



Z-100 Interlaced Video

The following explanation contains excerpts from an article by Frank T. Clark of Zenith Data Systems, published in Issue #59, December 1984, REMark Magazine.

The H/Z-100 video screen is normally composed of 25 rows by 80 columns. Each character row is made up of nine lines, giving us 225 lines of information in the display. These are referred to as scan lines, with each scan line divided into 640 pieces (pixels) which can be individually lighted or dark. There are eight pixels per character, thus permitting 80 characters.

The scan lines are drawn on the screen one at a time at a very high rate of speed, controlled by a device called a CRT-C (CRT Controller). The entire screen is drawn 60 times per second. If you look very closely at a screen with writing on it, you can see a small gap between every scan line. What the CRT-C can do is to shift the screen of information every other time it is refreshed (redisplayed on the screen) so that the space between the scan lines is filled in. This procedure of placing every other screen with the scan lines woven together is called Interlacing.

There are two ways the CRT-C can perform this interlacing procedure.

One way is for the two screens of information to be exactly the same, except for the fact that they are drawn in slightly different places on the screen. This is referred to as Interlaced Video Only.

The second way is for the two screens to be different parts of the same picture. This is referred to as Interlaced Video and Memory. The memory is of course the area where the picture image comes from.

The Interlaced Video Only method does not require any extra memory to store the video image, because it is simply redisplaying the same video in a slightly different location. The Interlaced Video and Memory Mode requires double the amount of memory for the same picture.

The Interlaced Video Only mode has the advantage of filling in the spaces between the pixels which gives the image on the screen a more solid appearance, but is not usually considered to be worth the work.

The Interlaced Video and Memory mode, however, provides double the resolution of the image that is drawn on the screen.

The normal mode of operation usually has somewhere near double the number of pixels per inch horizontally than it does vertically. So doubling the vertical resolution brings the number of pixels per inch to somewhere very nearly equal, which provides for a smoother picture and easier graphics calculations.

There are some problems with interlaced operation. The main problem is that the picture on the screen is now only updated every other time, so the refresh rate drops to 30 frames per second instead of 60. This rate is slow enough that on most screens the phosphor fades out before it is refreshed again. For some people this fading out is perceived as screen flicker, which can be annoying. There were video monitors available at the time which have a long persistence phosphor which fades out so slowly that the flickering was not an issue.

Since the Interlaced Video and Memory mode is the most useful and popular mode of operation, for the purpose of this discussion, we shall discuss only about this mode and refer to it as Interlaced.

In the Z-100, there are several things that must be considered when in Interlaced mode:

- * Since Interlaced operation required double the memory space, the 32k video RAM chips that were sufficient for normal video operation, were not sufficient for Interlaced operation. These must be changed to 64K chips for successful operation.
- * The normal Z-100 video monitor may flicker. This was, however, a problem of personal taste. Careful adjustment of monitor brightness, contrast, and the room lighting could minimize the problem.
- * None of the standard Z-100 software was designed with or even intended for adaptability to use in Interlaced mode.

In spite of all these glum problems, if desired, Interlace mode could be implemented fairly easily. Let's begin with a simple little program to turn the Interlaced video only mode ON and OFF again. This program is written in Assembly language and can be entered with EDLIN. The source code is:

```

Name    INTERLAC
Title.  INTERLAC - Turn Interlace Video
        Only Mode ON/OFF

CODE    Segment
        ASSUME  CS:CODE, DS:CODE, ES:CODE,
            SS:CODE
        ORG      0100h
BEGIN:
        CLI
        MOV     AL,8
        OUT     0DCh,AL
        MOV     AL,DS:[082h]
        AND     AL,1
        OUT     0DDh, AL
        STI
        RET
CODE    ENDS
        END      BEGIN

```

Now assemble using the following commands:

```

EDLIN INTERLAC.ASM
MASM INTERLAC;
LINK INTERLAC;
EXE2BIN INTERLAC.EXE INTERLAC.COM

```

The program is run by entering INTERLAC 1 to turn the Interlace Video Only Mode ON and INTERLAC 0 to turn it OFF.

This simple demonstration of the Interlace Video Only Mode should allow you to evaluate the flicker problem for yourself. Please note that the Interlaced Video and Memory mode will not be quite as bad because only half the dots will flicker at any time, which masks the problem.

There are certain limitations when we use Interlacing on the H/Z-100, however:

- * Since the H/Z-100 is a pure graphics machine, characters are drawn on the screen just like any other graphic item, such as a Line or a Circle. Drawing characters is much more complex though because of the considerations of the 96 character set, positioning on the screen and other such things. You cannot use the normal methods to write characters when you are in Interlaced mode. Therefore, one way of simplifying the problem is to forgo the use of characters or draw them just like anything else.

- * Consideration of characters aside, an Interlaced graphics program must operate independently of all normal video features. The program must initialize the CRT-C for Interlaced operation, perform all video display itself, and then restore normal operation. In particular, program input must not be allowed to echo because this would require the use of normal video.

This leaves us with a very primitive program, and yet, the level of computer software required to do even this very simple task is still very high, must be done in assembly language for speed, and would probably be beyond the abilities of the average hobbyist to program.

Finally, most companies would not believe there was enough interest to make it worth the cost of developing such software.

This article's conclusion was that Interlaced Graphics was considered an extremely interesting idea to play with, but regrettably, it was deemed as not commercially viable.

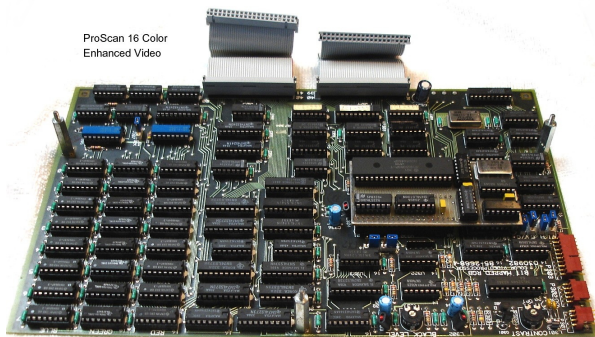
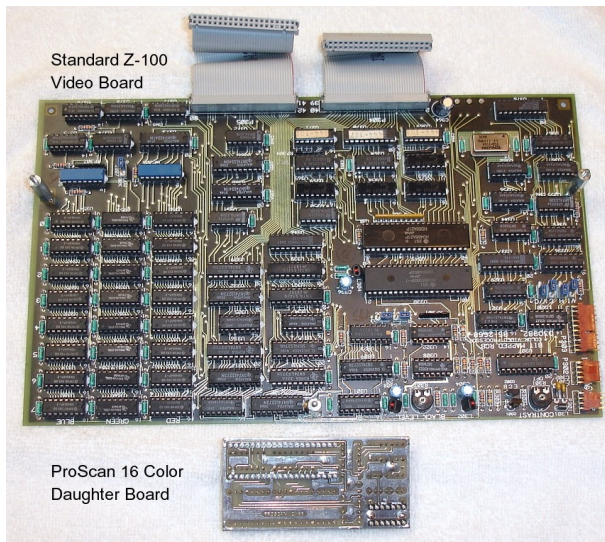
Nevertheless, a few companies did make a go of it in their software... ShowOff from Hogware, LACE87 and ED48 from Friendliware, Benoit from Village Software, the MicroWorld Data Bank II from Micro Doc, AutoCAD, WordStar (which could display 49 lines), Flexi-Graph from New Orleans General Data, WatchWord (which could display 43 lines) from S&K Technology, and Doodler-V from Paul Herman, to name a few.

A few companies also offered complete hardware options...



PROSCAN: Higher Resolution Z-100 Video

From William N. Locke's Z-100 Notebook
Article in Sextant Magazine #33, Early Spring, 1988.



ProScan Board Installed

ProScan was Travis Barfield's contribution to improved Z-100 graphics. This daughter board cost \$65 at the time; plugged into the Z-100's video board and required the use of NEC's MultiSync monitor.

ProScan gave you a resolution of 640 by 480 pixels - roughly twice the Z-100's usual 640x225 pixels, and the same number as operating the Z-100 in Interlace mode. But the annoying flicker associated with Interlace was gone because with ProScan the normal horizontal scan rate was nearly doubled.

Many products on the market back then for use with the Z-100's Interlace mode had been modified for use with ProScan. Those programs capable of using Interlace mode (listed above) quickly took advantage of ProScan. The Flexi-Graph graphics library from New Orleans General Data Services could also be used with ProScan and the MultiSync monitor.

ProScan also came with a program, PROSCAN.COM, that let standard word processors, such as WordStar, work with a display of more than 40 lines.

Installation:

Installation of ProScan was easy, if you were willing to disassemble the Z-100 to remove the video board. You had to remove the CRT-C controller chip at U330 and the SN74S86 chip at U344, install the CRT-C chip on the ProScan board, then install the ProScan board into those empty sockets on the video board.

The board changed the Z-100's normal horizontal scan rate from the standard 15.75 kilohertz to a new value of 27 KHz. This scan rate was higher than the Extended Graphics Adapter (EGA) standard for IBM compatibles, but still lower than that of the new Video Graphics Array (VGA) system used by IBM's new PS/2 computer series of the time.

Programming:

With ProScan, the NEC MultiSync monitor could be used as either a normal-scan-speed or a high-scan-speed monitor. The number of lines on the screen would be established by the software you used. It was also easy to include ProScan support in any programs that you wrote yourself.

Switching between the two modes was controlled by using direct access to the registers of the CRT-C chip. It could be done using any language that allowed direct access to the computer's ports, including 'C', Pascal, and BASIC.

The Z-100's CRT-C used two ports to communicate with the computer; one was at port address 0DCh; the other at address 0DDh. Programs wrote to 0DCh to select one of the CRT-C's 18 internal registers for access. Then they used 0DDh to write a value to the selected register.

ProScan would recognize the value of the third bit of the CRT-C's register 8 as the switch to toggle the high-speed scan rate operation. (Register 8 is the interlace and skew register, and bit 3 isn't used by normal Z-100 activities.

The following 'C' code fragment would turn ON high-speed operation:

```
OUTB(8,0xDC);  
OUTB(4,0xDD);
```

In BASIC, that would be:

```
10 OUT &HDC,8  
20 OUT &HDD,4
```

To turn ProScan OFF in C, you could use:

```
OUTB(8,0xDC);  
OUTB(0,0xDD);
```

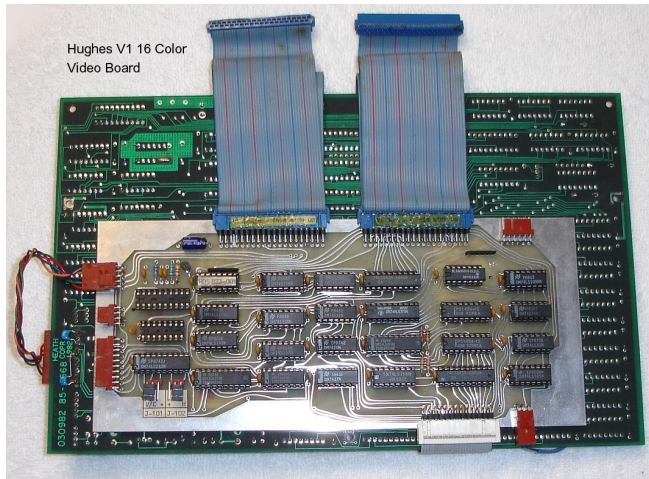
In BASIC, that would be:

```
10 OUT &HDC,8  
20 OUT &HDD,0
```


Hughes V1 Board: 16 Color Z-100 Video!

From William N. Locke's Z-100 Notebook Article in Sextant Magazine #31, Nov/Dec, 1987.

And from Stanley Schwartz's article "A More Colorful Z-100 Can Display EGA Video", published in Sextant Magazine #39, Early Winter, 1989.



Hughes Development Systems was offering the V1 Video Enhancement Kit, which gave the Z-100 an additional 64K bank of video memory and the intensity plane, for \$139.00. The Z-100 computer could then bank-switch the intensity plane with the blue plane, under the control of an I/O port. That memory permitted adding intensity to the primary colors red, green, and blue. Those three colors and the intensity could then be mixed to produce a possible 16 colors.

The blue plane and the intensity plane had the same memory address. The V1 control port located at 97 hexadecimal (151 decimal) determined which of the two were used.

Bit 5 of this port controlled whether or not the intensity plane was displayed on the monitor.

Bit 6 controlled whether the blue or intensity planes are accessed when writing to memory.

And bit 7 controlled whether the intensity plane should be written to automatically when any plane is written to.

If bit 7 was 1, then whenever a value was written to any plane, the same value would be written to the intensity plane. These three bits were the only bits used to control the V1 board.

The intensity plane was displayed and accessed only under the control of this V1 control port. The V1 board did not interfere with existing software. Stanley's article also has a full discussion on programming using the V1, using the V1 with the ZPC PC-emulator, and other comments outside the scope of this article.

(If you do not have access to either of these articles, I can email a copy to you - SWV)

The V1 also had a palette capability that consisted of a conversion circuit placed between the video memory and the monitor. This allowed mapping any color into another color without changing the contents of video memory, while making the color remapping invisible to any application program being run on the Z-100.

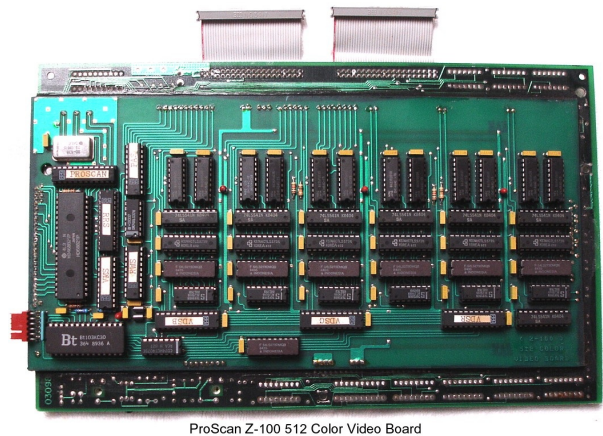
The V1 board and associated cables were physically installed between the video and main boards of the Z-100. No trace cutting or soldering were involved in the installation.

To see the extra colors, the monitor, such as the Zenith ZVM-135 color monitor, had to have the capability of using the intensity signal.

The basic configuration of the Video Enhancement system supported only the color-monitor (RGB) output. To add support for a monochrome monitor (composite) output, an optional hardware modification could be purchased from Hughes for \$29.00. Also, for those not having a Scottie Board (for PC-emulation), the V1 board could be purchased with a ZHS option for an extra \$29.00.

PROCOLOR: Higher Resolution Z-100 Video with 512 Colors!

From Paul Herman's Issue #9, Z-100 LifeLine, August 1990.



Travis Barfield developed another impressive video system for the Z-100. The new ProColor 512 color video upgrade for the Z-100. The production model of the ProColor video board consisted of a standard Z-100 video board with two additional circuit boards attached to it. They were all mounted together to make one 'thick' video board which mounted in the same way as the original video board.

Before you could use ProColor, however, you needed to have an auto-synchronizing monitor, like the NEC MultiSync or Logitech AutoSync.

At a minimum, you had to have a monitor that accepted analog RGB video signal. The **standard Zenith ZVM-134 or ZVM-135 monitor would NOT work**, because they were digital RGB monitors.

Another reason for going the MultiSync route was so you could use the high-resolution ProScan display mode, which was included with the color upgrade.

You could also use a composite monochrome monitor with ProColor, if a simple modification was made. The reason to do this was for use with the internal monitor on an All-in-one Z-100. Using the internal monitor, you could get 512 shades of grey (actually green or amber).

Programming was simple and could be done from any programming language, including BASIC. Once installed, the new video system gave you the equivalent of nine banks of video memory, instead of just three. This allowed you to have eight levels of each of the primary colors (8x8x8=512 colors).

ProColor allowed the use of three different video modes (standard, interlace, and ProScan), and one of two display formats (512 colors, or multiple pages of 8 colors).

On first power-up, ProColor is in normal video mode (640x225 pixels) with eight colors, just like a stock Z-100. But instead of having only one or two pages of video memory to work with, now there were six! This made it possible to display some interesting animated effects by switching between the six screens in succession. A demo program was included with ProColor that showed how this was done.

If you switched to a high-resolution video mode, such as Interlace or ProScan, you had three pages of video memory, compared to only one in a stock Z-100.

Multiple pages on video memory are neat, but the real fun began when you switched to the 512 color format. ProColor allowed access to each of the nine video planes separately, or write to multiple planes simultaneously. This was similar to the scheme used in the standard Z-100 video of having multiple access bits in the video control port. A separate 'clear screen' port was provided to make clearing all planes of video RAM just as simple as clearing one plane.

The main drawback to ProColor was the lack of software support. Only a few demo programs came with ProColor - the rest was up to the Z-100 community to develop.

Paul did create a GIF (Compuserve's 'Graphic Interchange Format') picture viewer and it is included in the Z-100 LifeLine Software Library.

For more information, see the original article. The production run was quite limited, even at the asking price of \$299.00 at the time.

A full Review of the ProColor 512 Board by Tim McCaffrey was published in Issue #21 of the Z-100 LifeLine, August 1992. It also included a means to get the ProColor board to work with a standard Z-135 color monitor.

I hope this helps clarify the use of Z-100 Interlace mode and Z-100 Graphics.

Cheers,

Steve Vagts

