registers R3-RF. (See operating system register table in Appendix B). Advanced programmers can even develop their own interpretive language tailored to special requirements. Direct execution of machine language code starting at location 0000 together with the expansion interface permits the COSMAC VIP system to be used as a low-cost development system as well as a personal recreational or educational computer.

V. Logic Description

A complete set of logic diagrams is provided in Appendix E. Power requirements for a system with 2048 bytes of RAM is 5 V DC at 350 mA. If you wish to expand the system you can use your own higher-current power supply.

This system is designed around the CDP1802 microprocessor (U1). Refer to the CDP1802 data sheet and User Manual for the CDP1802 COSMAC Microprocessor MPM-201A for a complete description of its operation. The CDP1802 requires a square-wave clock input at pin 1 for operation. This system uses a 1.7609-MHz clock. One half of U3 is connected as a free-running crystal-controlled oscillator. A 3.52180-MHz crystal is used in this circuit. The output of this 3.52180-MHz oscillator is then divided by 2 using U4 to provide the 1.7609-MHz input clock for the CDP1802. Because each CDP1802 machine cycle equals 8 clock cycles, each machine cycle is about 4.54 μs in duration. TPA and TPB are timing pulses generated once each machine cycle by the CDP1802 microprocessor.

How Memory Is Addressed

A debounced RUN level goes high when the RUN switch is flipped up. This signal causes the CDP1802 to begin fetching instructions from memory. When the RUN switch is down, the CDP1802 is held in a reset state and U6A (in Fig. E-2) is reset. U6B is held set by U6A. The CDP1802 starts fetching instructions from the ROM (U10) at location 8000 since U6B is being held set. The ROM contains the operating system program which uses a 64 instruction to generate an N2 pulse. This N2 pulse sets U6A so it no longer holds U6B in its set state. From this point on, the selection of RAM or ROM locations is controlled by the most significant address bit latched into U6B each cycle by TPA.

U8 latches an additional 4 address bits to provide the 12-bit address required in a 4096-byte RAM system. U9A decodes 2 of these address bits into 4 lines which are used to select up to four 1024-byte RAM sections. Each 1024-byte section of RAM consists of two 4 x 1024-bit RAM IC's (U16-U23 in Fig. E-4). Only the first two sections of RAM (U16-U19) are used in a 2048-byte system. U9B in Fig. E-2 is wired as a simple gate that inhibits selecting any section of RAM when either the ROM is selected or a positive RAM inhibit signal is generated on pin 19 of the expansion interface by external circuits.

Memory read ($\overline{\text{MRD}}$) and write ($\overline{\text{MWR}}$) signals are supplied to the RAM at appropriate times by the CDP1802. Data is transferred between memory, CDP1802, input, or output via an 8-bit data bus. Pull-up resistors are provided on this bus for compatibility with TTL signal swings provided by some RAMs.

How the Input/Output Works

U11 and U12 in Fig. E-3 are used to decode the input/output instruction codes used in the system.

U13 provides the hex keyboard interface. This interface permits a program to determine which key is pressed. A 62 machine instruction causes the least significant 4 bits of memory byte to be latched into U13. These 4 bits are decoded to bring one of the 16 U13 output lines low. If the key that corresponds to this output line is pressed, the CDP1802 EF3 input will go low. The 4-bit codes latched into U13 correspond to the equivalent key position. After the program sends a 4-bit code to U13, it subsequently examines the EF3 line to see if the key corresponding to this code is pressed or not. In this manner, a program can determine when any specific key is pressed or can sequentially scan all keys while waiting for any one to be pressed. Key debounce delays must be provided in the program when required. A program can also cause a speaker tone to occur when a key is pressed. Only one key at a time should be pressed with this method of interfacing the keyboard.

U15 generates an audible tone when pin 4 is high. The output on pin 3 drives a small speaker. The 10-ohm resistor R48 in series with the speaker output can be raised in value to lower the volume if desired. The CDP1802 latched Q-line output drives the tone generator and also turns on the Q light. Q can be set high (1) or low (0) by machine language instructions. The RC network connected to pins 2, 6, and 7 of U15 determines the frequency of the tone. You can increase or decrease the value of R to adjust this frequency to suit your taste.

Q is also shaped by U14A in Fig. E-3 to form a signal suitable for recording on an audio cassette. Audio cassette recorders can't cope with square waves. The divider on the output of U14A reduces the signal to about 50 mV which is suitable for the microphone input of most recorders. During recording, the operating system program in ROM converts memory bytes into bit serial form and transmits them to the recorder via the Q line. See the cassette data test page of Appendix A for the cassette data code used.

In playback, bit serial data from the cassette drives the tape light. The serial data is amplified and shaped into 5-volt pulses by U14B. The output of U14B is connected to the CDP1802 EF2 input line. The operating system reads tape data by examining the timing of the transitions on the EF2 input line. Cassette read and record timing is derived from the crystal-controlled clock so that no adjustments are necessary.

Video output is provided by the unique CDP1861 video display interface IC (U2 in Fig. E-1). Refer to the CDP1861 data sheet in Appendix G for a description of its operation. This chip provides one of the lowest cost and most useful display interface capabilities available for any microcomputer. The values of the resistors R1 and R4 in Fig. E-1 of Appendix E connected to output pins 6 and 7 of U2 can be adjusted for best results with your video display. 61 and 69 machine language instructions are used to generate the required on and off pulses for U2. The down position of the RUN switch resets the internal U2 circuits. When a program is initiated, by flipping RUN up, U2 will remain off until a 69 instruction is executed. No CDP1802 interrupt or DMA requests are generated by U2 until it is turned on by a 69 instruction. U1 and U2 are both driven by the same clock. They must remain in sync to provide proper operation of the display.

In general, the logic of this system has been kept simple and straight-forward by the use of software to replace hardware. This design not only yields a low-cost system, but one that should prove extremely reliable because of the reduced number of components that can cause failures. This system will not become obsolete for a long time. RAM, ROM, and microprocessor are all state-of-the-art devices and not obsolescent types that are about to be replaced by better ones. The cassette and video interfaces are optimum for long life. Also designed into the system are full expansion capability for added RAM, ROM, input, output, and full color graphics.

VI. Expansion Considerations and Connections

The COSMAC VIP was designed primarily as a self-contained graphic system for home use. Enough RAM and input/output features are provided for years of computer fun without adding anything to your system. If, however, you do want to expand your system, a variety of features have been included to make expansion as easy and inexpensive as possible. You can easily increase RAM to 4096 bytes by adding U20-U23 to your PC card. Use the same type or a compatible type of RAM as used for U16-U19. You may, however, have to add a higher-current power supply when expanding RAM.

Using the Byte Input/Output

First, you may wish to add some external computer-controlled devices such as relays, input sensing switches, or even a low-cost printer. The printer will require an 8-bit parallel input or output port and some "hand-shaking" signals. One parallel input port and one parallel output port are available on the PC card as shown in Fig. E-5 in Appendix E. These ports are provided by U24, U25, U26, and U27 along with the associated resistors and two 1N914 diodes. The 22 input/output port connection pads (A-Z) along the back right edge of the PC card are connected to a standard 44-pin card socket on the COSMAC VIP board. You can plug your external circuits or devices into this socket. Table II gives the input/output port terminal connections.

The 8 buffered output signals (M,N,P,R,S,T,U,V) will each drive up to 2 TTL loads. A 63 machine language instruction will latch a memory byte into U24 for output. The 8 latched output lines can be used to drive individual relay driver circuits, power amplifiers, lights, battery motor drivers, etc. The buffered Q output line (W) can be used as an output strobe for transferring the latched output byte to an external device such as a printer. The EF3 (X) and EF4 (L) input lines can be used to indicate the status of an external device. Don't forget that EF3 is shared with the hex keyboard.

Table II – Input/Output Port Terminal Connections (See Fig. E-5, Appendix E)

Pin	Signal	Description
A	IN 0	8-bit input bus
B	IN 1	
C	IN 2	
D	IN 3	
E	IN 4	
F	IN 5	
H	IN 6	
J	IN 7	
K	$\overline{\text{INST}}$	Input byte strobe to latch U25
L	$\overline{\text{EF4}}$	Input flag line #4
M	OUT 0	8-bit output bus
N	OUT 1	
P	OUT 2	
R	OUT 3	
S	OUT 4	
T	OUT 5	
U	OUT 6	
V	OUT 7	
W	$\overline{\text{Q}}$	Q flip-flop output line
X	$\overline{\text{EF3}}$	Input flag line #3 (also used for hex keyboard)
Y	+5 V	Optional power for external logic
Z	GND	

A single photocell input could be provided via the buffered EF4 line. You can attach the photocell directly between the L and Z pads. Experimentally adjust the pull-up resistor on pad L for best operation. No photocell amplifier should be required to drive the COS/MOS input. An externally supplied positive pulse on pins 2 and 14 of U25 can be used as an input byte strobe when you want to latch an input byte into U25. A 68 instruction can be used to store this input byte in RAM.

Using the Expansion Interface

The 44-pin card socket for the expansion interface pads along the back left edge of the PC board permits extensive expansion. If you expand beyond the capabilities of the power converter provided with the VIP, you will, of course, have to provide your own power supply. Output signals should only drive COS/MOS loads and must be externally buffered with a CD4050 or CD4049 IC to drive TTL loads. Keep any wires connected to the expansion pad signals as short as possible. Excessive stray capacitance on these signal lines can interfere with proper operation of the computer. Input signals should also be buffered with COS/MOS circuits. Refer to the machine language programming section (Section IV) and the logic diagrams (Appendix E) to avoid conflicts with normal COSMAC VIP use of these signals. The external option terminal connections are given in Table III.

You can latch up the required high order address bits with the trailing edge of TPA when adding external memory. You must provide a positive level on pad 19 to disable internal RAM when external RAM is addressed. The operating system will always use the highest page of internal (on-card) RAM, even when you add external RAM.

If you wish to substitute an external ROM or battery-powered COS/MOS RAM for U10, you can use the signal on pad X to select it. Remove U10 when substituting an external ROM. If you do use an external ROM for your own operating system you may no longer be able to use the CHIP-8 interpreter because it requires some of the operating system subroutines.

The expansion interface pads provide access to all CDP1802 signals so that you can add any desired external circuits.

Only 5 out of the possible 14 CDP1802 input/output instructions are used internally, so that you can externally decode the N0, N1, and N2 lines and use them with MRD to obtain the use of the remaining 9 input/output instruction codes. You can also latch high-order address bits to select external devices if desired. When using external circuits to generate DMA requests, interrupt requests, or input flag signals, isolate these signals with 1N914 diodes as shown for EF3 and EF4 in the optional parallel input/ output port logic. Refer to the User Manual for the CDP1802 COSMAC Microprocessor, MPM-201A, for specific examples of input/output attachment techniques.

Some Expansion Ideas

The August and September 1976 issues of Popular Electronics contain descriptions of a COSMAC ELF microcomputer using the CDP1802. These articles illustrate some input/output attachment techniques.

The following lists some things that with some exercise of your ingenuity could be added to your system at relatively low cost:

1. Manually operated photoelectric paper-tape strip reader. Only requires a tape guide and 8 photocells.
2. Scanning circuit for multiple input lines from sensing devices using CD4515 IC.
3. Full alphanumeric keyboard.
4. Low-cost printer.
5. Multi-digit numeric display.
6. Calculator chip.
7. Individual photocells or switches.
8. Output relays to control solenoids, bells, whistles, sirens, lights, or motors.
9. Sound-generating circuits that can be controlled by program.
10. Analog-to-digital input circuits.
11. Read-Only Memory for fixed program.
12. Digital-to-analog output circuits.
13. Alpha wave monitor input to control pictures on TV or output devices.
14. Temperature- or pressure-sensing devices.
15. Computer terminal.
16. A second hex keyboard for multi-player video games.

Table III – External Option Terminal Connections (See Fig. E-2, Appendix E)

Pin	Signal	Description
A	$\overline{\text{MWR}}$	Negative-going memory-write pulse
B	TPA	Early timing pulse for M address clocking, etc.
C	MA0	Memory address lines. High-order address byte appears on these lines during TPA time, followed by low-order address byte
D	MA1	
E	MA2	
F	MA3	
H	MA4	
J	MA5	
K	MA6	
L	MA7	
M	BUS 0	8-bit, 2-way tri-state data bus
N	BUS 1	
P	BUS 2	
R	BUS 3	
S	BUS 4	
T	BUS 5	
U	BUS 6	
V	BUS 7	
W	$\overline{\text{MRD}}$	Low for memory read machine cycles
X	CS	Chip select for operating system
Y	+5 V	Optional power for external logic
Z	GND	
1	CLOCK	CDP1802 clock output
2	$\overline{\text{EF4}}$	Flag input lines #3 and #4
3	$\overline{\text{EF3}}$	(Flag 3 also used for hex keyboard)
4	XTAL	Crystal frequency
5	$\overline{\text{EF1}}$	Flag input line #1
6	N0	Low-order 3 bits of N during 6N instruction
7	N1	
8	N2	
9	SPOT	Video spot output
10	$\overline{\text{SYNC}}$	Video sync output
11	TPB	Timing pulse for clocking memory byte out, etc.
12	SC0	State code bit (+5 V for S1/S3, GND for S0/S2)
13	$\overline{\text{INTERRUPT}}$	Pulling to GND causes interrupt (22-KΩ input)
14	SC1	State code bit (+5 V for S2/S3, GND for S0/S1)
15	$\overline{\text{DMA-OUT}}$	Pull to GND for DMA-OUT cycles
16	Q	Q flip-flop output line
17	$\overline{\text{DMA-IN}}$	Pull to GND for DMA-IN cycles
18	RUN	+5 V when running, GND when RUN switch down
19	INDIS	Internal RAM-disable input
20	$\overline{\text{CDEF}}$	GND when RAM pages C, D, E, and F selected
21	+5 V	Optional power for external logic (same as Y-Z)
22	GND	

Some possible applications for expanded systems include:

1. Counting packages, parts, cars, or people via photocell or switch input.
2. Composing poetry or pictures with printer output.
3. Video target games using photocell light gun.
4. Monitor burglar alarm switches.
5. Monitor water level and temperature in fish tank and regulate automatically.
6. Measure motor speed with photocell.
7. Monitor and control experiments in home, school, or lab. Use video display for real time bar graphs of multiple variables.
8. Provide a crystal-controlled, programmable pulse generator, clock, or timer.
9. Provide a programmable sequencer for light shows, advertising displays, holiday lighting, etc.
10. Automatic telephone dialer.
11. Model railroad controller.
12. Battery-operated toy or robot controller.
13. Detect tape-player tones and control slide projector.

You will soon discover that the potential applications of a computer such as the COSMAC VIP are only limited by your imagination and the ability to develop appropriate interface circuits.

VII. Troubleshooting Hints

This section is aimed at helping you diagnose and fix hardware problems should they occur. First, check all IC's to make sure they are properly inserted in the PC card. An IC inserted in the wrong direction can be permanently damaged. Check that the +5 V DC supply voltage ripple does not exceed 0.2 volt. Visually inspect the PC card for solder shorts or bad solder joints. Try to avoid zapping your PC card with static electricity charges. Discharge youself, if necessary, by touching a grounded object before touching any IC's or PC card wiring.

No Sound

If everything works but you don't hear any sound from the speaker you probably have a bad U15, bad speaker, or bad connection. Flip RUN up with key C down. Hold any key down and the Q light should come on. Check the Q line if it doesn't. The Q line should be at +5 V with a key held. If the Q light is on, but with no tone, check U15 and your speaker connections.

No Display

If you get no display but do get operating system key tones, check the video output signal. First, select the operating system to make sure video should be present. The video signal should be 0.5 volt peak to peak or higher. You should see negative-going vertical and horizontal sync pulses and positive-going video pulses. The sync pulses should be about 25% of the total swing. Check your display system and interconnections if you have the video signal present. Make sure you are using the correct high-impedance input setting, for example.

Other Problems

Using operating system mode 0, load bytes into RAM using all 16 hex keys. If a key doesn't work or shows the wrong value on the display screen, check the keyboard and U13.

If everything except the cassette interface works, check U14. Review the cassette recording guidelines in Appendix A. Use the cassette phase and data test procedures described in Appendix A to find out what's wrong.

If you can run some programs but not others, you may have a bad RAM bit. Load and use the memory test program provided in Appendix A. Try changing RAM chips, one at a time.

If nothing seems to work and you can't run the operating system, check your power supply and PC card wiring for shorts again. If everything still seems OK you will have to start signal tracing.

Signal Tracing

Check the U3 oscillator output. If not present, replace U3. If the 3.521280-MHz signal is present, check the U4 divider. Replace U4 if it isn't toggling. Make sure you use a 7474 type. With RUN up, you should see TPA and TPB pulses being generated at pins 33 and 34 of U1. If they are not present, check the RUN level to make sure the switch is working, then replace U1.

Check the output of U6B to make sure that the ROM is initially selected when RUN is first flipped up with key C down. With RUN up, check bus and address lines to see if any look different from the others. They will, of course, be at different levels or bouncing around but you might spot something suspicious that would indicate a short or open for one of these lines.

Try operating with only a 1024-byte RAM (U16 and U17). Try the other two RAM chips in these sockets. Check U5 inputs and outputs to verify that all stages are inverting properly.

If you don't get a pulse at pin 10 of U2 when you flip RUN up with key C down, U12 may be bad. This pulse is a difficult pulse to see and you might have to breadboard a latch or use a latching logic probe to catch it. If you get the display on pulse at pin 10 of U2, you should then see U2 output pulses on pins 2, 3, and 9. If you don't, try replacing U2.

Last Resorts

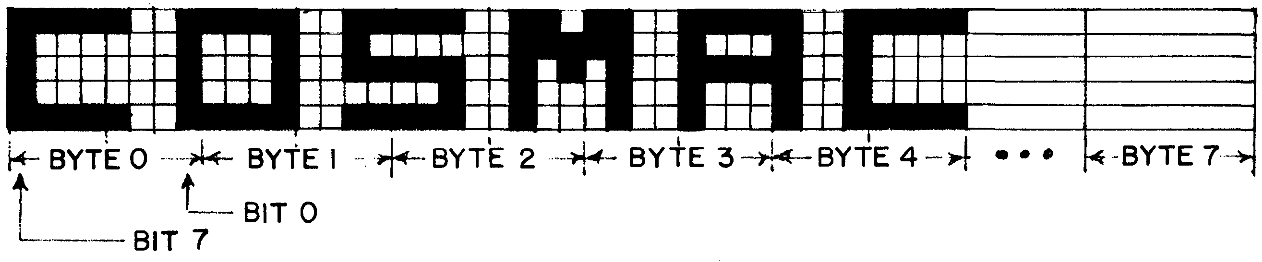
As a last resort, try replacing U1 and the ROM. Check the supply voltage at all chips. Examine the PC card for hairline breaks in the printed conductors. Fill up plated-through holes with solder to insure continuity. Check all signals. They should swing between ground and +4 or 5 volts. If you see a logic signal at some intermediate voltage, like +1 or 2 volts, check the source IC.

Once you get the operating system running, over 90% of the hardware will be operating properly. There are no critical adjustments to be made or maintained. All system timing is controlled by the crystal clock. With reasonable care your COSMAC VIP system should run for years without any problems.

Appendix A - Test and Operating Data

Byte Pattern for Displaying "COSMAC"

The following figure shows how the word "COSMAC" would be formed by spots (or bits) on the display screen.



The following bytes when loaded into memory will cause the word "COSMAC" to be shown on the display in a 2048-byte RAM system. Start pattern of bytes at location 0F00 in a 4096-byte system.

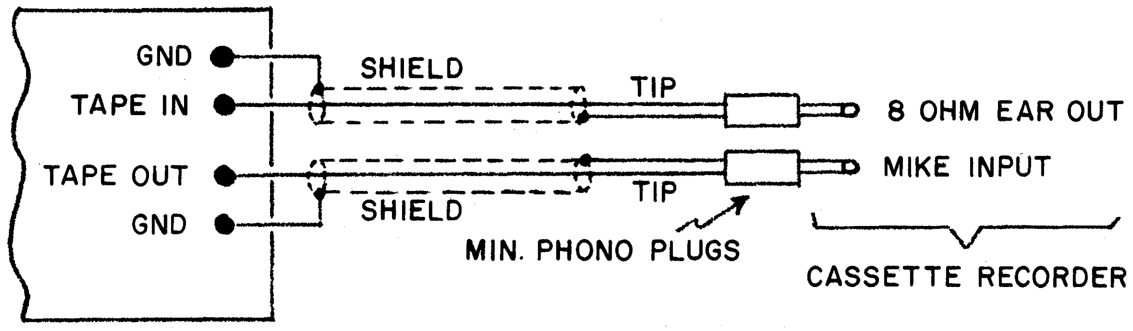
```
0700   F9  F3  E6  CF  9F  00  00  00
0708   81  12  07  C8  90  00  00  00
0710   81  13  E5  4F  90  00  00  00
0718   81  10  24  48  90  00  00  00
0720   F9  F3  E4  48  9F  00  00  00
0728   00  00  00  00  00  00  00  00
```

Beeper Program

This machine-language program flashes the Q light and beeps at a rate determined by the byte at location 0002. Change this byte for faster or slower rates.

```
0000   7A  F8  0F  BF  2F  9F  3A  04
0008   31  00  7B  30  01  00  00  00
```

Cassette Attachment Diagram



Cassette Phase Test

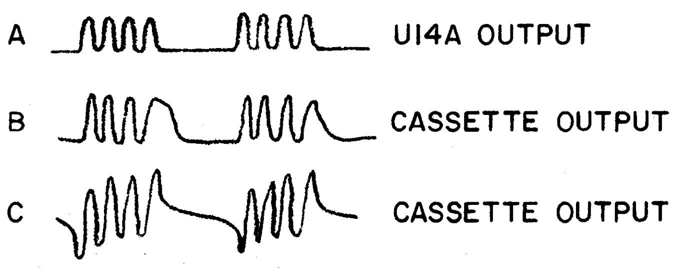
For best results your cassette recorder should not reverse the phase of an input signal on playback. When playing back a tape recorded on another recorder, it should not reverse the phase of the output signal. You may have to reverse the internal head connections on some cassette recorders to eliminate unwanted phase reversals.

To check for phase reversals, load the machine language test program, given below, into memory. Run this program to generate a phase test signal on the tape out line. Record one minute of this test signal, then play it back and observe the cassette recorder output on a scope. It should appear as shown in B or C below. Save this tape to test new recorders on which you want to play tapes you have recorded on a previously tested machine. If the playback signal appears upside down from that shown in B or C, you will have to reverse the internal head connection leads on the out-of-phase recorder.

Test Program

```
0000  F8 04 AA 7B F8 0C FF 01
0008  3A 06 7A F8 0C FF 01 3A
0010  0D 2A 8A 3A 03 F8 60 FF
0018  01 3A 17 30 00 00 00 00
```

Signals

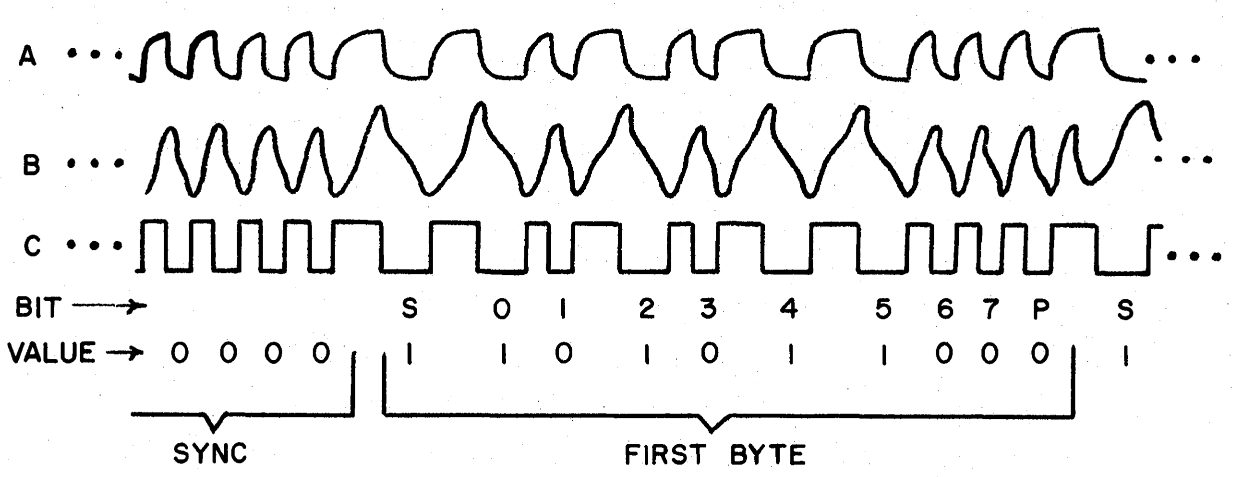


Cassette Data Test

Load the following machine language program into memory:

```
0000  90 B6 B3 F8 33 A6 F8 0A
0008  A3 D3 F8 6F AC F8 40 B9
0010  93 F6 DC 29 99 3A 10 F8
0018  10 A7 F8 08 A9 06 B7 F8
0020  80 FE DC 97 F6 B7 DC 29
0028  89 3A 23 17 87 F6 DC 30
0030  17 30 31 35 00 00 00 00
```

Rewind a blank cassette and put recorder into record mode. Wait 10 seconds and flip RUN up to initiate the program. The byte at location 0033 will be continuously recorded on tape. Flip RUN down to stop recording after a minute or so. You can play this tape to check the signals shown below. You can also load the tape into memory for testing purposes. Load 7 pages starting at 0100. You can use this tape to determine the proper volume control setting for your recorder. You can change the recorded byte at 0033 if desired. Bits on tape consist of one cycle at 2 kHz for "0" or one cycle at 0.8 kHz for "1". Data format is 4 seconds of continuous "0's" for sync followed by the specified number of data bytes. Bytes always begin with a "1" start bit (S) followed by 8 data bits (0-7), and end with a parity bit (P). Odd byte parity is used in this code. The waveforms below show how a 35 byte would appear on tape. The operating system translates memory bytes to bit serial output via the Q output line. Bit serial input from tape is received via input flag 2 and translated into parallel form for storage in memory by the operating system software.



A-OUTPUT OF U14A
B-OUTPUT FROM CASSETTE (TAP IN PAD ON CARD)
C-OUTPUT OF U14B

*WAVEFORMS SHOWN FOR PANASONIC MODEL RQ-413S RECORDER.

Cassette Recording Guidelines

1. Use high quality tape (Maxell UD or equivalent).
2. Use shortest tapes possible. You can shorten tapes to several minutes in length if you enjoy splicing.
3. Keep heads and pinch rollers clean.
4. Keep heads aligned for tape interchangability.
5. Avoid recording too close to beginning of tape.
6. Make sure cassette is properly seated in recorder.
7. If you have trouble with a cassette try others. You can have a bad spot on tape or a warped cassette.
8. Highest setting of tone control is usually best.
9. Set recorder volume control so that TAPE light glows steadily on playback. This setting should be lower than highest-volume setting. Excessive TAPE light flickering indicates a bad tape or misaligned heads.
10. A dirty recorder volume control can cause tape dropouts.
11. Make sure cassette connection plugs make good contact.
12. Rewind cassettes before removing them from recorder.
13. Store cassettes in dust-proof containers.
14. Avoid exposing cassettes to heat or magnetic fields.
15. Before recording, wind cassette to one end and fully rewind.
16. Cassette recorders will give you problems once in a while (they don't like certain cassettes, etc.). If one gives you problems most of the time replace it.
17. Make sure that MIKE plug is connected before recording. You will hear a tone even if MIKE plug is out. On most recorders the TAPE light will glow while recording.
18. When recording give the page key a short tap to start.
19. Use the last byte of a tape block as a program identification and check code. It will appear on the display screen after the tape is loaded.
20. When loading a cassette into memory, the tape must contain as many pages as you specify to be loaded. If you try to load 8 pages from a 7-page tape the loading operation won't terminate properly.
21. You may have to record with the EAR plug out for some tape recorders.
22. Always use AC adaptor with recorder for best results.

Memory Test Program

This machine language program should be loaded into 0000-007F. It checks RAM locations 0400-07FF (U18 and U19) for proper data storage. Flip RUN up to start test. Beeps sound during test. Entire 1024-byte section of RAM being tested is shown on screen. Program stops with tone on if a bad RAM bit is found. Error byte is at 007F. This byte should be 00 or FF for no error. For example, if byte is 01 or FE then bit 0 was bad. The error byte is also shown on the screen.

Set location 0020=00 and location 0023=80 to test RAM locations 0080-03FF (U16 and U17).

```
0000   90 B1 B2 B3 F8 17 A3 D3
0008   42 70 22 78 22 52 C4 C4
0010   C4 94 B0 91 A0 30 08 F8
0018   0A A1 F8 7F A2 E2 69 F8
0020   04 B4 F8 00 A4 94 B7 84
0028   A7 7A E2 F8 00 A5 F8 FF
0030   A6 85 57 94 BA 84 AA 8A
0038   52 87 F3 3A 45 9A 52 97
0040   F3 3A 45 30 47 86 5A 1A
0048   9A 52 94 FC 04 F3 3A 37
0050   07 52 85 F3 3A 6C F8 FF
0058   A5 93 A6 31 60 7B 30 31
0060   17 97 52 94 FC 04 F3 3A
0068   29 7A 30 6A 7B 30 6D 00
0070   00 00 00 00 00 00 00 00
0078   00 00 00 00 00 00 00 00
```

Appendix B - Operating System

Operating System Listing

The following shows the machine language code for the ROM operating system. ROM is addressed at 8000-81FF. This listing can be used to verify the contents of the ROM if required.

```
8000  F8 80 B2 F8 08 A2 E2 D2      8100  30 39 22 2A 3E 20 24 34
8008  64 00 62 0C F8 FF A1 F8      8108  26 28 2E 18 14 1C 10 12
8010  0F B1 F8 AA 51 01 FB AA      8110  F0 80 F0 80 F0 80 80 80
8018  32 22 91 FF 04 3B 22 B1      8118  F0 50 70 50 F0 50 50 50
8020  30 12 36 28 90 A0 E0 D0      8120  F0 80 F0 10 F0 80 F0 90
8028  E1 F8 00 73 81 FB AF 3A      8128  F0 90 F0 10 F0 10 F0 90
8030  29 F8 D2 73 F8 9F 51 81      8130  F0 90 90 90 F0 10 10 10
8038  A0 91 B0 F8 CF A1 D0 73      8138  10 60 20 20 20 70 A0 A0
8040  20 20 40 FF 01 20 50 FB      8140  F0 20 20 7A 42 70 22 78
8048  82 3A 3E 92 B3 F8 51 A3      8148  22 52 C4 19 F8 00 A0 9B
8050  D3 90 B2 BB BD F8 81 B1      8150  B0 E2 E2 80 E2 E2 20 A0
8058  B4 B5 B7 BA BC F8 46 A1      8158  E2 20 A0 E2 20 A0 3C 53
8060  F8 AF A2 F8 DD A4 F8 C6      8160  98 32 67 AB 2B 8B B8 88
8068  A5 F8 BA A7 F8 A1 AC E2      8168  32 43 7B 28 30 44 D3 F8
8070  69 DC D7 D7 D7 B6 D7 D7      8170  0A 3B 76 F8 20 17 7B BF
8078  D7 A6 D4 DC BE 32 F4 FB      8178  FF 01 3A 78 39 6E 7A 9F
8080  0A 32 EF DC AE 22 61 9E      8180  30 78 D3 F8 10 3D 85 3D
8088  FB 0B 32 C2 9E FB 0F 3A      8188  8F FF 01 3A 87 17 9C FE
8090  8F F8 6F AC F8 40 B9 93      8190  35 90 30 82 D3 E2 9C AF
8098  F6 DC 29 99 3A 97 F8 10      8198  2F 22 8F 52 62 E2 E2 3E
80A0  A7 F8 08 A9 46 B7 93 FE      81A0  98 F8 04 A8 88 3A A4 F8
80A8  DC 86 3A AD 2E 97 F6 B7      81A8  04 A8 36 A7 88 31 AA 8F
80B0  DC 29 89 3A AD 17 87 F6      81B0  FA 0F 52 30 94 00 00 00
80B8  DC 8E 3A 9E DC 69 26 D4      81B8  00 D3 DC FE FE FE FE AE
80C0  30 C0 F8 83 AC F8 0A B9      81C0  DC 8E F1 30 B9 D4 AA 0A
80C8  DC 33 C5 29 99 3A C8 DC      81C8  AA F8 05 AF 4A 5D 8D FC
80D0  3B CF F8 09 A9 A7 97 76      81D0  08 AD 2F 8F 3A CC 8D FC
80D8  B7 29 DC 89 3A D6 87 F6      81D8  D9 AD 30 C5 D3 22 06 73
80E0  33 E3 7B 97 56 16 86 3A      81E0  86 73 96 52 F8 06 AE F8
80E8  CF 2E 8E 3A CF 30 BD DC      81E8  D8 AD 02 F6 F6 F6 F6 D5
80F0  16 D4 30 EF D7 D7 D7 56      81F0  42 FA 0F D5 8E F6 AE 32
80F8  D4 16 30 F4 00 00 00 00      81F8  DC 3B EA 1D 1D 30 EA 01
```

Operating System Register Table

Memory Address	Register Byte	Memory Address	Register Byte
0XB0	–	0XC0	–
0XB1	–	0XC1	–
0XB2	–	0XC2	–
0XB3	R3.0	0XC3	R3.1
0XB4	R4.0	0XC4	R4.1
0XB5	R5.0	0XC5	R5.1
0XB6	R6.0	0XC6	R6.1
0XB7	R7.0	0XC7	R7.1
0XB8	R8.0	0XC8	R8.1
0XB9	R9.0	0XC9	R9.1
0XBA	RA.0	0XCA	RA.1
0XBB	RB.0	0XCB	RB.1
0XBC	RC.0	0XCC	RC.1
0XBD	RD.0	0XCD	RD.1
0XBE	RE.0	0XCE	RE.1
0XBF	RF.0	0XCF	RF.1

0X = 07 for 2048-byte RAM
0X = 0B for 3072-byte RAM
0X = 0F for 4096-byte RAM
R5 = CHIP-8 language program counter
RA = CHIP-8 language I pointer

Operating System Summary

1. RUN up with key C pressed selects operating system at 8000.

2. Enter four-digit address followed by mode digit:

A = MR (Memory Read)
0 = MW (Memory Write)
B = TR (Tape Read)
F = TW (Tape Write)

3. CDP1802 microprocessor registers are stored as shown in table above. They may be examined after a program is run by using operating system mode A.

4. Mode 0 can be used to insert temporary stops in a program for debugging purposes. Insert a "branch-to-itself" instruction at the desired stopping point.

5. The operating system uses the top 84 bytes of RAM (0XAC-0XFF). Avoid using these byte locations in your programs.

6. The operating system searches for and uses the top (highest) 256-byte page of on-card RAM. When RUN is flipped up to execute a program beginning at 0000, the following initial conditions exist:

P=0, Q=0, R0=0000, and R1=0XFF where 0X= highest page of on-card RAM.

Appendix C - CHIP-8 Interpreter

CHIP-8 Interpreter Listing

To use the CHIP-8 language you must first load the following interpreter program into memory locations 0000-01FF (2 pages). This interpreter will allow you to run the games in Appendix D or write your own programs using the CHIP-8 instruction set described in section III.

```
0000  91 BB FF 01 B2 B6 F8 CF
0008  A2 F8 81 B1 F8 46 A1 90
0010  B4 F8 1B A4 F8 01 B5 F8
0018  FC A5 D4 96 B7 E2 94 BC
0020  45 AF F6 F6 F6 F6 32 44
0028  F9 50 AC 8F FA 0F F9 F0
0030  A6 05 F6 F6 F6 F6 F9 F0
0038  A7 4C B3 8C FC 0F AC 0C
0040  A3 D3 30 1B 8F FA 0F B3
0048  45 30 40 22 69 12 D4 00
0050  00 01 01 01 01 01 01 01
0058  01 01 01 01 01 00 01 01
0060  00 7C 75 83 8B 95 B4 B7
0068  BC 91 EB A4 D9 70 99 05  E5
0070  06 FA 07 BE 06 FA 3F F6
0078  F6 F6 22 52 07 FA 1F FE
0080  FE FE F1 AC 9B BC 45 FA
0088  0F AD A7 F8 D0 A6 93 AF
0090  87 32 F3 27 4A BD 9E AE
0098  8E 32 A4 9D F6 BD 8F 76
00A0  AF 2E 30 98 9D 56 16 8F
00A8  56 16 30 8E 00 EC F8 D0
00B0  A6 93 A7 8D 32 D9 06 F2
00B8  2D 32 BE F8 01 A7 46 F3
00C0  5C 02 FB 07 32 D2 1C 06
00C8  F2 32 CE F8 01 A7 06 F3
00D0  5C 2C 16 8C FC 08 AC 3B
00D8  B3 F8 FF A6 87 56 12 D4
00E0  9B BF F8 FF AF 93 5F 8F
00E8  32 DF 2F 30 E5 00 42 B5
00F0  42 A5 D4 8D A7 87 32 AC
00F8  2A 27 30 F5 00 00 00 00
```

```
0100  00 00 00 00 00 45 A3 98
0108  56 D4 F8 81 BC F8 95 AC
0110  22 DC 12 56 D4 06 B8 D4
0118  06 A8 D4 64 0A 01 E6 8A
0120  F4 AA 3B 28 9A FC 01 BA
0128  D4 F8 81 BA 06 FA 0F AA
0130  0A AA D4 E6 06 BF 93 BE
0138  F8 1B AE 2A 1A F8 00 5A
0140  0E F5 3B 4B 56 0A FC 01
0148  5A 30 40 4E F6 3B 3C 9F
0150  56 2A 2A D4 00 22 86 52
0158  F8 F0 A7 07 5A 87 F3 17
0160  1A 3A 5B 12 D4 22 86 52
0168  F8 F0 A7 0A 57 87 F3 17
0170  1A 3A 6B 12 D4 15 85 22  [illegible]
0178  73 95 52 25 45 A5 86 FA
0180  0F B5 D4 45 E6 F3 3A 82
0188  15 15 D4 45 E6 F3 3A 88
0190  D4 45 07 30 8C 45 07 30
0198  84 E6 62 26 45 A3 36 88
01A0  D4 3E 88 D4 F8 F0 A7 E7
01A8  45 F4 A5 86 FA 0F 3B B2
01B0  FC 01 B5 D4 45 56 D4 45
01B8  E6 F4 56 D4 45 FA 0F 3A
01C0  C4 07 56 D4 AF 22 F8 D3
01C8  73 8F F9 F0 52 E6 07 D2
01D0  56 F8 FF A6 F8 00 7E 56
01D8  D4 19 89 AE 93 BE 99 EE
01E0  F4 56 76 E6 F4 B9 56 45
01E8  F2 56 D4 45 AA 86 FA 0F
01F0  BA D4 00 00 00 00 00 00
01F8  00 00 00 00 00 E0 00 4B
```

CHIP-8 Memory Map

Location	Use
0000 • • • 01FF	CHIP-8 LANGUAGE INTERPRETER
0200 • • •	User programs using CHIP-8 instruction set (1184 bytes available in 2048-byte system)
0YA0 • • • 0YCF	CHIP-8 stack (48 bytes max. for up to 12 levels of subroutine nesting)
0YD0 • • • 0YEF	Reserved for CHIP-8 INTERPRETER work area
0YF0	V0
0YF1	V1
0YF2	V2
0YF3	V3
0YF4	V4
0YF5	V5
0YF6	V6
0YF7	V7
0YF8	V8
0YF9	V9
0YFA	VA
0YFB	VB
0YFC	VC
0YFD	VD
0YFE	VE
0YFF	VF
0X00 • • • 0XFF	256-byte RAM area for display refresh

1;6

0X = Highest on-card RAM page (07 for 2048-byte system)
0Y = 0X – 1 (06 for 2048-byte system)

CDP1802 Register Use for CHIP-8 Interpreter

R0	= DMA pointer (page 0X for display refresh)
R1	= INTERRUPT routine program counter
R2	= Stack pointer
R3	= INTERPRETER subroutine program counter
R4	= CALL subroutine program counter
R5	= CHIP-8 instruction program counter
R6	= VX pointer (R6.1 must not be changed)
R7	= VY pointer (available for machine-language subroutines)
R8	= Timers (R8.1 = timer, R8.0 = tone duration)
R9	= Random number (+1 in INTERRUPT routine)
RA	= I pointer
RB	= Display page pointer (RB.1 = 0X)
RC	= Available
RD	= Available
RE	= Available
RF	= Available

CHIP-8/Operating System Standard Digit Display Format

HEX	ROM		BITS							
DIGIT	ADDRESS	BYTE	7	6	5	4	3	2	1	0
E-	8110	F0	■	■	■	■	□	□	□	□
	11	80	■	□	□	□	□	□	□	□
F-	8112	F0	■	■	■	■	□	□	□	□
	13	80	■	□	□	□	□	□	□	□
C-	8114	F0	■	■	■	■	□	□	□	□
	15	80	■	□	□	□	□	□	□	□
	16	80	■	□	□	□	□	□	□	□
	17	80	■	□	□	□	□	□	□	□
B-	8118	F0	■	■	■	■	□	□	□	□
	19	50	□	■	□	■	□	□	□	□
	1A	70	□	■	■	■	□	□	□	□
	1B	50	□	■	□	■	□	□	□	□
D-	811C	F0	■	■	■	■	□	□	□	□
	1D	50	□	■	□	■	□	□	□	□
	1E	50	□	■	□	■	□	□	□	□
	1F	50	□	■	□	■	□	□	□	□
5-	8120	F0	■	■	■	■	□	□	□	□
	21	80	■	□	□	□	□	□	□	□
2-	8122	F0	■	■	■	■	□	□	□	□
	23	10	□	□	□	■	□	□	□	□
6-	8124	F0	■	■	■	■	□	□	□	□
	25	80	■	□	□	□	□	□	□	□
8-	8126	F0	■	■	■	■	□	□	□	□
	27	90	■	□	□	■	□	□	□	□
9-	8128	F0	■	■	■	■	□	□	□	□
	29	90	■	□	□	■	□	□	□	□
3-	812A	F0	■	■	■	■	□	□	□	□
	2B	10	□	□	□	■	□	□	□	□
	2C	F0	■	■	■	■	□	□	□	□
	2D	10	□	□	□	■	□	□	□	□
A-	812E	F0	■	■	■	■	□	□	□	□
	2F	90	■	□	□	■	□	□	□	□
0-	8130	F0	■	■	■	■	□	□	□	□
	31	90	■	□	□	■	□	□	□	□
	32	90	■	□	□	■	□	□	□	□
	33	90	■	□	□	■	□	□	□	□
7-	8134	F0	■	■	■	■	□	□	□	□
	35	10	□	□	□	■	□	□	□	□
	36	10	□	□	□	■	□	□	□	□
	37	10	□	□	□	■	□	□	□	□
	38	10	□	□	□	■	□	□	□	□
1-	8139	60	□	■	■	□	□	□	□	□
	3A	20	□	□	■	□	□	□	□	□
	3B	20	□	□	■	□	□	□	□	□
	3C	20	□	□	■	□	□	□	□	□
	3D	70	□	■	■	■	□	□	□	□
4-	813E	A0	■	□	■	□	□	□	□	□
	3F	A0	■	□	■	□	□	□	□	□
	40	F0	■	■	■	■	□	□	□	□
	41	20	□	□	■	□	□	□	□	□
	42	20	□	□	■	□	□	□	□	□

CHIP-8 User Notes

1. Do not use any of the CDP1802 three-cycle machine language instructions in CHIP-8 programs.

2. CDP1802 R5 is used as the CHIP-8 instruction counter. It will be addressing the byte following a 0MMM instruction for machine language subroutines and can be used to pass 2-byte parameters. Refer to the operating system register table in Appendix B to examine this register during CHIP-8 program debugging.

3. Display page 0X is erased to all 0's before beginning CHIP-8 programs at 0200. To inhibit erasing page 0X, change 00E0 at location 01FC to 11FE.

4. To change the display page from 0X, use a machine language subroutine to set RB.1 equal to the new display page.

5. R7, RC, RD, RE, and RF can be used as working registers in machine language subroutines. Changing other registers can cause the CHIP-8 interpreter to malfunction.

6. Do not turn off the CDP1861 video display chip in machine language subroutines. This will interfere with proper operation of the CHIP-8 interpreter.

7. Program bugs can destroy the CHIP-8 interpreter at locations 0000-01FF. If you suspect that this has happened, reload the interpreter.

8. The CHIP-8 interpreter uses subroutines and digit patterns contained in the operating system ROM. If you modify this operating system, the CHIP-8 interpreter should not be used.

Appendix D - Video Games

This Appendix contains program listings for twenty video games. These games, which illustrate entertainment applications of COSMAC VIP, were developed by Joe Weisbecker (games 1 through 8), Joyce Weisbecker (games 9 and 10), Jef Winsor (games 11, 12, and 13), Tom Chen (games 14, 15, and 16), and Phil Baltzer (games 17 through 20).

In the listing for each game, the first column is the memory location at which the instruction bytes in the second column are stored. The comments in the third column indicate the function of the instruction byte. The comments are not stored in memory.

The game titles are listed below:

Game Title **Page**

1. VIP Kaleidoscope

This program uses the CHIP-8 INTERPRETER at 0000-01FF. Four spots appear in a group at the center of the screen. Press keys 2, 4, 6, or 8 to create a pattern. Keep your pattern smaller than 138 key depressions. Push key 0 to terminate pattern entry. Pushing key 0 causes your pattern to be continuously repeated forming a fascinating, changing kaleidoscope display on the screen. A "44444442220" key sequence provides a very nice effect. Experiment to find other nice patterns. The subroutine at 0232-0274 causes your pattern to be duplicated in the four quadrants of the screen.

```
0200 6000 V0=00
0202 6380 V3=80
0204 611F V1=1F
0206 620F V2=0F
0208 2232 DO 0232
020A A200 I=0200
020C F31E I=I+V3
020E F00A V0=KEY
0210 F055 MI=V0:V0
0212 4000 SKIP;V0 NE 00
0214 121C GO 021C
0216 7301 V3+01
0218 3300 SKIP;V3 EQ 00
021A 1208 GO 0208
021C 6380 V3=80
021E A200 I=0200
0220 F31E I=I+V3
0222 F065 V0:V0=MI
0224 4000 SKIP;V0 NE 00
0226 121C GO 021C
0228 7301 V3+01
022A 4300 SKIP;V3 NE 00
022C 121C GO 021C
022E 2232 DO 0232
0230 121E GO 021E
0232 4002 SKIP;V0 NE 02
0234 72FF V2+FF
0236 4004 SKIP;V0 NE 04
0238 71FF V1+FF
023A 4006 SKIP;V0 NE 06
```

```
023C 7101 V1+01
023E 4008 SKIP;V0 NE 08
0240 7201 V2+01
0242 A277 I=0277
0244 6AE0 VA=E0
0246 8A12 VA=VA&V1
0248 6B1F VB=1F
024A 81B2 V1=V1&VB
024C 3A00 SKIP;VA EQ 00
024E 7201 V2+01
0250 6AF0 VA=F0
0252 8A22 VA=VA&V2
0254 6B0F VB=0F
0256 82B2 V2=V2&VB
0258 3A00 SKIP;VA EQ 00
025A 7101 V1+01
025C 6B1F VB=1F
025E 81B2 V1=V1&VB
0260 D121 SHOW 1MI@V1V2
0262 8A10 VA=V1
0264 6B1F VB=1F
0266 8B25 VB=VB-V2
0268 DAB1 SHOW 1MI@VAVB
026A 6A3F VA=3F
026C 8A15 VA=VA-V1
026E DAB1 SHOW 1MI@VAVB
0270 8B20 VB=V2
0272 DAB1 SHOW 1MI@VAVB
0274 00EE RET
0276 0180
0278 0000
```

2. VIP Video Display Drawing Game

This program uses the CHIP-8 INTERPRETER at 0000-01FF. A flashing spot appears in the upper left corner of the screen. You can move the spot by holding key 2, 4, 6, or 8. Press key 5 and you can draw a picture with the spot. Press key 0 and the spot can be moved without drawing or used to erase a previously drawn line. 0245-024E is a list of initial values for V0-V9. In this program, locations 0300-03FF are used for the picture. After drawing a picture, you can change M(0208) from 00E0 to 120A. Write locations 0000-03FF (4 pages) to tape to save your picture. When you load these four pages back into memory you will see your original picture. Changing the 00E0 instruction in the program to 120A prevents your picture from being erased when the program is started.

```
0200 A245 I=0245
0202 F965 V0:V9=MI
0204 A24F I=024F
0206 0236 MLS@0236
0208 00E0 ERASE
020A F915 TIME=V9
020C FA07 VA=TIME
020E 3A00 SKIP;VA EQ 00
0210 120C GO 020C
0212 D121 SHOW 1MI@V1V2
0214 3F00 SKIP;VF EQ 00
0216 D121 SHOW 1MI@V1V2
0218 E3A1 SKIP;V3 NE KEY
021A 8030 V0=V3
021C E4A1 SKIP;V4 NE KEY
021E 8040 V0=V4
0220 4000 SKIP;V0 NE 00
0222 123C GO 023C
0224 E5A1 SKIP;V5 NE KEY
0226 72FF V2+FF
0228 E6A1 SKIP;V6 NE KEY
022A 71FF V1+FF
022C E7A1 SKIP;V7 NE KEY
022E 7101 V1+01
0230 E8A1 SKIP;V8 NE KEY
0232 7201 V2+01
0234 120A GO 020A
0236 01F8
0238 03BB
023A E2D4
023C D121 SHOW 1MI@V1V2
023E 4F00 SKIP;VF NE 00
0240 D121 SHOW 1MI@V1V2
0242 1224 GO 0224
0244 0100
0246 0000
0248 0005
024A 0204
024C 0608
024E 0880
```

3. VIP Wipe Off

This program uses the CHIP-8 INTERPRETER at 0000-01FF. Serve the ball by pressing any key. Move the paddle left or right by pressing key 4 or 6. Try to wipe out as many spots as possible. Each spot counts one point. You get 20 balls. You see your final score at the end of the game. You can make the paddle wider by changing the E0 byte at 02CD to F8 or FF.

```
0200 A2CC I=02CC
0202 6A07 VA=07
0204 6100 V1=00
0206 6B08 VB=08
0208 6000 V0=00
020A D011 SHOW 1MI@V0V1
020C 7008 V0+08
020E 7BFF VB+FF
0210 3B00 SKIP;VB EQ 00
0212 120A GO 020A
0214 7104 V1+04
0216 7AFF VA+FF
0218 3A00 SKIP;VA EQ 00
021A 1206 GO 0206
021C 6600 V6=00
021E 6714 V7=14
0220 A2CD I=02CD
0222 6020 V0=20
0224 611E V1=1E
0226 D011 SHOW 1MI@V0V1
0228 631D V3=1D
022A 623F V2=3F
022C 8202 V2=V2&V0
022E 77FF V7+FF
0230 4700 SKIP;V7 NE 00
0232 12AA GO 02AA
0234 FF0A VF=KEY
0236 A2CB I=02CB
0238 D231 SHOW 1MI@V2V3
023A 65FF V5=FF
023C C401 V4=RND
023E 3401 SKIP;V4 EQ 01
0240 64FF V4=FF
0242 A2CD I=02CD
0244 6C00 VC=00
0246 6E04 VE=04
0248 EEA1 SKIP;VE NE KEY
024A 6CFF VC=FF
024C 6E06 VE=06
024E EEA1 SKIP;VE NE KEY
0250 6C01 VC=01
0252 D011 SHOW 1MI@V0V1
0254 80C4 V0=V0+VC
0256 D011 SHOW 1MI@V0V1
0258 4F01 SKIP;VF NE 01
025A 1298 GO 0298
025C 4200 SKIP;V2 NE 00
025E 6401 V4=01
0260 423F SKIP;V2 NE 3F
0262 64FF V4=FF
0264 4300 SKIP;V3 NE 00
0266 6501 V5=01
0268 431F SKIP;V3 NE 1F
026A 12A4 GO 02A4
026C A2CB I=02CB
026E D231 SHOW 1MI@V2V3
0270 8244 V2=V2+V4
0272 8354 V3=V3+V5
0274 D231 SHOW 1MI@V2V3
0276 3F01 SKIP;VF EQ 01
0278 1242 GO 0242
027A 431E SKIP;V3 NE 1E
027C 1298 GO 0298
027E 6A02 VA=02
0280 FA18 TONE=VA
0282 7601 V6+01
0284 4670 SKIP;V6 NE 70
0286 12AA GO 02AA
0288 D231 SHOW 1MI@V2V3
028A C401 V4=RND
028C 3401 SKIP;V4 EQ 01
028E 64FF V4=FF
0290 C501 V5=RND
0292 3501 SKIP;V5 EQ 01
0294 65FF V5=FF
0296 1242 GO 0242
0298 6A03 VA=03
029A FA18 TONE=VA
029C A2CB I=02CB
029E D231 SHOW 1MI@V2V3
02A0 73FF V3+FF
02A2 1236 GO 0236
02A4 A2CB I=02CB
02A6 D231 SHOW 1MI@V2V3
02A8 1228 GO 0228
02AA A2CD I=02CD
02AC D011 SHOW 1MI@V0V1
02AE A2F0 I=02F0
02B0 F633 MI=V6(3DD)
02B2 F265 V0:V2=MI
02B4 6318 V3=18
02B6 641B V4=1B
02B8 F029 I=V0(LSDP)
02BA D345 SHOW 5MI@V3V4
02BC 7305 V3+05
02BE F129 I=V1(LSDP)
02C0 D345 SHOW 5MI@V3V4
02C2 7305 V3+05
02C4 F229 I=V2(LSDP)
02C6 D345 SHOW 5MI@V3V4
02C8 12C8 GO 02C8
02CA 0180
02CC 44E0
```

4. VIP Space Intercept

This program uses the CHIP-8 INTERPRETER at 0000-01FF. Launch your rocket by pressing key 4, 5, or 6. Hit the UFO's to score. The big UFO counts 5 points. The small UFO counts 15 points. You get 15 rockets as shown in the lower right corner of the screen. Your score is shown in the lower left corner of the screen.

```
0200 A2CD I=02CD          024C 8BD4 VB=VB+VD      0298 6D03 VD=03
0202 6938 V9=38           024E DBC3 SHOW 3MI@VBVC 029A FD18 TONE=VD
0204 6A08 VA=08           0250 3F00 SKIP;VF EQ 00 029C A2D3 I=02D3
0206 D9A3 SHOW 3MI@V9VA   0252 1292 GO 0292       029E D453 SHOW 3MI@V4V5
0208 A2D0 I=02D0          0254 A2CD I=02CD        02A0 1286 GO 0286
020A 6B00 VB=00           0256 D9A3 SHOW 3MI@V9VA 02A2 A2F8 I=02F8
020C 6C03 VC=03           0258 CD01 VD=RND        02A4 F733 MI=V7(3DD)
020E DBC3 SHOW 3MI@VBVC   025A 3D00 SKIP;VD EQ 00 02A6 6300 V3=00
0210 A2D6 I=02D6          025C 6DFF VD=FF         02A8 22B6 DO 02B6
0212 641D V4=1D           025E 79FE V9+FE         02AA 00EE RET
0214 651F V5=1F           0260 D9A3 SHOW 3MI@V9VA 02AC A2F8 I=02F8
0216 D451 SHOW 1MI@V4V5   0262 3F00 SKIP;VF EQ 00 02AE F833 MI=V8(3DD)
0218 6700 V7=00           0264 128C GO 028C       02B0 6332 V3=32
021A 680F V8=0F           0266 4E00 SKIP;VE NE 00 02B2 22B6 DO 02B6
021C 22A2 DO 02A2         0268 122E GO 022E       02B4 00EE RET
021E 22AC DO 02AC         026A A2D3 I=02D3        02B6 6D1B VD=1B
0220 4800 SKIP;V8 NE 00   026C D453 SHOW 3MI@V4V5 02B8 F265 V0:V2=MI
0222 1222 GO 0222         026E 4500 SKIP;V5 NE 00 02BA F029 I=V0(LSDP)
0224 641E V4=1E           0270 1286 GO 0286       02BC D3D5 SHOW 5MI@V3VD
0226 651C V5=1C           0272 75FF V5+FF         02BE 7305 V3+05
0228 A2D3 I=02D3          0274 8464 V4=V4+V6      02C0 F129 I=V1(LSDP)
022A D453 SHOW 3MI@V4V5   0276 D453 SHOW 3MI@V4V5 02C2 D3D5 SHOW 5MI@V3VD
022C 6E00 VE=00           0278 3F01 SKIP;VF EQ 01 02C4 7305 V3+05
022E 6680 V6=80           027A 1246 GO 0246       02C6 F229 I=V2(LSDP)
0230 6D04 VD=04           027C 6D08 VD=08         02C8 D3D5 SHOW 5MI@V3VD
0232 EDA1 SKIP;VD NE KEY  027E 8D52 VD=VD&V5      02CA 00EE RET
0234 66FF V6=FF           0280 4D08 SKIP;VD NE 08 02CC 017C
0236 6D05 VD=05           0282 128C GO 028C       02CE FE7C
0238 EDA1 SKIP;VD NE KEY  0284 1292 GO 0292       02D0 60F0
023A 6600 V6=00           0286 22AC DO 02AC       02D2 6040
023C 6D06 VD=06           0288 78FF V8+FF         02D4 E0A0
023E EDA1 SKIP;VD NE KEY  028A 121E GO 021E       02D6 F8D4
0240 6601 V6=01           028C 22A2 DO 02A2       02D8 6E01 VE=01
0242 3680 SKIP;V6 EQ 80   028E 7705 V7+05         02DA 6D10 VD=10
0244 22D8 DO 02D8         0290 1296 GO 0296       02DC FD18 TONE=VD
0246 A2D0 I=02D0          0292 22A2 DO 02A2       02DE 00EE RET
0248 DBC3 SHOW 3MI@VBVC   0294 770F V7+0F
024A CD01 VD=RND          0296 22A2 DO 02A2
```

5. VIP 4096-Bit Picture

This is a machine language program that shows a picture pattern stored at 0100-02FF on the screen. Load the following program into memory at 0000-002F:

```
0000  90 B1 B2 B3 F8 08 A3 D3
0008  F8 FF A2 F8 14 A1 E2 69
0010  30 10 42 70 22 78 22 52
0018  C4 C4 C4 F8 01 B0 91 A0
0020  80 E2 E2 20 A0 E2 80 A0
0028  E2 20 A0 3C 20 30 12 00
```

Store the following picture pattern at 0100-02FF. This picture is shown as an array of spots 64 wide by 64 high. You can substitute your own picture pattern at 0100-01FF.

```
0100  80 00 00 00 00 80 00 00
0108  00 00 00 00 00 00 00 00
0110  00 00 00 00 00 01 FF FF
0118  00 00 00 00 00 02 00 01
0120  00 00 10 00 30 02 00 02
0128  00 00 00 00 78 01 FF FE
0130  00 00 00 3F 87 F8 20 00
0138  00 00 00 20 30 08 20 00
0140  00 00 00 3F 87 F0 20 00
0148  00 00 00 00 7A 20 20 00
0150  00 40 00 00 31 10 20 20
0158  00 00 00 00 00 C8 20 00
0160  00 00 00 00 01 3F F8 00
0168  00 00 00 00 02 00 08 00
0170  00 00 00 40 06 FC 08 00
0178  00 03 80 00 02 01 F8 00
0180  0C 07 C0 00 01 0E 00 00
0188  1E 0F E0 00 00 F0 00 00
0190  3F BF F0 00 00 00 00 00
0198  37 FF F8 00 00 00 00 00
01A0  67 FF E8 02 00 00 00 00
01A8  67 FF F8 00 00 00 00 00
01B0  43 BF F8 00 00 04 00 00
01B8  43 BF F0 00 00 00 00 00
01C0  C3 FF E0 00 00 00 00 00
01C8  C3 FF C0 00 00 00 00 00
01D0  81 E0 00 00 00 00 00 01
01D8  81 C0 00 00 00 00 00 00
01E0  81 C0 00 00 00 00 00 00
01E8  81 C0 00 00 00 00 00 00
01F0  81 C0 00 00 00 00 00 00
01F8  81 E0 00 00 00 00 00 00
0200  C1 FE 00 00 00 00 00 00
0208  C1 FE 00 00 00 00 00 00
```

```
0210  63 FE 00 00 00 00 00 00
0218  63 FC 00 00 00 00 00 00
0220  3F F8 00 00 00 00 00 00
0228  3F F0 30 00 00 00 00 00
0230  03 F0 70 EE E3 BB AB B8
0238  03 F8 F0 A8 A2 2A 3A A0
0240  00 FD E0 E8 E2 2B BB A0
0248  00 FF C0 C8 A2 28 AA A0
0250  00 7F 80 AE A3 BB AA B8
0258  00 7F 00 00 00 00 00 00
0260  00 6F 00 00 00 00 00 00
0268  00 6F 80 00 00 00 00 00
0270  00 77 80 1D D5 D5 DD C0
0278  00 77 C0 11 5D 54 91 40
0280  00 5B C0 11 5D D4 99 C0
0288  00 5B C0 11 55 14 91 80
0290  00 6D C0 1D D5 1C 9D 40
0298  00 6D C0 00 00 00 00 00
02A0  00 6D C0 00 00 00 00 00
02A8  00 6D C0 00 00 00 00 00
02B0  00 33 80 00 00 00 00 00
02B8  00 3F 3F FF FF FF FF FF
02C0  00 1F 00 00 00 00 00 00
02C8  00 0E 00 00 00 00 00 00
02D0  00 0E 00 00 00 00 00 00
02D8  00 0E 00 00 00 00 00 00
02E0  00 0E 00 00 00 00 00 00
02E8  00 1F 80 00 00 00 00 00
02F0  00 3F C0 00 00 00 00 00
02F8  00 3F C0 00 00 00 00 00
```

6. VIP Figure Shooting at Moving Target

This program uses the CHIP-8 INTERPRETER at 0000-01FF. Fire the gun by pressing key 3(up), 6(straight), or 9(down) to hit the moving target. You get 25 shots (bottom number). Each hit scores 10 points (top number).

```
0200 6719 V7=19
0202 6800 V8=00
0204 22C2 DO 02C2
0206 22DE DO 02DE
0208 6525 V5=25
020A 660D V6=0D
020C 22E6 DO 02E6
020E D565 SHOW 5MI@V5V6
0210 CD01 VD=RND
0212 3D01 SKIP;VD EQ 01
0214 6D07 VD=07
0216 230C DO 030C
0218 6410 V4=10
021A 630B V3=0B
021C 83D4 V3=V3+VD
021E A2BF I=02BF
0220 D341 SHOW 1MI@V3V4
0222 6C00 VC=00
0224 6B80 VB=80
0226 6003 V0=03
0228 E0A1 SKIP;V0 NE KEY
022A 6BFF VB=FF
022C 6006 V0=06
022E E0A1 SKIP;V0 NE KEY
0230 6B00 VB=00
0232 6009 V0=09
0234 E0A1 SKIP;V0 NE KEY
0236 6B01 VB=01
0238 3B80 SKIP;VB EQ 80
023A 229A DO 029A
023C A2B3 I=02B3
023E D565 SHOW 5MI@V5V6
0240 8594 V5=V5+V9
0242 86A4 V6=V6+VA
0244 4520 SKIP;V5 NE 20
0246 22F6 DO 02F6
0248 453B SKIP;V5 NE 3B
024A 22FA DO 02FA
024C 4600 SKIP;V6 NE 00
024E 22FE DO 02FE
0250 461B SKIP;V6 NE 1B
0252 2308 DO 0308
0254 D565 SHOW 5MI@V5V6
0256 3F00 SKIP;VF EQ 00
0258 1280 GO 0280
025A 4C00 SKIP;VC NE 00
025C 1224 GO 0224
025E A2BF I=02BF
0260 D341 SHOW 1MI@V3V4

0262 333E SKIP;V3 EQ 3E
0264 126E GO 026E
0266 4700 SKIP;V7 NE 00
0268 1268 GO 0268
026A 230C DO 030C
026C 1210 GO 0210
026E 7302 V3+02
0270 4400 SKIP;V4 NE 00
0272 6B01 VB=01
0274 441D SKIP;V4 NE 1D
0276 6BFF VB=FF
0278 84B4 V4=V4+VB
027A D341 SHOW 1MI@V3V4
027C 4F00 SKIP;VF NE 00
027E 123C GO 023C
0280 6002 V0=02
0282 F018 TONE=V0
0284 A2BF I=02BF
0286 D341 SHOW 1MI@V3V4
0288 A2B3 I=02B3
028A D565 SHOW 5MI@V5V6
028C 22DE DO 02DE
028E 780A V8+0A
0290 22DE DO 02DE
0292 4700 SKIP;V7 NE 00
0294 1294 GO 0294
0296 230C DO 030C
0298 1208 GO 0208
029A 6C01 VC=01
029C 6007 V0=07
029E F018 TONE=V0
02A0 22C2 DO 02C2
02A2 77FF V7+FF
02A4 22C2 DO 02C2
02A6 00EE RET
02A8 017C
02AA 7CFE
02AC 7C7C
02AE 707C
02B0 387F
02B2 7F7C
02B4 7C7C
02B6 7C7C
02B8 3838
02BA 3838
02BC 383E
02BE E080
02C0 00D4
02C2 A3F8 I=03F8

02C4 F733 MI=V7(3DD)
02C6 6E1B VE=1B
02C8 6F10 VF=10
02CA F265 V0:V2=MI
02CC F029 I=V0(LSDP)
02CE DFE5 SHOW 5MI@VFVE
02D0 6F15 VF=15
02D2 F129 I=V1(LSDP)
02D4 DFE5 SHOW 5MI@VFVE
02D6 6F1A VF=1A
02D8 F229 I=V2(LSDP)
02DA DFE5 SHOW 5MI@VFVE
02DC 00EE RET
02DE A3F8 I=03F8
02E0 F833 MI=V8(3DD)
02E2 6E00 VE=00
02E4 12C8 GO 02C8
02E6 C901 V9=RND
02E8 3901 SKIP;V9 EQ 01
02EA 69FF V9=FF
02EC CA01 VA=RND
02EE 3A01 SKIP;VA EQ 01
02F0 6AFF VA=FF
02F2 A2B3 I=02B3
02F4 00EE RET
02F6 6901 V9=01
02F8 12EC GO 02EC
02FA 69FF V9=FF
02FC 12EC GO 02EC
02FE 6A01 VA=01
0300 C901 V9=RND
0302 3901 SKIP;V9 EQ 01
0304 69FF V9=FF
0306 00EE RET
0308 6AFF VA=FF
030A 1300 GO 0300
030C 6E08 VE=08
030E A2A9 I=02A9
0310 DDEF SHOW FMI@VDVE
0312 7E0F VE+0F
0314 A2B8 I=02B8
0316 DDE6 SHOW 6MI@VDVE
0318 6E10 VE=10
031A 6008 V0=08
031C 80D4 V0=V0+VD
031E 8F00 VF=V0
0320 A2BE I=02BE
0322 DFE2 SHOW 2MI@VFVE
0324 00EE RET
```

7. VIP Tick-Tack-Toe Game

This program uses the CHIP-8 INTERPRETER at 0000-01FF. You are "O", VIP is "X". You move first. Press key 1-9 to put your "O" into a square. Squares are in the same positions as keys 1-9. VIP then puts an "X" into an empty square. If you get three "O" 's in a row you win the game. If VIP gets three "X" 's in a row you lose the game. The game is a draw when all squares are filled without getting 3 in a row. You can beat VIP because it is programmed to make a mistake once in a while.

```
0200 02E4 MLS@02E4        024E 135A GO 035A        029C A3F5 I=03F5
0202 232E DO 032E         0250 6402 V4=02          029E F065 V0:V0=MI
0204 FD0A VD=KEY          0252 6C01 VC=01          02A0 3001 SKIP;V0 EQ 01
0206 6009 V0=09           0254 2390 DO 0390        02A2 1380 GO 0380
0208 9D00 SKIP;VD NE V0   0256 3510 SKIP;V5 EQ 10  02A4 A3F3 I=03F3
020A 1214 GO 0214         0258 1360 GO 0360        02A6 F065 V0:V0=MI
020C 70FF V0+FF           025A C703 V7=RND         02A8 4000 SKIP;V0 NE 00
020E 3000 SKIP;V0 EQ 00   025C 4700 SKIP;V7 NE 00  02AA 138C GO 038C
0210 1208 GO 0208         025E 1268 GO 0268        02AC 2314 DO 0314
0212 1204 GO 0204         0260 A3F5 I=03F5         02AE 22F8 DO 02F8
0214 A3F0 I=03F0          0262 F065 V0:V0=MI       02B0 1204 GO 0204
0216 FD1E I=I+VD          0264 4000 SKIP;V0 NE 00  02B2 0100
0218 F065 V0:V0=MI        0266 1364 GO 0364        02B4 1401
021A 3000 SKIP;V0 EQ 00   0268 A3F2 I=03F2         02B6 1C01
021C 1204 GO 0204         026A F065 V0:V0=MI       02B8 2401
021E 22F2 DO 02F2         026C 4001 SKIP;V0 NE 01  02BA 1409
0220 6130 V1=30           026E 1368 GO 0368        02BC 1C09
0222 6002 V0=02           0270 A3F4 I=03F4         02BE 2409
0224 F018 TONE=V0         0272 F065 V0:V0=MI       02C0 1411
0226 C007 V0=RND          0274 4001 SKIP;V0 NE 01  02C2 1C11
0228 F015 TIME=V0         0276 1368 GO 0368        02C4 2411
022A F007 V0=TIME         0278 A3F5 I=03F5         02C6 0104
022C 3000 SKIP;V0 EQ 00   027A F065 V0:V0=MI       02C8 0303
022E 122A GO 022A         027C 4001 SKIP;V0 NE 01  02CA 0203
0230 71FF V1+FF           027E 1368 GO 0368        02CC 0103
0232 3100 SKIP;V1 EQ 00   0280 C703 V7=RND         02CE 0701
0234 1222 GO 0222         0282 4700 SKIP;V7 NE 00  02D0 0401
0236 6403 V4=03           0284 1296 GO 0296        02D2 0101
0238 6C01 VC=01           0286 A3F6 I=03F6         02D4 0302
023A 2390 DO 0390         0288 F065 V0:V0=MI       02D6 4224
023C 3510 SKIP;V5 EQ 10   028A 4001 SKIP;V0 NE 01  02D8 1818
023E 134C GO 034C         028C 1374 GO 0374        02DA 2442
0240 2314 DO 0314         028E A3F8 I=03F8         02DC 7E42
0242 4D00 SKIP;VD NE 00   0290 F065 V0:V0=MI       02DE 4242
0244 1356 GO 0356         0292 4001 SKIP;V0 NE 01  02E0 427E
0246 6402 V4=02           0294 1374 GO 0374        02E2 FFFF
0248 6C02 VC=02           0296 C703 V7=RND         02E4 F803
024A 2390 DO 0390         0298 4700 SKIP;V7 NE 00  02E6 BFF8
024C 3510 SKIP;V5 EQ 10   029A 12A4 GO 02A4        02E8 F0AF
```

(Continued on next page)

7. VIP Tick-Tack-Toe Game (Continued)

```
02EA F800
02EC 5F1F
02EE 8F3A
02F0 EAD4
02F2 6C01 VC=01
02F4 22FC DO 02FC
02F6 00EE RET
02F8 6C02 VC=02
02FA 12F4 GO 02F4
02FC A3F0 I=03F0
02FE FD1E I=I+VD
0300 80C0 V0=VC
0302 F055 MI=V0:V0
0304 A2B2 I=02B2
0306 13D0 GO 03D0
0308 F165 V0:V1=MI
030A A2DC I=02DC
030C 3C01 SKIP;VC EQ 01
030E A2D6 I=02D6
0310 D016 SHOW 6MI@V0V1
0312 00EE RET
0314 6D00 VD=00
0316 6101 V1=01
0318 A3F0 I=03F0
031A F11E I=I+V1
031C F065 V0:V0=MI
031E 4000 SKIP;V0 NE 00
0320 132A GO 032A
0322 4109 SKIP;V1 NE 09
0324 00EE RET
0326 7101 V1+01
0328 1318 GO 0318
032A 8D10 VD=V1
032C 00EE RET
032E A2E2 I=02E2
0330 6303 V3=03
0332 6014 V0=14
0334 6100 V1=00
0336 6218 V2=18
0338 D011 SHOW 1MI@V0V1
033A 72FF V2+FF
033C 7101 V1+01
033E 3200 SKIP;V2 EQ 00
0340 1338 GO 0338
0342 73FF V3+FF
0344 7008 V0+08
0346 3300 SKIP;V3 EQ 00
0348 1334 GO 0334
034A 00EE RET
034C A2DC I=02DC
034E 601C V0=1C
0350 611A V1=1A
0352 D016 SHOW 6MI@V0V1
0354 1354 GO 0354
0356 1358 GO 0358
0358 1358 GO 0358
035A 22F8 DO 02F8
035C A2D6 I=02D6
035E 134E GO 034E
0360 22F8 DO 02F8
0362 1204 GO 0204
0364 6D05 VD=05
0366 1360 GO 0360
0368 A3F1 I=03F1
036A F065 V0:V0=MI
036C 3000 SKIP;V0 EQ 00
036E 1280 GO 0280
0370 6D01 VD=01
0372 1360 GO 0360
0374 A3F9 I=03F9
0376 F065 V0:V0=MI
0378 3000 SKIP;V0 EQ 00
037A 1296 GO 0296
037C 6D09 VD=09
037E 1360 GO 0360
0380 A3F2 I=03F2
0382 F065 V0:V0=MI
0384 3000 SKIP;V0 EQ 00
0386 12A4 GO 02A4
0388 6D02 VD=02
038A 1360 GO 0360
038C 6D03 VD=03
038E 1360 GO 0360
0390 6500 V5=00
0392 A2C6 I=02C6
0394 6603 V6=03
0396 F51E I=I+V5
0398 F165 V0:V1=MI
039A 8200 V2=V0
039C 6300 V3=00
039E 6D00 VD=00
03A0 A3F0 I=03F0
03A2 23BE DO 03BE
03A4 3600 SKIP;V6 EQ 00
03A6 13A0 GO 03A0
03A8 9340 SKIP;V3 NE V4
03AA 13B4 GO 03B4
03AC 7502 V5+02
03AE 4510 SKIP;V5 NE 10
03B0 00EE RET
03B2 1392 GO 0392
03B4 4403 SKIP;V4 NE 03
03B6 00EE RET
03B8 3D00 SKIP;VD EQ 00
03BA 00EE RET
03BC 13AC GO 03AC
03BE F21E I=I+V2
03C0 F065 V0:V0=MI
03C2 90C0 SKIP;V0 NE VC
03C4 7301 V3+01
03C6 4000 SKIP;V0 NE 00
03C8 8D20 VD=V2
03CA 8214 V2=V2+V1
03CC 76FF V6+FF
03CE 00EE RET
03D0 FD1E I=I+VD
03D2 FD1E I=I+VD
03D4 1308 GO 0308
```

8. VIP Spooky Spot

This program uses the CHIP-8 INTERPRETER at locations 0000-01FF. Now you can let the computer make your big decisions or predict the future just like government or industry leaders do.

Flip RUN up. You will see the words YES and NO at the right of the screen. Ask the computer any question that can be answered with YES or NO. Press KEY 0 and the spooky spot will show you the computer's answer. This program replaces your old fashioned mechanical OUIJA board.

```
0200 00E0 ERASE
0202 2242 DO 0242
0204 2254 DO 0254
0206 FA0A VA=KEY
0208 A290 I=0290
020A 6100 V1=00
020C 6210 V2=10
020E D121 SHOW 1MI@V1V2
0210 3F00 SKIP;VF EQ 00
0212 1236 GO 0236
0214 6A04 VA=04
0216 FA18 TONE=VA
0218 6A0A VA=0A
021A FA15 TIME=VA
021C FA07 VA=TIME
021E 3A00 SKIP;VA EQ 00
0220 121C GO 021C
0222 7101 V1+01
0224 CA01 VA=RND
0226 3A01 SKIP;VA EQ 01
0228 6AFF VA=FF
022A 82A4 V2=V2+VA
022C 4207 SKIP;V2 NE 07
022E 7201 V2+01
0230 4218 SKIP;V2 NE 18
0232 72FF V2+FF
0234 120E GO 020E
0236 6A10 VA=10
0238 8A22 VA=VA&V2
023A 3A00 SKIP;VA EQ 00
023C 1240 GO 0240
023E 225A DO 025A
0240 226A DO 026A
0242 A270 I=0270
0244 6408 V4=08
0246 6330 V3=30
0248 D348 SHOW 8MI@V3V4
024A 6A08 VA=08
```

```
024C FA1E I=I+VA
024E 7308 V3+08
0250 D348 SHOW 8MI@V3V4
0252 00EE RET
0254 A280 I=0280
0256 6410 V4=10
0258 1246 GO 0246
025A 6408 V4=08
025C 6331 V3=31
025E A290 I=0290
0260 D348 SHOW 8MI@V3V4
0262 7301 V3+01
0264 3340 SKIP;V3 EQ 40
0266 1260 GO 0260
0268 1268 GO 0268
026A 6410 V4=10
026C 125C GO 025C
026E 0101
0270 7F7F
0272 6A6A
0274 6276
0276 767F
0278 FFFF
027A 23EF
027C 63FB
027E 23FF
0280 7F76
0282 7270
0284 7476
0286 7F7F
0288 FF87
028A B7B7
028C B787
028E FFFF
0290 8080
0292 8080
0294 8080
0296 8080
0298 80D4
```

9. VIP Jackpot

This program uses the CHIP-8 INTERPRETER at 0000-01FF. You start with $10. It costs you $1 each time you play. Push any key to start the 3 wheels spinning. Push keys 1, 2, and 3 (one at a time) to stop the wheels. You win $3 if you stop with 3 different symbols. You win $5 if you stop with 3 identical symbols. You get a $5 bonus for 3 solid squares. You break the bank if you get your winnings up to $50.

```
0200 6E0A VE=0A
0202 00E0 ERASE
0204 601A V0=1A
0206 610B V1=0B
0208 A333 I=0333
020A D017 SHOW 7MI@V0V1
020C 22D6 DO 02D6
020E FF0A VF=KEY
0210 22D6 DO 02D6
0212 7EFF VE+FF
0214 22D6 DO 02D6
0216 6102 V1=02
0218 6216 V2=16
021A 631E V3=1E
021C 6426 V4=26
021E 6501 V5=01
0220 6602 V6=02
0222 6A01 VA=01
0224 6B01 VB=01
0226 6C01 VC=01
0228 6D03 VD=03
022A C70C V7=RND
022C C80C V8=RND
022E C90C V9=RND
0230 22BE DO 02BE
0232 22C6 DO 02C6
0234 22CE DO 02CE
0236 4A00 SKIP;VA NE 00
0238 1240 GO 0240
023A 22BE DO 02BE
023C C70C V7=RND
023E 22BE DO 02BE
0240 4B00 SKIP;VB NE 00
0242 124A GO 024A
0244 22C6 DO 02C6
0246 C80C V8=RND
0248 22C6 DO 02C6
024A 4C00 SKIP;VC NE 00
024C 1254 GO 0254
024E 22CE DO 02CE
0250 C90C V9=RND
0252 22CE DO 02CE
0254 6000 V0=00
0256 80A4 V0=V0+VA
0258 80B4 V0=V0+VB
025A 80C4 V0=V0+VC
```

```
025C 4000 SKIP;V0 NE 00
025E 126E GO 026E
0260 E5A1 SKIP;V5 NE KEY
0262 6A00 VA=00
0264 E6A1 SKIP;V6 NE KEY
0266 6B00 VB=00
0268 EDA1 SKIP;VD NE KEY
026A 6C00 VC=00
026C 1236 GO 0236
026E 6D00 VD=00
0270 8670 V6=V7
0272 8685 V6=V6-V8
0274 4600 SKIP;V6 NE 00
0276 1286 GO 0286
0278 8895 V8=V8-V9
027A 4800 SKIP;V8 NE 00
027C 1292 GO 0292
027E 8795 V7=V7-V9
0280 3700 SKIP;V7 EQ 00
0282 6D03 VD=03
0284 1292 GO 0292
0286 8895 V8=V8-V9
0288 3800 SKIP;V8 EQ 00
028A 1292 GO 0292
028C 4700 SKIP;V7 NE 00
028E 6D05 VD=05
0290 7D05 VD+05
0292 4D00 SKIP;VD NE 00
0294 129C GO 029C
0296 2302 DO 0302
0298 22F6 DO 02F6
029A 2302 DO 0302
029C 22F6 DO 02F6
029E 22D6 DO 02D6
02A0 8ED4 VE=VE+VD
02A2 22D6 DO 02D6
02A4 4E00 SKIP;VE NE 00
02A6 1326 GO 0326
02A8 6631 V6=31
02AA 86E5 V6=V6-VE
02AC 3F00 SKIP;VF EQ 00
02AE 1202 GO 0202
02B0 A368 I=0368
02B2 6419 V4=19
02B4 6518 V5=18
02B6 22EC DO 02EC
```

(Continued on next page)

9. VIP Jackpot (Continued)

```
02B8 22EC DO 02EC
02BA 22F6 DO 02F6
02BC 12B0 GO 02B0
02BE A33A I=033A
02C0 F71E I=I+V7
02C2 D214 SHOW 4MI@V2V1
02C4 00EE RET
02C6 A33A I=033A
02C8 F81E I=I+V8
02CA D314 SHOW 4MI@V3V1
02CC 00EE RET
02CE A33A I=033A
02D0 F91E I=I+V9
02D2 D414 SHOW 4MI@V4V1
02D4 00EE RET
02D6 A400 I=0400
02D8 FE33 MI=VE(3DD)
02DA F265 V0:V2=MI
02DC 601E V0=1E
02DE 630C V3=0C
02E0 F129 I=V1(LSDP)
02E2 D035 SHOW 5MI@V0V3
02E4 7005 V0+05
02E6 F229 I=V2(LSDP)
02E8 D035 SHOW 5MI@V0V3
02EA 00EE RET
02EC D455 SHOW 5MI@V4V5
02EE 6605 V6=05
02F0 F61E I=I+V6
02F2 7408 V4+08
02F4 00EE RET
02F6 6660 V6=60
02F8 F615 TIME=V6
02FA F607 V6=TIME
02FC 3600 SKIP;V6 EQ 00
02FE 12FA GO 02FA
0300 00EE RET
0302 A34A I=034A
0304 640D V4=0D
0306 6518 V5=18
0308 22EC DO 02EC
030A 22EC DO 02EC
030C 22EC DO 02EC
030E 22EC DO 02EC
0310 652A V5=2A
0312 6318 V3=18
```

```
0314 A400 I=0400
0316 FD33 MI=VD(3DD)
0318 F265 V0:V2=MI
031A F129 I=V1(LSDP)
031C D535 SHOW 5MI@V5V3
031E 7505 V5+05
0320 F229 I=V2(LSDP)
0322 D535 SHOW 5MI@V5V3
0324 00EE RET
0326 A35E I=035E
0328 6418 V4=18
032A 6518 V5=18
032C 22EC DO 02EC
032E 22EC DO 02EC
0330 1330 GO 0330
0332 0140
0334 E0C0
0336 E060
0338 E040
033A F0F0
033C F0F0
033E 60F0
0340 F060
0342 9060
0344 6090
0346 F090
0348 90F0
034A F42A
034C 2E2A
034E EAEA
0350 8C8C
0352 8AEA
0354 EEAA
0356 EA8A
0358 8EE0
035A 4040
035C 4040
035E 8E8A
0360 8A8A
0362 EEEE
0364 88EC
0366 28EE
0368 8B89
036A A9F9
036C DBA4
036E 343C
0370 2CA4
```

10. VIP Snake Race

This program uses the CHIP-8 INTERPRETER at 0000-01FF. Flip the RUN switch up to see the four snakes race to the finish line. You and your friends can have hours of fun betting on the winner.

```
0200 6400 V4=00
0202 6500 V5=00
0204 6101 V1=01
0206 F129 I=V1(LSDP)
0208 D455 SHOW 5MI@V4V5
020A 6102 V1=02
020C 6508 V5=08
020E F129 I=V1(LSDP)
0210 D455 SHOW 5MI@V4V5
0212 6103 V1=03
0214 6510 V5=10
0216 F129 I=V1(LSDP)
0218 D455 SHOW 5MI@V4V5
021A 6104 V1=04
021C 6518 V5=18
021E F129 I=V1(LSDP)
0220 D455 SHOW 5MI@V4V5
0222 6105 V1=05
0224 6205 V2=05
0226 6305 V3=05
0228 6405 V4=05
022A 6501 V5=01
022C 6609 V6=09
022E 6711 V7=11
0230 6819 V8=19
0232 A2CF I=02CF
0234 D153 SHOW 3MI@V1V5
0236 D263 SHOW 3MI@V2V6
0238 D373 SHOW 3MI@V3V7
023A D483 SHOW 3MI@V4V8
023C 693F V9=3F
023E 6A00 VA=00
0240 D9A8 SHOW 8MI@V9VA
0242 6A08 VA=08
0244 D9A8 SHOW 8MI@V9VA
0246 6A10 VA=10
0248 D9A8 SHOW 8MI@V9VA
024A 6A18 VA=18
024C D9A5 SHOW 5MI@V9VA
024E C901 V9=RND
0250 3901 SKIP;V9 EQ 01
0252 126E GO 026E
0254 F518 TONE=V5
0256 7101 V1+01
0258 D153 SHOW 3MI@V1V5
025A 603E V0=3E
025C 8015 V0=V0-V1
025E 3000 SKIP;V0 EQ 00
0260 126E GO 026E
0262 D153 SHOW 3MI@V1V5
0264 F715 TIME=V7
0266 FA07 VA=TIME
0268 3A00 SKIP;VA EQ 00
026A 1266 GO 0266
026C 1262 GO 0262
026E C901 V9=RND
0270 3901 SKIP;V9 EQ 01
0272 128E GO 028E
0274 F518 TONE=V5
0276 7201 V2+01
0278 D263 SHOW 3MI@V2V6
027A 603E V0=3E
027C 8025 V0=V0-V2
027E 3000 SKIP;V0 EQ 00
0280 128E GO 028E
0282 D263 SHOW 3MI@V2V6
0284 F715 TIME=V7
0286 FA07 VA=TIME
0288 3A00 SKIP;VA EQ 00
028A 1286 GO 0286
028C 1282 GO 0282
028E C901 V9=RND
0290 3901 SKIP;V9 EQ 01
0292 12AE GO 02AE
0294 F518 TONE=V5
0296 7301 V3+01
0298 D373 SHOW 3MI@V3V7
029A 603E V0=3E
029C 8035 V0=V0-V3
029E 3000 SKIP;V0 EQ 00
02A0 12AE GO 02AE
02A2 D373 SHOW 3MI@V3V7
02A4 F715 TIME=V7
02A6 FA07 VA=TIME
02A8 3A00 SKIP;VA EQ 00
02AA 12A6 GO 02A6
02AC 12A2 GO 02A2
02AE C901 V9=RND
02B0 3901 SKIP;V9 EQ 01
02B2 124E GO 024E
02B4 F518 TONE=V5
02B6 7401 V4+01
02B8 D483 SHOW 3MI@V4V8
02BA 603E V0=3E
02BC 8045 V0=V0-V4
02BE 3000 SKIP;V0 EQ 00
02C0 124E GO 024E
02C2 D483 SHOW 3MI@V4V8
02C4 F715 TIME=V7
02C6 FA07 VA=TIME
02C8 3A00 SKIP;VA EQ 00
02CA 12C6 GO 02C6
02CC 12C2 GO 02C2
02CE 0180
02D0 8080
02D2 8080
02D4 8080
02D6 80D4
```

11. VIP Card Matching Game

This program uses the CHIP-8 INTERPRETER at 0000-01FF. Two players, A and B, alternately try to match up pairs of symbols arranged in a four by four matrix. The positions in the matrix correspond to the arrangement of the input keyboard of the COSMAC VIP. The player whose turn it is will be shown at the left or right of the screen. When a player successfully matches a pair, his letter replaces the symbols and he goes again. The positions of the symbols are shown for a brief time at the beginning of the game. If it is too brief or too long a time, change location 0316 from 6020 to 60——.

```
0200 A385 I=0385
0202 6002 V0=02
0204 6102 V1=02
0206 6202 V2=02
0208 6302 V3=02
020A 6402 V4=02
020C 6502 V5=02
020E 6602 V6=02
0210 6702 V7=02
0212 F755 MI=V0:V7
0214 6300 V3=00
0216 A385 I=0385
0218 C107 V1=RND
021A F11E I=I+V1
021C F065 V0:V0=MI
021E 4000 SKIP;V0 NE 00
0220 1216 GO 0216
0222 70FF V0+FF
0224 A385 I=0385
0226 F11E I=I+V1
0228 F055 MI=V0:V0
022A A38E I=038E
022C F31E I=I+V3
022E 8010 V0=V1
0230 F055 MI=V0:V0
0232 7301 V3+01
0234 3310 SKIP;V3 EQ 10
0236 1216 GO 0216
0238 2314 DO 0314
023A C501 V5=RND
023C 22C4 DO 02C4
023E 6B00 VB=00
0240 6D10 VD=10
0242 F00A V0=KEY
0244 A375 I=0375
0246 F01E I=I+V0
0248 F065 V0:V0=MI
024A 90D0 SKIP;V0 NE VD
024C 1242 GO 0242
024E 8D00 VD=V0
0250 22D8 DO 02D8
0252 3B00 SKIP;VB EQ 00
0254 125E GO 025E
0256 6B0F VB=0F
0258 8CD0 VC=VD
025A 89A0 V9=VA
025C 1242 GO 0242
```

```
025E 6020 V0=20
0260 F015 TIME=V0
0262 F007 V0=TIME
0264 3000 SKIP;V0 EQ 00
0266 1262 GO 0262
0268 99A0 SKIP;V9 NE VA
026A 1278 GO 0278
026C 22C4 DO 02C4
026E 7501 V5+01
0270 6001 V0=01
0272 8502 V5=V5&V0
0274 22A0 DO 02A0
0276 123C GO 023C
0278 6020 V0=20
027A F018 TONE=V0
027C 7E01 VE+01
027E 22A0 DO 02A0
0280 A385 I=0385
0282 FA1E I=I+VA
0284 60DD V0=DD
0286 F055 MI=V0:V0
0288 4500 SKIP;V5 NE 00
028A 1296 GO 0296
028C A367 I=0367
028E D346 SHOW 6MI@V3V4
0290 A367 I=0367
0292 D126 SHOW 6MI@V1V2
0294 12B8 GO 02B8
0296 A33F I=033F
0298 D346 SHOW 6MI@V3V4
029A A33F I=033F
029C D126 SHOW 6MI@V1V2
029E 12B8 GO 02B8
02A0 22D8 DO 02D8
02A2 8130 V1=V3
02A4 8240 V2=V4
02A6 8DC0 VD=VC
02A8 22D8 DO 02D8
02AA 00EE RET
02AC A36D I=036D
02AE FA1E I=I+VA
02B0 F065 V0:V0=MI
02B2 A334 I=0334
02B4 F01E I=I+V0
02B6 00EE RET
02B8 3E07 SKIP;VE EQ 07
02BA 123E GO 023E
```

(Continued on next page)

11. VIP Card Matching Game (Continued)

```
02BC 22C4 DO 02C4
02BE 6060 V0=60
02C0 F018 TONE=V0
02C2 12C2 GO 02C2
02C4 6300 V3=00
02C6 6408 V4=08
02C8 A33F I=033F
02CA 4500 SKIP;V5 NE 00
02CC 12D2 GO 02D2
02CE 633A V3=3A
02D0 A367 I=0367
02D2 D346 SHOW 6MI@V3V4
02D4 00EE RET
02D6 5555 SKIP;V5 EQ V5
02D8 A38E I=038E
02DA FD1E I=I+VD
02DC F065 V0:V0=MI
02DE 8A00 VA=V0
02E0 A385 I=0385
02E2 F01E I=I+V0
02E4 F065 V0:V0=MI
02E6 40DD SKIP;V0 NE DD
02E8 1242 GO 0242
02EA 22AC DO 02AC
02EC 6310 V3=10
02EE 6400 V4=00
02F0 600C V0=0C
02F2 80D2 V0=V0&VD
02F4 4004 SKIP;V0 NE 04
02F6 6408 V4=08
02F8 4008 SKIP;V0 NE 08
02FA 6410 V4=10
02FC 400C SKIP;V0 NE 0C
02FE 6418 V4=18
0300 6003 V0=03
0302 80D2 V0=V0&VD
0304 4001 SKIP;V0 NE 01
0306 6318 V3=18
0308 4002 SKIP;V0 NE 02
030A 6320 V3=20
030C 4003 SKIP;V0 NE 03
030E 6328 V3=28
0310 D346 SHOW 6MI@V3V4
0312 00EE RET
0314 2324 DO 0324
0316 6020 V0=20
0318 F015 TIME=V0
031A F007 V0=TIME
031C 3000 SKIP;V0 EQ 00
031E 131A GO 031A
```

```
0320 2324 DO 0324
0322 00EE RET
0324 6D00 VD=00
0326 22D8 DO 02D8
0328 7D01 VD+01
032A 4D10 SKIP;VD NE 10
032C 1330 GO 0330
032E 1326 GO 0326
0330 00EE RET
0332 0101
0334 1010
0336 1E78
0338 0808
033A 1818
033C 7E7E
033E 1818
0340 2424
0342 3C24
0344 2466
0346 6618
0348 1866
034A 667E
034C 2424
034E 7E66
0350 4224
0352 1818
0354 2442
0356 7E52
0358 5252
035A 527E
035C 4242
035E 7E42
0360 7E14
0362 7C26
0364 643E
0366 287C
0368 243C
036A 2424
036C 7C00
036E 0611
0370 161C
0372 2227
0374 2D0D
0376 0001
0378 0204
037A 0506
037C 0809
037E 0A0C
0380 0E03
0382 070B
0384 0FD4
```

12. VIP Armored Vehicle Clash

This program uses the CHIP-8 INTERPRETER at 0000-01FF. At the start of the game and after every score change, the score, on the left, and number of shots remaining, on the right, are shown. The tank may be moved by pressing keys 2, 4, 6, or 8 for up, left, right, or down, respectively. To fire a shell press key F. After the score is shown the target will come on the screen at one of eight positions and change direction randomly. Every time you hit the target you score 10 points, but if you are hit by the target you lose 5 shots.

```
0200 6E00 VE=00
0202 6DA0 VD=A0
0204 6A08 VA=08
0206 6906 V9=06
0208 6804 V8=04
020A 6702 V7=02
020C 6619 V6=19
020E 6410 V4=10
0210 630C V3=0C
0212 6200 V2=00
0214 6106 V1=06
0216 A412 I=0412
0218 FA55 MI=V0:VA
021A 23A4 DO 03A4
021C 6040 V0=40
021E F015 TIME=V0
0220 F007 V0=TIME
0222 3000 SKIP;V0 EQ 00
0224 1220 GO 0220
0226 23A4 DO 03A4
0228 22DA DO 02DA
022A 2332 DO 0332
022C A412 I=0412
022E F565 V0:V5=MI
0230 227E DO 027E
0232 2296 DO 0296
0234 22BC DO 02BC
0236 3F01 SKIP;VF EQ 01
0238 22E4 DO 02E4
023A 3F01 SKIP;VF EQ 01
023C 22BC DO 02BC
023E 3F01 SKIP;VF EQ 01
0240 22BC DO 02BC
0242 3F01 SKIP;VF EQ 01
0244 224C DO 024C
0246 4F01 SKIP;VF NE 01
0248 1336 GO 0336
024A 1232 GO 0232
024C A412 I=0412
024E F565 V0:V5=MI
0250 4600 SKIP;V6 NE 00
0252 3500 SKIP;V5 EQ 00
0254 1258 GO 0258
0256 135C GO 035C
0258 E7A1 SKIP;V7 NE KEY
025A 6202 V2=02
025C E8A1 SKIP;V8 NE KEY
025E 6204 V2=04
0260 E9A1 SKIP;V9 NE KEY
0262 6206 V2=06
0264 EAA1 SKIP;VA NE KEY
0266 6208 V2=08
0268 4200 SKIP;V2 NE 00
026A 00EE RET
026C 227E DO 027E
026E 8120 V1=V2
0270 236A DO 036A
0272 237C DO 037C
0274 6C01 VC=01
0276 6200 V2=00
0278 6F00 VF=00
027A A412 I=0412
027C F555 MI=V0:V5
027E A3CF I=03CF
0280 4102 SKIP;V1 NE 02
0282 6000 V0=00
0284 4104 SKIP;V1 NE 04
0286 6013 V0=13
0288 4106 SKIP;V1 NE 06
028A 600D V0=0D
028C 4108 SKIP;V1 NE 08
028E 6006 V0=06
0290 F01E I=I+V0
0292 D347 SHOW 7MI@V3V4
0294 00EE RET
0296 600F V0=0F
0298 E09E SKIP;V0 EQ KEY
029A 00EE RET
029C 450F SKIP;V5 NE 0F
029E 00EE RET
02A0 650F V5=0F
02A2 76FF V6+FF
02A4 A412 I=0412
02A6 F555 MI=V0:V5
02A8 7403 V4+03
02AA 7303 V3+03
02AC 236A DO 036A
02AE 236A DO 036A
02B0 236A DO 036A
02B2 A423 I=0423
02B4 F555 MI=V0:V5
02B6 A3E9 I=03E9
02B8 D341 SHOW 1MI@V3V
02BA 00EE RET
02BC A423 I=0423
02BE F565 V0:V5=MI
02C0 4500 SKIP;V5 NE 0
02C2 00EE RET
02C4 A3E9 I=03E9
02C6 D341 SHOW 1MI@V3V
02C8 236A DO 036A
02CA 6C02 VC=02
02CC 238E DO 038E
02CE 4BBB SKIP;VB NE B
02D0 12DA GO 02DA
02D2 D341 SHOW 1MI@V3V
02D4 A423 I=0423
02D6 F555 MI=V0:V5
02D8 00EE RET
02DA 6500 V5=00
02DC 6000 V0=00
02DE A417 I=0417
02E0 F055 MI=V0:V0
02E2 12D4 GO 02D4
02E4 A41D I=041D
02E6 F565 V0:V5=MI
02E8 350F SKIP;V5 EQ 0
02EA 1314 GO 0314
02EC A3EA I=03EA
02EE D345 SHOW 5MI@V3V
02F0 3200 SKIP;V2 EQ 0
02F2 1302 GO 0302
02F4 C103 V1=RND
02F6 A419 I=0419
02F8 F11E I=I+V1
02FA F065 V0:V0=MI
02FC 8100 V1=V0
02FE C20F V2=RND
0300 7201 V2+01
0302 236A DO 036A
0304 A3EA I=03EA
0306 6C03 VC=03
0308 72FF V2+FF
030A 6F00 VF=00
```

(Continued on next page)

12. VIP Armored Vehicle Clash (Continued)

```
030C D345 SHOW 5MI@V3V4
030E A41D I=041D
0310 F555 MI=V0:V5
0312 00EE RET
0314 C407 V4=RND
0316 A3EF I=03EF
0318 F41E I=I+V4
031A F065 V0:V0=MI
031C 8300 V3=V0
031E A3F7 I=03F7
0320 F41E I=I+V4
0322 F065 V0:V0=MI
0324 8400 V4=V0
0326 A3EA I=03EA
0328 D345 SHOW 5MI@V3V4
032A 6020 V0=20
032C F018 TONE=V0
032E 650F V5=0F
0330 130E GO 030E
0332 6500 V5=00
0334 130E GO 030E
0336 4C01 SKIP;VC NE 01
0338 1400 GO 0400
033A 4C02 SKIP;VC NE 02
033C 1352 GO 0352
033E A423 I=0423
0340 F565 V0:V5=MI
0342 4500 SKIP;V5 NE 00
0344 1400 GO 0400
0346 A3E9 I=03E9
0348 D341 SHOW 1MI@V3V4
034A 6F00 VF=00
034C D341 SHOW 1MI@V3V4
034E 3F01 SKIP;VF EQ 01
0350 1400 GO 0400
0352 7E0A VE+0A
0354 6040 V0=40
0356 F018 TONE=V0
0358 00E0 ERASE
035A 121A GO 021A
035C 00E0 ERASE
035E 23A4 DO 03A4
0360 6060 V0=60
0362 F018 TONE=V0
0364 1364 GO 0364
0366 6E00 VE=00
0368 1354 GO 0354
036A 4102 SKIP;V1 NE 02
036C 74FF V4+FF
036E 4104 SKIP;V1 NE 04
0370 73FF V3+FF
0372 4106 SKIP;V1 NE 06
0374 7301 V3+01
0376 4108 SKIP;V1 NE 08
0378 7401 V4+01
037A 00EE RET
037C 4400 SKIP;V4 NE 00
037E 7401 V4+01
0380 4300 SKIP;V3 NE 00
0382 7301 V3+01
0384 4338 SKIP;V3 NE 38
0386 73FF V3+FF
0388 4418 SKIP;V4 NE 18
038A 74FF V4+FF
038C 00EE RET
038E 6B00 VB=00
0390 4400 SKIP;V4 NE 00
0392 139E GO 039E
0394 4300 SKIP;V3 NE 00
0396 139E GO 039E
0398 433F SKIP;V3 NE 3F
039A 139E GO 039E
039C 441F SKIP;V4 NE 1F
039E 6BBB VB=BB
03A0 6F00 VF=00
03A2 00EE RET
03A4 6308 V3=08
03A6 6408 V4=08
03A8 A429 I=0429
03AA FE33 MI=VE(3DD)
03AC F265 V0:V2=MI
03AE 23BC DO 03BC
03B0 6328 V3=28
03B2 A429 I=0429
03B4 F633 MI=V6(3DD)
03B6 F265 V0:V2=MI
03B8 23C2 DO 03C2
03BA 00EE RET
03BC F029 I=V0(LSDP)
03BE D345 SHOW 5MI@V3V4
03C0 7306 V3+06
03C2 F129 I=V1(LSDP)
03C4 D345 SHOW 5MI@V3V4
03C6 7306 V3+06
03C8 F229 I=V2(LSDP)
03CA D345 SHOW 5MI@V3V4
03CC 00EE RET
03CE 0110
03D0 547C
03D2 6C7C
03D4 7C44
03D6 7C7C
03D8 6C7C
03DA 5410
03DC 00FC
03DE 786E
03E0 78FC
03E2 003F
03E4 1E76
03E6 1E3F
03E8 0080
03EA A870
03EC F870
03EE A80B
03F0 1B28
03F2 3830
03F4 2010
03F6 0000
03F8 0000
03FA 081B
03FC 1B1B
03FE 13D4
0400 76FB V6+FB
0402 6020 V0=20
0404 8065 V0=V0-V6
0406 4F00 SKIP;VF NE 00
0408 6600 V6=00
040A 1354 GO 0354
```

13. VIP Hi-Lo

This program uses the CHIP-8 INTERPRETER at 0000-01FF. You have 10 chances to guess the value of a random number between 00 and 99 selected by the program. The number at the right of the screen shows the number of the guess you are using. Enter a two digit number and the computer tells you if you are high or low. Press any key to erase this number and then, try again. If you have failed after ten guesses, press any key and the number will be shown. If you are good you will never need more than seven guesses. If you are not so good, alter the program to allow more guesses by changing location 0292 from 4E0A to 4E99.

```
0200 6C09 VC=09
0202 CD0F VD=RND
0204 8CD5 VC=VC-VD
0206 4F00 SKIP;VF NE 00
0208 1200 GO 0200
020A 89D0 V9=VD
020C 6C09 VC=09
020E CD0F VD=RND
0210 8CD5 VC=VC-VD
0212 4F00 SKIP;VF NE 00
0214 120C GO 020C
0216 8AD0 VA=VD
0218 6E00 VE=00
021A A2AA I=02AA
021C 7E01 VE+01
021E FE33 MI=VE(3DD)
0220 F265 V0:V2=MI
0222 6B30 VB=30
0224 6C10 VC=10
0226 680F V8=0F
0228 F129 I=V1(LSDP)
022A DBC5 SHOW 5MI@VBVC
022C 7B05 VB+05
022E F229 I=V2(LSDP)
0230 DBC5 SHOW 5MI@VBVC
0232 4800 SKIP;V8 NE 00
0234 1254 GO 0254
0236 660A V6=0A
0238 F10A V1=KEY
023A 8165 V1=V1-V6
023C 3F00 SKIP;VF EQ 00
023E 1236 GO 0236
0240 710A V1+0A
0242 660A V6=0A
0244 F20A V2=KEY
0246 8265 V2=V2-V6
0248 3F00 SKIP;VF EQ 00
024A 1242 GO 0242
024C 720A V2+0A
024E 6B10 VB=10
0250 6800 V8=00
0252 1228 GO 0228
```

```
0254 8195 V1=V1-V9
0256 3100 SKIP;V1 EQ 00
0258 1272 GO 0272
025A 82A5 V2=V2-VA
025C 3200 SKIP;V2 EQ 00
025E 1286 GO 0286
0260 6B20 VB=20
0262 6518 V5=18
0264 F929 I=V9(LSDP)
0266 DBC5 SHOW 5MI@VBVC
0268 7B05 VB+05
026A FA29 I=VA(LSDP)
026C DBC5 SHOW 5MI@VBVC
026E FC18 TONE=VC
0270 1270 GO 0270
0272 65F0 V5=F0
0274 8152 V1=V1&V5
0276 3100 SKIP;V1 EQ 00
0278 128E GO 028E
027A A29F I=029F
027C 6B10 VB=10
027E 6C18 VC=18
0280 DBC5 SHOW 5MI@VBVC
0282 F60A V6=KEY
0284 1292 GO 0292
0286 65F0 V5=F0
0288 8252 V2=V2&V5
028A 4200 SKIP;V2 NE 00
028C 127A GO 027A
028E A2A4 I=02A4
0290 127C GO 027C
0292 4E0A SKIP;VE NE 0A
0294 129A GO 029A
0296 00E0 ERASE
0298 121A GO 021A
029A DBC5 SHOW 5MI@VBVC
029C 1260 GO 0260
029E 0197
02A0 92F2
02A2 9297
02A4 8F89
02A6 8989
02A8 EFD4
```

14. VIP Hex Reflex

This program uses the CHIP-8 INTERPRETER at 0000-01FF. Key 1 selects decimal-to-hexadecimal conversion. Key 2 selects binary-to-hexadecimal conversion. Convert the decimal or binary number as quickly as possible and press the corresponding hexadecimal key. UA is the random number counter. M(0225) is the limit of the count for the random numbers. By changing this memory location, the amount of random numbers per game can be increased or decreased. The score is a function of your response time. The faster you respond, the higher the score.

```
0200 F80A V8=KEY
0202 3801 SKIP;V8 EQ 01
0204 4802 SKIP;V8 NE 02
0206 120A GO 020A
0208 1200 GO 0200
020A 6700 V7=00
020C 6A00 VA=00
020E 00E0 ERASE
0210 A2DD I=02DD
0212 F733 MI=V7(3DD)
0214 A2DD I=02DD
0216 F665 V0:V6=MI
0218 F029 I=V0(LSDP)
021A D435 SHOW 5MI@V4V3
021C F129 I=V1(LSDP)
021E D535 SHOW 5MI@V5V3
0220 F229 I=V2(LSDP)
0222 D635 SHOW 5MI@V6V3
0224 4A0F SKIP;VA NE 0F
0226 1200 GO 0200
0228 C90F V9=RND
022A 4801 SKIP;V8 NE 01
022C 2296 DO 0296
022E 4802 SKIP;V8 NE 02
0230 22A8 DO 02A8
0232 6BFF VB=FF
0234 FB15 TIME=VB
0236 7A01 VA+01
0238 6000 V0=00
023A E0A1 SKIP;V0 NE KEY
023C 1252 GO 0252
023E 7001 V0+01
0240 4010 SKIP;V0 NE 10
0242 1238 GO 0238
0244 FB07 VB=TIME
0246 3B00 SKIP;VB EQ 00
0248 123A GO 023A
024A 6C80 VC=80
024C FC18 TONE=VC
024E 2280 DO 0280
0250 120E GO 020E
0252 5090 SKIP;V0 EQ V9
0254 124A GO 024A
0256 6C10 VC=10
0258 FC18 TONE=VC
025A A2E4 I=02E4
025C F265 V0:V2=MI
025E 63F0 V3=F0
0260 83B2 V3=V3&VB
0262 3300 SKIP;V3 EQ 00
0264 126A GO 026A
0266 7701 V7+01
0268 127C GO 027C
026A 81B2 V1=V1&VB
026C 8114 V1=V1+V1
026E 83F0 V3=VF
0270 8224 V2=V2+V2
0272 8234 V2=V2+V3
0274 7001 V0+01
0276 3004 SKIP;V0 EQ 04
0278 126C GO 026C
027A 8724 V7=V7+V2
027C 2280 DO 0280
027E 120E GO 020E
0280 A2E7 I=02E7
0282 F365 V0:V3=MI
0284 D013 SHOW 3MI@V0V1
0286 F929 I=V9(LSDP)
0288 D235 SHOW 5MI@V2V3
028A 6B80 VB=80
028C FB15 TIME=VB
028E FB07 VB=TIME
0290 3B00 SKIP;VB EQ 00
0292 128E GO 028E
0294 00EE RET
0296 A2DD I=02DD
0298 F933 MI=V9(3DD)
029A A2DA I=02DA
029C F565 V0:V5=MI
029E F429 I=V4(LSDP)
02A0 D025 SHOW 5MI@V0V2
02A2 F529 I=V5(LSDP)
02A4 D125 SHOW 5MI@V1V2
02A6 00EE RET
02A8 A2D3 I=02D3
02AA F665 V0:V6=MI
02AC 6E08 VE=08
02AE 22C6 DO 02C6
02B0 D045 SHOW 5MI@V0V4
02B2 6E04 VE=04
02B4 22C6 DO 02C6
02B6 D145 SHOW 5MI@V1V4
02B8 6E02 VE=02
02BA 22C6 DO 02C6
02BC D245 SHOW 5MI@V2V4
02BE 6E01 VE=01
02C0 22C6 DO 02C6
02C2 D345 SHOW 5MI@V3V4
02C4 00EE RET
02C6 8E92 VE=VE&V9
02C8 4E00 SKIP;VE NE 00
02CA F529 I=V5(LSDP)
02CC 3E00 SKIP;VE EQ 00
02CE F629 I=V6(LSDP)
02D0 00EE RET
02D2 0114
02D4 1A20
02D6 260D
02D8 0001
02DA 2026
02DC 0D00
02DE 0009
02E0 0030
02E2 363C
02E4 00F0
02E6 002B
02E8 0E30
02EA 0DE0
02EC 00E0
02EE 00D4
```

15. VIP Dot-Dash

This program uses the CHIP-8 INTERPRETER at 0000-01FF. The track or obstacle pattern is copied from 0400-04FF into the display page. The direction of the dot is controlled by keys 2, 4, 6, and 8. The dot is accelerated so long as key 0 is not pressed. Key 0 is used as a brake. New tracks or obstacle patterns can be created by using the VIP Video Display Drawing Game. When you create new patterns, the timer area (upper right corner) should be left blank. The dot starts at the left and the finish is any opening on the right edge of the display. The dot can wrap around at the top and bottom but not the right or left edges. A crash stops the clock and the dot blinks at the crash site. The finish blinks the final clock time.

```
0200 A2EC I=02EC
0202 FE65 V0:VE=MI
0204 FE18 TONE=VE
0206 00E0 ERASE
0208 A400 I=0400
020A 6400 V4=00
020C D451 SHOW 1MI@V4V5
020E F71E I=I+V7
0210 8464 V4=V4+V6
0212 633F V3=3F
0214 8342 V3=V3&V4
0216 3300 SKIP;V3 EQ 00
0218 120C GO 020C
021A 8574 V5=V5+V7
021C 631F V3=1F
021E 8352 V3=V3&V5
0220 3300 SKIP;V3 EQ 00
0222 120A GO 020A
0224 122A GO 022A
0226 22CE DO 02CE
0228 7C01 VC+01
022A 22CE DO 02CE
022C 6D15 VD=15
022E FD15 TIME=VD
0230 FD07 VD=TIME
0232 4D00 SKIP;VD NE 00
0234 1226 GO 0226
0236 A2EC I=02EC
0238 F365 V0:V3=MI
023A E0A1 SKIP;V0 NE KEY
023C 8B00 VB=V0
023E E1A1 SKIP;V1 NE KEY
0240 8B10 VB=V1
0242 E2A1 SKIP;V2 NE KEY
0244 8B20 VB=V2
0246 E3A1 SKIP;V3 NE KEY
0248 8B30 VB=V3
024A 1254 GO 0254
024C 7602 V6+02
024E FD07 VD=TIME
0250 56D0 SKIP;V6 EQ VD
0252 124E GO 024E
0254 7701 V7+01
0256 57AD SKIP;V7 EQ VA
0258 1230 GO 0230
025A 67FF V7=FF
025C 6100 V1=00
025E E1A1 SKIP;V1 NE KEY
0260 126A GO 026A
0262 7AFF VA+FF
0264 4AFF SKIP;VA NE FF
0266 6A00 VA=00
0268 1270 GO 0270
026A 7A01 VA+01
026C 4A00 SKIP;VA NE 00
026E 6AFF VA=FF
0270 A2FB I=02FB
0272 D891 SHOW 1MI@V8V9
0274 3F01 SKIP;VF EQ 01
0276 D891 SHOW 1MI@V8V9
0278 3B04 SKIP;VB EQ 04
027A 1282 GO 0282
027C 3800 SKIP;V8 EQ 00
027E 78FF V8+FF
0280 8000 V0=V0
0282 4B06 SKIP;VB NE 06
0284 7801 V8+01
0286 4B02 SKIP;VB NE 02
0288 79FF V9+FF
028A 4B08 SKIP;VB NE 08
028C 7901 V9+01
028E 6FFF VF=FF
0290 A2FB I=02FB
0292 D891 SHOW 1MI@V8V9
0294 4F01 SKIP;VF NE 01
0296 12BA GO 02BA
0298 383F SKIP;V8 EQ 3F
029A 1230 GO 0230
029C 6E80 VE=80
029E FE18 TONE=VE
02A0 22CE DO 02CE
02A2 6D20 VD=20
02A4 FD15 TIME=VD
02A6 FD07 VD=TIME
02A8 3D00 SKIP;VD EQ 00
02AA 12A6 GO 02A6
02AC 22CE DO 02CE
02AE 6D40 VD=40
02B0 FD15 TIME=VD
02B2 FD07 VD=TIME
02B4 3D00 SKIP;VD EQ 00
02B6 12B2 GO 02B2
02B8 12A0 GO 02A0
02BA A2FB I=02FB
02BC D891 SHOW 1MI@V8V9
02BE 6E10 VE=10
02C0 FE18 TONE=VE
02C2 6D20 VD=20
02C4 FD15 TIME=VD
02C6 FD07 VD=TIME
02C8 3D00 SKIP;VD EQ 00
02CA 12C6 GO 02C6
02CC 12BA GO 02BA
02CE A2FD I=02FD
02D0 FC33 MI=VC(3DD)
02D2 F265 V0:V2=MI
02D4 F229 I=V2(LSDP)
02D6 643C V4=3C
02D8 6500 V5=00
02DA D455 SHOW 5MI@V4V5
02DC 6436 V4=36
02DE F129 I=V1(LSDP)
02E0 D455 SHOW 5MI@V4V5
02E2 6430 V4=30
02E4 F029 I=V0(LSDP)
02E6 D455 SHOW 5MI@V4V5
02E8 00EE RET
02EA 0100
02EC 0406
02EE 0208
02F0 0000
02F2 0801
02F4 000E
02F6 1506
02F8 0030
02FA 2080
02FC D4D4
02FE 0100
```

(Continued on next page)

15. VIP Dot-Dash (Continued)

```
0400 FFC0        0456 0387        04AC 0100
0402 0000        0458 0070        04AE 0003
0404 3F82        045A 03E0        04B0 0000
0406 0000        045C 001F        04B2 C002
0408 8FC0        045E 0003        04B4 8500
040A 0000        0460 0020        04B6 70E1
040C 3F82        0462 03FF        04B8 0000
040E 0000        0464 FFFF        04BA C002
0410 DF00        0466 0003        04BC 4900
0412 3000        0468 0020        04BE F9F0
0414 3F82        046A 03E0        04C0 0000
0416 0000        046C 001F        04C2 C001
0418 DF00        046E 0001        04C4 3200
041A 0000        0470 0020        04C6 3F78
041C 7F82        0472 01C0        04C8 0000
041E 0000        0474 000E        04CA C001
0420 F800        0476 0001        04CC 8400
0422 7800        0478 07FF        04CE 0FF8
0424 FF82        047A 0080        04D0 0000
0426 0000        047C 0004        04D2 C000
0428 8800        047E 1DC0        04D4 7830
042A 7801        0480 03FE        04D6 0788
042C AA82        0482 0000        04D8 0000
042E 0000        0484 0000        04DA C000
0430 B83F        0486 0880        04DC 0078
0432 F800        0488 00F8        04DE C708
0434 2A83        048A 0000        04E0 0003
0436 FFFF        048C 7800        04E2 F00E
0438 883F        048E 3FC0        04E4 00FC
043A F800        0490 0020        04E6 7F08
043C 2A80        0492 0001        04E8 000F
043E 010F        0494 8600        04EA FC1F
0440 F800        0496 0880        04EC 00FC
0442 0000        0498 0020        04EE 1FF8
0444 2A00        049A 0001        04F0 003F
0446 010F        049C 0200        04F2 FFFF
0448 F000        049E 1DC0        04F4 8078
044A 0080        04A0 0020        04F6 07C0
044C 0804        04A2 C002        04F8 003F
044E 0387        04A4 4900        04FA FFFF
0450 F0F8        04A6 0001        04FC C031
0452 01C0        04A8 0000        04FE FFFF
0454 000E        04AA C002        0500 00D4
```

16. VIP A-Mazing

This program uses the CHIP-8 INTERPRETER at 0000-01FF. Key 1 or key 2 starts the program. Key 1 is used to generate a maze at 0400-04FF. Key 2 skips the generation of a maze. The maze pattern at 0400-04FF is copied into the display page. Traversing the maze is controlled by keys 2, 4, 6, or 8. The spot always starts on the left (the 15th line (0E-hex) from the top), and the finish is any opening on the right border. The maze wraps around at the top and bottom but not from left-to-right or right-to-left. An internal clock keeps track of the time used to traverse the maze but is also incremented whenever a collision occurs. This clock is displayed in the upper right corner when the end of the maze is reached. The background pattern may be changed by changing 0211 to: 8F for a checker-board pattern; 90 for a cross-hatch pattern; and 91 for a solid pattern. The starting location can be changed by setting 0381 and 0388 to the X-coordinate and setting 0382 and 0389 to the Y-coordinate. V6=M(0386) and V7=M(0387) are parameters used in generating a new maze. V6 is used to determine how often moving to the left of the screen is disallowed (1/V6). V7 is used to determine the length of randomly occurring excursions. M(0251) is the bit mask which is used to set the probability of excursions occurring. Maze patterns can be saved on cassette tape and reloaded into 0400-04FF using the operating system.

```
0200 6001 V0=01
0202 6102 V1=02
0204 E1A1 SKIP;V1 NE KEY
0206 1280 GO 0280
0208 E09E SKIP;V0 EQ KEY
020A 1204 GO 0204
020C 036C MLS@036C
020E 00E0 ERASE
0210 A391 I=0391
0212 6100 V1=00
0214 6000 V0=00
0216 D012 SHOW 2MI@V0V1
0218 7008 V0+08
021A 6240 V2=40
021C 8205 V2=V2-V0
021E 3200 SKIP;V2 EQ 00
0220 1216 GO 0216
0222 7102 V1+02
0224 6220 V2=20
0226 8215 V2=V2-V1
0228 3200 SKIP;V2 EQ 00
022A 1214 GO 0214
022C A380 I=0380
022E F965 V0:V9=MI
0230 CB03 VB=RND
0232 4600 SKIP;V6 NE 00
0234 1240 GO 0240
0236 73FF V3+FF
0238 3300 SKIP;V3 EQ 00
023A 1240 GO 0240
023C 8360 V3=V6
023E 7B01 VB+01
0240 A377 I=0377
0242 D891 SHOW 1MI@V8V9
0244 3F01 SKIP;VF EQ 01
0246 D891 SHOW 1MI@V8V9
0248 4700 SKIP;V7 NE 00
024A 1266 GO 0266
024C 5470 SKIP;V4 EQ V7
024E 125A GO 025A
0250 C501 V5=RND
0252 3500 SKIP;V5 EQ 00
0254 1266 GO 0266
0256 8180 V1=V8
0258 8290 V2=V9
025A 74FF V4+FF
025C 3400 SKIP;V4 EQ 00
025E 1266 GO 0266
0260 8810 V8=V1
0262 8920 V9=V2
0264 8470 V4=V7
0266 2332 DO 0332
0268 4FED SKIP;VF NE ED
026A 1278 GO 0278
026C D891 SHOW 1MI@V8V9
026E 3F01 SKIP;VF EQ 01
0270 D891 SHOW 1MI@V8V9
0272 2332 DO 0332
0274 3FED SKIP;VF EQ ED
0276 1230 GO 0230
0278 D891 SHOW 1MI@V8V9
027A 3F01 SKIP;VF EQ 01
027C D891 SHOW 1MI@V8V9
027E 0373 MLS@0373
0280 00E0 ERASE
0282 A400 I=0400
0284 6001 V0=01
0286 6200 V2=00
0288 6100 V1=00
028A D121 SHOW 1MI@V1V2
028C F01E I=I+V0
028E 7108 V1+08
0290 6540 V5=40
0292 8515 V5=V5-V1
0294 3500 SKIP;V5 EQ 00
0296 128A GO 028A
0298 7201 V2+01
029A 6520 V5=20
029C 8525 V5=V5-V2
029E 3500 SKIP;V5 EQ 00
02A0 1288 GO 0288
02A2 A380 I=0380
```

(Continued on next page)

16. VIP A-Mazing (Continued)

```
02A4 FE65 V0:VE=MI          02F4 D891 SHOW 1MI@V8V9     0344 6900 V9=00
02A6 A377 I=0377            02F6 4402 SKIP;V4 NE 02     0346 48FF SKIP;V8 NE FF
02A8 D891 SHOW 1MI@V8V9     02F8 6B01 VB=01             0348 6800 V8=00
02AA FE18 TONE=VE           02FA 4401 SKIP;V4 NE 01     034A 4840 SKIP;V8 NE 40
02AC 6D40 VD=40             02FC 6B02 VB=02             034C 683F V8=3F
02AE FD15 TIME=VD           02FE 4400 SKIP;V4 NE 00     034E 483F SKIP;V8 NE 3F
02B0 7C01 VC+01             0300 6B03 VB=03             0350 6FED VF=ED
02B2 FD07 VD=TIME           0302 4403 SKIP;V4 NE 03     0352 00EE RET
02B4 4D00 SKIP;VD NE 00     0304 6B00 VB=00             0354 0100
02B6 12AC GO 02AC           0306 7C01 VC+01             0356 9BBD
02B8 85A0 V5=VA             0308 FE18 TONE=VE           0358 F806
02BA 85D2 V5=V5&VD          030A 12E8 GO 02E8           035A ADAF
02BC 3500 SKIP;V5 EQ 00     030C D891 SHOW 1MI@V8V9     035C F800
02BE 12B2 GO 02B2           030E 4F01 SKIP;VF NE 01     035E 5D1D
02C0 D891 SHOW 1MI@V8V9     0310 12F4 GO 02F4           0360 5D8D
02C2 6BBD VB=BD             0312 12B2 GO 02B2           0362 FC07
02C4 6502 V5=02             0314 0356 MLS@0356          0364 AD2F
02C6 E5A1 SKIP;V5 NE KEY    0316 A378 I=0378            0366 8F3A
02C8 6B02 VB=02             0318 FC33 MI=VC(3DD)        0368 5CD4
02CA 6508 V5=08             031A A378 I=0378            036A 0100
02CC E5A1 SKIP;V5 NE KEY    031C F665 V0:V6=MI          036C 9BBE
02CE 6B01 VB=01             031E F029 I=V0(LSDP)        036E F804
02D0 6504 V5=04             0320 D435 SHOW 5MI@V4V3     0370 BBD4
02D2 E5A1 SKIP;V5 NE KEY    0322 F129 I=V1(LSDP)        0372 109E
02D4 6B00 VB=00             0324 D535 SHOW 5MI@V5V3     0374 BBD4
02D6 6506 V5=06             0326 F229 I=V2(LSDP)        0376 0180
02D8 E5A1 SKIP;V5 NE KEY    0328 D635 SHOW 5MI@V6V3     0378 0007
02DA 6B03 VB=03             032A 6E80 VE=80             037A 0500
02DC 84B0 V4=VB             032C FE18 TONE=VE           037C 3237
02DE 4BBD SKIP;VB NE BD     032E F00A V0=KEY            037E 3C00
02E0 12B2 GO 02B2           0330 1200 GO 0200           0380 0000
02E2 D891 SHOW 1MI@V8V9     0332 4B00 SKIP;VB NE 00     0382 0E02
02E4 3F01 SKIP;VF EQ 01     0334 78FF V8+FF             0384 0000
02E6 D891 SHOW 1MI@V8V9     0336 4B01 SKIP;VB NE 01     0386 0220
02E8 2332 DO 0332           0338 7901 V9+01             0388 000E
02EA 3FED SKIP;VF EQ ED     033A 4B02 SKIP;VB NE 02     038A 0F06
02EC 130C GO 030C           033C 79FF V9+FF             038C 0010
02EE D891 SHOW 1MI@V8V9     033E 4B03 SKIP;VB NE 03     038E 20AA
02F0 3F01 SKIP;VF EQ 01     0340 7801 V8+01             0390 55FF
02F2 1314 GO 0314           0342 4920 SKIP;V9 NE 20     0392 FFD4
```

17. VIP Deduce

This program uses the CHIP-8 INTERPRETER at 0000-01FF. This game is an old favorite, described as BAGELS in David Ahl's "101 Computer Games"; "What to Do After You Hit Return", p. 10 and 11 (People's Computer Company); and many other places. The computer is thinking of a secret three-digit number. You should determine this secret number in a minimum of turns, indicated in lower right corner. Enter your guess - using any number 0-9. Each digit will be examined in the same way. For example, the digit in the first location is checked to see if it is the same as in the secret number. If it is, it receives a score of 2; if not, but does occur elsewhere in number, it receives a score of 1; and if not at all, a score of 0. The computer then gives you the total score below your guess as a clue. A score of 6 indicates that you have determined the secret number.

```
0200 6E00 VE=00
0202 A3F0 I=03F0
0204 22A0 DO 02A0
0206 22A0 DO 02A0
0208 22A0 DO 02A0
020A 6500 V5=00
020C 6000 V0=00
020E 6100 V1=00
0210 6200 V2=00
0212 F255 MI=V0:V2
0214 22AE DO 02AE
0216 6534 V5=34
0218 22D0 DO 02D0
021A A3F6 I=03F6
021C 22E2 DO 02E2
021E 22E2 DO 02E2
0220 22E2 DO 02E2
0222 6500 V5=00
0224 22AE DO 02AE
0226 A3F6 I=03F6
0228 F265 V0:V2=MI
022A A3F3 I=03F3
022C F255 MI=V0:V2
022E 6500 V5=00
0230 22AE DO 02AE
0232 6402 V4=02
0234 6D00 VD=00
0236 A3F3 I=03F3
0238 22F4 DO 02F4
023A A3F3 I=03F3
023C F255 MI=V0:V2
023E 8500 V5=V0
0240 A3F0 I=03F0
0242 22F4 DO 02F4
0244 A3F0 I=03F0
0246 F255 MI=V0:V2
0248 9500 SKIP;V5 NE V0
024A 1300 GO 0300
024C 9510 SKIP;V5 NE V1
024E 1252 GO 0252
0250 9520 SKIP;V5 NE V2
0252 7D01 VD+01
0254 4400 SKIP;V4 NE 00
0256 125C GO 025C
0258 74FF V4+FF
025A 1236 GO 0236
025C 6508 V5=08
025E 22D0 DO 02D0
0260 6534 V5=34
0262 22D0 DO 02D0
0264 7E01 VE+01
0266 6534 V5=34
0268 22D0 DO 02D0
026A 4D06 SKIP;VD NE 06
026C 1288 GO 0288
026E 4E63 SKIP;VE NE 63
0270 1282 GO 0282
0272 61C0 V1=C0
0274 F115 TIME=V1
0276 F107 V1=TIME
0278 3100 SKIP;V1 EQ 00
027A 1276 GO 0276
027C 6508 V5=08
027E 22D0 DO 02D0
0280 121A GO 021A
0282 A3F0 I=03F0
0284 652C V5=2C
0286 22AE DO 02AE
0288 6108 V1=08
028A 6002 V0=02
028C F018 TONE=V0
028E 6F10 VF=10
0290 71FF V1+FF
0292 FF15 TIME=VF
0294 FF07 VF=TIME
0296 3F00 SKIP;VF EQ 00
0298 1294 GO 0294
029A 3100 SKIP;V1 EQ 00
029C 128A GO 028A
029E 129E GO 029E
02A0 6409 V4=09
02A2 C00F V0=RND
02A4 8405 V4=V4-V0
02A6 4F00 SKIP;VF NE 00
02A8 12A0 GO 02A0
02AA F055 MI=V0:V0
02AC 00EE RET
02AE 6600 V6=00
02B0 3500 SKIP;V5 EQ 00
02B2 12C6 GO 02C6
02B4 A3F3 I=03F3
02B6 F265 V0:V2=MI
02B8 F029 I=V0(LSDP)
02BA 22CA DO 02CA
02BC F129 I=V1(LSDP)
02BE 22CA DO 02CA
02C0 F229 I=V2(LSDP)
02C2 22CA DO 02CA
02C4 00EE RET
02C6 A3F0 I=03F0
02C8 12B6 GO 02B6
02CA D565 SHOW 5MI@V5V6
02CC 7508 V5+08
02CE 00EE RET
02D0 6618 V6=18
02D2 3508 SKIP;V5 EQ 08
02D4 12DA GO 02DA
02D6 FD29 I=VD(LSDP)
02D8 12CA GO 02CA
02DA A3F6 I=03F6
02DC FE33 MI=VE(3DD)
02DE F265 V0:V2=MI
02E0 12BC GO 02BC
02E2 F00A V0=KEY
02E4 400F SKIP;V0 NE 0F
02E6 1282 GO 0282
02E8 6109 V1=09
02EA 8105 V1=V1-V0
02EC 4F00 SKIP;VF NE 00
02EE 12E2 GO 02E2
02F0 F055 MI=V0:V0
02F2 00EE RET
02F4 F265 V0:V2=MI
02F6 8300 V3=V0
02F8 8010 V0=V1
02FA 8120 V1=V2
02FC 8230 V2=V3
02FE 00EE RET
0300 7D02 VD+02
0302 1254 GO 0254
```

18. VIP Shooting Stars

This program uses the CHIP-8 INTERPRETER at 0000-01FF. Each location in universe is either a Black Hole or a Star. The goal is to obtain a central Black Hole surrounded by all Stars in a minimum number of turns. To shoot Star, press corresponding number (1-9) on keyboard. When Star is shot, it will turn into a Black Hole and all other states in its galaxy are complemented. If your universe becomes all Black Holes, you lose and are given a score of 99. For further discussion of game, see "What to Do After You Hit Return", p. 54, 55 (People's Computer Company) and BYTE Magazine, May 1976, p. 42-49, W. I. Nico.

```
0200 00E0 ERASE
0202 6E00 VE=00
0204 A2E9 I=02E9
0206 67FF V7=FF
0208 6801 V8=01
020A 228E DO 028E
020C 6900 V9=00
020E 6A00 VA=00
0210 A2E4 I=02E4
0212 228E DO 028E
0214 225A DO 025A
0216 4E63 SKIP;VE NE 63
0218 12D2 GO 02D2
021A 3900 SKIP;V9 EQ 00
021C 1222 GO 0222
021E 4A00 SKIP;VA NE 00
0220 12CC GO 02CC
0222 3A00 SKIP;VA EQ 00
0224 122A GO 022A
0226 49FF SKIP;V9 NE FF
0228 12D2 GO 02D2
022A A2EE I=02EE
022C 2270 DO 0270
022E 4100 SKIP;V1 NE 00
0230 122C GO 022C
0232 71FF V1+FF
0234 8114 V1=V1+V1
0236 8114 V1=V1+V1
0238 F11E I=I+V1
023A F365 V0:V3=MI
023C 6402 V4=02
023E F418 TONE=V4
0240 8092 V0=V0&V9
0242 3000 SKIP;V0 EQ 00
0244 124C GO 024C
0246 81A2 V1=V1&VA
0248 4100 SKIP;V1 NE 00
024A 122A GO 022A
024C 8720 V7=V2
024E 8830 V8=V3
0250 A2E4 I=02E4
0252 228E DO 028E
0254 225A DO 025A
0256 7E01 VE+01
0258 1214 GO 0214
025A 6520 V5=20
025C 6608 V6=08
025E A3FD I=03FD
0260 FE33 MI=VE(3DD)
0262 F265 V0:V2=MI
0264 F129 I=V1(LSDP)
0266 D565 SHOW 5MI@V5V6
0268 7505 V5+05
026A F229 I=V2(LSDP)
026C D565 SHOW 5MI@V5V6
026E 00EE RET
0270 6109 V1=09
0272 E1A1 SKIP;V1 NE KEY
0274 00EE RET
0276 4100 SKIP;V1 NE 00
0278 00EE RET
027A 71FF V1+FF
027C 1272 GO 0272
027E 60FF V0=FF
0280 61FF V1=FF
0282 8035 V0=V0-V3
0284 8125 V1=V1-V2
0286 8202 V2=V2&V0
0288 8132 V1=V1&V3
028A 8211 V2=V2/V1
028C 00EE RET
028E 8370 V3=V7
0290 8290 V2=V9
0292 227E DO 027E
0294 8920 V9=V2
0296 8380 V3=V8
0298 82A0 V2=VA
029A 227E DO 027E
029C 8A20 VA=V2
029E 6409 V4=09
02A0 6610 V6=10
02A2 6510 V5=10
02A4 4400 SKIP;V4 NE 00
02A6 00EE RET
02A8 4405 SKIP;V4 NE 05
02AA 12C6 GO 02C6
02AC 4406 SKIP;V4 NE 06
02AE 12B4 GO 02B4
02B0 3403 SKIP;V4 EQ 03
02B2 12B8 GO 02B8
02B4 76F8 V6+F8
02B6 6510 V5=10
02B8 8774 V7=V7+V7
02BA 4F00 SKIP;VF NE 00
02BC 12C0 GO 02C0
02BE D565 SHOW 5MI@V5V6
02C0 75F8 V5+F8
02C2 74FF V4+FF
02C4 12A4 GO 02A4
02C6 4800 SKIP;V8 NE 00
02C8 12C0 GO 02C0
02CA 12BE GO 02BE
02CC 225A DO 025A
02CE 6E63 VE=63
02D0 225A DO 025A
02D2 6002 V0=02
02D4 F018 TONE=V0
02D6 6F10 VF=10
02D8 FF15 TIME=VF
02DA FF07 VF=TIME
02DC 3F00 SKIP;VF EQ 00
02DE 12DA GO 02DA
02E0 12D2 GO 02D2
02E2 01FF
02E4 1C3E
02E6 3E3E
02E8 1C14
02EA 2200
02EC 2214
02EE 0100
02F0 0B01
02F2 0200
02F4 0700
02F6 0400
02F8 1601
02FA 0800
02FC 2900
02FE 0001
0300 5A01
0302 1000
0304 9400
0306 2000
0308 6801
030A 4000
030C E000
030E 8000
0310 D001
0312 00D4
```

19. VIP Strike-9

This program uses the CHIP-8 INTERPRETER at 0000-01FF. STRIKE-9 is based on the roll of dice. To roll dice, press key "0". Select from the numbers 1-9 those adding up to total on dice, then roll again. To win you must just eliminate all the starting nine numbers. You are given up to 4 seconds to hit any valid key. Refer to Creative Computing, Vol. 3, 88 (1977), Bruce Grembowski.

```
0200 00E0 ERASE
0202 6401 V4=01
0204 60FF V0=FF
0206 22EA DO 02EA
0208 F055 MI=V0:V0
020A 7401 V4+01
020C 340A SKIP;V4 EQ 0A
020E 1206 GO 0206
0210 6401 V4=01
0212 22EA DO 02EA
0214 F265 V0:V2=MI
0216 F429 I=V4(LSDP)
0218 D125 SHOW 5MI@V1V2
021A 7401 V4+01
021C 340A SKIP;V4 EQ 0A
021E 1212 GO 0212
0220 6108 V1=08
0222 22B4 DO 02B4
0224 6112 V1=12
0226 22B4 DO 02B4
0228 6000 V0=00
022A 22FA DO 02FA
022C 6109 V1=09
022E 22BC DO 02BC
0230 8700 V7=V0
0232 6113 V1=13
0234 22BC DO 02BC
0236 8800 V8=V0
0238 8980 V9=V8
023A 8974 V9=V9+V7
023C 60FF V0=FF
023E F015 TIME=V0
0240 F007 V0=TIME
0242 4000 SKIP;V0 NE 00
0244 1284 GO 0284
0246 6401 V4=01
0248 E4A1 SKIP;V4 NE KEY
024A 1254 GO 0254
024C 7401 V4+01
024E 340A SKIP;V4 EQ 0A
```

```
0250 1248 GO 0248
0252 1240 GO 0240
0254 22EA DO 02EA
0256 F265 V0:V2=MI
0258 4000 SKIP;V0 NE 00
025A 1240 GO 0240
025C F618 TONE=V6
025E F429 I=V4(LSDP)
0260 D125 SHOW 5MI@V1V2
0262 22EA DO 02EA
0264 6000 V0=00
0266 F055 MI=V0:V0
0268 8945 V9=V9-V4
026A 4F00 SKIP;VF NE 00
026C 1284 GO 0284
026E 3900 SKIP;V9 EQ 00
0270 12A4 GO 02A4
0272 22AC DO 02AC
0274 620A V2=0A
0276 6109 V1=09
0278 F729 I=V7(LSDP)
027A D125 SHOW 5MI@V1V2
027C 6113 V1=13
027E F829 I=V8(LSDP)
0280 D125 SHOW 5MI@V1V2
0282 1228 GO 0228
0284 A310 I=0310
0286 128A GO 028A
0288 A301 I=0301
028A 6005 V0=05
028C 6218 V2=18
028E 6108 V1=08
0290 D125 SHOW 5MI@V1V2
0292 6110 V1=10
0294 F01E I=I+V0
0296 D125 SHOW 5MI@V1V2
0298 6118 V1=18
029A F01E I=I+V0
029C D125 SHOW 5MI@V1V2
029E 6077 V0=77
02A0 F018 TONE=V0
```

(Continued on next page)

19. VIP Strike-9 (Continued)

```
02A2 12A2 GO 02A2
02A4 22D8 DO 02D8
02A6 3000 SKIP;V0 EQ 00
02A8 123C GO 023C
02AA 1284 GO 0284
02AC 22D8 DO 02D8
02AE 3000 SKIP;V0 EQ 00
02B0 00EE RET
02B2 1288 GO 0288
02B4 A33A I=033A
02B6 6208 V2=08
02B8 D129 SHOW 9MI@V1V2
02BA 00EE RET
02BC 6601 V6=01
02BE 620A V2=0A
02C0 6001 V0=01
02C2 F029 I=V0(LSDP)
02C4 D125 SHOW 5MI@V1V2
02C6 F618 TONE=V6
02C8 C307 V3=RND
02CA 4300 SKIP;V3 NE 00
02CC 00EE RET
02CE 7001 V0+01
02D0 D125 SHOW 5MI@V1V2
02D2 3007 SKIP;V0 EQ 07
02D4 12C2 GO 02C2
02D6 12C0 GO 02C0
02D8 6401 V4=01
02DA 22EA DO 02EA
02DC F065 V0:V0=MI
02DE 3000 SKIP;V0 EQ 00
02E0 00EE RET
02E2 7401 V4+01
02E4 340A SKIP;V4 EQ 0A
02E6 12DA GO 02DA
02E8 00EE RET
02EA 6203 V2=03
02EC 8340 V3=V4
02EE A31F I=031F
02F0 4301 SKIP;V3 NE 01
02F2 00EE RET
02F4 F21E I=I+V2
02F6 73FF V3+FF
02F8 12F0 GO 02F0
02FA E09E SKIP;V0 EQ KEY
02FC 12FA GO 02FA
02FE 00EE RET
0300 0189
0302 89A9
0304 A9F9
0306 2232
0308 2A26
030A 2222
030C 2222
030E 0022
0310 8382
0312 8282
0314 F3CF
0316 484F
0318 41CF
031A 3C20
031C 3820
031E 3C00
0320 2808
0322 0030
0324 0800
0326 3808
0328 0028
032A 1000
032C 3010
032E 0038
0330 1000
0332 2818
0334 0030
0336 1800
0338 3818
033A FCFC
033C FCFC
033E FCFC
0340 FCFC
0342 FC00
```

20. VIP Card Game (like the well-known acey-ducey)

This program uses the CHIP-8 INTERPRETER at 0000-01FF. ACEY-DUCEY is a card game in which the dealer shows two cards from deck. You bet (from 1 to 9) that the next dealer card lies between or equal to the first two cards in face value (ACES are low).

In order to obtain a new deal, press the zero key, and then bet as before. Try for a score of 100 or greater.

```
0200 A350 I=0350
0202 600A V0=0A
0204 F055 MI=V0:V0
0206 00E0 ERASE
0208 2284 DO 0284
020A 6113 V1=13
020C 22A2 DO 02A2
020E 8540 V5=V4
0210 6127 V1=27
0212 22A2 DO 02A2
0214 8740 V7=V4
0216 6801 V8=01
0218 E8A1 SKIP;V8 NE KEY
021A 1224 GO 0224
021C 7801 V8+01
021E 380A SKIP;V8 EQ 0A
0220 1218 GO 0218
0222 1216 GO 0216
0224 A350 I=0350
0226 F065 V0:V0=MI
0228 8085 V0=V0-V8
022A 3F01 SKIP;VF EQ 01
022C 1216 GO 0216
022E 6002 V0=02
0230 F018 TONE=V0
0232 22C4 DO 02C4
0234 611D V1=1D
0236 22A2 DO 02A2
0238 8640 V6=V4
023A 8565 V5=V5-V6
023C 4500 SKIP;V5 NE 00
023E 124E GO 024E
0240 3F00 SKIP;VF EQ 00
0242 126C GO 026C
0244 8675 V6=V6-V7
0246 4600 SKIP;V6 NE 00
0248 124E GO 024E
024A 3F00 SKIP;VF EQ 00
024C 1276 GO 0276
024E 6000 V0=00
0250 8085 V0=V0-V8
0252 8085 V0=V0-V8
0254 8800 V8=V0
0256 22C4 DO 02C4
0258 6164 V1=64
025A A350 I=0350

025C F065 V0:V0=MI
025E 8015 V0=V0-V1
0260 3F00 SKIP;VF EQ 00
0262 127E GO 027E
0264 6000 V0=00
0266 E09E SKIP;V0 EQ KEY
0268 1266 GO 0266
026A 1206 GO 0206
026C 8675 V6=V6-V7
026E 4600 SKIP;V6 NE 00
0270 124E GO 024E
0272 3F00 SKIP;VF EQ 00
0274 124E GO 024E
0276 A350 I=0350
0278 F065 V0:V0=MI
027A 3000 SKIP;V0 EQ 00
027C 1264 GO 0264
027E 6040 V0=40
0280 F018 TONE=V0
0282 1282 GO 0282
0284 A350 I=0350
0286 F065 V0:V0=MI
0288 F033 MI=V0(3DD)
028A 641B V4=1B
028C 6318 V3=18
028E F265 V0:V2=MI
0290 F029 I=V0(LSDP)
0292 D345 SHOW 5MI@V3V4
0294 7306 V3+06
0296 F129 I=V1(LSDP)
0298 D345 SHOW 5MI@V3V4
029A 7306 V3+06
029C F229 I=V2(LSDP)
029E D345 SHOW 5MI@V3V4
02A0 00EE RET
02A2 C40F V4=RND
02A4 4400 SKIP;V4 NE 00
02A6 12A2 GO 02A2
02A8 600D V0=0D
02AA 8045 V0=V0-V4
02AC 3F01 SKIP;VF EQ 01
02AE 12A2 GO 02A2
02B0 6002 V0=02
02B2 F018 TONE=V0
02B4 6200 V2=00
02B6 A311 I=0311
02B8 D127 SHOW 7MI@V1V2

02BA 7201 V2+01
02BC 7101 V1+01
02BE 22D4 DO 02D4
02C0 D125 SHOW 5MI@V1V2
02C2 00EE RET
02C4 2284 DO 0284
02C6 A350 I=0350
02C8 F065 V0:V0=MI
02CA 8085 V0=V0-V8
02CC A350 I=0350
02CE F055 MI=V0:V0
02D0 2284 DO 0284
02D2 00EE RET
02D4 6001 V0=01
02D6 8045 V0=V0-V4
02D8 4000 SKIP;V0 NE 00
02DA 12F2 GO 02F2
02DC 6009 V0=09
02DE 8045 V0=V0-V4
02E0 3F00 SKIP;VF EQ 00
02E2 12EE GO 02EE
02E4 A2ED I=02ED
02E6 F41E I=I+V4
02E8 F065 V0:V0=MI
02EA F01E I=I+V0
02EC 00EE RET
02EE F429 I=V4(LSDP)
02F0 00EE RET
02F2 A303 I=0303
02F4 00EE RET
02F6 0114
02F8 0205
02FA 0C10
02FC 1010
02FE 90F0
0300 9090
0302 B0F0
0304 90F0
0306 9090
0308 B0E0
030A B090
030C F060
030E 6060
0310 60FC
0312 FCFC
0314 FCFC
0316 FCFC
```

Appendix E - Logic Diagrams

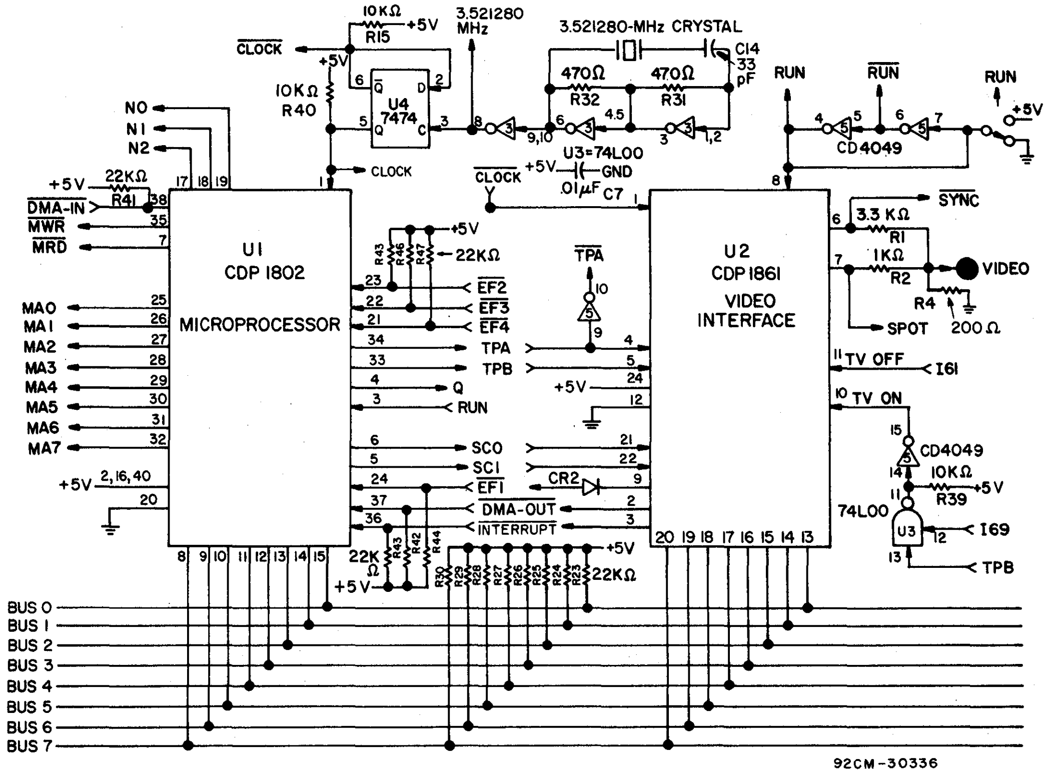


Fig. E-1 - Microprocessor and Display Interface Circuits

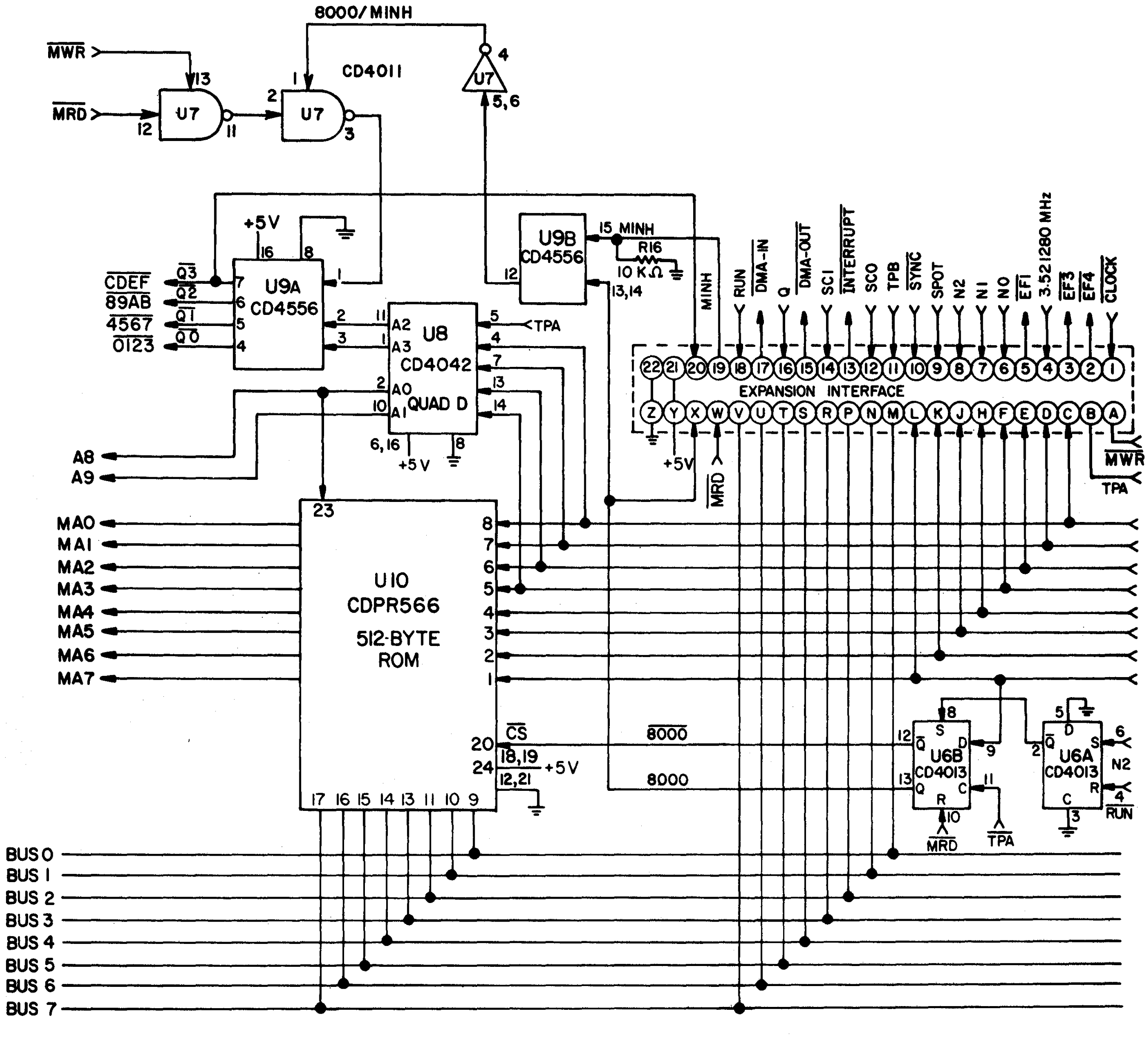


Fig. E-2 - ROM Circuits and Expansion Interface

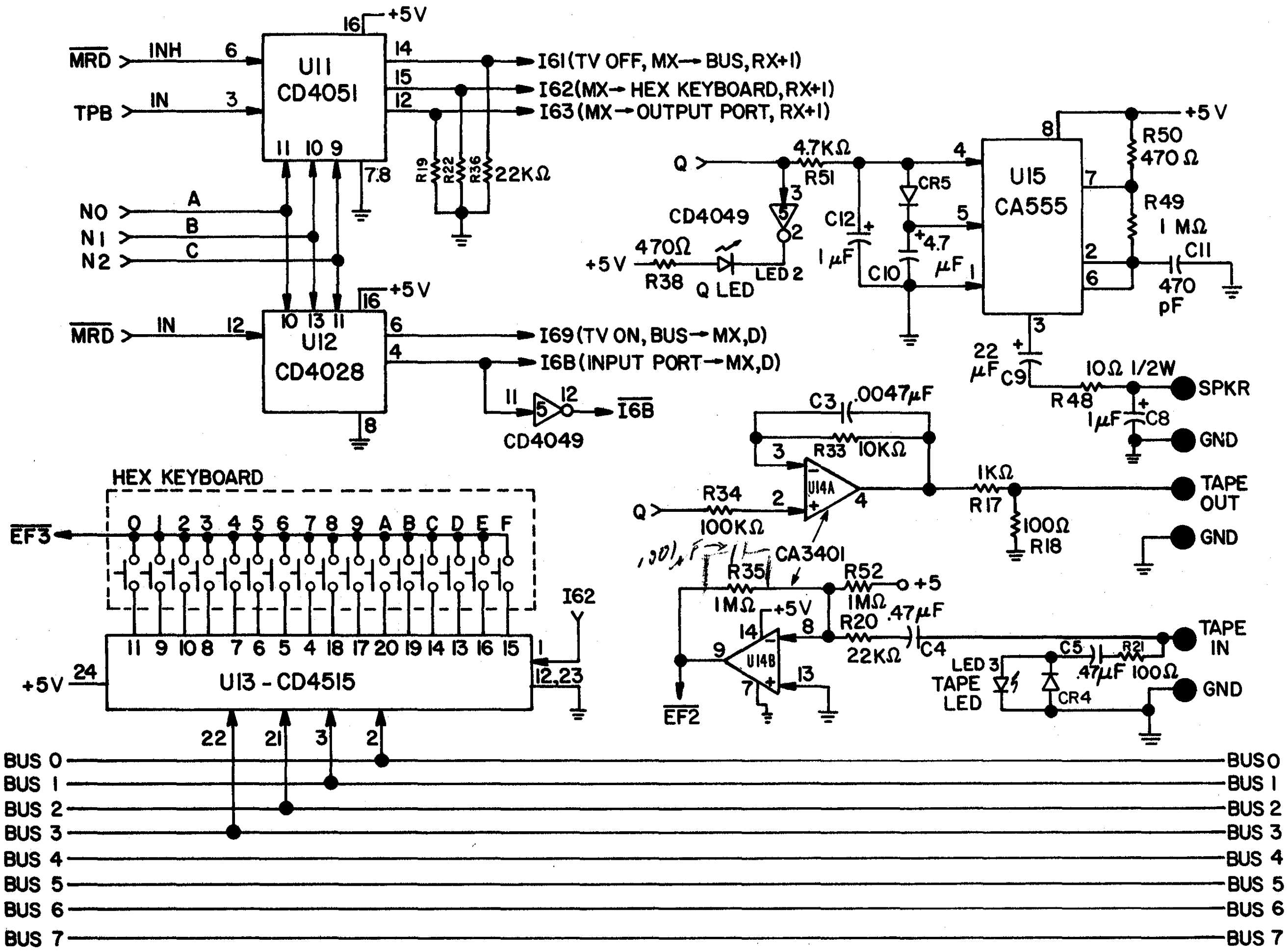


Fig. E-3 - Keyboard, Decoding, Audio Oscillator, and Cassette Interface Circuits

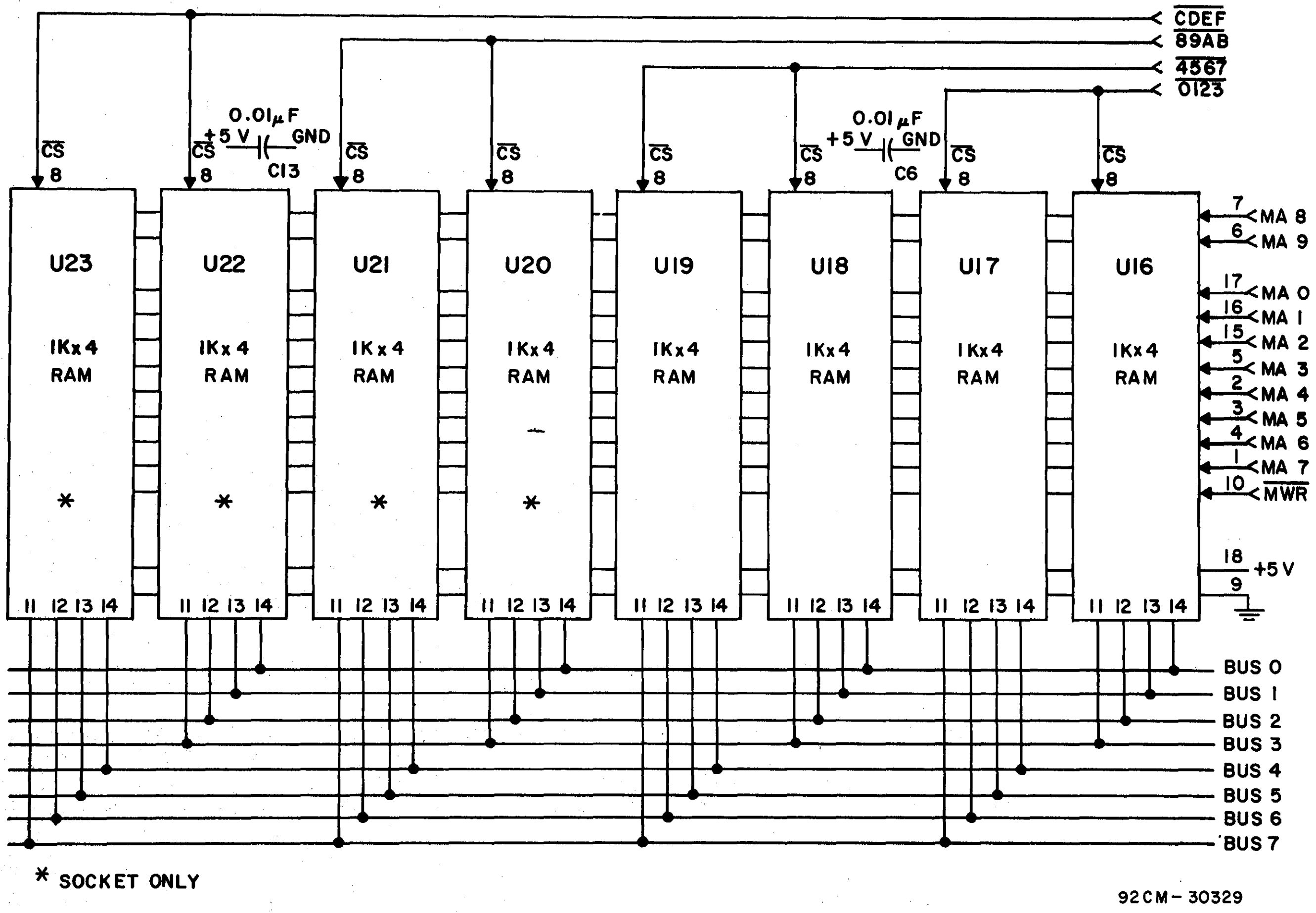
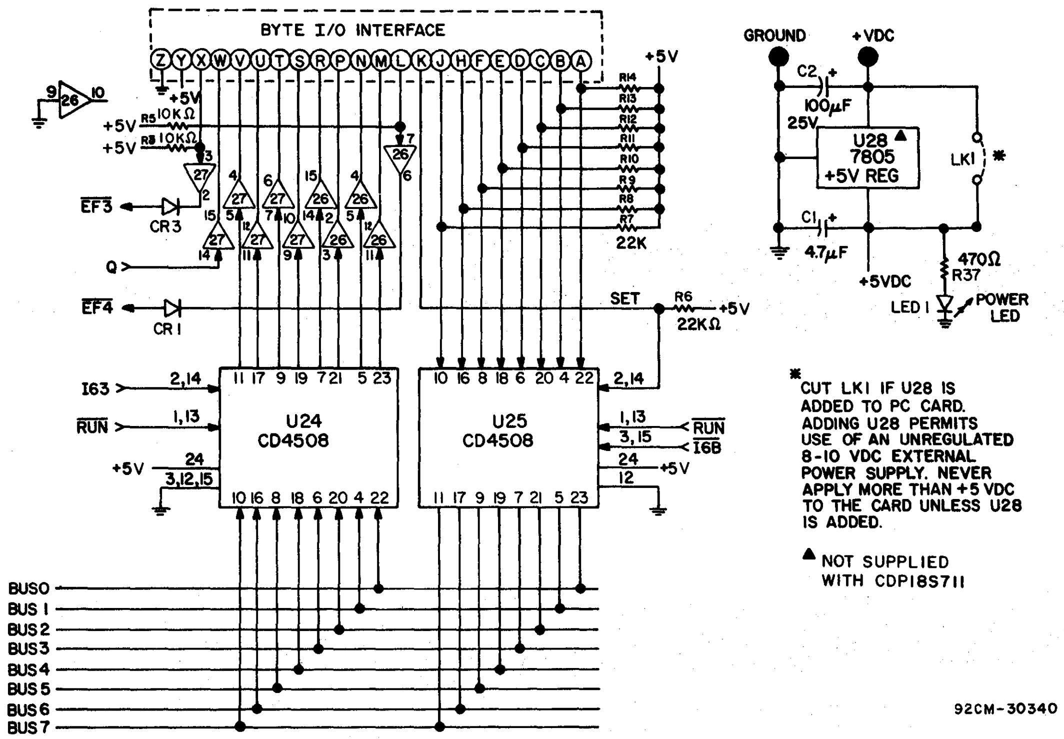


Fig. E-4 - RAM Circuits

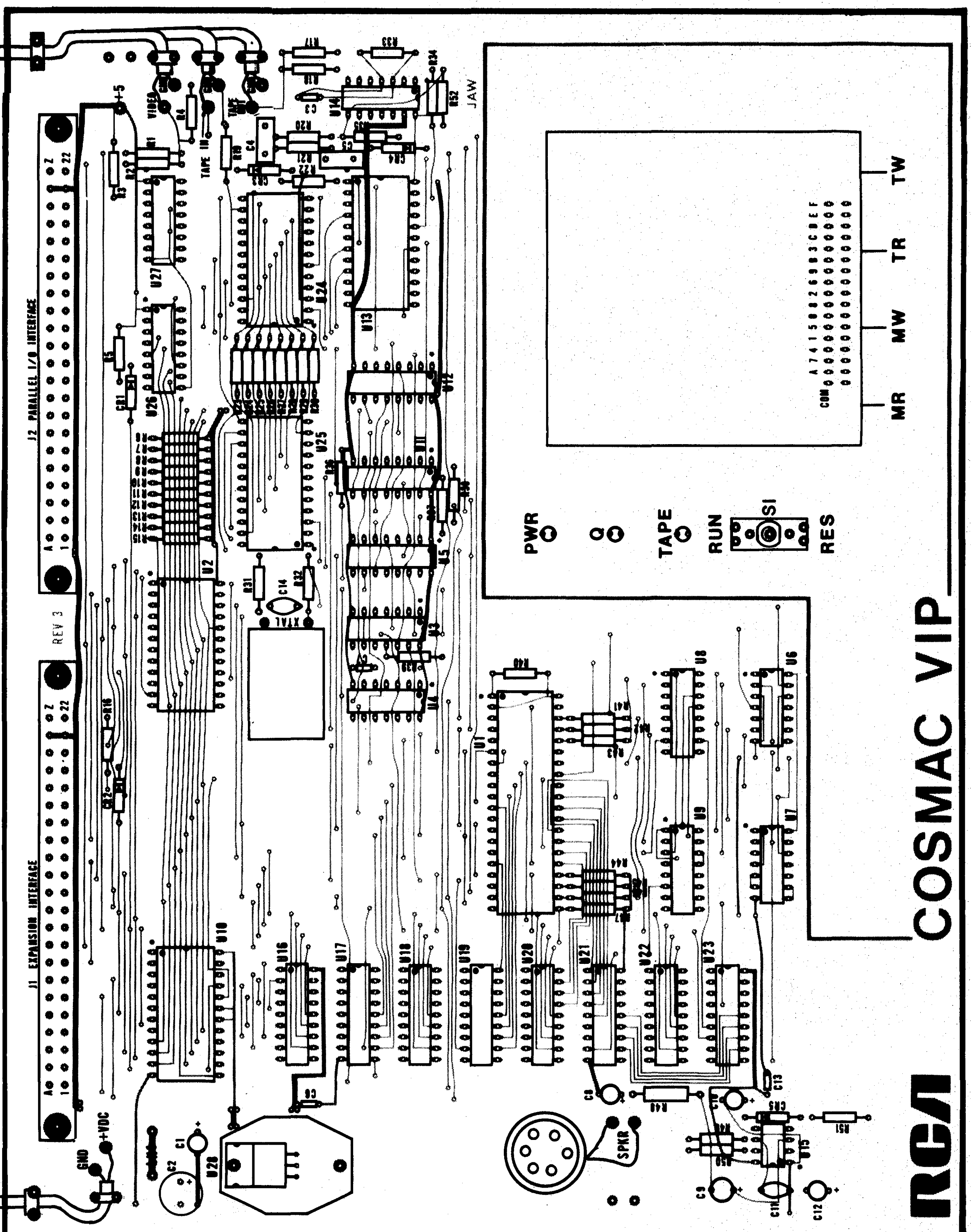


✱CUT LK1 IF U28 IS ADDED TO PC CARD. ADDING U28 PERMITS USE OF AN UNREGULATED 8-10 VDC EXTERNAL POWER SUPPLY. NEVER APPLY MORE THAN +5 VDC TO THE CARD UNLESS U28 IS ADDED.

▲ NOT SUPPLIED WITH CDP18S711

Fig. E-5 - Power Supply Circuit and Byte Input/Output Interface

Appendix F - Board Layout, Parts List, and Expansion Notes



92CS-30330

1. Printed Circuit Board Layout

2. Parts List for RCA COSMAC VIP CDP18S711

Type	Number	Qty.	Description
Integrated Circuits – Supplied			
CDP1802	U1	1	COSMAC Microprocessor
CDP1861	U2	1	Video Interface
SN74L00N	U3	1	Quad NAND Low Power
SN7474N	U4	1	Dual D-Type Flip-Flop
CD4049	U5	1	Hex Inverting Buffer
CD4013	U6	1	Dual D-Type Flip-Flop
CD4011	U7	1	Quad 2-Input NAND Gate
CD4042	U8	1	Quad Clocked "D" Latch
CD4556	U9	1	Dual Binary 1 of 4 Decoder
CDPR566	U10	1	512 x 8-Bit Static ROM (Programmed CDP1832)
CD4051	U11	1	Binary 1 of 8 Decoder
CD4028	U12	1	BCD-to-Decimal Decoder
CD4515	U13	1	4-Bit Latch/1 of 16 Decoder
CA3401	U14	1	Quad Single-Supply Op-Amp
CA555CE	U15	1	Timer
2114 or TMS4045	U16-U19	4	1K x 4-Bit Static RAM
CD4508	U24, U25	2	Dual 4-Bit Latch
CD4050	U26, U27	2	Hex Buffer
Integrated Circuits – Optional			
2114 or TMS4045	U20-U23	4	1K x 4-Bit Static RAM
Capacitors – Supplied			
	C1, C10	2	4.7 μF 35 V Electrolytic
	C2	1	100 μF 16 V Electrolytic
	C3	1	0.0047 μF 50 V Poly Film (472)
	C4, C5	2	0.47 μF 50 V
	C6, C7, C13	3	0.01 μF 50 V Poly Film (103)
	C8, C12	2	1 μF 50 V Electrolytic
	C9	1	22 μF 16 V Electrolytic
	C11	1	470 pF 500 V Disc
	C14	1	33 pF ± 10% 1 kV

(Continued on next page)

2. Parts List for RCA COSMAC VIP CDP18S711 (Continued)

Type	Number	Qty.	Description
Resistors – Supplied (1/4 W except as noted)			
	R1	1	3.3 KΩ
	R3, R5, R15, R16, R33, R39, R40	7	10 KΩ
	R2, R17	2	1 KΩ
	R6-R14, R19, R20, R22-R30, R36, R41-R47	28	22 KΩ
	R18, R21	2	100 Ω
	R31, R32, R37, R38, R50	5	470 Ω
	R34	1	100 KΩ
	R35, R49, R52	3	1 megohm
	R48	1	10 Ω 1/2 W
	R51	1	4.7 KΩ
	R4	1	200 Ω
Miscellaneous – Supplied			
1N914	CR1 through CR5	5	Diode
HP5082-4494		3	Red LED
		1	3.521280 MHz Crystal
7101-S-D-V30-B	S1	1	SPDT Toggle Switch; C&K
E7807		1	Panel Dress Nut for Switch; C&K
18S022		1	Printed Circuit Board
M-1651-0		1	Keyboard, Centralab
		1	Cover, Thermoplastic
		7	Rubber Feet
C931802		4	18-Pin IC Socket
C932402		1	24-Pin IC Socket
C934002		1	40-Pin IC Socket
CDP18S023		1	Power Supply, Regulated; 5 V dc, 600 mA; 110 V 50/60 Hz
		6	Cable Straps
		1	Speaker
	J1, J2	2	44-Pin Connector
MPM-201		1	User Manual for the CDP1802 COSMAC Microprocessor
MPM-920		1	Instruction Summary for the CDP1802 COSMAC Microprocessor
VIP-311		1	RCA COSMAC VIP Instruction Manual
			Connectors, cables, hardware
Miscellaneous – Optional			
LAD66A2CD		1	Heat Sink; IERC
		2	4–40 ¼" Binder Hd. Machine Screws and Nuts
7805	U28	1	Voltage Regulator

3. COSMAC VIP Expansion Notes

a. Soldering the PC Board

In the event you wish to make some changes or add components requiring soldering, you should have some experience building electronic kits. The PC board pads are small and close together requiring extra caution when soldering to avoid shorts or solder bridges. Use a low-heat, small-tipped, grounded soldering iron. Keep it clean. Use small gauge, rosin-core 60/40 solder. Preheat the connection and apply just enough solder to "wet" the connection. Avoid using excessive amounts of solder because it will flow through the plated-through holes and form "blobs" on the top of the card. Excessive or protracted heat from the soldering iron can damage some of the components.

b. Voltage Regulator Option

An unregulated 8-10 volt DC power supply can be used with the COSMAC VIP card if desired. Cut LK1 on the PC card. Add U28 (a 7805, 5-volt regulator IC) to the card together with an appropriate heat sink. Make sure the U28 lead pads on the PC card don't touch the heat sink. Disconnect the +5 V DC supply at the + V DC and GND pads and connect your unregulated 8-10 V DC power supply to these pads. This on-card regulator will handle up to 1 ampere of current and is useful for system expansion. Do not use a plastic cover for your PC card when this on-card regulator option is used. Air flow is needed to permit the regulator to operate properly.

c. Additional 2048-Byte RAM Option

To increase your COSMAC VIP RAM to a total of 4096 bytes, add U20-U23 to the PC card by plugging units into the four sockets provided. Measure the power supply current to be sure it does not exceed the capacity of the +5 V DC power pack supplied with the VIP (600 mA). If you require additional power supply current use a regulated +5 V DC supply capable of supplying 1 ampere or use an unregulated 8-10 V DC supply with the voltage regulator option on the cards.

Appendix G - Data Sheets

CDP1832 512-Word x 8-Bit Static Read-Only Memory*

CDP1861 Video Display Controller (Video Interface)

CDP1802 COSMAC Microprocessor

*The CDPR566 supplied with the VIP is a mask-programmed CDP1832.

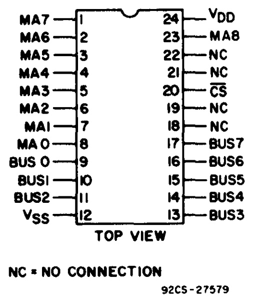
RCA

Solid State Division

Microprocessor Products

Preliminary Data▲

CDP1832D CDP1832CD



Terminal Assignment

512-Word x 8-Bit Static Read-Only Memory

Features:

- Static Silicon-Gate CMOS circuitry—CD4000-series compatible
- Compatible with CDP1800-series microprocessors at maximum speed
- Fast access time: 400 ns typ. at $V_{DD} = 10$ V
- Single voltage supply
- Full military temperature range (−55°C to +125°C)
- Functional replacement for industry type 8704 512 x 8 PROM
- Three-state outputs
- Low quiescent and operating power

The RCA-CDP1832D and CDP1832CD are static 4096-bit mask-programmable COS/MOS read-only memories organized as 512 words x 8 bits and designed for use in CDP1800-series microprocessor systems. The CDP1832 ROM's are completely static—no clocks are required.

A Chip-Select input ($\overline{CS}$) is provided for memory expansion. Outputs are enabled when $\overline{CS}$=0.

The CDP1832 is a pin-for-pin compatible replacement for the industry types 2704/8704 Reprogrammable Read-Only Memories.

The CDP1832D is functionally identical to the CDP1832CD. The CDP1832D has a recommended operating voltage range of 3 to 12 volts, and the CDP1832CD has a recommended operating voltage range of 4 to 6 volts.

The CDP1832D and CDP1832CD are supplied in 24-lead, hermetic, dual-in-line ceramic packages.

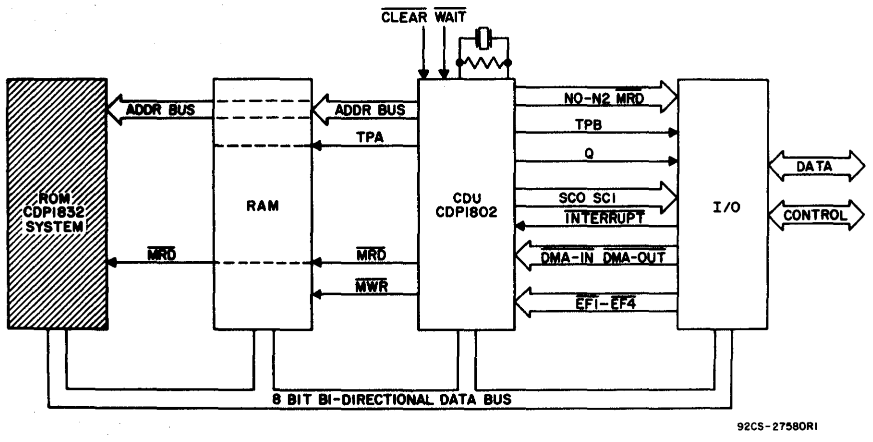


Fig. 1—Typical CDP1802 microprocessor system.

▲The Preliminary Data are intended for guidance purposes in evaluating the device for equipment design. The device is now being designed for inclusion in our standard line of commercially available products. For current information on the status of this program, please contact your RCA Sales Office.

Printed in USA/6-77
Supersedes data issued 2-76

MAXIMUM RATINGS, *Absolute-Maximum Values*

Storage-Temperature Range (T_{stg})
. . . . –65 to +150°C
Operating-Temperature Range (T_A)
. . . . –55 to +125°C
DC Supply-Voltage Range (V_{DD})
(All voltage values referenced to V_{SS} terminal)
CDP1832D . . . . –0.5 to +15 V
CDP1832CD . . . . –0.5 to +7 V
Power Dissipation Per Package (P_D):
For T_A=–55 to +100°C
. . . . 500 mW
For T_A=+100 to +125°C
. . . . Derate Linearly to 200 mW
Device Dissipation Per Output Transistor:
For T_A=–55°C to +125°C . . . . 100 mW
Input Voltage Range, All Inputs
. . . . –0.5 to V_{DD} +0.5 V
Lead Temperature (During Soldering):
At distance 1/16 ± 1/32 inch (1.59 ± 0.79 mm) from case for 10 s max. . . . . +265°C

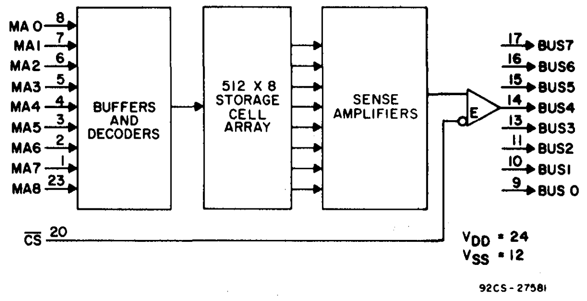
OPERATING CONDITIONS at T_A=25°C Unless Otherwise Specified

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

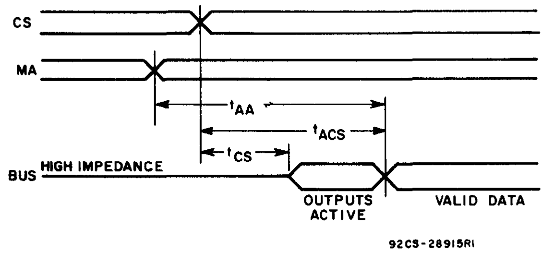
CHARACTERISTIC	CONDITIONS	LIMITS				UNITS
	V_{DD}	CDP1832D		CDP1832CD		
	(V)	Min.	Max.	Min.	Max.	
Static						
Supply-Voltage Range (At T_A=Full Package-Temperature Range)	–	3	12	4	6	V
Recommended Input Voltage Range	–	V_{SS}	V_{DD}	V_{SS}	V_{DD}	V

ELECTRICAL CHARACTERISTICS at T_A=25°C

CHARACTERISTIC	TEST CONDITIONS		LIMITS		UNITS
			CDP1832D	CDP1832CD	
	V_O (V)	V_{DD} (V)	TYPICAL VALUES	TYPICAL VALUES	
Static					
Quiescent Device Current, I_L	–	5	100	500	μA
	–	10	500	–	
	–	15	1000	–	
Output Drive Current: N-Channel (Sink), I_DN	0.4	5	0.8	0.8	mA
	0.5	10	1.8	–	
P-Channel (Source), I_DP	4.6	5	–0.8	–0.8	
	9.5	10	–1.8	–	
Dynamic: t_r,t_f=10 ns, C_L=50 pF					
Access Time From Address Change, t_{AA}	–	5	850	850	ns
	–	10	400	–	
Access Time From Chip Select, t_{ACS}	–	5	400	400	ns
	–	10	200	–	
Chip Select Delay, t_{CS}	–	5	400	400	ns
	–	10	200	–	



CDP1832
Functional Diagram



CDP1832
Timing Diagram

OPERATING & HANDLING CONSIDERATIONS

1. **Handling**

   All inputs and outputs of this device have a network for electrostatic protection during handling. Recommended handling practices for COS/MOS devices are described in ICAN-6000 "Handling and Operating Considerations for MOS 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

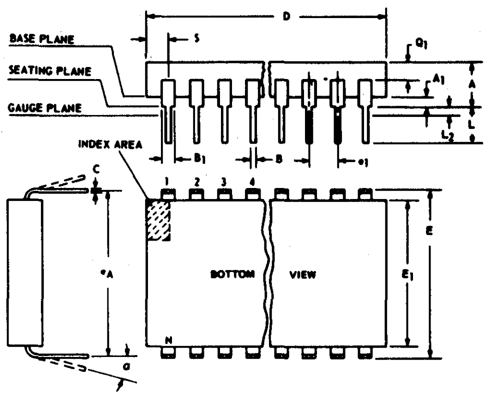
A connection must be provided at every input terminal. All unused input terminals must be connected to either V_{DD} or V_{SS}, whichever is appropriate.

Output Short Circuits

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

DIMENSIONAL OUTLINE

D Suffix
24-Lead Dual-In-Line Ceramic Package
JEDEC MO-015-AG



92CS-19948

SYMBOL	INCHES		NOTE	MILLIMETERS	
	MIN.	MAX.		MIN.	MAX.
A	0.090	0.150		2.29	3.81
A_1	0.020	0.065	2	0.51	1.65
B	0.015	0.020		0.381	0.508
B_1	0.045	0.055		1.143	1.397
C	0.008	0.012	1	0.204	0.304
D	1.15	1.22		29.21	30.98
E	0.600	0.625		15.24	15.87
E_1	0.480	0.520		12.20	13.20
e_1	0.100 TP		3	2.54 TP	
e_A	0.600 TP		3	15.24 TP	
L	0.100	0.180		2.54	4.57
L_2	0.000	0.030	3	0.00	0.76
a	0°	15°	4	0°	15°
N	24		5	24	
N_1	0		6	0	
Q_1	0.020	0.080		0.51	2.03
S	0.020	0.060		0.51	1.52

NOTES:

Refer to JEDEC Publication No. 13 for Rules for Dimensioning Axial Lead Product Outlines.

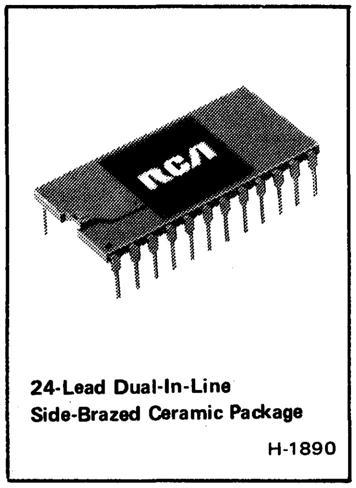
1. When this device is supplied solder dipped, the maximum lead thickness (narrow portion) will not exceed 0.013" (0.33 mm).
2. When base of body is to be attached to heat sink, terminal lead standoffs are not required and A_1 = 0. When A_1 = 0, the leads emerge from the body with the B_1 dimension and reduce to the B dimension above the seating plane.
3. e_1 and e_A apply in zone L_2 when unit is installed. Leads within 0.005 " (0.127 mm) radius of True Position (TP) at gauge plane with maximum material condition.
4. Applies to spread leads prior to installation.
5. N is the maximum quantity of lead positions.
6. N_1 is the quantity of allowable missing leads.

When Incorporating RCA Solid State Devices in equipment, it is recommended that the designer refer to "Operating Considerations for RCA Solid State Devices", Form No. 1CE-402, available on request from RCA Solid State Division, Box 3200, Somerville, N.J. 08876.

RCA **Solid State Division**

Microprocessor Products

Preliminary Data▲ **CDP1861CD**



24-Lead Dual-In-Line
Side-Brazed Ceramic Package
H-1890

Video Display Controller

Features:

- **Static silicon-gate CMOS circuitry**
- **Interfaces directly with CDP1802 microprocessor**
- **Supports bit-mapped video display for graphic flexibility**
- **Generates composite horizontal and vertical sync**
- **Programmable vertical resolution for matrix display of up to 64 x 128 segments**
- **Real-time interrupt generator**
- **Clear input**
- **External display control**
- **Single voltage supply (4 - 6 volts)**
- **Low quiescent and operating power**
- **Full military operating temperature range (–55 to +125°C)**

The RCA-CDP1861C is a video display controller designed for use in CDP1800-series microprocessor systems. It is compatible with the CDP1802 microprocessor and will interface directly with the CDP1802 as shown in the system diagram (Fig. 1).

The CDP1861C utilizes many of the features of the CDP1802 to simplify control and minimize the need for external components. The DMA feature of the CDP1802 may be used for direct data transfers from memory to the CDP1861C. The INTERRUPT input and the I/O command lines may be used to perform the necessary handshaking between the CDP-1802 and the CDP1861C. Timing may be simplified by operating the microprocessor at a clock frequency of 1.76064-MHz (the standard color frequency of 3.58 MHz, divided by 2, may also be used in some applications). The clock and the CDP1802 timing signals (TPA and TPB) may then be used to set the interface timing as shown in the system diagram. In general, the clock frequency equals the number of fields per second (60), times the number of lines per field (262), times the number of machine cycles per line (14), times the number of bits per byte (8). In DMA operation, each machine cycle is a memory access.

Flexibility in vertical resolution may be obtained by synchronizing the CDP1861C with the CDP1802, and employing direct program control over the DMA process in real time. The actual video display takes place during a "window" of 4.6 milliseconds out of each 16.7-millisecond TV field. Throughout each such display window, a CDP1802 interrupt program may be used to manipulate the DMA pointer, re-issuing a given line of the display several times to save memory storage at the expense of reduced vertical resolution.

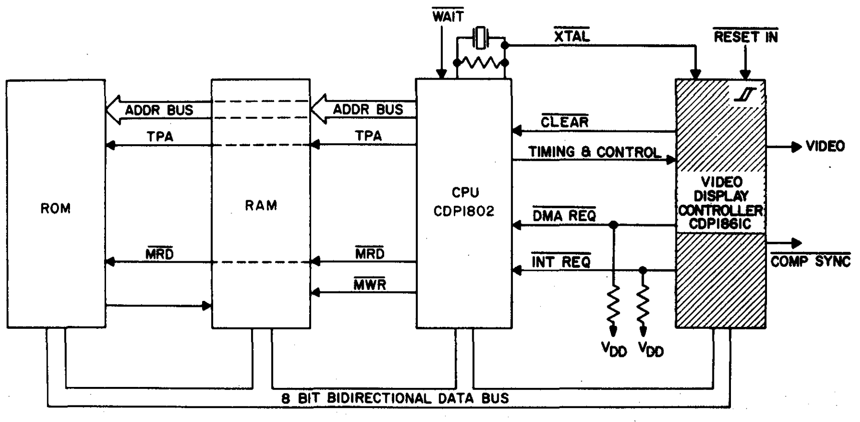


Fig. 1 – Typical CDP1802 microprocessor system.

▲The Preliminary Data are intended for guidance purposes in evaluating the device for equipment design. The device is now being designed for inclusion in our standard line of commercially available products. For current information on the status of this program, please contact your RCA Sales Office.

Printed in USA/6-77

The CDP1861C generates composite vertical and horizontal sync plus luminance signals which can be combined externally to create an NTSC compatible composite video signal. This composite vertical and horizontal sync output signal ($\overline{\text{COMP SYNC}}$) is generated from the sync reference (SYNC REF) and LOAD inputs. Vertical sync is derived from horizontal sync by dividing the horizontal sync frequency by 262. The composite sync signal generates timing for a non-interlace video display of 262 lines per field.

The CDP1861C generates an interrupt request ($\overline{\text{INT REQ}}$) once per field, 60 lines after the trailing edge of vertical sync and two lines before the raster has reached a "display window" (see Fig. 5). This request alerts the CDP1802 (or other control system) to prepare for DMA (direct memory access) activity. The CDP1861C DISP STATUS output goes low during the 4 lines before the display window, and again during the last 4 lines of the window. This signal may be used to give early warning of the display window and to release the control system from monitoring the DMA activity.

Beginning in the third machine cycle of each line of the display window, and lasting for 8 cycles, the CDP1861C asserts the $\overline{\text{DMA REQ}}$ output to request a sequence of eight 8-bit bytes, which are then used to generate the $\overline{\text{VIDEO}}$ signal. Then, when control signals A and B are low and high respectively, each assertion of the LOAD input causes the CDP-1861C to read a byte from the BUS lines, and immediately to shift it out on the $\overline{\text{VIDEO}}$ output, high-order bit first. A DMA pointer defines an area of memory which is accessed by the CDP1861C to provide a bit-mapped display.

The display on (DISP ON) and display off (DISP OFF) inputs set and reset an internal control flip-flop in the CDP1861C. When this flip-flop is set, $\overline{\text{DMA REQ}}$ and $\overline{\text{INT REQ}}$ are enabled; when reset, they are disabled.

The reset input ($\overline{\text{RESET IN}}$) is a Schmitt trigger input that resets the CDP1861C. The $\overline{\text{CLEAR}}$ output is a conditioned output pulse which can be used to reset the external system.

The CDP1861C is supplied in a 24-lead dual-in-line ceramic package.

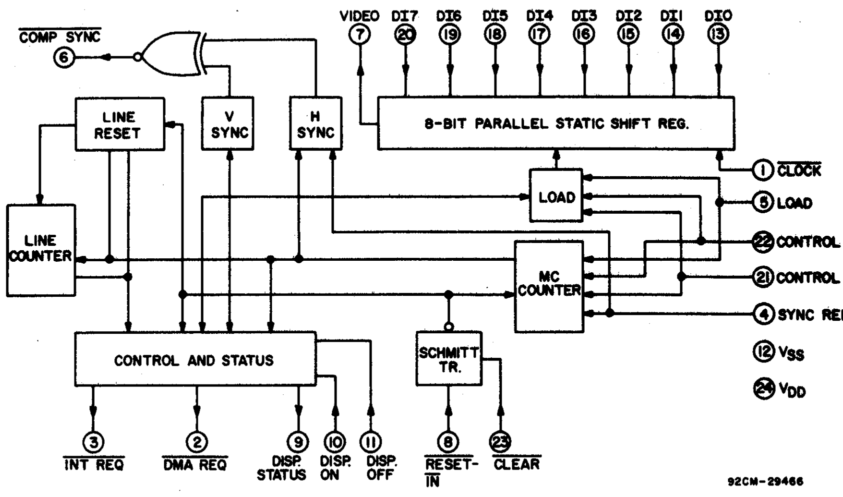
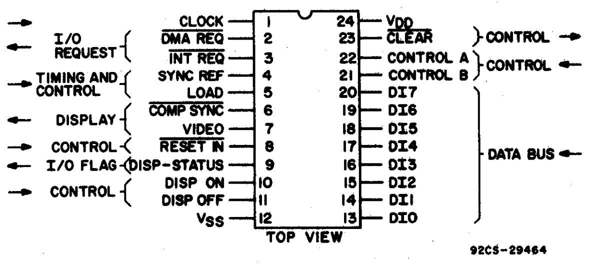


Fig. 2 — CDP1861C block diagram.



TERMINAL ASSIGNMENT

MAXIMUM RATINGS, *Absolute-Maximum Values:*

DC SUPPLY-VOLTAGE RANGE (V_{DD}) (All voltage values referenced to V_{SS} terminal)	–0.5 to +7 V
INPUT VOLTAGE RANGE, ALL INPUTS	–0.5 to V_{DD} +0.5 V
POWER DISSIPATION PER PACKAGE (P_D):	
For T_A = –55 to +100°C	500 mW
For T_A = +100 to +125°C	Derate Linearly to 200 mW
DEVICE DISSIPATION PER OUTPUT TRANSISTOR:	
For T_A = –55°C to +125°C	100 mW
STORAGE-TEMPERATURE RANGE (T_{stg})	–65 to +150°C
OPERATING-TEMPERATURE RANGE (T_A)	–55 to +125°C
LEAD TEMPERATURE (DURING SOLDERING):	
At distance 1/16 ± 1/32 inch (1.59 ± 0.79 mm) from case for 10 s max.	+265°C

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)	TYPICAL VALUES	UNITS
Supply-Voltage Range (For T_A = Full Package-Temperature Range)	–	4 - 6	V
Input Voltage Range	–	$V_{SS} - V_{DD}$	V
Input Signal Rise or Fall Time	5	5	μs
Clock Input Frequency, f_{CL}	5	0 - 2.5	MHz

ELECTRICAL CHARACTERISTICS AT T_A = 25 °C

CHARACTERISTIC		CONDITIONS V_O (V)	CONDITIONS V_{DD} (V)	TYPICAL VALUES	UNITS
Maximum Quiescent Device Current, I_L		–	5	500	μA
Minimum Output Drive Current:					
Video or Sync	N-Channel (Sink), I_DN	0.4	5	1.2	mA
	P-Channel (Source), I_DP	4.5	5	1	
Reset Out or Flag	N-Channel (Sink), I_DN	0.4	5	–0.4	mA
	P-Channel (Source), I_DP	4.5	5	0.4	
I/O Requests;	N-Channel (Sink), I_DN	0.4	5	–0.2	mA
Reset-In:					
Positive Trigger Threshold, V_P		–	5	2.5	V
Negative Trigger Threshold, V_N		–	5	1.7	V
Hysteresis Voltage, V_H		–	5	0.8	V

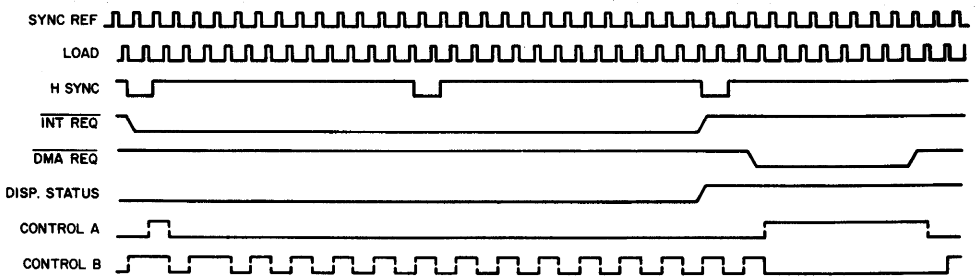


Fig. 3 — Horizontal sync timing diagram.

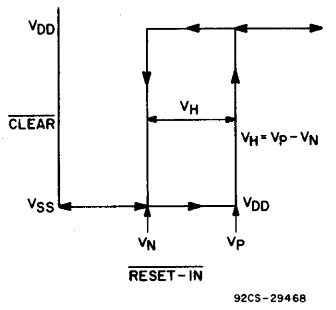


Fig. 4 — Reset transfer characteristics.

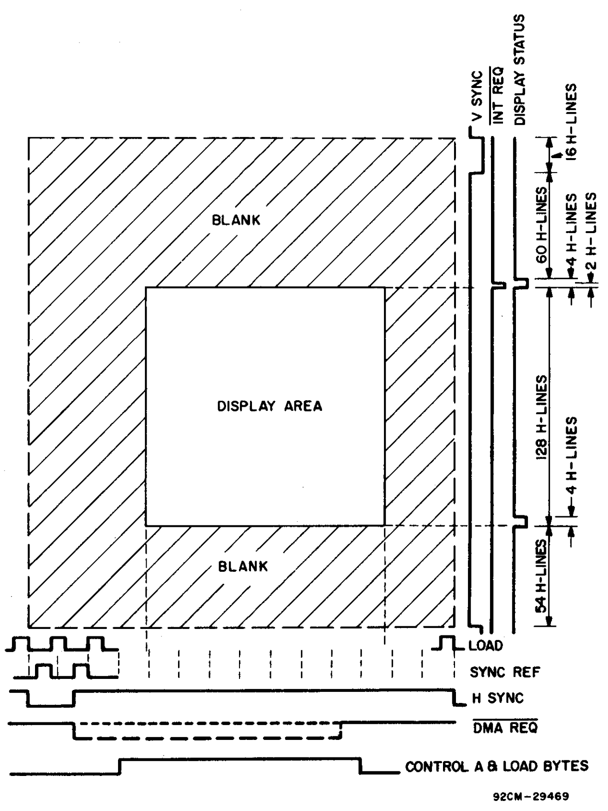


Fig. 5 — Spatial diagram of one video display field (not to scale).

Application Information (CDP1861C directly controlled by the CDP1802 microprocessor)

Figure 6 shows a simple graphic display system using the CDP1802 and the CDP1861C. The CDP1861C uses both the INTERRUPT and direct memory access (DMA) output channel of the microprocessor for display refresh. The microprocessor specifies the area of memory displayed via the interrupt routines, and the DMA output channel is the mechanism which transfers the data from memory to the CDP1861C via the 8-bit data bus. The data are then shifted out one bit at a time at the clock frequency to generate the video ($\overline{\text{VIDEO}}$) signal.

The composite sync ($\overline{\text{COMP SYNC}}$) signal creates a 262-line-per-field, 60-field-per-second non-interlace video picture. The non-interlaced picture frame for this display consists of two even fields of 262 horizontal lines each. This format differs slightly from the National Television Standard (NTSC) which has a 525-line interlaced picture frame of one odd field and one even field. The vertical sync pulse generated at $\overline{\text{COMP SYNC}}$ of the CDP-1861C has no equalizing pulses but is serrated to maintain horizontal synchronization during the vertical blanking time. The VIDEO and $\overline{\text{COMP SYNC}}$ pulses are resistively coupled to create the composite video, which can be supplied directly to a video monitor, a modified TV receiver, or an FCC approved rf modulator.

A clock source of 3.58 MHz, the NTSC color frequency, if divided by 2, may be used for some applications in place of the 1.76-MHz crystal shown in Fig. 6. Deviations from the NTSC frequencies are as follows:

NTSC		Clock Frequencies (MHz)		
		1.76064	**1.764000**	**3.579545/2**
Line Freq.	15750	15720	15750	15980
Field Freq.	60	60	60.11	60.99

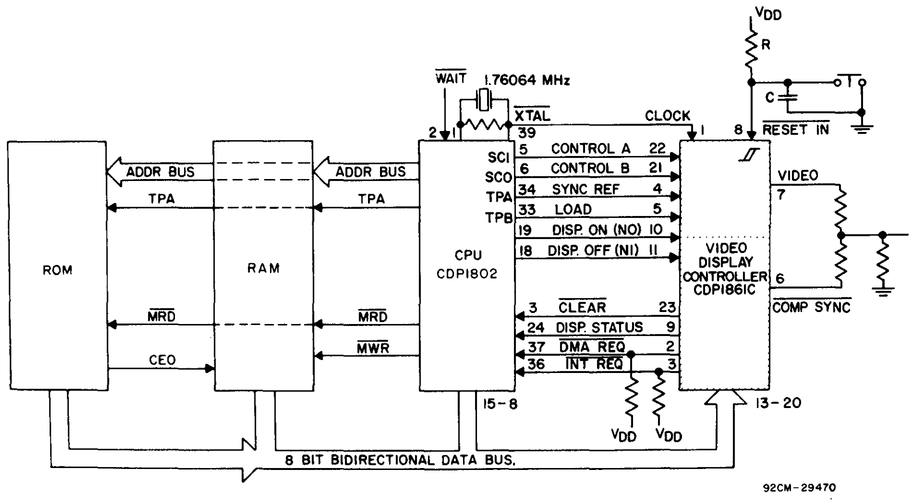


Fig. 6 — Typical CDP1802/CDP1861C video display system.

The user should determine which choice of frequencies provides an optimal cost/performance trade-off for his application. Generally, video CRT's are more sensitive to line frequency accuracy than to field frequency accuracy.

The display is a bit map of memory. Each bit in the display memory corresponds to one spot on the video screen. Logical 1 (V_{DD}) bits in memory correspond to white or lighted spots in the display. The highest resolution that may be produced without any hardware modifications is 128 vertical by 64 horizontal segments. This resolution requires 1024 bytes of memory for the display. The upper left-most spot that can be displayed on the video screen is the most significant bit of the first byte in the display refresh memory buffer. The starting location of the display buffer is initialized in the INTERRUPT routine and may be anywhere in addressable memory (ROM, RAM, or both). The lower right-most spot that can be displayed is the least significant bit of the last byte of the display bit map. For each of the 128 horizontal display lines, 8 bytes of memory are sequentially accessed and displayed from left to right on the video screen. Adjacent illuminated spots appear contiguous both in the horizontal and in the vertical directions. All display manipulations are accomplished by changing the data within the display buffer or by changing display buffers.

To control the CDP1861C as shown in Fig. 6, the CDP1802 must be in synchronization with the CDP1861C during the display window. Exactly six machine cycles must be executed beyond the eight DMA cycles during each line, and an even number of cycles (262 x 14) must be executed from the start of one display window to the start of the next. These requirements insure that the DMA bursts will not be delayed one cycle waiting for an instruction to finish—this delay would cause jitter on the screen. These requirements can be accomplished in two steps: 1) the main program must not execute any 3-cycle instructions (i.e., SKIPS, LONG BRANCHES, and NOP), and 2) the interrupt routine, including the interrupt cycle itself, must employ an even number of cycles, and must be synchronized with the DMA bursts. There must be 29 cycles between the INTERRUPT cycle (S3) and the first burst of eight DMA cycles. This timing is accomplished by executing an early 3-cycle instruction to compensate for the INTERRUPT cycle. Furthermore, exactly three 2-cycle instructions must be executed between each successive burst. Occasionally these restrictions may be ignored at the expense of jitter on the screen.

For the 128 x 64 display, the CDP1802 software requirement is straightforward. The DISP STATUS/$\overline{\text{EF1}}$ line is not required, and EF1 may be used for other purposes. A simple interrupt routine merely resets the DMA pointer, RO, to the beginning of the display buffer area (see Fig. 5)—note the 3-cycle NOP instruction at the beginning which compensates for the 1-cycle interrupt. The first burst of eight DMA cycles occurs just as this routine finishes, as indicated by the bracket following the RETURN instruction (70). Exactly 29 cycles separate the interrupt request cycle and the first DMA burst. The interrupt routine must last at least 28 cycles, because the interrupt request line is held up that long by the CDP1861C.

When less RAM is to be used (less resolution), a more complicated interrupt routine is used. The interrupt routine is protracted for the full duration of the display window, and the six free cycles in each line are used to execute three instructions, which maintain control over the DMA pointer, RO.1. In the simplest cases, each line of 8 bytes is repeated n times to give 128/n vertical resolution. With n = 4, for example, 64 x 32 resolution is obtained. Such an interrupt routine is shown in Fig. 8. The code from the entry at INTERRUPT to DISPLAY is as in the last example. The use of three instructions per line does not leave time to control a loop, so each of four copies of the line corresponds to three instructions in the main loop, starting at DISPLAY STATUS. The DISPLAY STATUS signal, applied to EF1, is used to RO.1 in the last pass through the loop, when RO advances into the next page after each burst.

For other values of n, similar routines can be devised. For n = 2, the 64 x 64 format, the last 4 lines need special treatment (see Fig. 7). Other schemes are possible, resulting in other resolutions which vary on command from the main program, or even resolutions which vary through the display window.

In general, additional functions may be implemented in the routine before returning to the main program. For example, a real-time clock can be maintained by incrementing a counter once on each interrupt, i.e., once per 1/60 second. Another example is vertical "scrolling" of the display, wherein the starting address in a display file is incremented or reincremented at regular intervals.

Signal Definitions

Signal	**Term.No.**	**Definition**
$\overline{\text{RESET IN}}$	8	An input signal which, when low (V_{SS}), initializes the counters, inhibits the display, and places all control outputs in the high (V_{DD}) state. The $\overline{\text{RESET IN}}$ terminal is a Schmitt-trigger-type input which permits the use of an external RC network to provide a power-on reset.
$\overline{\text{CLEAR}}$	23	The output of the Schmitt trigger (reset input circuitry) provides high speed transitions that may be used to reset other devices. It may be connected to the $\overline{\text{CLEAR}}$ terminal of the CDP1802 microprocessor.
DISPLAY-ON DISPLAY-OFF	10 11	Positive input signals that control the display. When enabled (DISPLAY-ON = V_{DD}), data transfers, DMA, and interrupt requests are permitted. These operations are inhibited by the low-to-high transition of the DISPLAY-OFF input signal if DISPLAY-ON = V_{SS}. The $\overline{\text{RESET IN}}$ signal also inhibits the display. When inhibited, the internal counters remain operational. Sync and display status signals are generated. Video output becomes low when the register is emptied. Table I indicates the enable/disable conditions.

Table I

STATE	SIGNAL		
	$\overline{\text{RESET-IN}}$	DISPLAY-ON	DISPLAY-OFF
RESET	L	L	X
INVALID	L	H	X
TV ENABLE	H	H	X
TV DISABLE	H	L	↑

Signal Definitions (Cont'd)

Signal	Term.No.	Definition
		The DISPLAY-ON and DISPLAY-OFF signals may be provided by the I/O commands (N bits) of the CDP1802 microprocessor.
CLOCK	1	The input for an externally generated single-phase clock which determines the clock rate for the 8-bit data shift register. Data are shifted on the high-to-low transition of the CLOCK input signal, most significant bit first. A low level ("0") is shifted into the least significant bit. The CLOCK signal may be derived directly from the CDP1802 microprocessor by connecting the CLOCK terminal of the CDP1861C to the XTAL terminal of the CDP1802.
SYNC REFERENCE LOAD	4 5	Positive timing pulses each occurring once for every 8 clock pulses. The SYNC REFERENCE signal precedes the LOAD signal. The SYNC REFERENCE signal is used as the clock for the horizontal line counter. The LOAD signal is used as a strobe for gating the output of the counter and for loading data into the data register. They are normally connected to the TPA and TPB terminals of the CDP1802 microprocessor.
$\overline{\text{COMP SYNC}}$	6	Negative (high going low) output signal resulting from the exclusive "OR" of the output of the horizontal and vertical counters. $\overline{\text{COMP SYNC}}$ can be combined with the $\overline{\text{VIDEO}}$ output to form a composite video signal. The $\overline{\text{COMP SYNC}}$ output frequency and pulse duration are determined by the SYNC REFERENCE and LOAD input signals. A horizontal sync pulse is initiated by the trailing edge of the LOAD input signal following the thirteenth or fourteenth SYNC REFERENCE input, as determined by the status of the CONTROL A and CONTROL B input signals, and is terminated on the leading edge of the subsequent second count of the SYNC REFERENCE input. Vertical timing is generated coincident with the 262 horizontal timing pulse and is present for six horizontal clock cycles. Idealized timing is illustrated in Figs. 3 and 5.
$\overline{\text{INTERRUPT REQUEST}}$	3	A low (V_{SS}) output signal two horizontal cycles prior to the display, as shown in Figs. 3 and 5. This signal is the output of the "open drain" of an n-channel transistor and requires an external pull-up resistor to V_{DD}. The $\overline{\text{INTERRUPT REQUEST}}$ output signal is normally connected to the INTERRUPT input terminal of the CDP1802 microprocessor. In a CDP1802-based system 29 machine cycles occur from initiation of an $\overline{\text{INTERRUPT REQUEST}}$ until the $\overline{\text{DMA REQUEST}}$.
DISPLAY STATUS	9	A low (V_{SS}) output signal which occurs for a period of four horizontal cycles prior to the beginning and end of the 128-line display window, as illustrated in Figs. 3 and 5. The signal can be used by the program software routines to indicate the boundaries of the display area. It is normally connected to a CDP1802 FLAG input terminal.

Signal Definitions (Cont'd)

Signal	**Term.No.**	**Definition**
$\overline{\text{DMA REQUEST}}$	2	A low output (V_{SS}) that requests an 8-bit data transfer. The output signal is from the "open drain" of an n-channel transistor and requires an external pull-up resistor to V_{DD}. Depending upon the status of the CONTROL A and CONTROL B input signals at horizontal sync time, DMA requests are initiated on the leading edge of the second SYNC REFERENCE input signal following the horizontal sync output. This feature is necessary in order to reference the data requests to the program's ability to respond to them, insuring that data will always be initiated at the same point on the display.
		The system should respond to a $\overline{\text{DMA REQUEST}}$ by setting CONTROL B high (V_{DD}), and CONTROL A low (V_{SS}) permitting data transfer. Data will be loaded on the subsequent 8 LOAD input signals. $\overline{\text{DMA REQUEST}}$ will be terminated on the ninth sync pulse, at which time CONTROL B should be set low (V_{SS}) prior to the next LOAD command. Timing is illustrated in Figs. 3 and 5. The $\overline{\text{DMA REQUEST}}$ output signal may be connected to the DMA IN terminal of the CDP1802 microprocessor, which responds as discussed above.
CONTROL A CONTROL B	22 21	Input signals used to synchronize the operation of the CDP1861C with its controller. They should be initiated prior to the SYNC REFERENCE input and terminate after the LOAD input pulse.
		The CONTROL signals are sampled at two different times: 1) During the horizontal sync output when the SYNC REFERENCE input is present, the CDP1861C expects to see CONTROL A = 1 (V_{DD}), and CONTROL B = 0 (V_{SS}). Any other combination will result in the skipping of one of the normal 14 cycles per line. This feature allows the CDP1802 to force initial instruction fetch/execute sync with the CDP1861C, and assures sync in case it is later lost for any reason. 2) In the 8 cycles following the CDP1861C $\overline{\text{DMA REQUEST}}$ assertion, the CDP1861C expects to see CONTROL A = 0, and CONTROL B = 1. Any other combination will prevent the CDP1861C from loading data from the bus.
		These signals may be connected to the STATE CODE outputs of the CDP1802 microprocessor; CONTROL A to SC0 and CONTROL B to SC1.
DI7 - DI0	13-20	Input signals to the data register. Data are loaded during the high-to-low transition of the CLOCK only when LOAD = V_{DD} and the CDP1861C is enabled. DISPLAY-ON = 1 (V_{DD}), CONTROL A = 0 (V_{SS}), and CONTROL B = 1 (V_{DD}).
		The data input signals are normally connected to the 8-bit microprocessor data bus.
VIDEO	7	Output from the most significant bit of the data register. It is used to determine the luminance level and may be combined externally with the $\overline{\text{COMP SYNC}}$ output signal to form a composite video signal.

```
Machine Code   Assembly Language            Comments
  72           INTRET:  LDXA                .. RESTORE D
  70                    RET                 .. RETURN
  C4           INT   :  NOP                 .. 3 CYC.INSTR.FOR PGM.SYNC
  22                    DEC R2              .. R2 IS STACK PTR
  78                    SAV                 .. T→STACK
  22                    DEC R2
  52                    STR R2              .. D→STACK
  F8__B0                A.1(DISMEM)→RO.1    .. DISMEM IS START ADDR
  F8__A0                A.0(DISMEM)→RO.0    .. OF DISPLAY MEMORY
  C4, C4                NOP; NOP            .. NOPS FOR PGM SYNC
  E2                    SEX2
  80]          DISP  :  GLO RO              .. NEW LINE
  E2                    SEX2                .. NOP
  20                    DEC RO              .. RESTORES RO.1 IF PASS PG
  A0]                   PLO RO              .. REPEATS SAME LINE
  E2                    SEX2                .. NOP
  3C__                  BN1 DISP            .. LOOP 60 TIMES
  80]          DISEF :  GLO RO              .. LAST 4 VIDEO LINES
  E2                    SEX2                .. NOP
  20 A0]                DEC RO; PLO RO
  E2                    SEX2                .. NOP
  34__                  B1 DISEF
  30__                  BR INTRET           .. END OF DISPLAY
```

Fig. 7 — Interrupt routine for 64 x 64 format (2 pgs mem).

```
Machine Code   Assembly Language            Comments
  72           INTRET:  LDXA                .. RESTORE D
  70                    RET                 .. RETURN
  C4           INT   :  NOP                 .. 3 CYC. INSTR. USED
                                            .. FOR PGM. SYNC
  22                    DEC R2              .. R2 IS STACK PTR
  78                    SAV                 .. T→STACK
  22                    DEC R2
  52                    STR R2              .. D →STACK
  F8__B0                A.1(DISMEM)→RO.1    .. LOAD RO WITH
  F8__A0                A.0(DISMEM)→RO.0    .. START.ADDR.OF DISP.MEM
  C4, C4                NOP; NOP            .. NOPS USED FOR SYNC
  E2           DISP  :  SEX2
  80]                   GLO RO              .. LINE START ADDR.→D
  E2                    SEX2                .. NOP
  20                    DEC RO              .. RESET RO.1 IF PASS PG
  A0]                   PLO RO              .. LINE START ADDR.→RO.0
  E2                    SEX2                .. NOP
  20                    DEC RO              .. RESET RO.1 IF PASS PG
  A0]                   PLO RO              .. LINE START ADDR.→RO.0
  E2                    SEX2                .. NOP
  20                    DEC RO              .. RESET RO.1 IF PASS PG
  A0                    PLO RO              .. REPEATS SAME LINE
  3C__                  BN1 DISP            .. LOOPS 32 TIMES
  30__                  BR INTRET           .. END OF DISPLAY
```

Fig. 8 — Interrupt routine for 64 x 32 format (1 pg mem).

```
Machine Code   Assembly Language                Comments
  72           INTRET:  LDXA                    .. RESTORE D
  70]                   RET                     .. RETURN
  C4           INT   :  NOP                     .. ENTRY POINT
  22                    DEC R2                  .. R2 = STACK PTR
  78                    SAV                     .. T → STACK
  22                    DEC R2
  52                    STR R2                  .. D → STACK
  E2, E2                SEX R2; SEX R2          .. NOP
  F8__B0                A.1(DISMEM) → RO.1      .. LOAD RO WITH
  F8__A0                A.0(DISMEM) → RO.0      .. START ADDR OF DISP.MEM.
  30__                  BR INTRET               .. OR INSERT OTHER COMMENT
```

Fig. 9 — Interrupt routine for 64 x 128 (4 pgs mem).

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

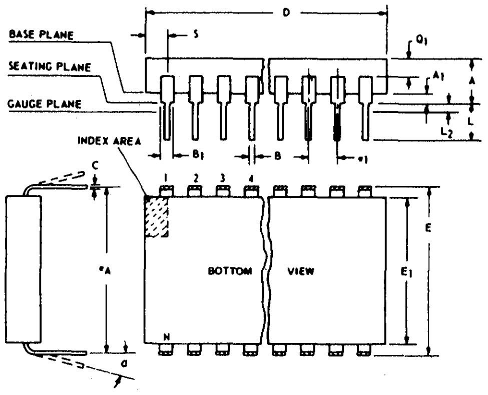
A connection must be provided at every input terminal. All unused input terminals must be connected to either V_{DD} or V_{SS}, whichever is appropriate.

Output Short Circuits

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

DIMENSIONAL OUTLINE

D Suffix
24-Lead Dual-In-Line Ceramic Package
JEDEC MO-015-AG



92CS19948R1

SYMBOL	INCHES		NOTE	MILLIMETERS	
	MIN.	MAX.		MIN.	MAX.
A	0.090	0.150		2.29	3.81
A_1	0.020	0.065	2	0.51	1.65
B	0.015	0.020		0.381	0.508
B_1	0.045	0.055		1.143	1.397
C	0.008	0.012	1	0.204	0.304
D	1.15	1.22		29.21	30.98
E	0.600	0.625		15.24	15.87
E_1	0.480	0.520		12.20	13.20
e_1	0.100 TP		3	2.54 TP	
e_A	0.600 TP		3	15.24 TP	
L	0.100	0.180		2.54	4.57
L_2	0.000	0.030	3	0.00	0.76
α	0°	15°	4	0°	15°
N	24		5	24	
N_1	0		6	0	
Q_1	0.020	0.080		0.51	2.03
S	0.020	0.060		0.51	1.52

NOTES:

Refer to JEDEC Publication No. 95 for Rules for Dimensioning Axial Lead Product Outlines.

1. When this device is supplied solder dipped, the maximum lead thickness (narrow portion) will not exceed 0.013" (0.33 mm).
2. When base of body is to be attached to heat sink, terminal lead standoffs are not required and $A_1 = 0$. When $A_1 = 0$, the leads emerge from the body with the B_1 dimension and reduce to the B dimension above the seating plane.
3. e_1 and e_A apply in zone L_2 when unit is installed. Leads within 0.005 " (0.127 mm) radius of True Position (TP) at gauge plane with maximum material condition.
4. Applies to spread leads prior to installation.
5. N is the maximum quantity of lead positions.
6. N_1 is the quantity of allowable missing leads.

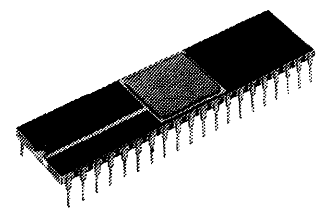
When incorporating RCA Solid State Devices in equipment, it is recommended that the designer refer to "Operating Considerations for RCA Solid State Devices", Form No. 1CE-402, available on request from RCA Solid State Division, Box 3200, Somerville, N. J. 08876.

RCA
Solid State Division

Microprocessor Products
CDP1802D
CDP1802CD

Preliminary Data

COSMAC Microprocessor



40-Lead Dual-In-Line Ceramic Package (D)

CDP1802D
CDP1802CD H-1847

Features:

- **Instruction fetch-execute time of 2.5 or 3.75 μs at V_{DD} = 10 V; 5.0 or 7.5 μs at V_{DD} = 5 V**
- **Static silicon-gate CMOS circuitry — no minimum clock frequency**
- **Full military temperature range (–55 to +125°C)**
- **High noise immunity, wide operating-voltage range**
- **Single voltage supply**
- **Low power**
- **Single-phase clock; optional on-chip crystal-controlled oscillator**
- **TTL compatible**
- **On-chip DMA**
- **Simple control of reset, run, and pause**
- **8-bit parallel organization with bidirectional data bus**
- **Any combination of standard RAM and ROM**
- **Memory addressing up to 65,536 bytes**
- **Flexible programmed I/O mode**
- **Program interrupt mode**
- **Four I/O flag inputs directly tested by branch instructions**
- **Programmable output port**
- **91 easy-to-use instructions**
- **16 x 16 matrix of registers for use as multiple program counters, data pointers, or data registers**

The RCA-CDP1802 is an LSI COS/MOS 8-bit register-oriented central-processing unit (CPU) designed for use as a general-purpose computing or control element in a wide range of stored-program systems or products.

The CDP1802 includes all of the circuits required for fetching, interpreting, and executing instructions which have been stored in standard types of memories. Extensive input/output (I/O) control features are also provided to facilitate system design.

The COSMAC architecture is designed with emphasis on the total microcomputer system as an integral entity so that systems having maximum flexibility and minimum cost can be realized. The COSMAC CPU also provides a synchronous interface to memories and external controllers for I/O devices, and minimizes the cost of interface controllers. Further, the I/O interface is capable of supporting devices operating in polled, interrupt-driven, or direct memory-access modes.

The CDP1802D and CDP1802CD are functionally identical. They differ in that the CDP1802D has a recommended operating voltage range of 4-12 volts, and the CDP 1802CD, a recommended operating voltage range of 4-6 volts. These types are supplied in 40-lead dual-in-line ceramic packages (D suffix).

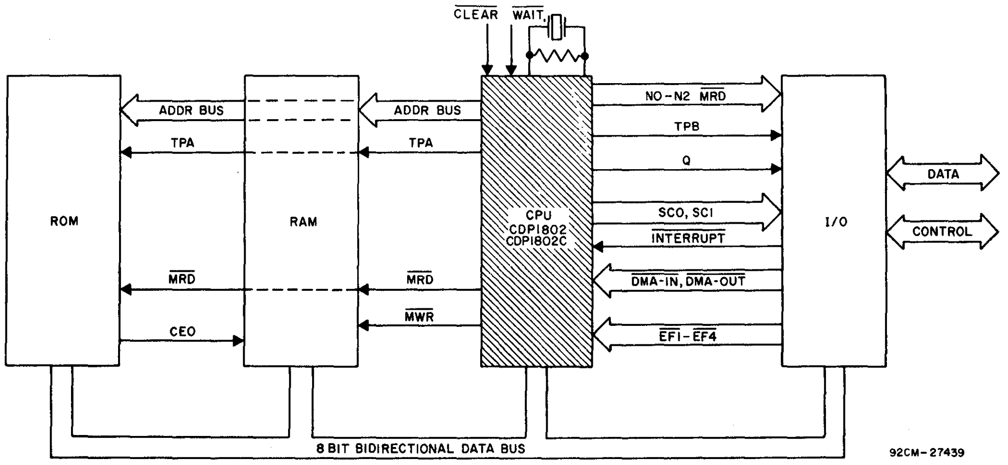


Fig. 1 — Typical CDP1802 microprocessor system.

The Preliminary Data are intended for guidance purposes in evaluating the device for equipment design. The device is now being designed for inclusion in our standard line of commercially available products. For current information on the status of this program, please contact your RCA Sales Office.

Printed in USA/8-77

Supersedes preliminary data issued 2/76

MAXIMUM RATINGS, *Absolute-Maximum Values:*

DC SUPPLY-VOLTAGE RANGE, (V_{CC}, V_{DD})	
(All voltage values referenced to V_{SS} terminal)	
$V_{CC} \leqslant V_{DD}$:	
CDP1802D	−0.5 to +15 V
CDP1802CD	−0.5 to +7 V
INPUT VOLTAGE RANGE, ALL INPUTS	−0.5 to V_{DD} +0.5 V
DC INPUT CURRENT, ANY ONE INPUT	±10 mA
POWER DISSIPATION PER PACKAGE (P_D):	
For T_A = −55 to +100°C	500 mW
For T_A = +100 to +125°C	Derate Linearly at 12 mW/°C to 200 mW
DEVICE DISSIPATION PER OUTPUT TRANSISTOR FOR T_A = FULL PACKAGE-TEMPERATURE RANGE	100 mW
OPERATING-TEMPERATURE RANGE (T_A)	−55 to +125°C
STORAGE TEMPERATURE RANGE (T_{stg})	−65 to +150°C
LEAD TEMPERATURE (DURING SOLDERING):	
At distance 1/16 ± 1/32 inch (1.59 ± 0.79 mm) from case for 10 s max.	+265°C

CHARACTERISTIC	CONDITIONS			LIMITS AT INDICATED TEMPERATURES (°C)							UNITS
	V_O (V)	V_{IN} (V)	V_{CC}, V_{DD} (V)	VALUES				+25			
				−55	−40	+85	+125	Min.	Typ.	Max.	
Quiescent Device Current, I_L Max. CDP1802D	—	—	5	—	—	—	—	—	1	100	µA
	—	—	10	—	—	—	—	—	10	500	µA
	—	—	15	—	—	—	—	—	—	1000	µA
CDP1802CD	—	—	5	—	—	—	—	—	—	500	µA
Output Low Drive (Sink) Current, I_{OL} Min. (Except $\overline{XTAL}$)	0.4	0,5	5	1.98	1.89	1.14	0.90	1.5	2.2	—	mA
	0.5	0,10	10	3.70	3.53	2.13	1.68	2.8	5.2	—	mA
$\overline{XTAL}$ Output I_{OL} Min.	0.4	5	5	132	126	76	60	100	—	—	µA
Output High Drive (Source Current) I_{OH} Min. (Except $\overline{XTAL}$)	4.6	0,5	5	−0.46	−0.44	−0.27	−0.21	−0.35	−0.51	—	mA
	9.5	0,10	10	−1.12	−1.07	−0.65	−0.51	−0.85	−1.3		mA
$\overline{XTAL}$ Output I_{OH} Min.	4.6	0	5	−66	−63	−38	−30	−50	—	—	µA
Output Voltage Low-Level V_{OL} Max.	—	0,5	5	0.05				—	0	0.05	V
	—	0,10	10	0.05				—	0	0.05	V
Output Voltage High Level, V_{OH} Min.	—	0,5	5	4.95				4.95	5	—	V
	—	0,10	10	9.95				9.95	10	—	V
Input Low Voltage V_{IL} Max.	0.5,4.5	—	5	1.5				—	—	1.5	V
	0.5,4.5	—	5,10	1				—	—	1	V
	1,9	—	10	3				—	—	3	V
Input High Voltage V_{IL} Min.	0.5,4.5	—	5	3.5				3.5	—	—	V
	0.5,4.5	—	5,10	4				4	—	—	V
	1,9	—	10	7				7	—	—	V
Input Leakage Current I_{IN} Max.	Any Input	0,15	15	±1				—	—	±1	µA
3-State Output Leakage Current I_{OUT} Max.	0,15	0,15	15	±1	±1	±12	±12	—	$\pm 10^{-4}$	±1	µA

RECOMMENDED OPERATING CONDITIONS at T_A = 25°C Unless Otherwise Specified

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

CHARACTERISTIC	CONDITIONS		LIMITS AT 25°C		UNITS
	V_{CC}[1] (V)	V_{DD} (V)	CDP1802D	CDP1802CD	
Supply-Voltage Range	–	–	4 to 12	4 to 6	V
Input Voltage Range	–	–	V_{SS} to V_{CC}	V_{SS} to V_{CC}	V
Maximum Clock Input Rise or Fall Time, t_r or t_f	4–12	4–12	1	1	μs
Instruction Time[2] (See Fig. 8)	5	5	5	5	μs
	5	10	4	–	
	10	10	2.5	–	
Maximum DMA Transfer Rate	5	5	400	400	KBytes/sec
	5	10	500	–	
	10	10	800	–	
Maximum Clock Input Frequency, f_{CL}[3]	5	5	DC – 3.2	DC – 3.2	MHz
	5	10	DC – 4	–	
	10	10	DC – 6.4	–	

NOTES:
1: $V_{CC} \leqslant V_{DD}$; for CDP1802CD, $V_{DD} = V_{CC}$ = 5 volts.
2. Equals 2 machine cycles – one Fetch and one Execute operation for all instructions except Long Branch and Long Skip, which require 3 machine cycles – one Fetch and two Execute operations.
3. Load Capacitance (C_L) = 50 pF.

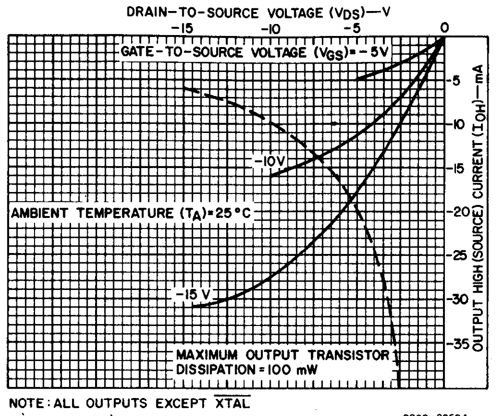


Fig. 2 – Typical output high (source) current characteristics.

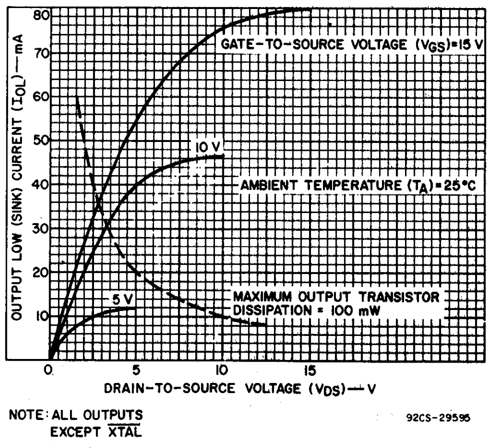


Fig. 3 – Typical output low (sink) current characteristics.

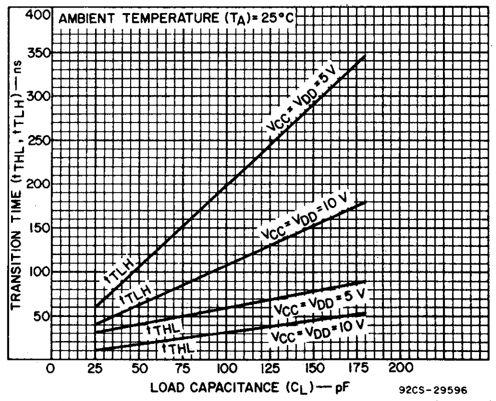


Fig. 4 – Typical transistion time vs. load capacitance.

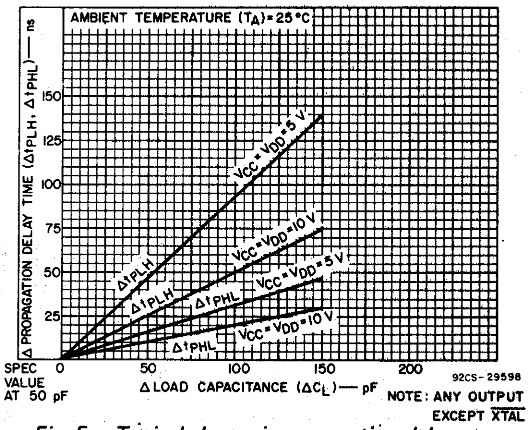


Fig. 5 – Typical change in propagation delay as a function of a change in load capacitance.

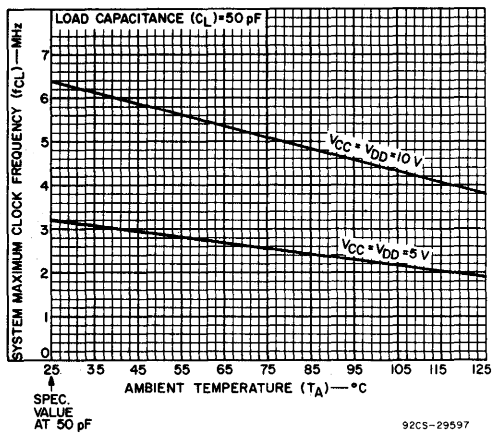


Fig. 6 — Typical maximum clock frequency as a function of temperature.

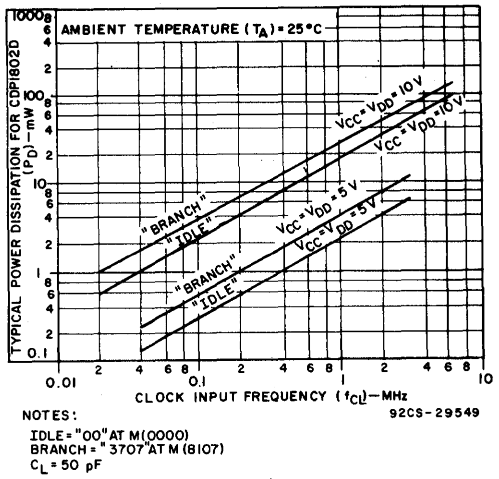


Fig. 7 — Typical power dissipation as a function of clock frequency for BRANCH instruction and IDLE instruction for CDP1802D.

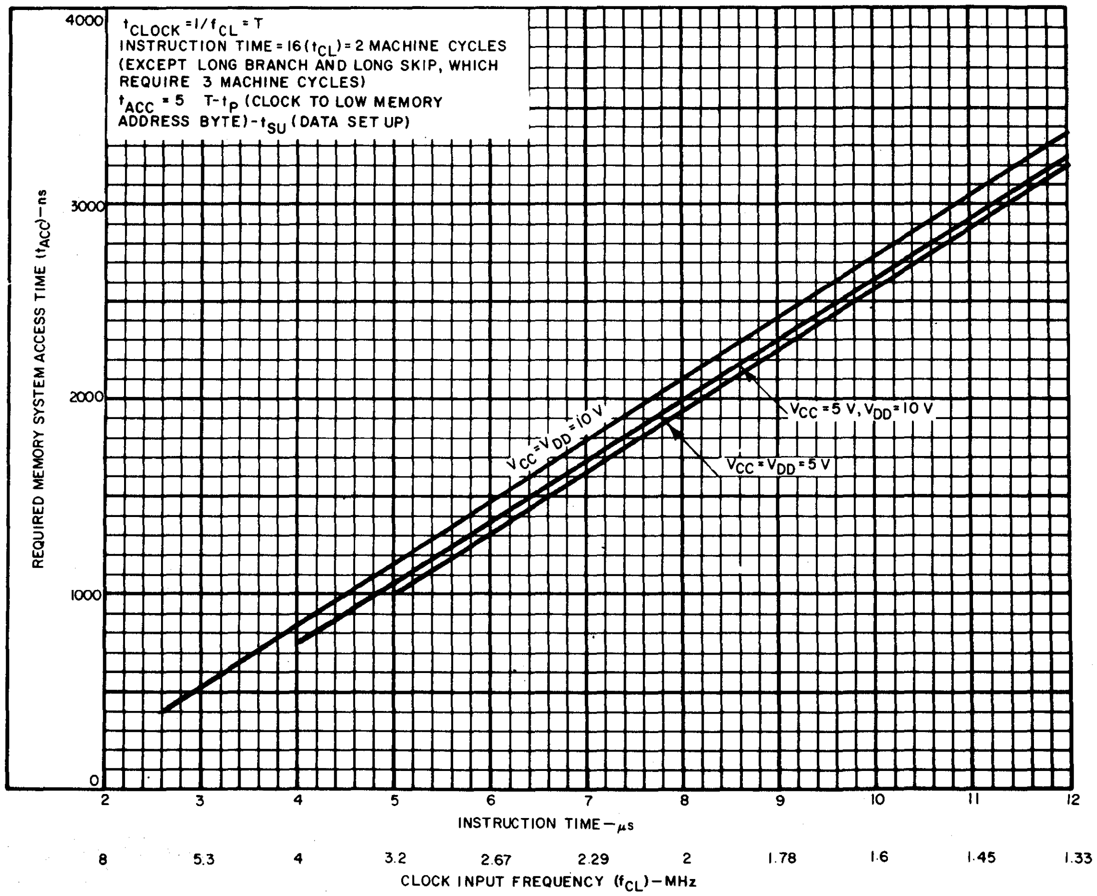


Fig. 8 — Required memory system address time as a function of instruction time. 92CL-29643

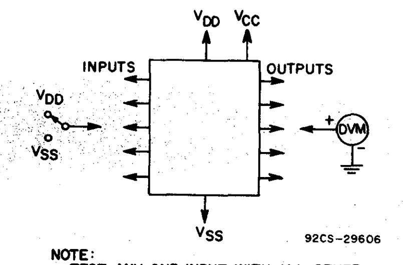

VDD
VCC
INPUTS
OUTPUTS
VDD
VSS
DVM
VSS
92CS-29606
NOTE:
TEST ANY ONE INPUT WITH ALL OTHER INPUTS AT "NOISE" VOLTAGE LEVELS.

Fig. 9 — Noise immunity test circuit.

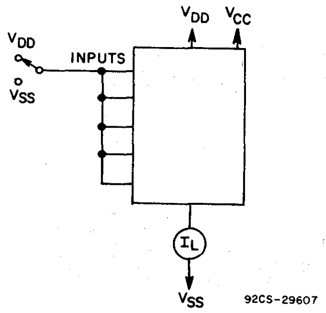


Fig. 10 — Quiescent-device leakage current test circuit.

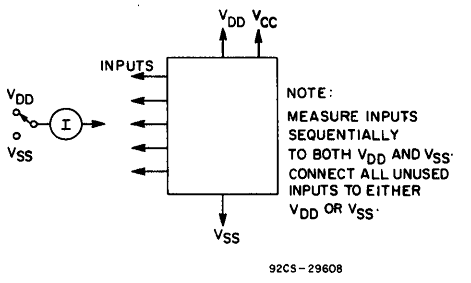

VDD VCC
INPUTS
VDD
VSS
NOTE:
MEASURE INPUTS SEQUENTIALLY TO BOTH VDD AND VSS. CONNECT ALL UNUSED INPUTS TO EITHER VDD OR VSS.
VSS
92CS-29608

Fig. 11 — Input leakage current test circuit.

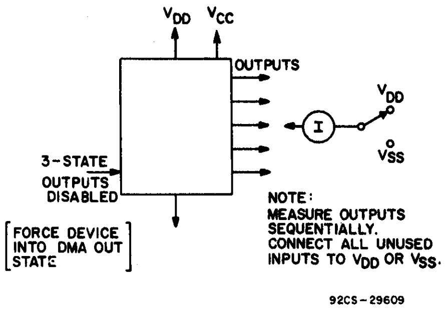
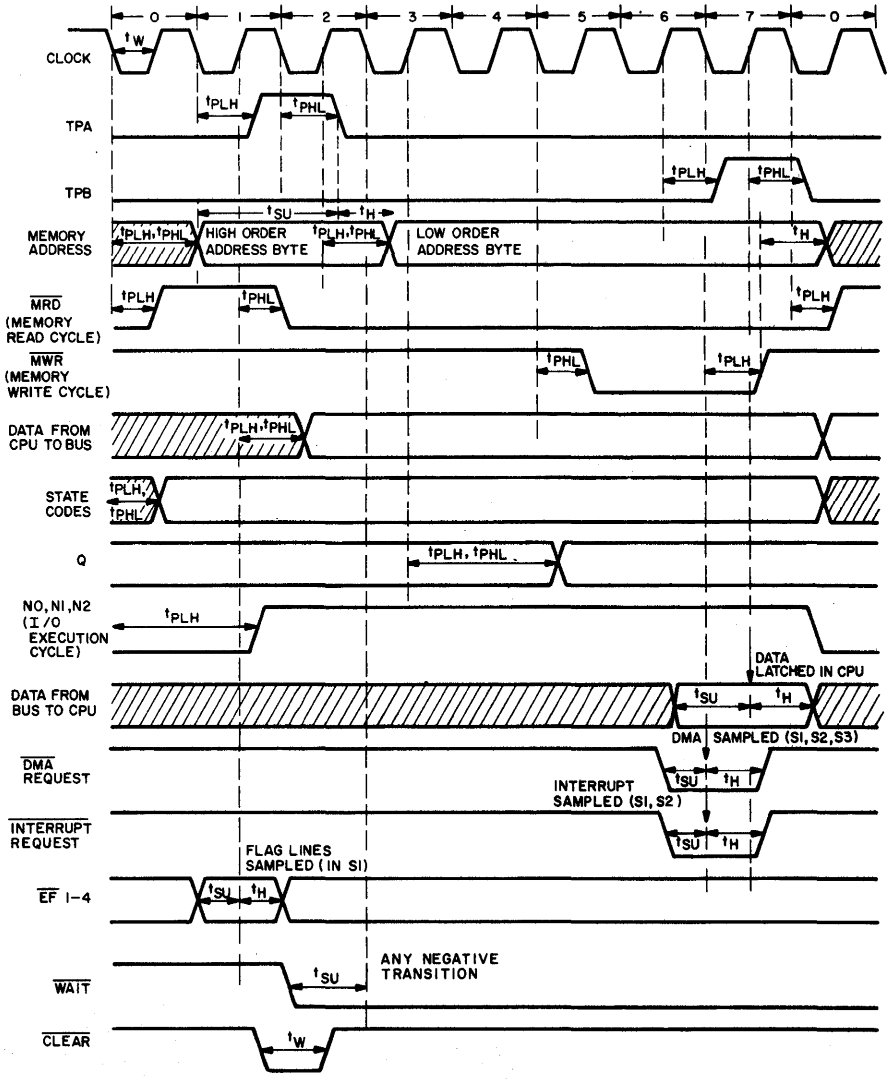

VDD VCC
OUTPUTS
VDD
VSS
3-STATE OUTPUTS DISABLED
FORCE DEVICE INTO DMA OUT STATE
NOTE:
MEASURE OUTPUTS SEQUENTIALLY. CONNECT ALL UNUSED INPUTS TO VDD OR VSS.
92CS-29609

Fig. 12 — Three-state output leakage (data bus) test circuit.



NOTES:
1. THIS TIMING DIAGRAM IS USED TO SHOW SIGNAL RELATIONSHIPS ONLY AND DOES NOT REPRESENT ANY SPECIFIC MACHINE CYCLE
2. ALL MEASUREMENTS ARE REFERENCED TO 50% POINT OF THE WAVEFORMS
3. SHADED AREAS INDICATE "DON'T CARE" OR UNDEFINED STATE; MULTIPLE TRANSITIONS MAY OCCUR DURING THIS PERIOD

92CL-29599

Fig. 13 — Timing waveforms.

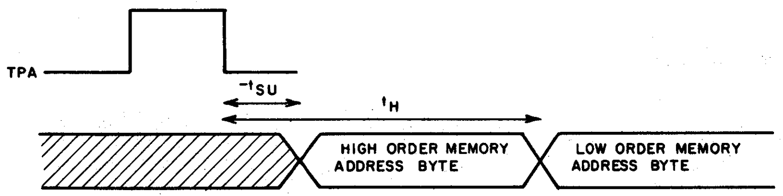
DYNAMIC ELECTRICAL CHARACTERISTICS at T_A = 25°C, C_L = 50 pF

CHARACTERISTIC		V_{CC} (V)	V_{DD} (V)	LIMITS			UNITS
				Min.	Typ.	Max.	
Propagation Delay Time, t_{PLH}, t_{PHL}: Clock to TPA, TPB		5	5	–	300	450	ns
		5	10	–	250	400	
		10	10	–	150	250	
Clock-to-Memory High-Address Byte		5	5	–	800	1200	ns
		5	10	–	600	900	
		10	10	–	400	600	
Clock-to-Memory Low-Address Byte		5	5	–	300	550	ns
		5	10	–	250	500	
		10	10	–	150	350	
Clock to $\overline{\text{MRD}}$, t_{PLH}		5	5	–	300	450	ns
		5	10	–	250	400	
		10	10	–	150	300	
Clock to $\overline{\text{MRD}}$, t_{PHL}		5	5	–	300	450	ns
		5	10	–	250	400	
		10	10	–	150	300	
Clock to $\overline{\text{MWR}}$, t_{PLH}, t_{PHL}		5	5	–	300	450	ns
		5	10	–	200	300	
		10	10	–	150	250	
Clock to CPU DATA to BUS		5	5	–	350	600	ns
		5	10	–	300	500	
		10	10	–	200	400	
Clock to State Code		5	5	–	400	600	ns
		5	10	–	200	400	
		10	10	–	150	300	
Clock to Q		5	5	–	300	700	ns
		5	10	–	150	400	
		10	10	–	100	300	
Clock to N(0-2), t_{PLH}		5	5	–	450	800	ns
		5	10	–	300	600	
		10	10	–	200	400	
High-Order Memory-Address Byte Set Up, t_{SU} (See Note)	f = 4 MHz	5	10	0	–	–	ns
	f = 6.4 MHz	10	10	–50	–	–	
	f = 2 MHz	5	5	50	–	–	
	f = 5 MHz	10	10	30	–	–	
High-Order Memory-Address Byte Hold t_H	f = 4 MHz	5	10	120	–	–	ns
	f = 6.4 MHz	10	10	75	–	–	
	f = 2 MHz	5	5	200	–	–	
	f = 5 MHz	10	10	100	–	–	
Low-Order Memory-Address Hold	f = 4 MHz	5	10	100	–	–	ns
	f = 6.4 MHz	10	10	50	–	–	

DYNAMIC ELECTRICAL CHARACTERISTICS (cont'd)

CHARACTERISTIC	V_{CC} (V)	V_{DD} (V)	LIMITS			UNITS
			Min.	Typ.	Max.	
Set-Up and Hold Times, t_{SU}, t_H Data Set Up	5	5	0	−50	–	ns
	5	10	25	0	–	
	10	10	50	0	–	
Data Hold	5	5	300	150	–	
	5	10	200	100	–	
	10	10	150	75	–	
$\overline{\text{DMA}}$ Set Up	5	5	100	0	–	ns
	5	10	125	25	–	
	10	10	150	50	–	
$\overline{\text{DMA}}$ Hold	5	5	250	150	–	
	5	10	200	100	–	
	10	10	150	75	–	
Interrupt Set Up	5	5	100	0	–	ns
	5	10	125	25	–	
	10	10	150	50	–	
Interrupt Hold	5	5	250	150	–	
	5	10	200	100	–	
	10	10	150	75	–	
$\overline{\text{WAIT}}$ Set Up	5	5	100	0	–	ns
	5	10	125	25	–	
	10	10	150	50	–	
$\overline{\text{EF1-4}}$ Set Up	5	5	100	0	–	ns
	5	10	125	25	–	
	10	10	150	50	–	
$\overline{\text{EF1-4}}$ Hold	5	5	250	150	–	
	5	10	200	100	–	
	10	10	150	75	–	
Pulse Width, t_{WL} $\overline{\text{CLEAR}}$ Pulse Width	5	5	600	300	–	ns
	5	10	400	200	–	
	10	10	300	150	–	
$\overline{\text{CLOCK}}$ Pulse Width, t_{WL}	5	5	160	–	–	ns
	5	10	125	–	–	
	10	10	80	–	–	
Typical Total Power Dissipation f = 2 MHz	5	5	–	4	–	mW
Idle "00" at M(0000), C_L = 50 pF f = 4 MHz	10	10	–	60	–	
Effective Input Capacitance, C_{IN} Any Input			–	5	–	pF
Effective 3-State Terminal Capacitance DATA BUS			–	7.5	–	pF

NOTE: Negative set-up indicates the addresses can change after the falling edge of TPA, as shown below:



92CS-29644

ARCHITECTURE

The COSMAC block diagram is shown in Fig. 14. The principal feature of this system is a register array (R) consisting of sixteen 16-bit scratchpad registers. Individual registers in the array (R) are designated (selected) by a 4-bit binary code from one of the 4-bit registers labeled N, P, and X. The contents of any register can be directed to any one of the following three paths:

1. the external memory (multiplexed, higher-order byte first, on to 8 memory address lines);
2. the D register (either of the two bytes can be gated to D);
3. the increment/decrement circuit where it is increased or decreased by one and stored back in the selected 16-bit register.

The three paths, depending on the nature of the instruction, may operate independently or in various combinations in the same machine cycle.

With two exceptions, COSMAC instructions consist of two 8-clock-pulse machine cycles. The first cycle is the fetch cycle, and the second—and third, if necessary—are execute cycles. During the fetch cycle the four bits in the P designator select one of the 16 registers R(P) as the current program counter. The selected register R(P) contains the address of the memory location from which the instruction is to be fetched. When the instruction is read out from the memory, the higher-order 4 bits of the instruction byte are loaded into the I register and the lower-order 4 bits into the N register. The content of the program counter is automatically incremented by one so that R(P) is now "pointing" to the next byte in the memory.

The X designator selects one of the 16 registers R(X) to "point" to the memory for an operand (or data) in certain ALU or I/O operations.

The N designator can perform the following five functions depending on the type of instruction fetched:

1. designate one of the 16 registers in R to be acted upon during register operations;
2. indicate to the I/O devices a command code or device-selection code for peripherals;
3. indicate the specific operation to be executed during the ALU instructions, types of tests to be performed during the Branch instructions, or the specific operation required in a class of miscellaneous instructions (70-73 and 78-7B);
4. indicate the value to be loaded into P to designate a new register to be used as the program counter R(P);

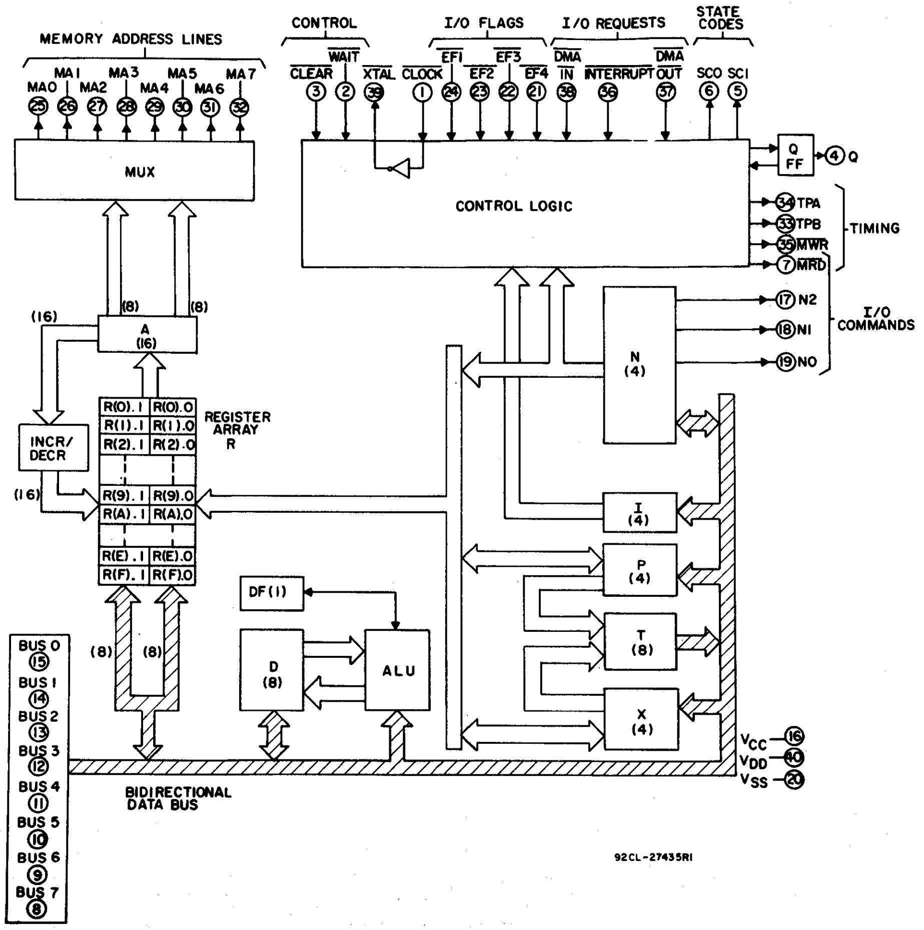


Fig. 14 – CDP1802 block diagram.

5. indicate the value to be loaded into X to designate a new register to be used as data pointer R(X).

The registers in R can be assigned by a programmer in three different ways: as program counters, as data pointers, or as scratchpad locations (data registers) to hold two bytes of data.

Program Counters

Any register can be the main program counter; the address of the selected register is held in the P designator. Other registers in R can be used as subroutine program counters. By a single instruction the contents of the P register can be changed to effect a "call" to a subroutine. When interrupts are being serviced, register R(1) is used as the program counter for the user's interrupt servicing routine. After reset, and during a DMA operation, R (O) is used as the program counter. At all other times the register designated as program counter is at the discretion of the user.

Data Pointers

The registers in R may be used as data pointers to indicate a location in memory. The register designated by X (i.e., R(X)) points to memory for the following instructions (see Table I):

1. ALU operations F1-F5, F7, 74, 75, 77;
2. output instructions 61 through 67;
3. input instructions 69 through 6F;
4. certain miscellaneous instructions—70-73, 78.60, FO.

The register designated by N (i.e., R(N)) points to memory for the "load D from memory" instructions 0N and 4N and the "Store D" instruction 5N. The register designated by P (i.e., the program counter) is used as the data pointer for ALU instructions F8-FD, FF, 7C, 7D, 7F. During these instruction executions, the operation is referred to as "data immediate".

Another important use of R as a data pointer supports the built-in Direct-Memory-Access (DMA) function. When a DMA-In or DMA-Out request is received, one machine cycle is "stolen". This operation occurs at the end of the execute machine cycle in the current instruction. Register R(0) is always used as the data pointer during the DMA operation. The data is read from (DMA-Out) or written into (DMA-In) the memory location pointed to by the R(0) register. At the end of the transfer, R(0) is incremented by one so that the processor is ready to act upon the next DMA byte transfer request. This feature in the COSMAC architecture saves a substantial amount of logic when fast exchanges of blocks of data are required, such as with magnetic discs or during CRT-display-refresh cycles.

A program load facility, using the DMA-In channel, is provided to enable users to load programs into the memory. This facility provides a simple, one-step means for initially entering programs into the microprocessor system and eliminates the requirement for specialized "bootstrap" ROM's.

Data Registers

When registers in R are used to store bytes of data, four instructions are provided which allow D to receive from or write into either the higher-order- or lower-order-byte portions of the register designated by N. By this mechanism (together with loading by data immediate) program pointer and data pointer designations are initialized. Also, this technique allows scratchpad registers in R to be used to hold general data. By employing increment or decrement instructions, such registers may be used as loop counters.

The Q Flip Flop

An internal flip flop, Q, can be set or reset by instruction and can be sensed by conditional branch instructions. The output of Q is also available as a microprocessor output.

Interrupt Servicing

Register R(1) is always used as the program counter whenever interrupt servicing is initiated. When an interrupt request comes in and the interrupt is allowed by the program (again, nothing takes place until the completion of the current instruction) the contents of the X and P registers are stored in the temporary register T, and X and P are set to new values; hex digit 2 in X and hex digit 1 in P. Interrupt enable is automatically deactivated to inhibit further interruptions. The user's interrupt routine is now in control; the contents of T may be saved by means of a single instruction (78) in the memory location pointed to by R (X). At the conclusion of the interrupt, the user's routine may restore the pre-interrupted value of X and P with a single instruction (70 or 71). The interrupt-enable flip-flop can be activated to permit further interrupts or can be disabled to prevent them.

COSMAC Register Summary

D	8 Bits	Data Register (Accumulator)
DF	1 Bit	Data Flag (ALU Carry)
R	16 Bits	1 of 16 Scratchpad Registers
P	4 Bits	Designates which register is Program Counter
X	4 Bits	Designates which register is Data Pointer

N	4 Bits	Holds Low-Order Instr. Digit
I	4 Bits	Holds High-Order Instr. Digit
T	8 Bits	Holds old X, P after Interrupt (X is high byte)
IE	1 Bit	Interrupt Enable
Q	1 Bit	Output Flip Flop

INSTRUCTION SET

The COSMAC instruction summary is given in Table I. Hexadecimal notation is used to refer to the 4-bit binary codes.

In all registers bits are numbered from the least significant bit (LSB) to the most significant bit (MSB) starting with 0.

R(W): Register designated by W, where W=N or X, or P

R(W).0: Lower-order byte of R(W)
R(W).1: Higher-order byte of R(W)
N0 = Least significant Bit of N Register

Operation Notation
M(R(N))→D; R(N) + 1

This notation means: The memory byte pointed to by R(N) is loaded into D, and R(N) is incremented by 1.

TABLE I – INSTRUCTION SUMMARY
(For Notes, see page 109)

INSTRUCTION	MNEMONIC	OP CODE	OPERATION
MEMORY REFERENCE			
LOAD VIA N	LDN	0N	M(R(N))→D; FOR N NOT 0
LOAD ADVANCE	LDA	4N	M(R(N))→D; R(N) +1
LOAD VIA X	LDX	F0	M(R(X))→D
LOAD VIA X AND ADVANCE	LDXA	72	M(R(X))→D; R(X) +1
LOAD IMMEDIATE	LDI	F8	M(R(P))→D; R(P) +1
STORE VIA N	STR	5N	D→M(R(N))
STORE VIA X AND DECREMENT	STXD	73	D→M(R(X)); R(X) –1
REGISTER OPERATIONS			
INCREMENT REG N	INC	1N	R(N) +1
DECREMENT REG N	DEC	2N	R(N) –1
INCREMENT REG X	IRX	60	R(X) +1
GET LOW REG N	GLO	8N	R(N).0→D
PUT LOW REG N	PLO	AN	D→R(N).0
GET HIGH REG N	GHI	9N	R(N).1→D
PUT HIGH REG N	PHI	BN	D→R(N).1
LOGIC OPERATIONS♦♦			
OR	OR	F1	M(R(X)) OR D→D
OR IMMEDIATE	ORI	F9	M(R(P)) OR D→D; R(P) +1
EXCLUSIVE OR	XOR	F3	M(R(X)) XOR D→D
EXCLUSIVE OR IMMEDIATE	XRI	FB	M(R(P)) XOR D→D; R(P) +1
AND	AND	F2	M(R(X)) AND D→D
AND IMMEDIATE	ANI	FA	M(R(P)) AND D→D; R(P) +1
SHIFT RIGHT	SHR	F6	SHIFT D RIGHT, LSB(D)→DF, 0→MSB(D)
SHIFT RIGHT WITH CARRY	SHRC	76♦	SHIFT D RIGHT, LSB(D)→DF, DF→MSB(D)
RING SHIFT RIGHT	RSHR		
SHIFT LEFT	SHL	FE	SHIFT D LEFT, MSB(D)→DF, 0→LSB(D)
SHIFT LEFT WITH CARRY	SHLC	7E♦	SHIFT D LEFT, MSB(D)→DF, DF→LSB(D)
RING SHIFT LEFT	RSHL		

♦NOTE: THIS INSTRUCTION IS ASSOCIATED WITH MORE THAN ONE MNEMONIC. EACH MNEMONIC IS INDIVIDUALLY LISTED.

♦♦NOTE: THE ARITHMETIC OPERATIONS AND THE SHIFT INSTRUCTIONS ARE THE ONLY INSTRUCTIONS THAT CAN ALTER THE DF.
AFTER AN ADD INSTRUCTION:
DF = 1 DENOTES A CARRY HAS OCCURRED
DF = 0 DENOTES A CARRY HAS NOT OCCURRED
AFTER A SUBTRACT INSTRUCTION:
DF = 1 DENOTES NO BORROW. D IS A TRUE POSITIVE NUMBER
DF = 0 DENOTES A BORROW. D IS TWO'S COMPLEMENT
THE SYNTAX "–(NOT DF)" DENOTES THE SUBTRACTION OF THE BORROW

TABLE I — INSTRUCTION SUMMARY (CONT'D)

INSTRUCTION	MNEMONIC	OP CODE	OPERATION
ARITHMETIC OPERATIONS♦♦			
ADD	ADD	F4	M(R(X)) +D→DF, D
ADD IMMEDIATE	ADI	FC	M(R(P)) +D→DF, D; R(P) +1
ADD WITH CARRY	ADC	74	M(R(X)) +D +DF→DF, D
ADD WITH CARRY, IMMEDIATE	ADCI	7C	M(R(P)) +D +DF→DF, D R(P) +1
SUBTRACT D	SD	F5	M(R(X))–D→DF, D
SUBTRACT D IMMEDIATE	SDI	FD	M(R(P))–D→DF, D; R(P) +1
SUBTRACT D WITH BORROW	SDB	75	M(R(X))–D–(NOT DF)→DF, D
SUBTRACT D WITH BORROW, IMMEDIATE	SDBI	7D	M(R(P))–D–(NOT DF)→DF, D; R(P) +1
SUBTRACT MEMORY	SM	F7	D–M(R(X))→DF, D
SUBTRACT MEMORY IMMEDIATE	SMI	FF	D–M(R(P))→DF, D; R(P) +1
SUBTRACT MEMORY WITH BORROW	SMB	77	D–M(R(X))–(NOT DF)→DF, D
SUBTRACT MEMORY WITH BORROW, IMMEDIATE	SMBI	7F	D–M(R(P))–(NOT DF)→DF, D R(P) +1
BRANCH INSTRUCTIONS—SHORT BRANCH			
SHORT BRANCH	BR	30	M(R(P))→R(P).0
NO SHORT BRANCH (SEE SKP)	NBR	38♦	R(P) +1
SHORT BRANCH IF D=0	BZ	32	IF D=0, M(R(P))→R(P).0 ELSE R(P) +1
SHORT BRANCH IF D NOT 0	BNZ	3A	IF D NOT 0, M(R(P))→R(P).0 ELSE R(P) +1
SHORT BRANCH IF DF=1	BDF	33♦	IF DF=1, M(R(P))→R(P).0 ELSE R(P) +1
SHORT BRANCH IF POS OR ZERO	BPZ		
SHORT BRANCH IF EQUAL OR GREATER	BGE		
SHORT BRANCH IF DF=0	BNF	3B♦	IF DF=0, M(R(P))→R(P).0 ELSE R(P) +1
SHORT BRANCH IF MINUS	BM		
SHORT BRANCH IF LESS	BL		
SHORT BRANCH IF Q=1	BQ	31	IF Q=1, M(R(P))→R(P).0 ELSE R(P) +1
SHORT BRANCH IF Q=0	BNQ	39	IF Q=0, M(R(P))→R(P).0 ELSE R(P) +1
SHORT BRANCH IF EF1=1 (1 = V_{SS})	B1	34	IF EF1=1, M(R(P))→R(P).0 ELSE R(P) +1
SHORT BRANCH IF EF1=0 (0 = V_{CC})	BN1	3C	IF EF1=0, M(R(P))→R(P).0 ELSE R(P) +1
SHORT BRANCH IF EF2=1 (1 = V_{SS})	B2	35	IF EF2=1, M(R(P))→R(P).0 ELSE R(P) +1
SHORT BRANCH IF EF2=0 (0 = V_{CC})	BN2	3D	IF EF2=0, M(R(P))→R(P).0 ELSE R(P) +1
SHORT BRANCH IF EF3=1 (1 = V_{SS})	B3	36	IF EF3=1, M(R(P))→R(P).0 ELSE R(P) +1
SHORT BRANCH IF EF3=0 (0 = V_{CC})	BN3	3E	IF EF3=0, M(R(P))→R(P).0 ELSE R(P) +1
SHORT BRANCH IF EF4=1 (1 = V_{SS})	B4	37	IF EF4=1, M(R(P))→R(P).0 ELSE R(P) +1
SHORT BRANCH IF EF4=0 (0 = V_{CC})	BN4	3F	IF EF4=0, M(R(P))→R(P).0 ELSE R(P) +1

♦NOTE: THIS INSTRUCTION IS ASSOCIATED WITH MORE THAN ONE MNEMONIC. EACH MNEMONIC IS INDIVIDUALLY LISTED.

♦♦NOTE: THE ARITHMETIC OPERATIONS AND THE SHIFT INSTRUCTIONS ARE THE ONLY INSTRUCTIONS THAT CAN ALTER THE DF.
AFTER AN ADD INSTRUCTION:
DF = 1 DENOTES A CARRY HAS OCCURRED
DF = 0 DENOTES A CARRY HAS NOT OCCURRED
AFTER A SUBTRACT INSTRUCTION:
DF = 1 DENOTES NO BORROW. D IS A TRUE POSITIVE NUMBER
DF = 0 DENOTES A BORROW. D IS TWO'S COMPLEMENT
THE SYNTAX "–(NOT DF)" DENOTES THE SUBTRACTION OF THE BORROW

TABLE I – INSTRUCTION SUMMARY (CONT'D)

INSTRUCTION	MNEMONIC	OP CODE	OPERATION
BRANCH INSTRUCTIONS–LONG BRANCH			
LONG BRANCH	LBR	C0	M(R(P))→R(P).1 M(R(P) +1)→R(P).0
NO LONG BRANCH (SEE LSKP)	NLBR	C8♦	R(P) +2

LONG BRANCH IF D=0	LBZ	C2	IF D=0, M(R(P))→R(P).1
			M(R(P) +1)→R(P).0
			ELSE R(P) +2

LONG BRANCH IF D NOT 0	LBNZ	CA	IF D NOT 0, M(R(P))→R(P).1
			M(R(P) +1)→R(P).0
			ELSE R(P) +2

LONG BRANCH IF DF=1	LBDF	C3	IF DF=1, M(R(P))→R(P).1
			M(R(P) +1)→R(P).0
			ELSE R(P) +2

LONG BRANCH IF DF=0	LBNF	CB	IF DF=0, M(R(P))→R(P).1
			M(R(P) +1)→R(P).0
			ELSE R(P) +2

LONG BRANCH IF Q=1	LBQ	C1	IF Q=1, M(R(P))→R(P).1
			M(R(P) +1)→R(P).0
			ELSE R(P) +2

LONG BRANCH IF Q=0	LBNQ	C9	IF Q=0, M(R(P))→R(P).1
			M(R(P) +1)→R(P).0
			ELSE R(P) +2

SKIP INSTRUCTIONS			
SHORT SKIP (SEE NBR)	SKP	38♦	R(P) +1
LONG SKIP (SEE NLBR)	LSKP	C8♦	R(P) +2

LONG SKIP IF D=0	LSZ	CE	IF D=0, R(P) +2
			ELSE CONTINUE

LONG SKIP IF D NOT 0	LSNZ	C6	IF D NOT 0, R(P) +2
			ELSE CONTINUE

LONG SKIP IF DF=1	LSDF	CF	IF DF=1, R(P) +2
			ELSE CONTINUE

LONG SKIP IF DF=0	LSNF	C7	IF DF=0, R(P) +2
			ELSE CONTINUE

LONG SKIP IF Q=1	LSQ	CD	IF Q=1, R(P) +2
			ELSE CONTINUE

LONG SKIP IF Q=0	LSNQ	C5	IF Q=0, R(P) +2
			ELSE CONTINUE

LONG SKIP IF IE=1	LSIE	CC	IF IE=1, R(P) +2
			ELSE CONTINUE

CONTROL INSTRUCTIONS			
IDLE	IDL	00#	WAIT FOR DMA OR INTERRUPT; M(R(0))→BUS
NO OPERATION	NOP	C4	CONTINUE
SET P	SEP	DN	N→P
SET X	SEX	EN	N→X
SET Q	SEQ	7B	1→Q
RESET Q	REQ	7A	0→Q
SAVE	SAV	78	T→M(R(X))
PUSH X,P TO STACK	MARK	79	(X,P)→T; (X,P)→M(R(2)) THEN P→X; R(2)–1
RETURN	RET	70	M(R(X))→(X,P); R(X) +1 1→IE
DISABLE	DIS	71	M(R(X))→(X,P); R(X) +1 0→IE

#An idle instruction initiates a repeating S1 cycle. The processor will continue to idle until an I/O request ($\overline{\text{INTERRUPT}}$, $\overline{\text{DMA-IN}}$, or $\overline{\text{DMA-OUT}}$) is activated. When the request is acknowledged, the IDLE cycle is terminated and the I/O request is serviced, and then normal operation is resumed.

♦NOTE: THIS INSTRUCTION IS ASSOCIATED WITH MORE THAN ONE MNEMONIC. EACH MNEMONIC IS INDIVIDUALLY LISTED.

TABLE I – INSTRUCTION SUMMARY (CONT'D)

INSTRUCTION	MNEMONIC	OP CODE	OPERATION
INPUT–OUTPUT BYTE TRANSFER			
OUTPUT 1	OUT 1	61	M(R(X))→BUS; R(X) +1; N LINES = 1
OUTPUT 2	OUT 2	62	M(R(X))→BUS; R(X) +1; N LINES = 2
OUTPUT 3	OUT 3	63	M(R(X))→BUS; R(X) +1; N LINES = 3
OUTPUT 4	OUT 4	64	M(R(X))→BUS; R(X) +1; N LINES = 4
OUTPUT 5	OUT 5	65	M(R(X))→BUS; R(X) +1; N LINES = 5
OUTPUT 6	OUT 6	66	M(R(X))→BUS; R(X) +1; N LINES = 6
OUTPUT 7	OUT 7	67	M(R(X))→BUS; R(X) +1; N LINES = 7
INPUT 1	INP 1	69	BUS→M(R(X)); BUS→D; N LINES = 1
INPUT 2	INP 2	6A	BUS→M(R(X)); BUS→D; N LINES = 2
INPUT 3	INP 3	6B	BUS→M(R(X)); BUS→D; N LINES = 3
INPUT 4	INP 4	6C	BUS→M(R(X)); BUS→D; N LINES = 4
INPUT 5	INP 5	6D	BUS→M(R(X)); BUS→D; N LINES = 5
INPUT 6	INP 6	6E	BUS→M(R(X)); BUS→D; N LINES = 6
INPUT 7	INP 7	6F	BUS→M(R(X)); BUS→D; N LINES = 7

1. Long-Branch, Long-Skip and No Op instructions are the only instructions that require three cycles to complete (1 fetch + 2 execute).

   Long-Branch instructions are three bytes long. The first byte specifies the condition to be tested; and the second and third byte, the branching address.

   The long-branch instructions can:
   a) Branch unconditionally
   b) Test for D=0 or D≠0
   c) Test for DF=0 or DF=1
   d) Test for Q=0 or Q=1
   e) effect an unconditional no branch

   If the tested condition is met, then branching takes place; the branching address bytes are loaded in the high-and-low-order bytes of the current program counter, respectively. This operation effects a branch to any memory location.

   If the tested condition is not met, the branching address bytes are skipped over, and the next instruction in sequence is fetched and executed. This operation is taken for the case of unconditional no branch (NLBR).

2. The short-branch instructions are two bytes long. The first byte specifies the condition to be tested, and the second specifies the branching address.

   The short-branch instructions can:
   a) Branch unconditionally
   b) Test for D=0 or D≠0
   c) Test for DF=0 or DF=1
   d) Test for Q=0 or Q=1
   e) Test the status (1 or 0) of the four EF flags
   f) Effect an unconditional no branch

   If the tested condition is met, then branching takes place; the branching address byte is loaded into the low-order byte position of the current program counter. This effects a branch with the current 256-byte page of the memory, i.e., the page which holds the branching address. If the tested condition is not met, the branching address byte is skipped over, and the next instruction in sequence is fetched and executed. This same action is taken in the case of unconditional no branch (NBR)

3. The skip instructions are one byte long. There is one Unconditional Short-Skip (SKP) and eight Long-Skip instructions.

   The Unconditional Short-Skip instruction takes 2 cycles to complete (1 fetch + 1 execute). Its action is to skip over the byte following it. Then the next instruction in sequence is fetched and executed. This SKP instruction is identical to the unconditional no-branch instruction (NBR) except that the skipped-over byte is not considered part of the program.

   The Long-Skip instructions take three cycles to complete (1 fetch + 2 execute).

   They can:
   a) Skip unconditionally
   b) Test for D=0 or D≠0
   c) Test for DF=0 or DF=1
   d) Test for Q=0 or Q=1
   e) Test for IE=1

   If the tested condition is met, then Long Skip takes place; the current program counter is incremented twice. Thus two bytes are skipped over and the next instruction in sequence is fetched and executed. If the tested condition is not met, then no action is taken. Execution is continued by fetching the next instruction in sequence.

SIGNAL DESCRIPTIONS

BUS 0 to BUS 7 (Data Bus)

8-bit directional DATA BUS lines. These lines are used for transferring data between the memory, the microprocessor, and I/O devices.

N0 to N2 (I/O Lines)

Activated by an I/O instruction to signal the I/O control logic of a data transfer between memory and I/O interface. These lines can be used to issue command codes or device selection codes to the I/O devices (independently or combined with the memory byte on the data bus when an I/O instruction is being executed). The N bits are low at all times except when an I/O instruction is being executed. During this time their state is the same as the corresponding bits in the N register.

The direction of data flow is defined in the I/O instruction by bit N3 (internally) and is indicated by the level of the $\overline{\text{MRD}}$ signal.

$\overline{\text{MRD}} = V_{CC}$: Data from I/O to CPU and Memory

$\overline{\text{MRD}} = V_{SS}$: Data from Memory to I/O

$\overline{\text{EF1}}$ to $\overline{\text{EF4}}$ (4 Flags)

These inputs enable the I/O controllers to transfer status information to the processor. The levels can be tested by the conditional branch instructions. They can be used in conjunction with the INTERRUPT request line to establish interrupt priorities. These flags can also be used by I/O devices to "call the attention" of the processor, in which case the program must routinely test the status of these flag(s). The flag(s) are sampled at the beginning of every S1 cycle.

$\overline{\text{INTERRUPT}}$, $\overline{\text{DMA-IN}}$, $\overline{\text{DMA-OUT}}$ (3 I/O Requests)

These inputs are sampled by the CDP1802 during the interval between the leading edge of TPB and the leading edge of TPA.

Interrupt Action: X and P are stored in T after executing current instruction; designator X is set to 2; designator P is set to 1; interrupt enable is reset to 0 (inhibit); and instruction execution is resumed. The interrupt action requires one machine cycle (S3).

DMA Action: Finish executing current instruction; R(0) points to memory area for data transfer; data is loaded into or read out of memory; and increment R(0).

Note: In the event of concurrent DMA and INTERRUPT requests, DMA-IN has priority followed by DMA-OUT and then INTERRUPT.

SC0, SC1, (2 State Code Lines)

These outputs indicate that the CPU is: 1 fetching an instruction, or 2) executing an instruction, or 3) processing a DMA request, or 4) acknowledging an interrupt request. The levels of state code are tabulated below. All states are valid at TPA. H = V_{CC}, L = V_{SS}.

State Type	State Code Lines	
	SC1	SC0
S0 (Fetch)	L	L
S1 (Execute)	L	H
S2 (DMA)	H	L
S3 (Interrupt)	H	H

TPA, TPB (2 Timing Pulses)

Positive pulses that occur once in each machine cycle (TPB follows TPA). They are used by I/O controllers to interpret codes and to time interaction with the data bus. The trailing edge of TPA is used by the memory system to latch the higher-order byte of the 16-bit memory address. TPA is suppressed in IDLE when the CPU is in the load mode.

MA0 to MA7 (8 Memory Address Lines)

The higher-order byte of a 16-bit COSMAC memory address appears on the memory address lines MA0-7 first. Those bits required by the memory system can be strobed into external address latches by timing pulse TPA. The low-order byte of the 16-bit address appears on the address lines after the termination of TPA. Latching of all 8 higher-order address bits would permit a memory system of 64K bytes.

$\overline{\text{MWR}}$ (Write Pulse)

A negative pulse appearing in a memory-write cycle, after the address lines have stabilized.

$\overline{\text{MRD}}$ (Read Level)

A low level on $\overline{\text{MRD}}$ indicates a memory read cycle. It can be used to control three-state outputs from the addressed memory which may have a common data input and output bus. If a memory **does** not have a three-state high-impedance output, $\overline{\text{MRD}}$ is useful for driving memory/bus separator gates. It is also used to indicate the direction of data transfer during an I/O instruction. For additional information see Table I.

Q

Single bit output from the CPU which can be set or reset under program control. During SEQ or REQ instruction execution, Q is set or reset between the trailing edge of TPA and the leading edge of TPB.

CLOCK

Input for externally generated single-phase clock. A typical clock frequency is 6.4 MHz at $V_{CC} = V_{DD} = 10$ volts. The clock is counted down internally to 8 clock pulses per machine cycle.

$\overline{\text{XTAL}}$

Connection to be used with clock input terminal, for an external crystal, if the on-chip oscillator is utilized. The crystal is connected between terminals 1 and 39 (CLOCK and $\overline{\text{XTAL}}$) in parallel with a resistance (10 megohms typ.). Frequency trimming capacitors may be required at terminals 1 and 39. For additional information see ICAN-6565.

$\overline{\text{WAIT}}$, $\overline{\text{CLEAR}}$
(2 Control Lines)

Provide four control modes as listed in the following truth table:

$\overline{\text{CLEAR}}$	$\overline{\text{WAIT}}$	MODE
L	L	Load
L	H	Reset
H	L	Pause
H	H	Run

The function of the modes are defined as follows:

Load

Holds the CPU in the IDLE execution state and allows an I/O device to load the memory without the need for a "bootstrap" loader. It modifies the IDLE condition so that DMA-IN operation does not force execution of the next instruction.

Reset

Registers I, N, Q are reset, IE is set and 0's (V_{SS}) are placed on the data bus. TPA and TPB are suppressed while reset is held and the CPU is placed in S1. The first machine cycle after termination of reset is an initialization cycle which requires 9 clock pulses. During this cycle the CPU remains in S1 and registers X, P, and R(0) are reset. Interrupt and DMA servicing are suppressed during the initialization cycle. The next cycle is an S0, S1, or an S2 but never an S3. With the use of a 71 instruction followed by 00 at memory locations 0000 and 0001, this feature may be used to reset IE, so as to preclude interrupts until ready for them. Power-up reset can be realized by connecting a buffered RC network to $\overline{\text{CLEAR}}$. For additional information see ICAN-6581.

Pause

Stops the internal CPU timing generator on the first negative high-to-low transition of the input clock. The oscillator continues to operate, but subsequent clock transitions are ignored.

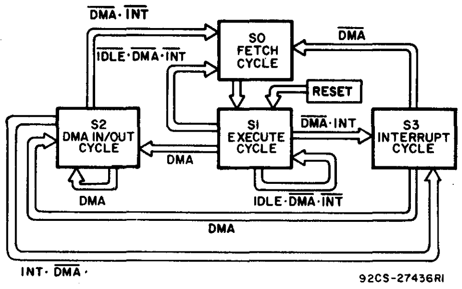
Run

May be initiated from the Pause or Reset mode functions. If initiated from Pause, the CPU resumes operation on the first negative high-to-low transition of the input clock. When initiated from the Reset operation, the first machine cycle following Reset is always the initialization cycle. The initialization cycle is then followed by a DMA (S2) cycle or fetch (S0) from location 0000 in memory.

V_{DD}, V_{SS}, V_{CC} (Power Levels)	The internal voltage supply V_{DD} is isolated from the Input/Output voltage supply V_{CC} so that the processor may operate at maximum speed while interfacing with various external circuit technologies, including T^2L at 5 volts. V_{CC} must be less than or equal to V_{DD}. All outputs swing from V_{SS} to V_{CC}. The recommended input voltage swing is V_{SS} to V_{CC}.

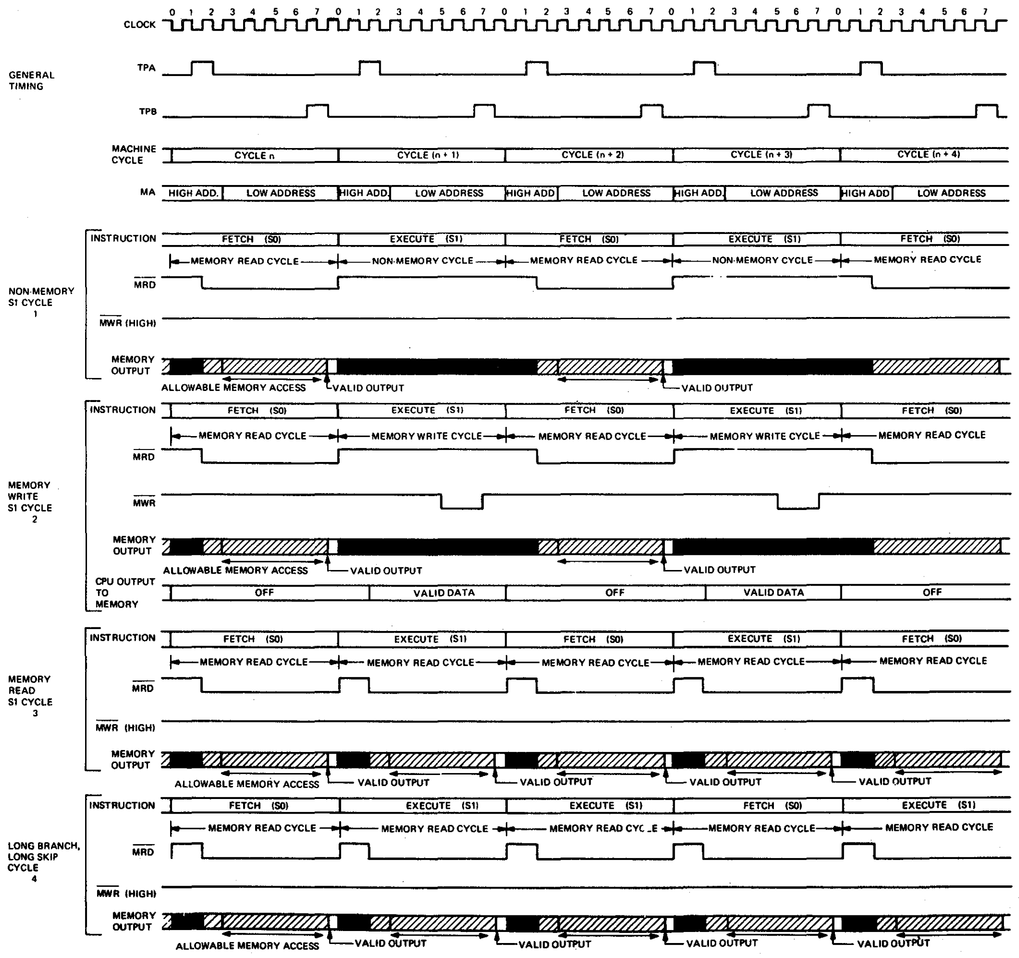
RUN-MODE STATE TRANSITIONS

The CDP1802 and CDP1802C CPU state transitions when in the RUN mode are shown in Fig. 15. Each machine cycle requires the same period of time 8 clock pulses except the initialization cycle, which requires 9 clock pulses. The execution of an instruction requires either two or three machine cycles, S0 followed by a single S1 cycle or two S1 cycles. S2 is the response to a DMA request and S3 is the interrupt response. Table II shows the conditions on Data Bus and Memory-Address lines during all machine states.



$\overline{DMA}\cdot\overline{INT}$
$\overline{IDLE}\cdot\overline{DMA}\cdot\overline{INT}$
S0 FETCH CYCLE
$\overline{DMA}$
RESET
S2 DMA IN/OUT CYCLE
S1 EXECUTE CYCLE
$\overline{DMA}\cdot INT$
S3 INTERRUPT CYCLE
DMA
DMA
$IDLE\cdot\overline{DMA}\cdot\overline{INT}$
DMA
$INT\cdot\overline{DMA}\cdot$
92CS-27436RI

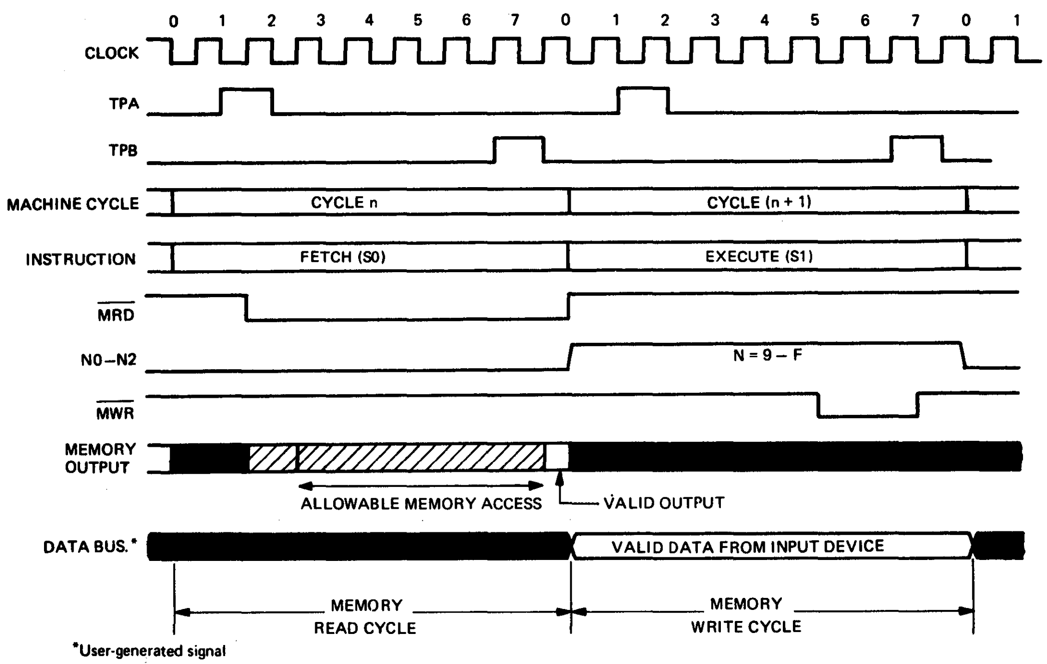
Fig. 15 – CDP1802 microprocessor state transitions (Run Mode).



▨ "Don't Care" or internal delays.

■ High-impedance state

Fig. 16 – Timing diagram for machine cycle types Nos. 1 through 4.



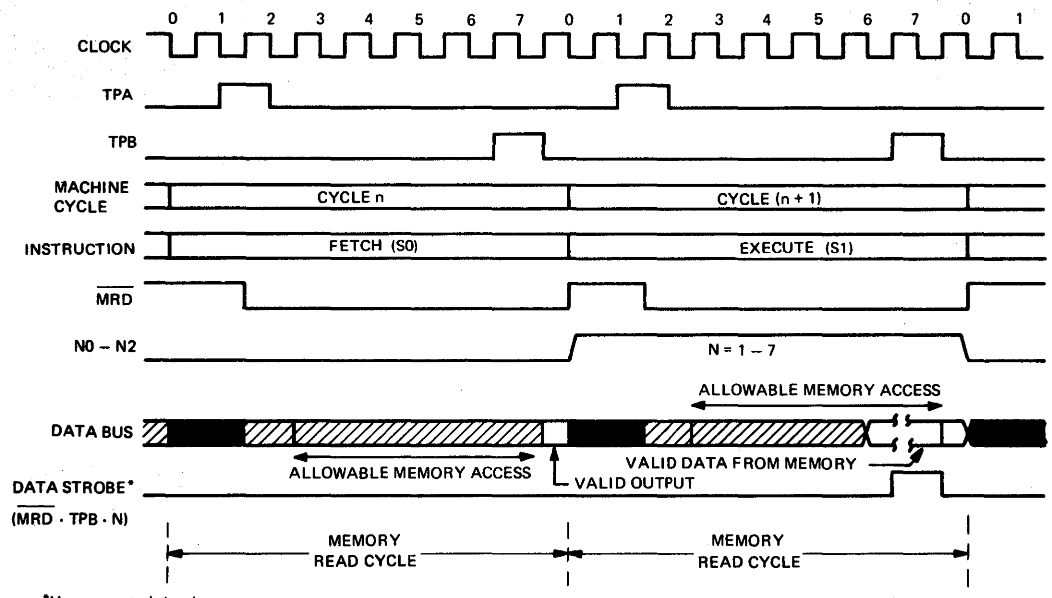
*User-generated signal

▨ "Don't Care" or internal delays.

■ High-impedance state

92CS-29601

Fig. 17 – Timing diagram for machine cycle type No. 5.



*User-generated signal

▨ "Don't Care" or internal delays.

■ High-impedance state

92CS-29602

Fig. 18 – Timing diagram for machine cycle type No. 6.

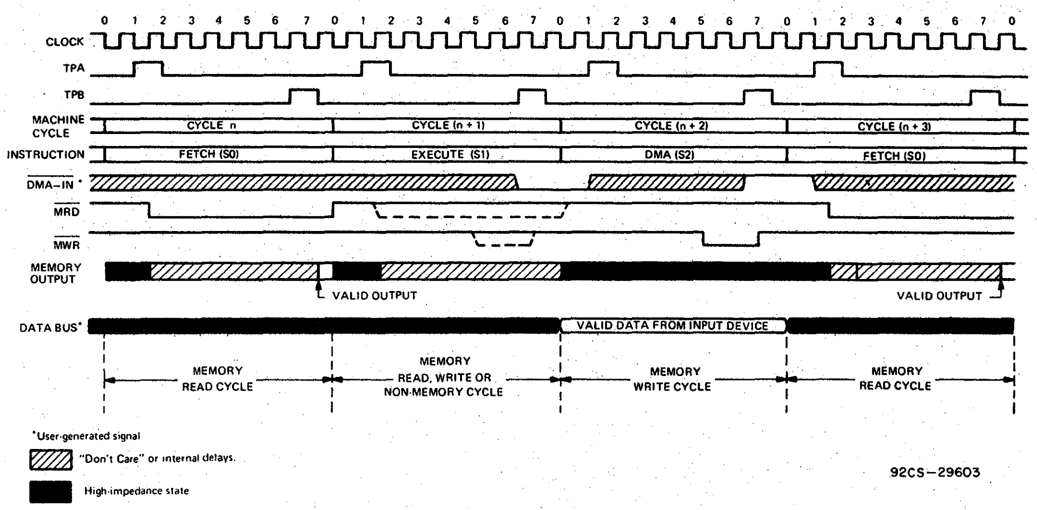


Fig. 19 — Timing diagram for machine cyle type No. 7.

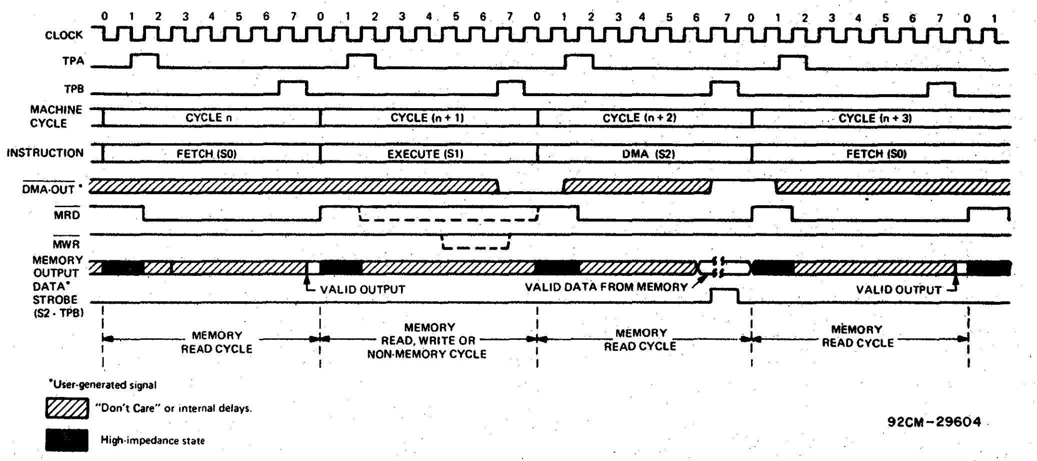


Fig. 20 — Timing diagram for machine cyle type No. 8.

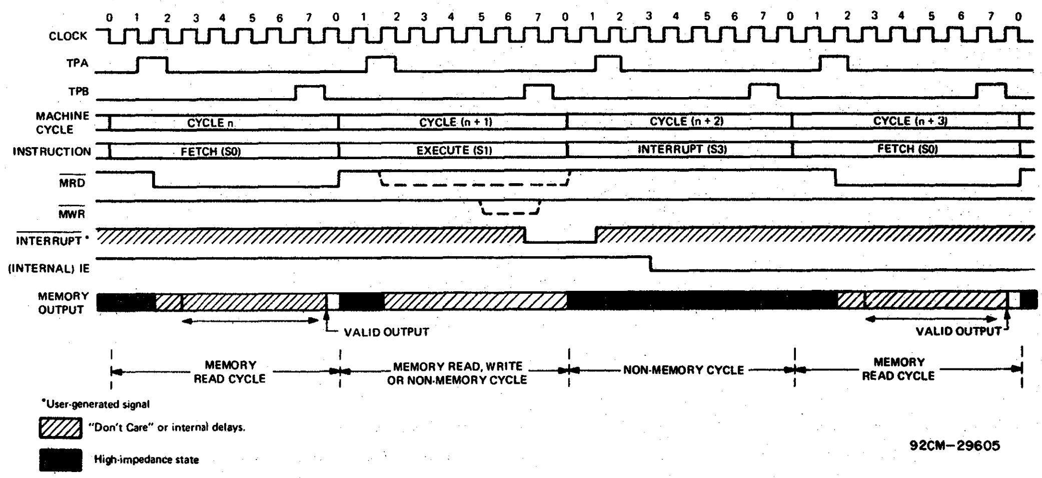


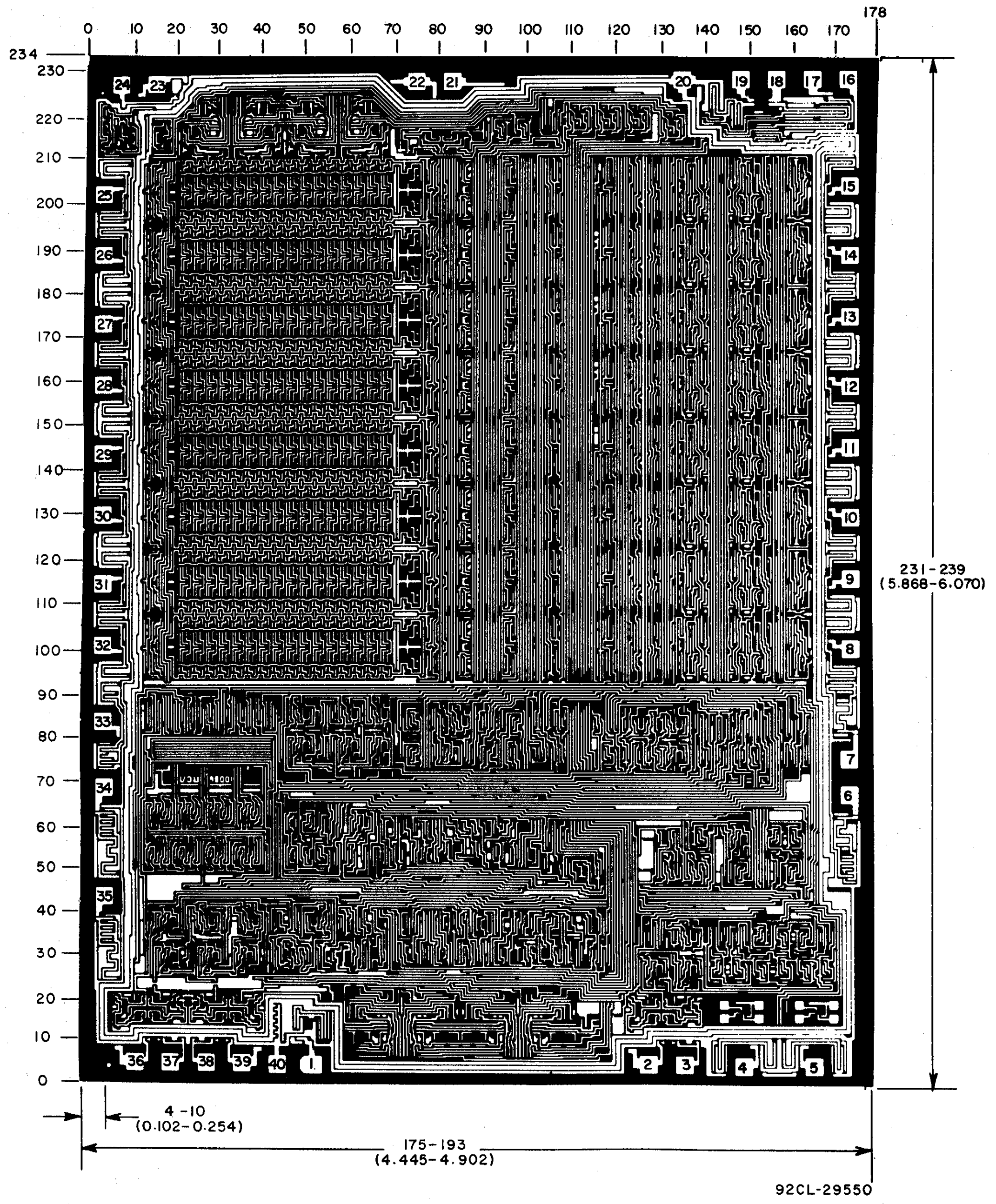
Fig. 21 — Timing diagram for machine cycle type No. 9.

TABLE II. CONDITIONS ON DATA BUS AND MEMORY ADDRESS LINES DURING ALL MACHINE STATES

STATE	I	N	MNEMONIC	INSTRUCTION	OPERATION	DATA BUS	MEMORY ADDRESS	$\overline{\text{MRD}}$	NOTES[G]
S1	RESET				JAM: I,N,Q,X,P = 0 IE = 1	0	R (0) UNDEFINED	1	A
	FIRST CYCLE AFTER RESET NOT PROGRAMMER ACCESSIBLE				INITIALIZE	0	R (0) UNDEFINED	1	B
S0	FETCH				M(R(P))→I.N R(P)+1	M(R(P))	R(P)	0	C
S1 (Execute)	0	0	IDL	IDLE	[Load = 0 (Program Idle)]	M (R(0))	R (0)	0	D,3
					[Load = 1 (Load Mode)]	M(R(0))	PREVIOUS ADDRESS	0	E,3
		N≠0	LDN	LOAD D VIA N	M(R(N))→D	M(R(N))	R(N)	0	3
	1	N	INC	INCREMENT	R(N)+1	FLOAT	R(N)	1	1
	2	N	DEC	DECREMENT	R(N)−1	FLOAT	R(N)	1	1
	3	N	–	SHORT BRANCH	[BRANCH NOT TAKEN]	M(R(P))	R(P)	0	3
					[BRANCH TAKEN]	M(R(P))	R(P)	0	
	4	N	LDA	LOAD ADVANCE	M(R(N))→D R(N)+1	M(R(N))	R(N)	0	3
	5	N	STR	STORE VIA N	D→M(R(N))	D	R(N)	1	3
	6	0	IRX	INC REG X	R(X)+1	M(R(X))	R(X)	0	3
		N=1–7	OUT N	OUTPUT	M(R(X))→BUS R(X)+1	M(R(X))	R(X)	0	6
		N=9–F	INP N	INPUT	BUS→M(R(X)), D	I/O DEVICE	R(X)	1	5
	7	0	RET	RETURN	M(R(X))→(X,P) R(X)+1; 1→IE	M(R(X))	R(X)	0	3
		1	DIS	DISABLE	M(R(X))→(X,P) R(X)+1; 0→IE	M(R(X))	R(X)	0	3
		2	LDXA	LOAD VIA X AND ADVANCE	M(R(X))→D P(X)−1	M(R(X))	R(X)	0	3
		3	STXD	STORE VIA X AND DECREMENT	D→M(R(X)) R(X)−1	D	R(X)	1	2
		4,5,7	–		ALU OPERATION	M(R(X))	R(X)	0	3
		6	–		ALU OPERATION	FLOAT	R(X)	1	1
		8	SAV	SAVE	T→M(R(X))	T	R(X)	1	2
		9	MARK	MARK	(X,P)→T, M(R(2)) P → X; R(2)−1	T	R(2)	1	2
		A	REQ	RESET Q	Q = 0	FLOAT	R(P)	1	1
		B	SEQ	SET Q	Q = 1	FLOAT	R(P)	1	1
		C,D,F			ALU OPERATION IMMEDIATE	M(R(P))	R(P)	0	3
		E			ALU OPERATION	FLOAT	R(X)	1	1
	8	N	GLO	GET LOW	R(N) .0→D	R(N) .0	R(N)	1	1
	9	N	GHI	GET HIGH	R(N) .1→D	R(N) .1	R(N)	1	1
	A	N	PLO	PUT LOW	D→R(N).0	D	R(N)	1	1
	B	N	PHI	PUT HIGH	D→R(N) .1	D	R(N)	1	1
	C	0,1,2 3,8,9 A,B		LONG BRANCH	[BRANCH NOT TAKEN]	M(R(P))	R(P)	0	4
					[BRANCH TAKEN]	M(R(P))	R(P)	0	4
		5,6,7 C,D,E F		LONG SKIP	[SKIP NOT TAKEN]	M(R(P))	R(P)	0	4
					[SKIP TAKEN]	M(R(P))	R(P)	0	4
		4	NOP	NO OPERATION	NO OPERATION	M(R(P))	R(P)	0	4
	D	N	SEP	SET P	N→P	N N	R(N)	1	1
	E	N	SEX	SET X	N→X	N N	R(N)	1	1
	F	0	LDX	LOAD VIA X	M(R(X))→D	M(R(X))	R(X)	0	3
		1,2,3 4,5,7			ALU OPERATION	M(R(X))	R(X)	0	3
		6	SHR	SHIFT RIGHT	SHIFT D RIGHT LSB(D)→DF 0 → MSB(D)	FLOAT	R(X)	1	1
		8	LDI	LOAD IMMEDIATE	M(R(P))→D R(P)+1	M(R(P))	R(P)	0	3
		9,A,B C,D,F			ALU OPERATION IMMEDIATE	M(R(P))	R(P)	0	3
		E	SHL	SHIFT LEFT	ALU OPERATION	FLOAT	R(P)	1	1
S2	IN REQUEST			DMA IN	BUS→M(R(0))	I/O DEVICE	R (0)	1	F,7
	OUT REQUEST			DMA OUT	M(R(0))→BUS	M(R(0))	R (0)	0	F,8
S3	INTERRUPT				X,P→T, 0→IE 2→X, 1→P	FLOAT	R(N)	1	9

NOTES:

A. IE = 1; TPA, TPB suppressed, state = S1
B. BUS = 0 for entire cycle
C. Next state always S1
D. Wait for DMA or INTERRUPT
E. Suppress TPA, wait for DMA
F. IN REQUEST has priority over OUT REQUEST
G. Numbers refer to machine cycles types — refer to timing diagrams, Figs. 16 through 20.



Dimensions in parentheses are in millimeters and are derived from the basic inch dimensions as indicated. Grid graduations are in mils (10^{-3} inch).

The photographs and dimensions of each COS/MOS chip represent a chip when it is part of the wafer. When the wafer is cut into chips, the cleavage angles are 57° instead of 90° with respect to the face of the chip. Therefore, the isolated chip is actually 7 mils (0.17 mm) larger in both dimensions.

Dimensions and pad layout for CDP1802

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

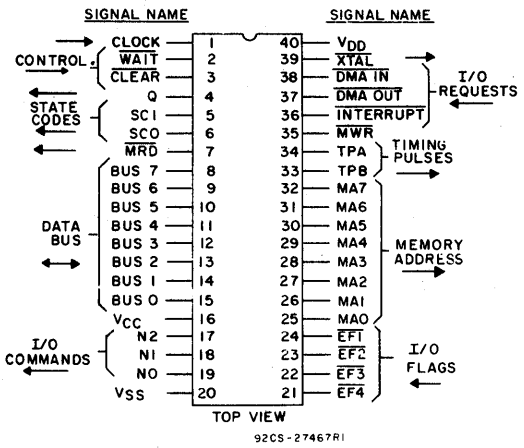
   To prevent damage to the input protection circuit, input signals should never be greater than V_{CC} nor less than V_{SS}. Input currents must not exceed 10 mA even when the power supply is off.

   Unused Inputs

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

   Output Short Circuits

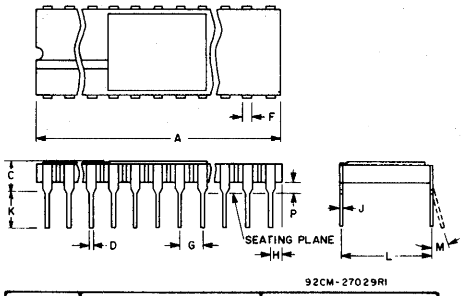
   Shorting of outputs to V_{DD}, V_{CC}, or V_{SS} may damage COS/MOS devices by exceeding the maximum device dissipation.



When incorporating RCA Solid State Devices in equipment, it is recommended that the designer refer to "Operating Considerations for RCA Solid State Devices", Form No. 1CE-402, available on request from RCA Solid State Division, Box 3200, Somerville, N. J. 08876.

DIMENSIONAL OUTLINE

CDP1802D, CDP1802CD
40-Lead Dual-In-Line Ceramic



DIM.	MILLIMETERS		INCHES	
	MIN.	MAX.	MIN.	MAX.
A	50.30	51.30	1.980	2.020
C	2.42	3.93	0.095	0.155
D	0.43	0.56	0.017	0.023
F	1.27 REF.		0.050 REF.	
G	2.54 BSC		0.100 BSC	
H	0.76	1.78	0.030	0.070
J	0.20	0.30	0.008	0.012
K	3.18	4.45	0.125	0.175
L	14.74	15.74	0.580	0.620
M	–	7°	–	7°
P	0.64	1.27	0.025	0.050
N	40		40	

NOTES:
1. Leads within 0.13 mm (0.005) radius of true position at maximum material condition.
2. Dimension "L" to center of leads when formed parallel.
3. When this device is supplied solder-dipped, the maximum lead thickness (narrow portion) will not exceed 0.013 in. (0.33 mm).