AMERICAN INTERACTIVE MEDIA

AIM Technical Notes

TN #39: NTSC Coloring Problems

Written by: Rodney E. Wood May 23, 1989

CD-I applications designers often make assumptions about output from consumer CD-I players. The player may be connected to an RGB monitor or to an NTSC television. There are differences between small pixel patterns on an RGB monitor and on an NTSC television. We offer some caveats regarding the use of bit patterns in CD-I.

Anyone who has used the composite output from the JNMS player or from any other computer system has undoubtedly questioned the accuracy of the output color. Patterns that appear crisp and monochromatic in RGB form seem to take on coloration when they are encoded/decoded to/from NTSC. This aberration is known, and some computers even take advantage of it.

The problem stems from the very nature of NTSC video. Unlike RGB where all the main components of the video signal are available at all times (excluding sync times), NTSC video may be thought of as time multiplexed. Although the vector math involved is beyond the scope of this technical note, the color signal may be thought of as broken into bands of blue, green, and red. This is similar to the display of the vertical stripes of color on a Sony color television screen (although the screen's stripes are much finer). A complete set of blue, green, and red bands takes approximately 279 nanoseconds (1/3.579545 MHz) to display. The complete set is required to reproduce an accurate white, although it does not have to be phase-coherent to reproduce white in NTSC.

As shown in Figure 1 below, as the time period for a pixel is reduced from 1/3.579 µsec, its color becomes more saturated (e.g., less "white"). A "white" dot at 1/7.159 µsec would be perceived as a color, such as turquoise or orange, on an NTSC television or monitor. However, it would continue to be perceived as white on an RGB monitor. The exact color depends on its phase relationship to a 3.579546 MHz signal that synchronizes the decoding at the beginning of each scan line called the "color burst."

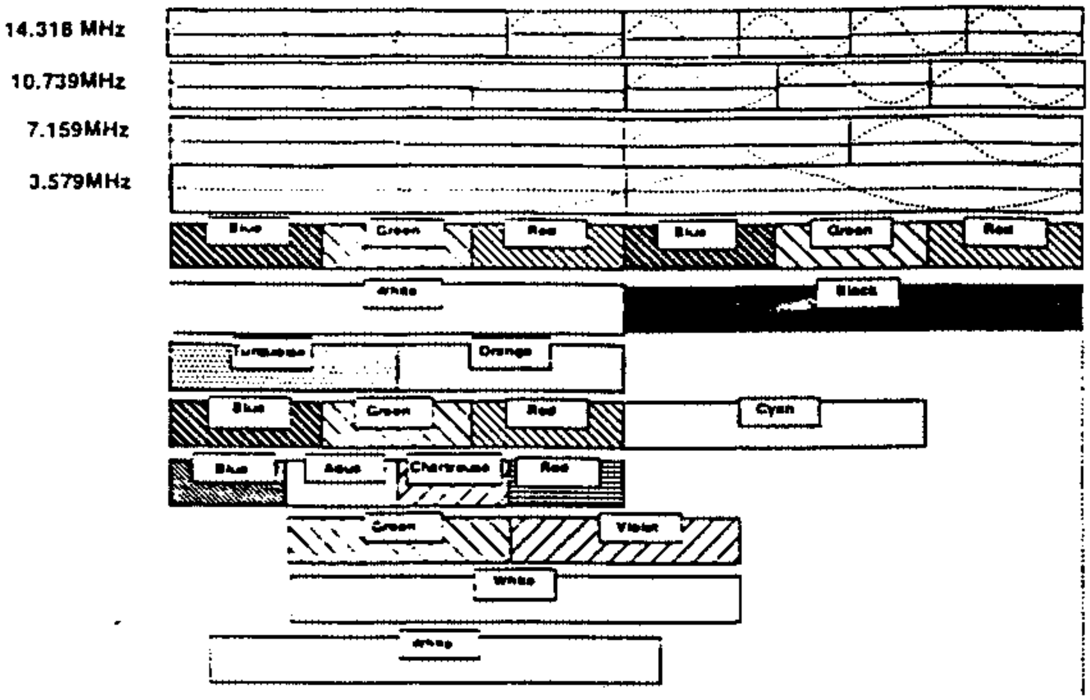

Figure 1: Clocks & Color

Figure 2 illustrates the multiplexing that occurs during NTSC encoding. Once every 279 nano-seconds, the "switch" makes a complete revolution, sampling each of the three primary colors. If the incoming signal is of sufficient duration, all three signals are multiplexed, as in the top example of "white." In the second example, a "cyan" color with only blue and green components is present. The resulting output is "shorter", since no red is present. In the third example, a "short white" is present followed by a "short black." This results in a "turquoise" being encoded. Notice how similarly the NTSC standard versions of "short white" and "cyan" are encoded.

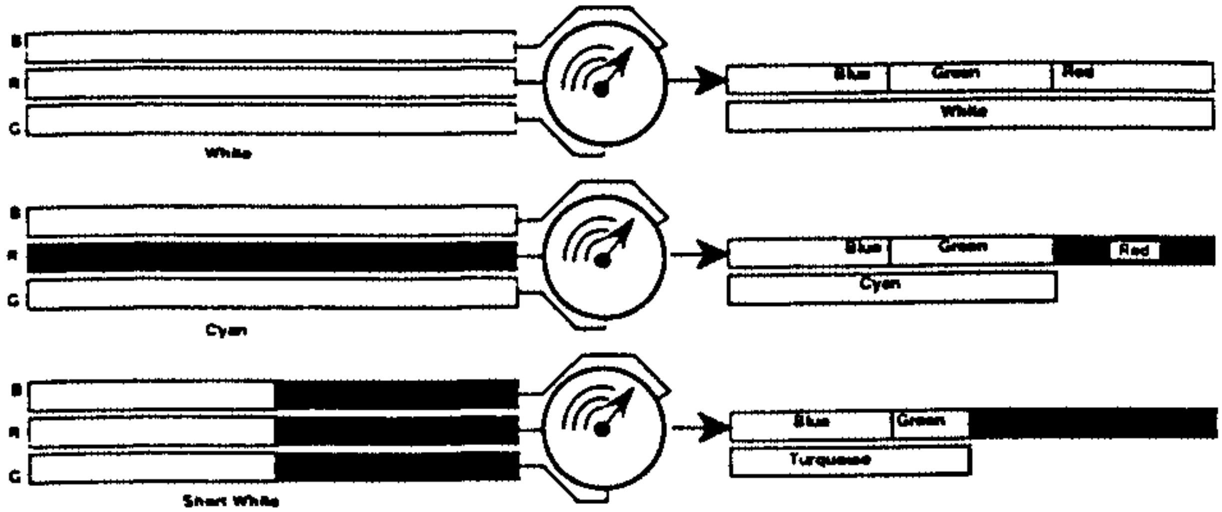

Figure 2: Time Multiplexing

This phase relationship may be used to advantage in some personal computers. The original Apple computers (6502-based) relied on this phenomenon to produce the user's choice of green/violet or orange/blue pixels by selecting the pixels that should be "on" or "off". The main processor and video monitor derived their clocks from a single 14.318 MHz (4 * 3.58 MHz) clock. By phase adjusting the output dot, the four colors were selectable. If two adjacent pixels were "on", then the result was "white", and if the two were "off" then the result was "black" (Figure 3). Phase adjustment was derived from the most significant bit of a byte: the least significant seven bits were used for the color information.

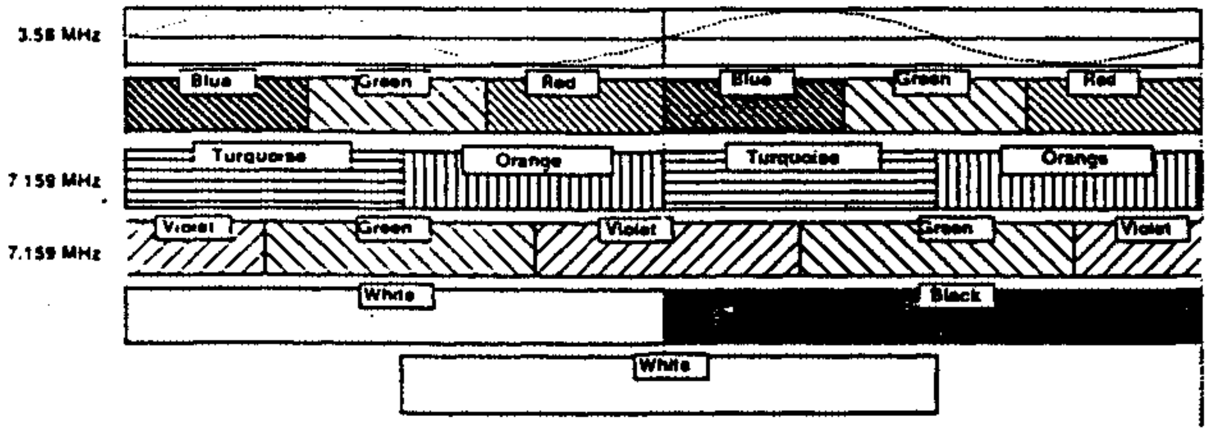

Figure 3: Apple II Color Implementation

Similarly, when IBM built the original PC, they realized that standard video could be derived from a 14.318 MHz clock. But unlike Apple, they chose to make all standard character sets with two-pixel wide verticals to eliminate the color aberrations that would occur using one-pixel wide verticals in their characters. Color in NTSC is derived from special clocking and gates, requiring an additional two TTL logic chips. Normal dot-clock rate is 7.159 MHz (14.318 / 2). Character verticals are normally two pixels wide (3.58 MHz). Thus, the IBM Color Graphics Adapter (CGA) produces forty columns of clean white characters. The gating logic determines the expected color and gates the signal "on" and "off". Since IBM's gating logic is similar to that illustrated in Figure 2, double pixel widths are required to assure that the desired color is gated.

Because they were familiar with the hardware, some software designers took matters into their own hands regarding color generation and purposely created graphics in a high-resolution mode (using the 14.318 MHz clock). They were able to produce a multitude of colors by selecting the proper phase in which a pixel should be "on" or "off". Thus, some software simulated not only the colors found in the Apple II, but additional colors not found in that computer.

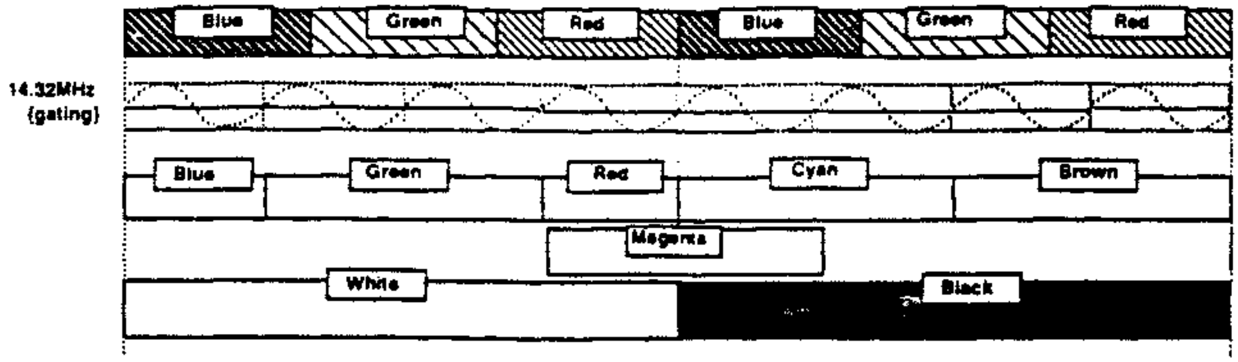

Figure 4: IBM PC Color Implementation
[showing gating for NTSC color generation]

To date only one other home computer system did not base its color video signal on a 14.318 MHz clock. That computer was the Texas Instruments TI 99/4. Its Video Display Processor (TI 9918) was fed a 10.739 MHz clock, which it used to generate a 3.579 MHz color burst (÷3) and its 5.370 MHz dot clock (÷2). The 10.739 MHz input clock was three-quarters of 14.318 MHz. Its internal NTSC mechanism was similar to that shown in Figure 2; thus, pure primary colors were very true. However, single pixels of white ususally contained some hue other than white. Only triple pixels could achieve true colors. Although the TI computer is no longer in production, their Video Display Processor is still available.

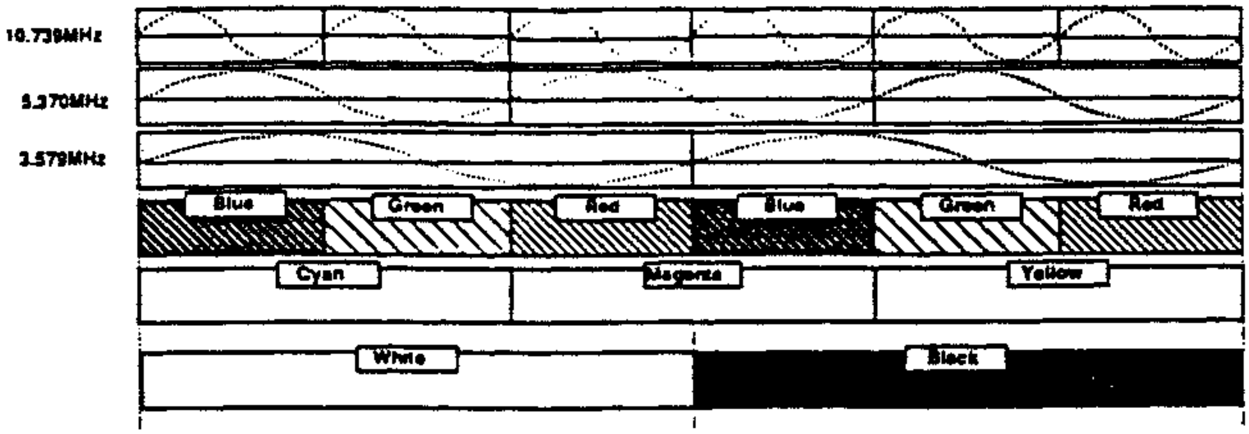

Figure 5: TI 99/4 Pixel Clock and its Colorations

Finally, we can consider CD-I. Philips and Sony have chosen to create a dot-clock rate of 7.5 MHz (normal resolution) instead of the usual 14.318 MHz, 7.159 MHz, or even 3.58 MHz. Although CD-I's unusual clock rate results in a higher horizontal resolution (384 versus 320 for the IBM PC and 280 for the Apple II), the color values in NTSC are unpredictable. Single pixel "white" verticals (or, worse, diagonals) become a multitude of colors, depending on their exact location on the screen. Even twin-pixel verticals have a pale hue. Although the precise details of how NTSC color is generated in a CD-I player are not generally known (the implementation is left to the hardware manufacturer as an exercise), it can be said that single pixel colors will not appear correctly on an NTSC monitor. Even double-pixel colors may vary in brightness across the screen.

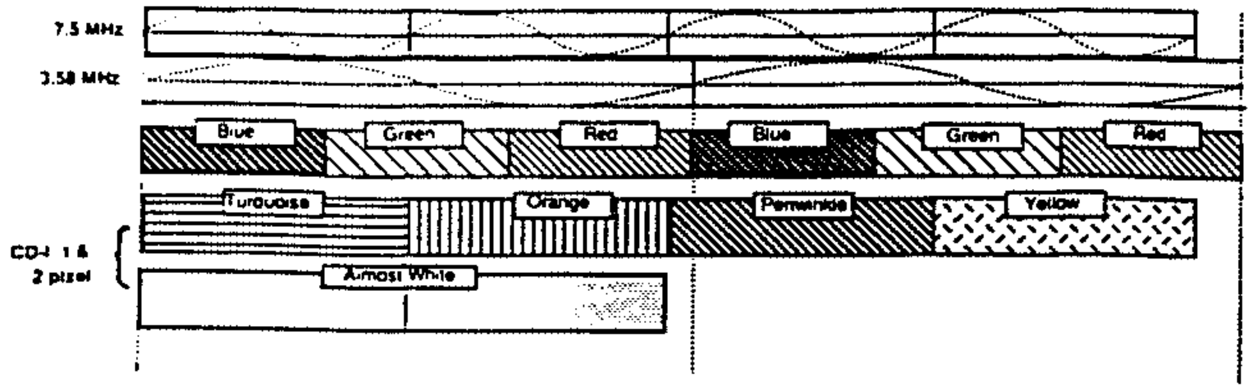

Figure 6: CD-I Color Implementation
[7.5 MHz dot, 3.579 MHz color clocks]

For this reason, even multibit (antialiased) fonts take on hue problems when their verticals are less than two pixels wide. Small fonts and those with single-pixel detailing may be illegible or suffer from color banding. One-bit fonts suffer even more in NTSC than their multibit counterparts. The CD-I producer should take this into account when designing screens for use in a consumer environment. This will become particularly important as the price of the CD-I player drops and it becomes available in homes with ordinary television sets. Acceptable fonts in RGB may not display as well on a television set. Characters that may exhibit particularly unpleasant effects include (but are not limited to) 'M', 'X', '#', 'A', 'V', and '@.' Similarly, fine patterns and highly saturated colors should be avoided for backgrounds; they also suffer when viewed on an NTSC monitor. Large areas of color are relatively unaffected by this phenomenon. If the screen designer varies only the luminance of patterns (white on black or light cyan on dark cyan), the problems outlined above can be minimized. Sticking with monochromatic backgrounds and lower contrasts (light/dark grey vs. white/black) also helps. In addition, the designer should be aware that the problem exists for all image coding methods in CD-I, not just CLUT (Color Look-Up Table).

Notes:

1) For a more detailed explanation of Apple II color generation, see the **Applesoft manual.**

2) For a more detailed explanation of IBM PC color generation, see the original **IBM Technical Manual.**

2) For a more complete explanation of the Texas Instruments TI 99/4 (TI 9918) color generation, see the **TI 9918 Data Manual.**

4) All figures are purely for illustrative purposes; however, the relative time periods are quite accurate. For example, the true NTSC phasing begins somewhere between red and blue, but closer to the blue area.

5) Truly "square" pixels may be obtained in "NTSC" mode with a dot clock rate of approximately 6.0 MHz, but no popular consumer computer uses this rate.