Blackfin® Processor's Parallel Peripheral Interface Simplifies LCD Connection in Portable Multimedia

By David Katz [david.katz@analog.com]
Ching Lam [ching.lam@analog.com]
Rick Gentile [richard.gentile@analog.com]

As low-power, fixed-point processors such as ADI's Blackfin family increase in performance and popularity, they can serve more and more multimedia applications. Many of these applications require small, low-power *liquid-crystal-display* (LCD) panels that have, in general, lower video resolutions than the full NTSC/PAL video used for broadcast TV. These panels are usually controlled either by a microcontroller or a dedicated LCD controller chip. But today, Blackfin processors have sufficient performance to handle both signal processing and control functions, and also to interface directly to the LCD displays—considerably reducing system cost and complexity. This article will discuss how the ADSP-BF561 Blackfin processor's *parallel peripheral interface* (PPI) integrates LCD display capability into the world of high-performance media processing, allowing a single processor to be used for both system processing and display driving.

Passive vs. Active

There are two major categories of LCD array technology—*passive*-matrix and *active*-matrix.

In the former, a glass substrate imprinted with rows forms a liquid-crystal sandwich with a substrate imprinted with columns. Pixels are defined at row-column intersections. To activate a given pixel, a timing circuit energizes the pixel's column while grounding its row. The resulting voltage differential renders the liquid crystal opaque in the vicinity of that pixel location, blocking light from coming through.

Although it is straightforward, passive matrix technology does have some shortcomings. For one, screen refresh times are relatively slow (which can result in *ghosting* for fast-moving images). Also, there is a tendency for the voltage field at a row-column intersection to *bleed* over into neighboring pixels, partly untwisting the liquid crystals and blocking some light from passing through the surrounding pixel area. The effect is to blur edges in the image and reduce contrast.

Active-matrix LCD technology, using an IC-like manufacturing process, is a considerable improvement. Each pixel has a capacitor, to retain charge between refresh cycles, and a transistor switch (giving rise to the popular term, *thin-film-transistor*—TFT—display). To address a particular pixel, its row is enabled, and a voltage is applied to its column. This has the effect of isolating only the pixel of interest, so others in the vicinity are not influenced. The current drawn in controlling a given pixel is reduced, so pixels can be switched at a faster rate, leading to faster refresh rates for TFTs compared to passive displays. What's more, modulating the voltage level applied to the pixel allows many discrete levels of brightness. Today, it is common to have 256 levels, corresponding to 8 bits of intensity.

For color displays, each pixel actually has three subpixels—with red, green and blue (R-G-B) filters—that the human eye sees as a single-color spot. For example, a 320 × 240 pixel display actually has 960 × 240 subpixels, accounting for the R, G and B components. Each subpixel has 8 bits of intensity, thus forming the basis of the common 24-bit color LCD display.

Since LCD technology relies on regulating the passage of light at the pixel level, one might wonder where the light would be generated. Many small, low-cost monochrome LCDs are *reflective*, meaning that external light reflects off the substrates but is blocked in areas where a liquid crystal segment is charged.

Since TFT color displays have millions of transistors that filter the incoming light, reflective displays would not be effective in active-matrix technology. Instead, the displays are backlit (or *transmissive*); typically a fluorescent light—or a white *light-emitting-diode* (LED) array, integrated into the display—generates light that is modulated as it is transmitted through the various layers of the LCD "sandwich". Unfortunately, the large surface area consumed by the transistors necessitates a greater light output from the backlight. In addition, each transistor of a TFT display dissipates power, so active-matrix displays are somewhat power-hungry compared with their passive cousins.

Components of a TFT-LCD System

Connecting to a TFT-LCD panel can seem complicated, considering all of the different components involved. First, there's the *panel* itself, which houses an array of pixels arranged for strobing by row and column at high speed, referenced to the pixel-clock frequency.

The backlight is often a *cold-cathode fluorescent lamp* (CCFL). In a CCFL, excited gas molecules emit bright light while generating very little heat. This low dissipation, plus their durability, long life, and straightforward drive requirements, make them ideal for LCD panel applications. As mentioned above, LEDs are also a popular backlight method, mainly for small- to mid-sized panels. They have the advantages of low cost, low operating voltage, long life, and good intensity control. However, in larger panels, LED backlights can draw a lot of power compared to CCFL solutions.

An LCD *controller* contains most of the circuitry needed to convert an input video signal into the proper format for display on the LCD panel. It usually includes a *timing generator*, which controls the synchronization and pixel-clock timing of the individual pixels on the panel. Additionally, it can offer a wide variety of extra features—such as on-screen display, graphics overlay blending, color lookup tables, dithering, and image-rotation. The more elaborate chips can be very expensive, often surpassing the cost of the processor to which they're connected. Some media processors, like ADI's Blackfin family, have ports that act electrically as an LCD interface—without requiring an external chip.

An *LCD driver* chip is necessary to generate the proper voltage levels to the LCD panel. It serves as the *translator* between the output of the LCD controller and the LCD panel. The rows and columns are usually driven separately, with timing controlled by the timing generator. Since dc currents will stress the crystal structure and ultimately cause deterioration, liquid crystals must be driven with periodic polarity inversions. Therefore, depending on the implementation, the voltage polarity applied to each pixel varies on either a per-frame, per-line, or per-pixel basis.

Connecting to TFT-LCD Modules

With the trend toward smaller, cheaper multimedia devices, there has been a push to combine the driver, controller, and LCD panel. Today, integrated TFT-LCD modules include timing generation and drive circuitry—thus requiring only a data-bus connection, clocking/synchronization lines, and power supplies. However, in order to meet panel-thickness and cost requirements in smaller PDA-type LCD panels, the timing generator often cannot be integrated into the LCD module. In this case, a separate external timing ASIC is required to produce timing signals to drive the individual rows and columns of the LCD panel.

Nevertheless, the ADSP-BF561 Blackfin Processor can *directly* connect to many TFT-LCD modules through its *parallel peripheral interface* (PPI). The PPI is a multifunction parallel interface that can be configured between 8- and 16 bits in width. Supporting bidirectional data flow, it includes three synchronization lines and a clock pin for connection to an externally supplied clock. In addition to connecting to LCD panels, the PPI can gluelessly decode ITU-R BT.656 data and can also interface to ITU-R BT.601 video streams.

Because the ADSP-BF561 provides many general-purpose timers with *pulse-width-modulation* (PWM) capability, it can be configured to provide the proper LCD timing to a module, thus eliminating the need for an external timing ASIC. Figure 1 shows a block diagram of the basic connection between the Blackfin Processor and a TFT-LCD module. Also shown is the ADSP-BF561 EZ-KIT Lite evaluation board; its many conveniences provide an easy way to get started with a wide variety of Blackfin applications, including the one discussed here.

Power Requirements

TFT-LCD panels typically need two separate power supplies. First, the panel itself has a power supply line. Although the voltage supply requirement varies among LCD panels, the usual values are either 3.3 V or 5 V. Second, CCFL backlights need a high-voltage supply to excite the gas molecules to fluorescence. This voltage is usually generated with a dc-ac inverter on a separate circuit board within the TFT-LCD module. On the other hand, LED backlights, not requiring a high-voltage ac supply, can usually be powered directly from a 5-V or 12-V dc source.

Clocking and Synchronization

The pixel clock period defines the pixel sampling rate, so speeds vary depending on panel resolution and refresh interval. For instance, a VGA panel (640 × 480 active pixels) with a 60-Hz refresh rate would require a 250-MHz clock, whereas a QVGA panel (320 × 240 active pixels) could run at 5 MHz.

The synchronization lines control the time during which each line and video frame is scanned and displayed on the LCD. There are two scanning methods, *interlacing* and *progressive* scan. In interlacing, the odd lines of the video frame are first drawn onto the screen, and then the even lines are filled in. In progressive scan, the video lines are displayed continuously in sequence.

Many newer progressive scan TFT-LCD panels use the synchronization lines to control where each line and frame begins and ends. The horizontal sync (HSYNC) indicates the beginning of each new line, while the vertical sync (VSYNC) denotes the beginning of each new frame. They ensure the generation of an aligned and viewable image. The polarity of the HSYNC and VSYNC pulses and the durations of the pulse widths vary among panels.

The ADSP-BF561 generates the HSYNC and VSYNC signals with configurable PWM outputs in order to achieve the greatest flexibility. This allows adjustments for the polarity, pulse-width, and period specified by a particular TFT panel.

Often, LCD timing requirements specify an *invalid* data period between the assertion of the horizontal sync signal and the actual displayed image data. The ADSP-BF561's PPI can handle this timing by allowing outgoing data to be delayed by a specified number of clock cycles after the HSYNC signal is received.

Data Lines

Although the module's data interface is straightforward, there are many things to consider in choosing the appropriate RGB data format. The three most common configurations use either 8 bits per channel for RGB (8-8-8 format), 6 bits per channel (6-6-6 format), or 5 bits per channel for R and B—and 6 bits for G (5-6-5 format).

The 8-8-8 RGB data format provides the greatest color clarity. With a total of 24 bits of resolution, more than 16 million shades of color are available. This format offers the precision and resolution needed for high-performance LCD TVs.

The 6-6-6 format is popular in portable electronics. The 18 bits of resolution provide over 262,000 shades of color. However, because the 18-pin (6+6+6) data bus doesn't conform nicely to 16-bit processor data paths, a popular industry compromise is to use 5 bits each of R and B, and 6 bits of G (5+6+5 = 16) to match the 16-bit data bus. This scenario works well because, of the three, green is the most visually important color. The least-significant bits of both red and blue are tied to their respective most-significant bits at the panel. This ensures a full dynamic range for each color channel (from full saturation to total black).

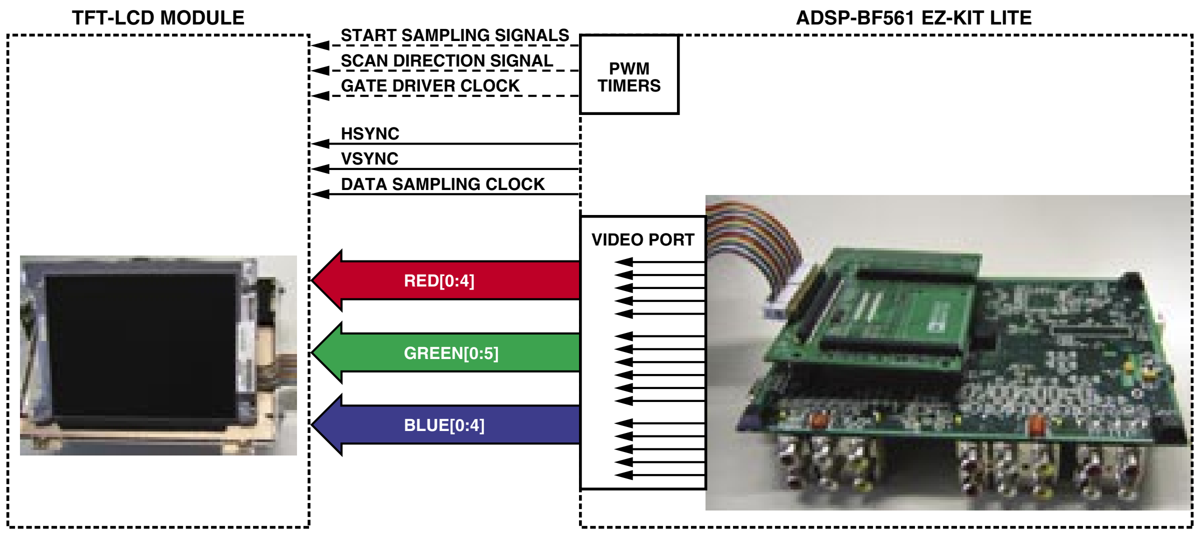

Figure 1. 5-6-5 LCD connection: The ADSP-BF561 eliminates the need for a timing ASIC by supplying the dotted-line connections.

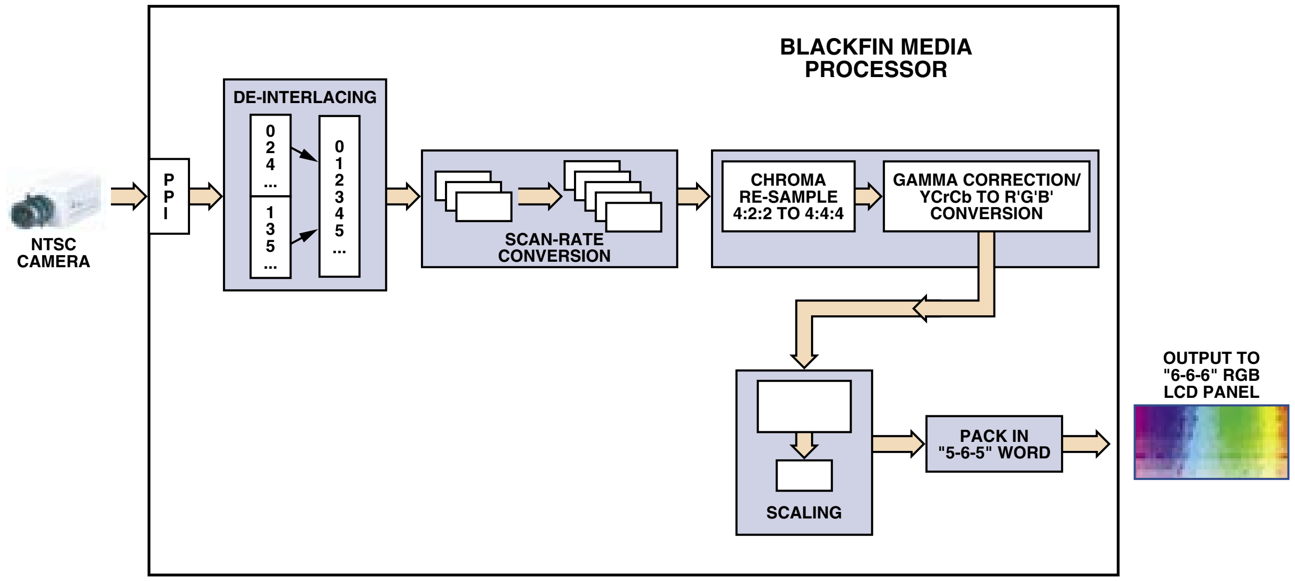

Figure 2. Example of system flow: converting a signal from a camera source to an LCD display output.

System Algorithm Flow

To understand what's involved in emulating an LCD controller on a media processor (in order to replace an external device), let's take a look at the system flow involved in displaying an incoming raw video stream on an integrated TFT-LCD module. Consider the example of Figure 2, where the digitized output of an NTSC camera provides the image stream applied to the video port of the ADSP-BF561 processor. We will discuss each of the steps shown in the figure.

De-interlacing

In interlaced video, used by the NTSC camera in the example, odd and even fields are separated, so that all odd lines in a given frame are transferred before any even lines. For this example, the video stream from the camera must be de-interlaced after it enters the video port. This is done in one of several ways, depending on the desired output quality. The simplest method is *line doubling*, which copies each odd line onto the subsequent even line, effectively eliminating the even field in favor of a shifted version of the odd field. Because this creates noticeable artifacts, more processing-intensive methods are often used. These include *linear interpolation*, motion *compensation* and *median filtering*. This latter method replaces each pixel's intensity value with the median gray-scale value of its immediate neighbors to help eliminate high-frequency noise in the image.

Scan-Rate Conversion

Once the video has been de-interlaced, a scan-rate conversion process may be necessary in order to insure that the input frame rate matches the output display-refresh rate. In order to equalize the two, fields may need to be dropped or duplicated. As with de-interlacing, some sort of filtering is desirable to smooth out high-frequency artifacts caused by creating abrupt frame transitions.

Chroma Resampling and Color Conversion (YCrCb → RGB)

Some cameras supply pixel information in raw form, exactly as the image sensor supplies it. This could mean one red, blue, and green value for each pixel in the sensor, or one Y, Cr, and Cb value for each pixel. Y, Cr, and Cb are mathematically related to the RGB values, but being less inter-correlated than RGB data, they allow better compression ratios. More commonly, though, the camera outputs a condensed stream that takes advantage of the physiology of the eye, providing greater weighting for green (in the RGB case) or for intensity (Y) in the YCrCb space. In the example of Figure 2, the video stream enters the PPI in 4:2:2 YCrCb format. "4:2:2" implies that there are four luma (Y) intensity values for every two chroma (Cr and Cb) values on a given video line. Each (Y,Cb) or (Y,Cr) 16-bit pair represents one pixel value.

For display on an LCD panel, the data stream ultimately needs to be converted to RGB space. More correctly, it needs to be transformed to R'G'B' space, which is a *gamma*-corrected version of RGB space. Gamma correction adjusts for the nonlinear properties of the LCD panel, since the brightness of a given pixel is not a linear function of the voltage applied at that pixel site. Varying gamma changes the ratios of red to green to blue in an image, as well as image brightness. Figure 3 shows a sample equation set for converting between YCrCb space and R'G'B' coordinates.

$$
\begin{aligned}
Y &= (0.301)R' + (0.586)G' + (0.113)B' \\
Cb &= -(0.172)R' - (87/256)G' + (0.512)B' + 128 \\
Cr &= (0.512)R' - (0.430)G' - (0.082)B' + 128
\end{aligned}
$$

$$
\begin{aligned}
R' &= Y + 1.371(Cr - 128) \\
G' &= Y - 0.698(Cr - 128) - 0.336(Cb - 128) \\
B' &= Y + 1.732(Cb - 128)
\end{aligned}
$$

Figure 3. Sample of conversion equations between gamma-corrected RGB and YCrCb color spaces (assuming 8-bit pixel components).

Prior to R'G'B' conversion, the Cb and Cr channels must be resampled to achieve a 4:4:4 format, where one byte each of Y, Cb, and Cr represents one pixel value, as shown in Figure 4. A clear-cut way to resample is to interpolate the missing chroma values from their nearest neighbors by simple averaging. Higher-order filtering might be necessary for some applications, but this simplified approach is often sufficient. In reality, the steps of chroma resampling and color space conversion can both be performed as a single operation, since each discrete step involves linear pixel operations.

Scaling

Video scaling, the next step, is very important because it allows the generation of an output stream whose resolution is different from that of the input format. Ideally, the fixed scaling requirements (input data resolution, output panel resolution) are known ahead of time, in order to avoid the computational load of arbitrary scaling between input and output streams. As a much simpler, cheaper option, the processed image can be cropped to fit within the confines of a smaller LCD panel.

Depending on the application, scaling can be done either upwards or downwards. It is important to understand the nature of the image content to be scaled (e.g., the presence of text and thin lines). Improper scaling can make text unreadable or cause some horizontal lines to disappear in the scaled image.

The most straightforward methods of scaling involve either dropping pixels or duplicating existing pixels. That is, when scaling down to a lower resolution, a number of pixels on each line (and/or some number of lines per frame) can be discarded. While this represents a low processing load, the results will yield aliasing and visual artifacts.

A small step upward in complexity uses linear interpolation to improve the image quality. For example, when scaling down an image, interpolation in either the horizontal or vertical directions provides a new output pixel to replace the pixels used in the interpolation process. As with the previous technique, information is still thrown away, so artifacts and aliasing will again be present.

If the image quality is paramount, there are other ways to perform scaling—without reducing quality. These methods strive to maintain the high frequency content of the image consistent with the horizontal and vertical scaling, while reducing the effects of aliasing. For example, assume an image is to be scaled by a factor of $Y \times X$. To accomplish this scaling, the image could be up-sampled (interpolated) by a factor, Y, filtered to eliminate aliasing, and then down-sampled (decimated) by a factor X. In practice, these two sampling processes can be combined within a single multirate filter.

Bit Extraction/Byte Packing

As described earlier, it is preferable to transfer 16 bits on each outgoing LCD clock cycle. This 5-6-5 bit packing can be accomplished with the source data. The Blackfin architecture offers a choice between two efficient methods to create the desired byte stream. The first is simply to shift the appropriate bits from each color (red, blue, and green), into a target register. The second is to make use of an EXTRACT/DEPOSIT instruction pair to pull out some number of bits, beginning at a specific bit location, and deposit the result in a target register.

Application note EE-256 provides a detailed description of a system where the processor, mounted on an ADSP-BF561 EZ-KIT Lite evaluation board, receives a streaming video input from a DVD player and connects to a TFT-LCD module. The Blackfin generates all necessary timing and performs decimation, color conversion, resampling, and output formatting. System data flows and buffer management are described in detail, and sample code for a working application with a specific LCD module is provided for download.

CONCLUSION

Due to its performance and popularity, members of the Blackfin processor family are serving in increasing numbers of multimedia applications. They are especially useful in system designs calling for displays that require small, low-power, moderate-resolution *liquid-crystal-display* (LCD) panels. For many of these applications, Blackfin processors have sufficient performance to handle both signal processing and control functions, and also to interface directly to the LCD displays—considerably reducing system cost and complexity. This article has suggested how such a system can be accomplished, by employing a portion of the ADSP-BF561 Blackfin processor's spare computing power and its *parallel peripheral interface* for display driving.

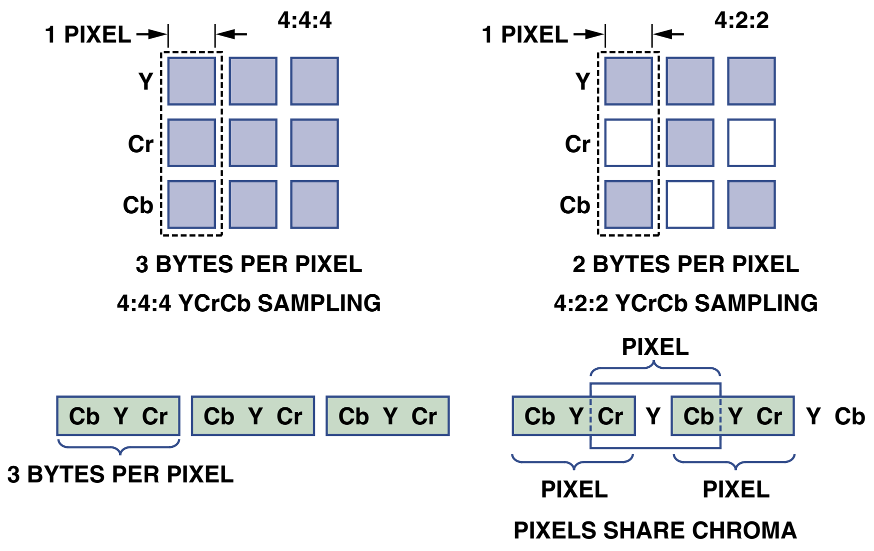

Figure 4. Illustration of 4:4:4 and 4:2:2 YCrCb sampling.