Technical notes on using Analog Devices DSPs, processors and development tools
Visit our Web resources http://www.analog.com/ee-notes and http://www.analog.com/processors or e-mail processor.support@analog.com or processor.tools.support@analog.com for technical support

Using the ADSP-BF561 Blackfin® Processor as a TFT-LCD Controller

Contributed by C. Lam

Rev 2 – March 7, 2006

Introduction

The ADSP-BF561 Blackfin® processor has two parallel peripheral interfaces (PPIs) that enable video to be simultaneously input and output from the device. The PPI can interface to video decoders that convert standard analog television signals, compatible with NTSC and PAL standards, into 4:2:2 component video data. The PPI can also directly connect to several TFT LCD panels that accept RGB video data. This EE-Note describes the hardware and software requirements necessary to develop a video application that simultaneously streams data to the ADSP-BF561 Blackfin processor and displays it on a LCD panel with a RGB interface.

Hardware Interface

The following hardware was used in this implementation of the application:

- ADSP-BF561 EZ-KIT Lite® board
- Logic Product Development Zoom Display Kit (with Sharp LQ035Q7DB02 LCD Panel)
- Input video source (DVD player, video camera, or any other input video source that outputs signals compatible with NTSC standards)

The selection of this Sharp LCD panel is arbitrary. This particular LCD panel is very popular in PDA applications and has a "low level interface," which means most of the drive electronics normally implemented in the LCD module have been removed. Removing these electronic components from the module saves both cost and space. Because the drive electronics are not integrated in the LCD module, an additional timing ASIC is usually needed to generate the specific timing signals required for the row and column drivers.

However, because the ADSP-BF561 processor provides many general-purpose PWM timers, it can be configured to provide the proper LCD timing, thus eliminating the need for the external timing ASIC.

To simplify the development process, Logic Product Development produces an evaluation module for this particular Sharp LCD panel, called the Zoom Display Kit (ZDK). In this application, the EZ-KIT Lite board interfaces directly to the ZDK.

A DVD player serves as the video input source for this application. The S-Video output of the DVD player connects to the RCA video input jack (J6) on the EZ-KIT Lite board.

Figure 1 in the Appendix shows a block diagram of the hardware connections between the ADSP-BF561 processor, the input video source, and the output display panel. Figure 2 shows the pin interface between the ADSP-BF561 processor and the LCD panel.

Software Interface

The software interface consists of three main parts:

- Receiving the input data
- Processing the input data
- Transmitting the processed data

The dual-core ADSP-BF561 Blackfin processor has 32 KB of L1 instruction memory SRAM available for each core. In partitioning the software components, we choose to have Core A handle the DMA of the input data and Core B handle the core processing and DMA of the output data. In this scheme, Core A is unused for any core processing and is available for additional tasks, if necessary.

Figure 3 in the Appendix shows a block diagram of the software flow: receiving the input video data, processing and formatting the data, and sending the data out to be displayed.

Receiving the Input Data

Configuring the processor to receive the input data requires setting up input source buffers in SDRAM memory, configuring the DMA channel, and configuring the PPI port. Because the EZ-KIT Lite design maps the video decoder to PPI0, this port is designated to be used for video input.

Source Buffers in SDRAM Memory

There are two source buffers in SDRAM memory defined to receive the incoming data. Having two buffers allows time for processing of the first buffer while simultaneously receiving data from the next video frame.

Direct Memory Access (DMA)

There are two peripheral DMA controllers on the ADSP-BF561 processor. Each DMA controller has a set of channels linked to different peripherals. Since PPI0 is linked to DMA controller 1, channel 0 by default, this channel must be configured to transfer the data from PPI0's port to the SDRAM source buffers.

Because there are two source buffers to fill with the incoming data, the small model descriptor list is the DMA mode of choice. This mode instructs the DMA channel to transfer data to the first buffer and then continue to fill the second buffer upon completion of the first.

This mode requires both setting up descriptors in memory and writing directly to memory-mapped registers (MMRs). The following registers are written directly:

- X and Y count registers: these registers specify the number of words to transfer. The word size can be 8, 16, or 32 bits, configured by writing the `WDSIZE` bits in the configuration register. In this application, the input frame size is 1716 bytes * 525 lines. Since the word size used is 32 bits, the X count register is set to 429 (1716/4), and the Y count register is set to 525.
- X and Y modify registers: these registers specify the number of bytes to modify after reading in a word. Since the word size used is 32 bits, the number of bytes to modify after each read is 4 (32 bits = 4 bytes).
- Next descriptor pointer register: this register contains the address of the next descriptor location to load after the data transfers specified by current descriptor have been processed. As soon as the DMA channel is enabled, it will read from the next descriptor pointer register, get the address of the next descriptor, and fetch the DMA elements from the descriptor. In this application, the next descriptor pointer register is assigned the address of the first receive descriptor.
- Configuration register: this register contains all the parameters and operating modes. In this register, we define the DMA mode to be small model descriptor list, word size to be 32 bits, descriptor size (number of elements

in each descriptor) to be 4, DMA direction to receive data, and interrupts to be enabled.

The receive descriptors are set up in L1 Data Bank A memory. There are two descriptors, one for each input source buffer. As described in the descriptor size of the configuration register, there are four elements in each descriptor. These elements define the next descriptor pointer location, the lower 16 bits of the receive buffer's starting address, the upper 16 bits of the receive buffer's starting address, and the DMA configuration value. The first descriptor points to the second descriptor, and the second descriptor points to the first descriptor. This creates a loop for the receive DMA to fill the first source buffer, then the second source buffer, and then repeat continuously.

Figure 4 in the Appendix shows a diagram of the receive DMA descriptors and how they are linked.

Parallel Peripheral Interface 0 (PPI0)

The PPI provides a parallel interface to the video decoder. After the decoder digitizes the incoming analog, NTSC, or PAL signal, it outputs a parallel 8- or 16-bit data stream. The PPI receives this data in ITU-R 656 format.

The registers necessary to configure the PPI to receive data from the decoder are:

- Frame register: this register holds the number of lines that make up a frame. If the programmed value in this register does not match the actual received lines, the frame track error (`FT_ERR`) bit is set in the PPI status register. In this particular application, the value programmed in this register is 525.
- Control register: this register sets the PPI parameters and operating modes. Through this register, the PPI is configured to receive data in ITU-R 656 mode, with a data length of eight bits.

Processing the Input Data

Once the first receive DMA completes, it generates an interrupt. In the interrupt service routine, a flag is set to indicate that the first input source buffer has been filled and is ready to be processed. The processing contains two routines:

- Decimation and de-interlacing
- Color space conversion and formatting

Decimation and De-interlacing

The input video source is an NTSC-compatible DVD player. An NTSC active video frame size dimensions of 720 words/line * 480 lines/frame and contains two fields/frame. All of Field 1's data (odd lines) is received first, followed by all of Field 2's data (even lines). However, typical LCD panels require a progressive data format, in which line one of Field 1 is followed by line one of Field 2, followed by line two of Field 1, and so on. The LCD panel used in this application has dimensions of 240 words/line * 320 lines/frame and requires a progressive data format.

The decimation and de-interlacing routine resolves the issues of the two different frame sizes and data arrangements.

Figure 5 in the Appendix shows a diagram of the data arrangement of interlaced and non-interlaced data.

Color Space Conversion

When the DVD player outputs an analog signal to the video decoder, the video decoder, in turn, outputs YCbCr component video data. These components represent the luminance and chrominance of the pixels. Because this LCD panel accepts data in RGB components, a color space conversion process is necessary.

The color space conversion routine converts the YCbCr components to RGB components using the following standard equations:

```
R = Y + 1.371(Cr-128)
  = Y + (Cr-128) + 0.371(Cr-128)
G = Y - 0.698(Cb-128)- 0.336(Cr- 128)
B = Y + 1.732(Cb-128)
  = Y + (Cb-128) + 0.732(Cb-128)
```

After conversion to RGB, each color component is represented by eight bits, producing a 24-bit pixel. Because the PPI port has a maximum data length of 16 bits, the converted pixel data must be scaled from 24 bits to 16 bits. Much research has been done on the importance and value of each color component. It has been determined that the green component provides more visual content than either the red or blue components. Therefore, in scaling the data to 16 bits, a 5-6-5 scheme is used. This scheme scales the 8-bit red and blue components to five bits each and the 8-bit green component to six bits, producing a 16-bit pixel.

Transmitting the Processed Data

Configuring the processor to transmit the output data requires setting up output destination buffers in SDRAM memory, configuring the DMA channel, configuring the PPI port, and configuring the necessary timing signals required by this particular LCD panel. Because PPI0 has already been used for input, the PPI1 port will be designated to transmit the output data and interface to the LCD panel.

Destination Buffers in SDRAM Memory

Similar to the source buffers allocated, two destination buffers are defined in SDRAM memory to accommodate the processed data from the two source buffers.

Direct Memory Access (DMA)

Since PPI1 is the designated transmit channel and linked to DMA controller 1, channel 1, this channel must be configured to transfer the data from the SDRAM destination buffers to PPI1's port.

The DMA configuration for the transmission of data is almost identical to the DMA receive configuration described above. The two main differences are the direction of transfer and the management of the descriptors.

Again, for the DMA transmit configuration, MMRs are written directly, as well as indirectly, via descriptors set up in memory. The following registers are written directly:

- X and Y count registers: in this application, the output frame size is 480 bytes * 329 lines. Since the word size used is 32 bits, the X count register is set to 120 (480/4), and the Y count register is set to 329.
- X and Y modify registers: since the word size used is 32 bits, the number of bytes to modify after each read is 4 (32 bits = 4 bytes).
- Next descriptor pointer register: in this application, the next descriptor pointer register is assigned the address of the first receive descriptor.
- Configuration register: in this register, we define the DMA mode to be small model descriptor list, word size to be 32 bits, descriptor size (number of elements in each descriptor) to be 4, DMA direction to transmit data, and interrupts to be enabled.

The transmit descriptors are set up in L1 Data Bank B memory. There are two descriptors, one for each output destination buffer. As specified in the descriptor size field of the DMA configuration register, there are four elements in each descriptor. These elements define the next descriptor pointer location, the lower 16 bits of the transmit buffer's starting address, the upper 16 bits of the transmit buffer's starting address, and the DMA configuration value. Both the first descriptor and the second descriptor point to themselves. This creates a loop for the transmit DMA to display the first or second output buffer

continuously until the descriptor has been modified.

Figure 6 in the Appendix shows a diagram of the transmit DMA descriptors and how they are linked.

Parallel Peripheral Interface 1 (PPI1)

The PPI provides a parallel interface to the LCD panel. After the input data has been processed and scaled, a 16-bit RGB word is sent out through the PPI. The PPI sends this data in general-purpose mode with one frame sync to indicate the start of each transmitting line.

The registers necessary to configure the PPI to correctly send data to the LCD panel are:

- Delay register: this register contains the number of PPI1 clock cycles to delay after the assertion of the `PPI1_FS1` frame sync pin before transmitting the output data. In this application, the value programmed in this register is 0x0. In TX modes, there is a one-cycle delay in addition to the value programmed in this register, therefore, the transmit data will be available one cycle after the assertion of `PPI1_FS1`.
- Count register: this register holds the number of words to write out through the PPI per line, minus one. Since there are 240 words per line in the LCD panel, the value programmed in this register is 239.
- Control register: through this register, the PPI is configured to send data in general-purpose transmit mode with 1 frame sync and a data length of 16 bits.

Timers and Programmable Flags

Because of the ADSP-BF561 processor's twelve general-purpose timers with pulse-width modulation (PWM) capabilities, this application uses those timers to replace the additional timing ASIC normally required to generate specific timing signals required for the row and column drivers of the LCD panel.

Six timing signals are required for this particular LCD panel. All of the timing signal requirements are met using five general-purpose timers and one programmable flag pin. In configuring the timers to generate the appropriate signals, three registers must be carefully configured:

- Period register: this register holds the period of the waveform. This register must be configured according to the datasheet of the LCD panel for the exact timing required to be generated.
- Width register: this register holds the pulse width of the waveform. Again, care must be taken to ensure that the pulse width is programmed according to the timing requirements specified in the LCD panel's data sheet.
- Configuration register: this register specifies the operating mode of the timers. It is through this register that the timers are configured for PWM mode, `SCLK` is selected as the clock counter, and the polarity of the pulse is configured as either positive or negative.

Since one of the timing signals does not have strict periodic requirements, a programmable flag pin (rather than a timer pin) is used to generate this particular signal. The registers necessary to program the flag pin are as follows:

- Direction control register: this register specifies whether the flag pin is used as an input or output. By setting bit 8 in this register, the `PF8` pin is configured to be an output pin.
- Flag data register: this register directly writes a value to the flag pins. By writing a 1 to bit 8 in this register, the `PF8` pin is set to a logic level of 1.

Management of Descriptors and Interrupt Service Routine

Once the very first processed buffer has been transmitted, an interrupt is generated. Since the transmit DMA is in descriptor mode, the next DMA has started already by the time the processor flow goes to the interrupt service routine. As described above in the transmit DMA section, the next descriptor pointers of both the first and second output buffers point to themselves. Therefore, even though the next DMA has started upon completion of the previous DMA, it is still transmitting from the same buffer, and thus displaying the same image to the LCD panel.

In the interrupt service routine, two actions occur. First, processing of the next input buffer starts. Second, upon completion of the processing, the current descriptor pointer is modified to point to the other descriptor, hence changing the displayed image to the next frame.

References

[1] *ADSP-BF561 Blackfin Processor Hardware Reference.* Rev 1.0, July 2005. Analog Devices, Inc.

[2] *ADSP-BF561 Blackfin Embedded Symmetric Multi-Processor Data Sheet.* Rev. 0, January 2005. Analog Devices, Inc.

[3] *ADSP-BF53x/BF56x Blackfin Processor Programming Reference.* Rev 1.1, February 2006. Analog Devices, Inc.

[4] *ADSP-BF561 EZ-KIT Lite Evaluation System Manual.* Rev 2.0, January 2005. Analog Devices, Inc.

[5] *Sharp LQ035Q7DB02 Technical Datasheet.* March 2002. Sharp Corporation.

[6] *Logic Product Development Zoom Display Kit QuickStart Guide*

Appendix

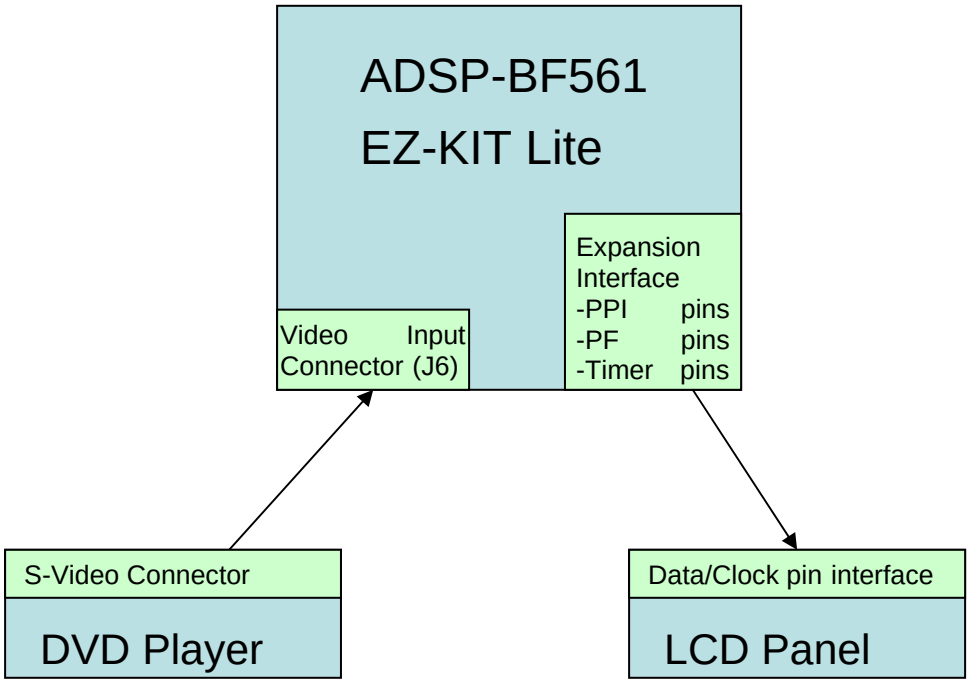

Figure 1. Hardware Connections Block Diagram

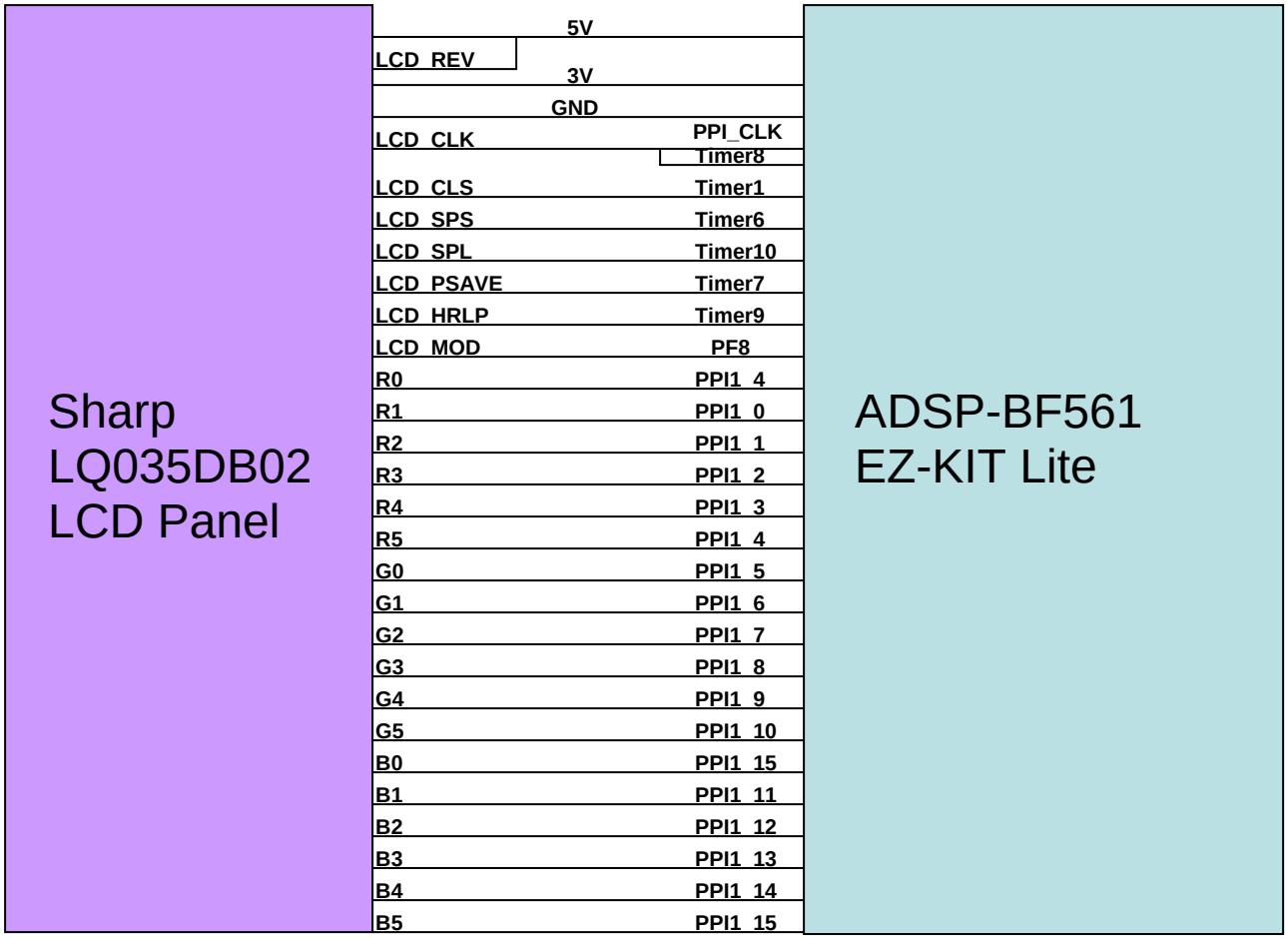

Figure 2. Blackfin Processor to LCD Panel Pin Interface

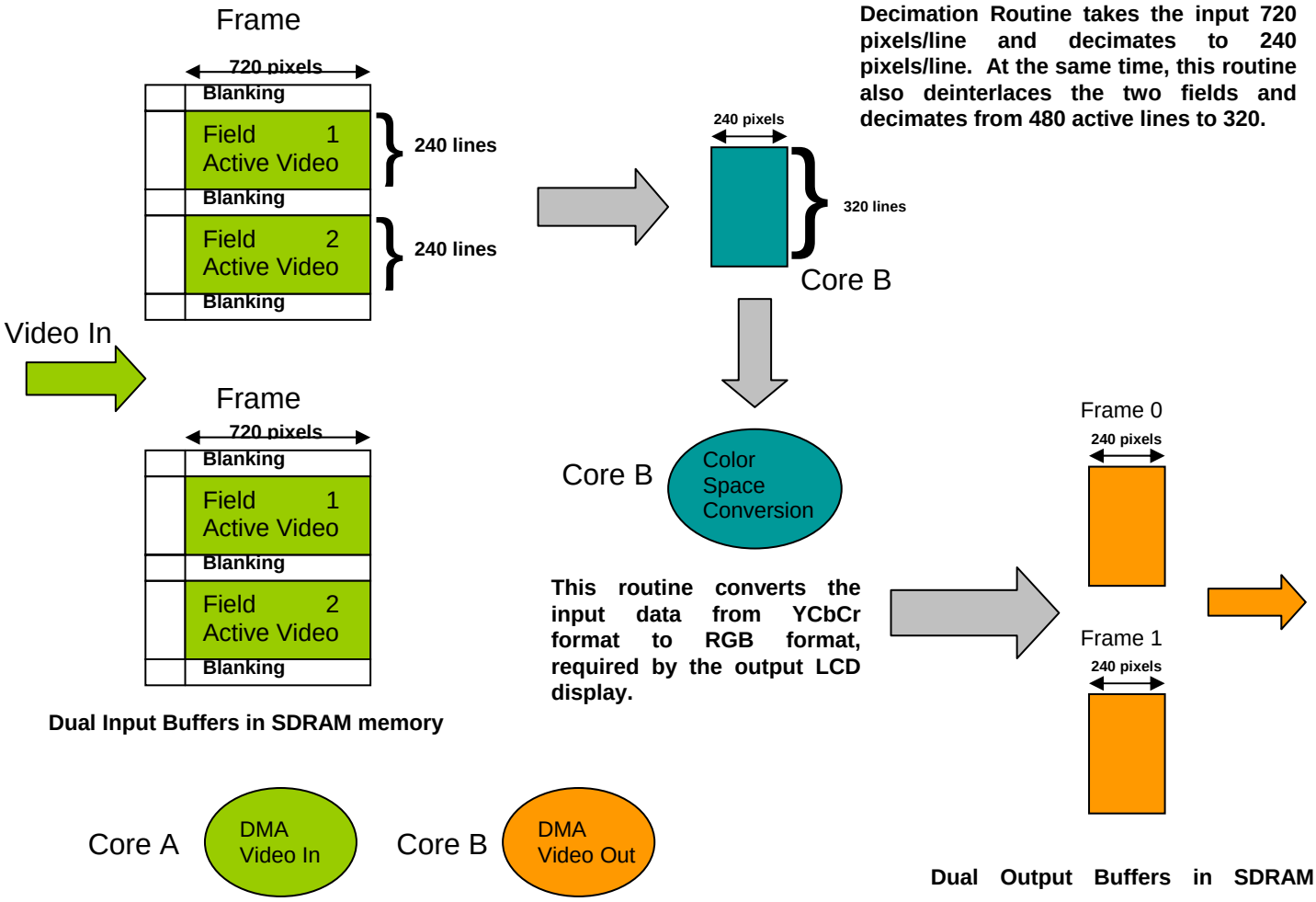

Figure 3. Software Flow Block Diagram

descript_buf_rx

0xFF80 0000	**0x0020**	lower addr of source buffer 1	**Descriptor 1**
0xFF80 0010	upper addr of source buffer 1	Config word for descriptor 1	
0xFF80 0020	**0x0000**	lower addr of source buffer 2	**Descriptor 2**
0xFF80 0030	upper addr of source buffer 2	Config word for descriptor 2	

Figure 4. Receive DMA Descriptors

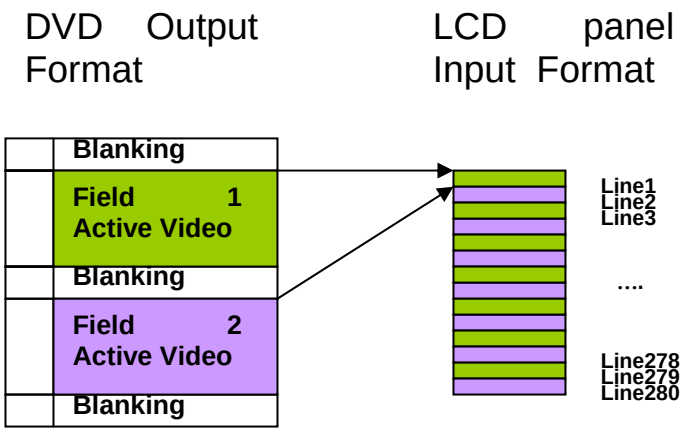

Figure 5. Interlaced vs. Non-Interlaced Data Arrangement

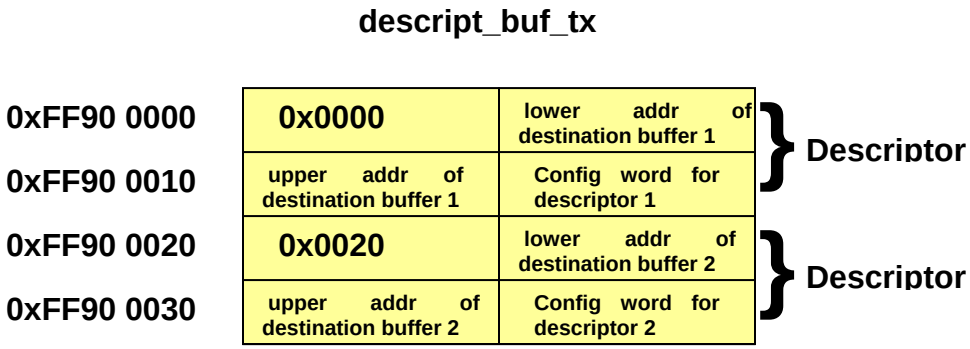

Figure 6. Transmit DMA Descriptors

Document History

Version	Description
Rev 2 – March 7, 2006 by C. Lam	Shortened title Corrected error in Figure 2
Rev 1 – December 9, 2004 by C. Lam	Initial Release as "Using the ADSP-BF561 Blackfin® Processor as a TFT-LCD Controller Eliminates Need for Timing ASIC"