A Method for Interfacing Digital Line Cameras to Field-Programmable Gate Array-Centric Data Processing Systems

by Thomas Kottke and Julian D. Fleniken

Approved for public release; distribution is unlimited.
NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.
A Method for Interfacing Digital Line Cameras to Field-Programmable Gate Array-Centric Data Processing Systems

Thomas Kottke and Julian D. Fleniken
Weapons and Materials Research Directorate, ARL

Approved for public release; distribution is unlimited.
This report presents a method for interfacing a digital line camera with a field-programmable gate array (FPGA)-centric data processing system. Although a specific camera and FPGA are used in this example, the presented techniques are general and can be readily applied to generic components. First, the operational details of the digital line camera are considered to highlight the types of data signals it produces and their formats. Next, the FPGA hardware is discussed, with an emphasis on what components and features are available to input and organize the data from the digital line camera. Finally, the software that configures and operates the FPGA is presented and discussed in detail.
Contents

List of Figures iv
List of Tables v
Acknowledgments vi
1. Introduction 1
2. Line Camera Operational Details 2
3. FPGA Data Acquisition Hardware 8
4. FPGA Data Acquisition Software 10
   4.1 Description of PIXEL_GRABBER Software Module ................................... 10
   4.2 Description of Phase Locked Loop Software Module ................................... 12
   4.3 Description of PIXEL_ORGANIZER Software Module ................................. 13
   4.4 Software Module Integration ....................................................................... 15
5. Summary 16
6. References 17

Appendix A. Code Listing for PIXEL_GRABBER Module 19

Appendix B. Code Listing for PIXEL_ORGANIZER Module 23

Distribution List 27
List of Figures

Figure 1. Digital camera signals and camera link transmission format........................................3
Figure 2. Digital line camera LVAL and DVAL synchronization signals........................................5
Figure 3. Close-up of digital line camera LVAL and DVAL synchronization signals.........................6
Figure 4. Digital line camera LVAL, DVAL, and data output signals. ...........................................7
Figure 5. Close-up of digital line camera LVAL, DVAL, and data output signals. .........................8
Figure 6. PIXEL_GRABBER graphical block diagram.................................................................11
Figure 7. Relationship of the video synchronization, trigger, and toggle signals for the
PIXEL_GRABBER software module.........................................................................................12
Figure 8. Phase-locked loop graphical block diagram.................................................................13
Figure 9. PIXEL_ORGANIZER graphical block diagram............................................................13
Figure 10. Relationship between video synchronization, trigger, and toggle signals for the
PIXEL_ORGANIZER software module. .....................................................................................15
Figure 11. Schematic circuit diagram illustrating integration of software module graphical
block diagrams to provide frame-grabber functionality. .........................................................16
List of Tables

Table 1. Names and locations of signals from line camera to FPGA integrated circuit. ...............10
Acknowledgments

The first author would like to thank the second author for the opportunity and support to work on this project. Also, the authors would like to thank Benjamin A. Breech (Advanced Weapons Concepts Branch, Weapons and Materials Research Directorate [WMRD], Barbara E. Ringers, (Applied Physics Branch, WMRD), and Mark Gatlin (Technical Publications Office) for reviewing and improving this manuscript.
1. Introduction

The Applied Physics Branch of the U.S. Army Research Laboratory’s Weapons and Materials Research Directorate is investigating sensor techniques for the detection and tracking of high-speed projectiles. The extremely restrictive timelines associated with these sensing activities limit both the amount of data that can be collected from the sensing elements and the amount of processing that can be performed on these data. Some potential measurement techniques utilize digital line cameras as the sensing elements. Unlike conventional digital cameras that use an array of light-sensitive elements, or pixels, to render a two-dimensional (2-D) image, line cameras make use of only a single line of pixels to produce a one-dimensional image. The field-of-view (FOV) from a line camera, when coupled with standard optics, is intrinsically limited to a plane containing both the linear array of pixels and the optical axis of the associated lens system. If the line camera’s FOV contains an object of interest, the object’s location is already known to exist in the plane of the FOV. By determining which pixel of the linear array is viewing the object of interest, the location can be further localized to a unique line contained in the line camera’s FOV. A second line camera with a FOV in the same plane as the first, but with a different optical axis, can determine a second line containing the object of interest. The point where these two lines in the plane intersect uniquely determines the object’s position, i.e., triangulation of the image’s position in space.

Of course, conventional digital cameras with 2-D arrays of pixels can also be used to triangulate the position of an object. The advantages of optical linear arrays vs. planar arrays in sensor systems is the reduction in the amount of data that must be collected and processed, and the associated increase in sensor speed and performance. Consider the example of a linear array with N pixels vs. a planar array with N · N pixels. To determine an object’s line-of-bearing at a single point in time, a planar array must acquire and process $N^2$ pixel’s worth of data. By contrast, a single linear array can measure the line-of-bearing to an object of interest at a single point in time by acquiring and processing only N pixel’s worth of data. Therefore, the ratio of data acquisition and processing times between a system with a planar array and a system with a linear array is $N$. With the availability of optical linear and planar arrays having pixel dimensions on the order of hundreds, or even thousands, the logistical burden of data acquisition and processing for linear arrays is much less than that for comparable systems with planar arrays. The inherent limitation of line cameras is the fact that a pair of these cameras with coplanar FOVs can provide only a single position measurement. Therefore, to determine a projectile’s trajectory, at least two pairs of line cameras must be used to measure the required two points to define a straight line. Conveniently, line cameras are relatively inexpensive, so the increased hardware burden of the linear camera trajectory measurement system is more than compensated by the reduced burden in data acquisition and processing time.
The sensor requirement for high-speed operation can be satisfied by pairing up the data output from an optical linear array with the processing capability of a field-programmable gate array (FPGA) (1). FPGAs are integrated circuits containing multiple logic blocks that can be physically interconnected to perform operations ranging from simple logic gates through complex combinational functions. This “hard-wired” connectivity allows multiple operations to be performed on the FPGA at the same time, with operating speeds limited only by component level switching times. This feature contrasts favorably with microprocessors and microcontrollers, where operations are software driven and limited by the speed of a program clock. Unlike traditional application-specific integrated circuits, which are manufactured to perform a single set of nonvarying operations, the FPGA is designed to be reconfigured by the user and thus, is far more flexible.

This report presents a method for interfacing a digital line camera with an FPGA-centric data processing system. Although a specific camera and FPGA are used in this example, the presented techniques are general and can be readily applied to generic components. Therefore, the information in this report is not only useful for documenting and reproducing the current smart sensor system, but can serve as a foundation for future enhancements. First, the operational details of the digital line camera will be considered to highlight the types of data signals it produces and their formats. Next, the FPGA hardware will be discussed with an emphasis on what components and features are available to input and organize the data from the digital line camera. Finally, the software that configures and operates the FPGA will be presented and discussed in detail.

2. Line Camera Operational Details

The specific line camera considered in this report is from the SmartBlue* series of charge couple device (CCD), line-scan cameras manufactured by Perkin Elmer Optoelectronics.‡ Specifically, part number SB0440CLG-011 is utilized with a resolution of 512 pixels (2). The SmartBlue line cameras were chosen because they are designed for applications where small size, low cost, and high scan rates are requirements (3). These qualities are consistent with smart sensor applications.

Digital cameras are generally applied using a third-party “frame-grabber” that serves as an interface between the digital camera and a personal computer (PC). Typical frame-grabbers provide paths for control signals to pass from a PC to the digital camera for configuration and

---

*SmartBlue is a registered trademark of PerkinElmer, Inc.
‡PerkinElmer Optoelectronics, Inc., 44370 Christy St., Fremont, CA 94538-3180.
control as well as additional paths from the camera to the PC to pass digital imagery data for display and long-term storage. Frame-grabbers also include generic driver software to execute these camera configuration and data transfer functions. The line camera application presented in this report does not utilize a third-party frame-grabber as an interface because generic frame-grabbers and their associated software do not offer the required flexibility and speed. In fact, part of the function of the FPGA data acquisition system is to serve as an application-specific frame-grabber. Therefore, a clear understanding of the function and format of the control signals input by the line camera, as well as the data signals output by the camera, is required for system integration.

SmartBlue cameras use a common high-speed video serial communication protocol known as Camera Link (4). A Channel Link chipset, consisting of a National Semiconductor* DS90CR287 transmitter and a DS90CR288A receiver, are used to transfer digital data at a maximum clock speed of 85 MHz. As illustrated in figure 1, this protocol allows 28 bits of complimentary metal-oxide semiconductor logic/transistor-transistor logic information and a strobe clock signal to be output from the digital camera.

---

*Texas Instruments, formerly National Semiconductor, 2900 Semiconductor Dr., Santa Clara, CA 95051.
These 28 channels comprise 24 pixel data lines, three video synchronization signals, and a single spare. In addition to these output signals, four discrete control signals are provided as input to the camera and two additional channels are provided for two-way, asynchronous serial communication.

These various camera input and output signals are not transferred along independent, parallel, single-ended lines. Rather, a collection of low-voltage differential signaling (LVDS) pairs transmit information as the difference between the voltages across a pair of wires. This arrangement provides high-speed, low-power communication with enhanced noise immunity (5). Four LVDS wire pairs are used to transmit the 28 channels of camera data output. This is accomplished by increasing the data clocking frequency by a factor of seven and serializing seven data signals through each LVDS pair. The strobe clock signal is transmitted through a dedicated LVDS pair. Similarly, each of the four discrete control signals and the two asynchronous serial communication channels has a dedicated LVDS pair. Thus, there are a total of 11 LVDS wire pairs. Four additional ground lines are included, for a total of 26 conductors in the standard Camera Link cable. After the signals pass through this cable, the LVDS voltages and frequencies must be converted back to their original format. Therefore, the hardware that serves as a frame-grabber must be capable of accepting Camera Link signals and converting them to single-ended, ground-referenced signals.

As mentioned, three video synchronization signals are provided by the digital camera to assist with image data transfer. These signals indicate which segment or section of image data is being transmitted. An entire 2-D array of pixel data from a conventional digital camera is considered to be a single data frame. This frame is composed of individual lines of data; in turn, the lines of data are composed of individual pixels of data. The frame valid (FVAL) synchronization line goes high, and remains high, whenever valid frame data is being transmitted. Similarly, the line valid (LVAL) synchronization line is high whenever a valid line of data is being transmitted within the data frame. Finally, the data valid (DVAL) synchronization line also goes high whenever valid pixel data is transmitted within a valid line of data. For digital line cameras, a frame of data consists of only a single line of data. Therefore, the FVAL signal is redundant for digital line cameras, and only the LVAL and DVAL signals need to be considered.

Figure 2 illustrates the LVAL and DVAL synchronization signals during the transfer of a single line of digital line camera data. On the left side of this figure, the rise of the upper dark-blue L_VAL signal denotes the start of valid line data transfer. The vertical cursor highlights this event. Similarly, the drop of the dark blue L_VAL signal, highlighted by cursor b on the right side of this figure, indicates the termination of this line data transfer. The SB0440CLG-011 SmartBlue line camera’s data rate is 40 MHz. Therefore, the data for a single pixel requires 25 nS or 0.025 μS to be transmitted. The time required to transfer all 512 pixels of data should therefore be:

$$ (512 \text{ pixels}) \cdot (0.025 \ \mu S / \text{ pixel}) = 12.8 \ \mu S. $$

(1)
The cursor data window in the upper right of figure 2 displays the temporal positions of the vertical cursors and confirms the duration of the $L_{VAL}$ synchronization signal to be 12.8 μS. During this time interval, the $D_{VAL}$ synchronization signal is oscillating at the 40-MHz data rate. At the time scale of figure 2, the individual oscillations of $D_{VAL}$ cannot be resolved, and the light-blue trace of this signal at the bottom of the figure is merely a blur. However, at the expanded time scale illustrated in figure 3, the first few valid $D_{VAL}$ signals are visible along with the onset of the $L_{VAL}$ signal. The vertical cursors in this figure confirm the data period of this line camera as 25 nS. Note that the $D_{VAL}$ signal continues to oscillate even when the $L_{VAL}$ signal is low. Therefore, both the $L_{VAL}$ and $D_{VAL}$ synchronization signals must be monitored to determine when valid pixel data is available for capture.
Figure 3. Close-up of digital line camera LVAL and DVAL synchronization signals.

Figure 4 displays an example of the activity on the line camera data lines during an active L_VAL period. In addition to the L_VAL and D_VAL signals at the top of this figure, a number of the data signals are shown with the most significant bit, D7, toward the top of the figure. The least significant bit, D0, is at the bottom of the figure. These are the signals that need to be captured by some sort of frame-grabber. These data were collected while the line camera viewed an image graded from high to low light intensity. At the beginning of the line scan, the camera was viewing the bright section of the image, as indicated by the high values of the more significant bits. Later in the scan, as the viewed image becomes darker, the measured pixel data values become smaller, as indicated by the drop-out of the more significant bits. Taking a closer look at the data lines, figure 5 shows a close-up of the data lines and their relation to the D_VAL synchronization signal. In this figure, the individual data lines can be seen to update only when the D_VAL synchronization is in an active high state.
The function and format of the rudimentary linear camera output signals has been discussed. Next, the FPGA-centric data processing system that provides, among other functions, the frame-grabber capability will be considered.

Figure 4. Digital line camera $LVAL$, $DVAL$, and data output signals.
3. FPGA Data Acquisition Hardware

FPGA integrated circuits are generally fabricated with hundreds of external pin connections (6). Because of the small size of these integrated circuits, there is not sufficient space around the edges of the package to fit this number of pins. Therefore, an array of solder bump connections are fabricated on the bottom surface of the integrated circuit that rests on the printed circuit board. This arrangement does not lend itself to easy prototype fabrication of FPGA components because when in place on the printed circuit board, the solder bump connections are not accessible (7). Therefore, FPGAs are generally applied using third-party development boards that have the FPGA premounted on a printed circuit board along with addition components and circuitry to facilitate communications, power management, expansion, etc. An example of such an FPGA development board is the Terasic* DE2-115 development and education board (8).

*Terasic Technology L.L.C., 3500 South Dupont Hwy., Dover, DE 19901.
This development board combines a high-end Altera® Cyclone IV EP4CE115 FPGA device (6) with a cornucopia of external circuitry, some of which will be discussed later. A primary factor in the decision to use this particular development board for line camera interfacing is the availability of a Camera Link receiver daughter card. As discussed, part of the function of the FPGA data acquisition system is to provide frame-grabber functionality. The Terasic CLR-HSMC Camera Link receiver daughter card offers a convenient method for connecting the line camera to the FPGA development board (9). An unfortunate consequence of using this daughter card is the resulting torturous signal path from the line camera to the FPGA integrated circuit.

As an example of this convoluted signal path, consider the journey of one of the line camera’s pixel data output signals, $D6$. Before leaving the line camera, this digital signal is serialized and converted to the LVDS format. The Camera Link protocol supports three different configurations—base, medium, and full. An appropriate Camera Link configuration is chosen based on the total number of digital data bits to be transferred. For this application, the standard base configuration is sufficient. The choice of Camera Link configuration determines which output ports and bits contain the digital data transferred from the line camera. For the base Camera Link configuration, the $D6$ data signal is assigned to port $A6$. The position of bit $A6$ in the serialized LVDS data stream is designated $TX27$. This designation is important because it is required to follow the $D6$ signal path after the camera signals pass through the Camera Link cable. As illustrated in figure 1, the line camera signals pass through the Camera Link cable in the LVDS format to the frame grabber, or in this case, the Camera Link receiver daughter card. On the daughter card, these LVDS signals are first routed to a National Semiconductor DS90CR288A LVDS link receiver that converts the LVDS back to ground-referenced, independent digital signals (10). The $TX27$ data bit in the LVDS data stream exits the DS90CR288A integrated circuit on pin 7 and is now designated $rx\_base27$. A schematic diagram supplied with the CLR-HSMC Camera Link receiver daughter card indicates the $rx\_base27$ signal is routed to pin 73 of the high-speed mezzanine connector (HSMC). The schematic diagram for the DE2-115 development and education board lists the signal leaving its HSMC at pin 73 as $TX\_D\_N[4]$. Finally, the DE2-115 User Manual (11) tabulates the Cyclone IV connections and reveals that the $TX\_D\_N[4]$ HSMC signal is connected to the FPGA through pin $K28$. For the reader’s convenience, the signal names and locations along the path from the line camera to the FPGA integrated circuit are listed in table 1.

---

*Altera Corporation, 101 Innovation Dr., San Jose, CA 95134.*
Table 1. Names and locations of signals from line camera to FPGA integrated circuit.

<table>
<thead>
<tr>
<th>Camera Signal Name</th>
<th>Camera Link Allocation</th>
<th>LVDS Designation</th>
<th>DS90CR288A Pin-Out</th>
<th>HSMC Pin-Out</th>
<th>DE2-115 Signal Name</th>
<th>FPGA Pin Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>LVAL</td>
<td>LVAL</td>
<td>TX/RX 24</td>
<td>3</td>
<td>83</td>
<td>TX D P[6]</td>
<td>K21</td>
</tr>
<tr>
<td>DVAL</td>
<td>DVAL</td>
<td>TX/RX 26</td>
<td>6</td>
<td>77</td>
<td>TX D P[5]</td>
<td>M27</td>
</tr>
<tr>
<td>STROBE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D0</td>
<td>A0</td>
<td>TX/RX 0</td>
<td>27</td>
<td>39</td>
<td>CLKOUT0</td>
<td>AD28</td>
</tr>
<tr>
<td>D1</td>
<td>A1</td>
<td>TX/RX1</td>
<td>29</td>
<td>40</td>
<td>CLKIN0</td>
<td>AH15</td>
</tr>
<tr>
<td>D2</td>
<td>A2</td>
<td>TX/RX 2</td>
<td>30</td>
<td>41</td>
<td>D[0]</td>
<td>AE26</td>
</tr>
<tr>
<td>D3</td>
<td>A3</td>
<td>TX/RX 3</td>
<td>32</td>
<td>43</td>
<td>D[1]</td>
<td>AE27</td>
</tr>
<tr>
<td>D4</td>
<td>A4</td>
<td>TX/RX 4</td>
<td>33</td>
<td>47</td>
<td>TX D P[0]</td>
<td>D27</td>
</tr>
<tr>
<td>D5</td>
<td>A5</td>
<td>TX/RX 6</td>
<td>35</td>
<td>53</td>
<td>TX D P[1]</td>
<td>E27</td>
</tr>
<tr>
<td>D7</td>
<td>A7</td>
<td>TX/RX 5</td>
<td>34</td>
<td>49</td>
<td>TX D N[0]</td>
<td>D28</td>
</tr>
<tr>
<td>D8</td>
<td>B0</td>
<td>TX/RX 7</td>
<td>37</td>
<td>42</td>
<td>D[1]</td>
<td>AE28</td>
</tr>
<tr>
<td>D9</td>
<td>B1</td>
<td>TX/RX 8</td>
<td>38</td>
<td>44</td>
<td>D[3]</td>
<td>AF27</td>
</tr>
<tr>
<td>D10</td>
<td>B2</td>
<td>TX/RX9</td>
<td>39</td>
<td>48</td>
<td>RX D P[0]</td>
<td>F24</td>
</tr>
<tr>
<td>D11</td>
<td>B3</td>
<td>TX/RX 12</td>
<td>43</td>
<td>54</td>
<td>RX D P[1]</td>
<td>D26</td>
</tr>
</tbody>
</table>

4. FPGA Data Acquisition Software

Software for the Cyclone IV FPGA on the DE2-115 application board can be developed using Altera’s Quartus II programmable logic device design package. This integrated development environment supports a variety of hardware design languages (HDL) for the design and description of electronic circuits, including Verilog (J2), very-high-speed integrated circuits hardware HDL (J3), and a graphical block design editor (J4). Following the electronic design process, Quartus can be used to compile the design, download the design to the target application, and perform timing analysis. Quartus II is available in two versions—a web edition that can be downloaded for free and a subscription edition that requires the purchase of a license for full functionality (J5). Conveniently, the low-cost Cyclone family of FPGAs is fully supported by the web edition. Software will now be presented that can be used to acquire data from the digital line camera and transfer these data to the Cyclone IV FPGA.

4.1 Description of PIXEL_GRABBER Software Module

The first software component to be presented is the PIXEL_GRABBER module. As the name suggests, this module is designed to acquire pixel data values from a digital line camera. Figure 6 illustrates the graphical block diagram with inputs entering on the left and outputs exiting on the right. A listing of the associated Verilog HDL code used to generate this graphical block diagram is presented in appendix A. The digital line camera’s DVAL, LVAL, and pixel data
signals are routed to the \textit{D\_VAL\_IN}, \textit{L\_VAL\_IN}, and \textit{PIXEL\_IN[11..0]} inputs, respectively. Copies of the \textit{DVAL} and \textit{LVAL} signals are output as \textit{D\_VAL\_OUT} and \textit{L\_VAL\_OUT} to provide subsequent modules with copies of these video synchronization signals that include any inherent delays introduced by the \textit{PIXEL\_GRABBER} module. Data acquisition is event-driven by a rising edge on the signal supplied to the \textit{CLOCK\_IN} input. For proper operation, the \textit{CLOCK\_IN} signal must have a frequency twice that of the line camera’s data rate, with a possible phase shift to compensate for signal transmission times. A convenient way to generate this \textit{CLOCK\_IN} signal is to use one of the phase look loop (PLL) modules on the FPGA. The generation and details of this triggering signal will be presented in the next section. For now, it is sufficient to realize that due to the doubling of the frequency, the PLL-generated daughter signal supplied to the \textit{CLOCK\_IN} input will transition high once during both the high and low periods of the \textit{DVAL} signal, as illustrated in figure 7. If the \textit{LVAL} signal is low when the \textit{CLOCK\_IN} signal triggers this module, the line camera is not outputting a line of pixel data, so there is no camera data to be grabbed and the \textit{PIXEL\_READY} output is simply recleared. This condition is denoted as events A in figure 7. If the \textit{LVAL} signal is high when the \textit{CLOCK\_IN} signal triggers this module, the \textit{DVAL} signal is considered. If the \textit{DVAL} signal is high under this condition, there is new pixel data available. The current values at the \textit{PIXEL\_IN[11..0]} inputs are latched to the \textit{PIXEL\_OUT[11..0]} outputs, and the \textit{PIXEL\_READY} output line is pulled high to indicate new pixel data is available. Figure 7 denotes these events as \textit{B}. Conversely, if the \textit{DVAL} signal is low under this condition, there is no new pixel data available, and the \textit{PIXEL\_READY} output line is cleared low as highlighted by events \textit{C} on figure 7.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{pixel_grabber_diagram.png}
\caption{\textit{PIXEL\_GRABBER} graphical block diagram.}
\end{figure}
4.2 Description of Phase Locked Loop Software Module

The PLL driver software is conveniently provided as a canned module with the Quartus II application design package. A user-friendly graphical interface walks the user through a configuration process to define input sources, output frequencies, phase shifts, duty cycles, etc. This module then generates a graphic block design and an HDL source file. The graphic block design can then be used in schematic electronic circuit design and the source code can be compiled along with other user generated module code. Figure 8 shows the PLL block design for a PLL with a 40-MHz input clock frequency and an 80-MHz output clock signal with a 75° phase shift and a 50% duty cycle.
4.3 Description of PIXEL_ORGANIZER Software Module

The previously discussed PIXEL_GRABBER software module allows pixel data values to be acquired from a digital line camera. It is necessary for these pixel values to be associated with the specific pixel that generated the data. This functionality is provided by the PIXEL_ORGANIZER software module. Figure 9 illustrates the graphical block diagram for this module, and a listing of the associated Verilog HDL code used to generate this block diagram is presented in appendix B. A copy of the PLL module output signal c0, output as CLOCK_OUT from the PIXEL_GRABBER software module, is routed to the CLOCK_IN input of the PIXEL_ORGANIZER module. Similarly, a copy of the line valid signal L_VAL_OUT and the PIXEL_READY toggle from PIXEL_GRABBER are routed to NEW_LINE and NEW_PIXEL inputs, respectively, of the PIXEL_ORGANIZER module. Additionally, the grabbed pixel value PIXEL_OUT[11..0] from PIXEL_GRABBER is also input to PIXEL_ORGANIZER at inputs PIXEL_VALUE[11..0].
Operation of the PIXEL_ORGANIZER software module is event-driven by a falling edge on the signal supplied to the \textit{CLOCK\_IN} input. The negative edge of the \textit{CLOCK\_IN} signal is used because it falls squarely in both the high and low segments of the \textit{NEW\_PIXEL} signal, as illustrated in figure 10. If the \textit{NEW\_LINE} input signal is high during the falling edge of \textit{CLOCK\_IN}, a line-status flag internal to the software module is polled. If this line-status flag is low, the \textit{NEW\_LINE} input signal has just recently gone high, as indicated by position \textit{A} in figure 10. For this condition, the line-status flag is set high, the internal pixel location index value is set to a prerollover binary value of 111111111, and the \textit{PIXEL\_ORG} toggle output is redundantly cleared. Next, the state of the \textit{NEW\_PIXEL} line is considered. If the \textit{NEW\_PIXEL} line is low, new pixel data is not currently available, so the \textit{PIXEL\_ORG} toggle is cleared and the pixel location index value is incremented for the next available pixel data value. For the situation indicated by \textit{A} in figure 10, the prerollover binary value of 111111111 will, in fact, roll over to a value of 000000000 in preparation for the input of the first pixel value of the new line of data. For subsequent events where \textit{NEW\_LINE} is high and \textit{NEW\_PIXEL} is low, designated as \textit{B} in figure 10, the \textit{PIXEL\_ORG} toggle is cleared and the pixel location index value is simply incremented. If the \textit{CLOCK\_IN} signal falls when \textit{NEW\_LINE} and \textit{NEW\_PIXEL} are both high, as shown by \textit{C} in figure 10, then new pixel data is available. The new pixel value is passed through from the \textit{PIXEL\_VALUE[11..0]} input to the \textit{PIXEL\_VAL[11..0]} output bus, the current internal pixel location index value is output on the \textit{PIXEL\_LOC[8..0]} output bus, and the \textit{PIXEL\_ORG} output toggle line is pulled high to notify subsequent modules.

If the \textit{NEW\_LINE} signal is low during an event, the internal line-status flag is also polled. If the line-status flag is high, this indicates the \textit{NEW\_LINE} input signal has just fallen (not shown in figure 10). For this circumstance, the line-status flag is cleared, the pixel organizer toggle is cleared, and the latched pixel location value is also cleared. If the line-status line is low, as shown by \textit{D} in figure 10, no action is taken.

The PIXEL\_ORGANIZER software module outputs the pixel data value, the pixel identity, and a toggle signal to denote when this information is available. Subsequent software modules can use these signals to display, process, and evaluate the data provided by the digital line camera.
Figure 10. Relationship between video synchronization, trigger, and toggle signals for the PIXEL_ORGANIZER software module.

4.4 Software Module Integration

Figure 11 illustrates how the previously discussed software modules can be interconnected and configured to provide frame-grabber functionality. The various digital line camera signals are routed to the appropriate FPGA input ports using the assignments provided in table 1. Of particular interest is the use of the line camera STROBE signal to drive the PLL and D_VAL_IN input of the PIXEL_GRABBER module rather than the DVAL signal. Many manufacturers of high-speed digital cameras default to using the STROBE signal as a DVAL signal because the STROBE signal defines the fastest possible data transfer rate. This is the case for the SmartBlue CCD line scan cameras.
5. Summary

A method for interfacing a digital line camera with an FPGA-centric data processing system has been presented. Although a specific camera and FPGA have been used in this example, the presented techniques are general and can be readily applied to generic components. First, the operational details of the digital line camera were considered to highlight the types of data signals that are produced and their formats. Next, the FPGA hardware was discussed with an emphasis on what components and features are available to input and organize the data from the digital line camera. Finally, the software that configures and operates the FPGA was presented and discussed in detail.
6. References


3. SmartBlue CCD Line Scan Camera Series Camera Instruction Manual; Publication no. 055-0467-MAN; PerkinElmer Optoelectronics: Fremont, CA.


INTENTIONALLY LEFT BLANK.
Appendix A. Code Listing for PIXEL_GRABBER Module

This appendix appears in its original form, without editorial change.
This module latches PIXEL data from a line camera. The process is event driven by the rising edge of CLOCK_IN which is assumed to be a PLL generated daughter of the camera STROBE clock signal with twice the frequency, and possibly a phase shift to compensate for PLL delay. Thus, the CLOCK_IN signal will go high during both the high and low segments of the DVAL input clock signal. If the L_VAL_IN line data valid signal from the camera is low during an event, there is no new PIXEL data to be grabbed and the PIXEL READY output toggle line is simply recleared. If the L_VAL_IN line is high and the D_VAL_IN pixel data valid input signal from the camera is high, new PIXEL data is available which is transferred to the PIXEL_OUT bus and the PIXEL READY toggle line is set high. If the L_VAL_IN line is high but the D_VAL_IN line is low, there is no new pixel data available so the PIXEL READY toggle line is cleared.

module PIXEL_GRABBER(
input wire CLOCK_IN , //event clock
input wire D_VAL_IN , //camera pixel valid signal
input wire L_VAL_IN , //camera line valid signal
input wire[11:0] PIXEL_IN , //camera pixel data
output wire CLOCK_OUT , //pass through copy of event clock
output wire D_VAL_OUT , //pass through copy of pixel valid
output wire L_VAL_OUT , //pass through copy of line valid
output reg PIXEL_READY , //new pixel value ready toggle
output reg [11:0] PIXEL_OUT , //new pixel value data
);

/*
MODULE NAME: PIXEL_GRABBER
MODULE LANGUAGE: Verilog HDL
MODULE TARGET: Altera Cyclone IV E on Terasic DE2-115 w/ CLR_HSMC
MODULE AUTHOR: Kottke
MODULE REV. DATE: Feb 2012

MODULE INPUTS:
CLOCK_IN event signal
D_VAL_IN line camera data valid signal
L_VAL_IN line camera line valid signal
PIXEL_IN() 12 bit wide line camera pixel data bus

MODULE OUTPUTS:
CLOCK_OUT event signal copy
D_VAL_OUT line camera data valid signal copy
L_VAL_OUT line camera line valid signal copy
PIXEL_READY new pixel value ready toggle
PIXEL_OUT() latched 12 bit line camera pixel data
*/
initial
begin
  PIXEL_READY <= 0; //initialize pixel ready toggle as clear
end

assign  CLOCK_OUT = CLOCK_IN; //assign pass through signals
assign  D_VAL_OUT = D_VAL_IN;
assign  L_VAL_OUT = L_VAL_IN;

always @ (posedge CLOCK_IN) //execute on positive edge of CLOCK_IN
begin
  if (L_VAL_IN == 1) //if camera line valid signal is high...
    begin
      if (D_VAL_IN == 1) //if camera pixel valid signal is high...
        begin
          PIXEL_OUT <= PIXEL_IN; //load pixel output data
          PIXEL_READY <= 1; //set pixel value ready toggle
        end
      else //if camera DATA VALID line is low...
        begin
          PIXEL_READY <= 0; //clear pixel value ready toggle
        end
    end
  else //if camera LINE VALID line is low...
    begin
      PIXEL_READY <= 0; //reclear pixel value ready toggle
    end
end
endmodule
Appendix B. Code Listing for PIXEL_ORGANIZER Module

This appendix appears in its original form, without editorial change.
This module organizes the pixel values from the PIXEL_GRABBER module by pairing each new pixel value with an appropriate pixel location value. The process is event driven by the falling edge of the CLOCK_IN input which is supplied by the pass-through clock signal from the PIXEL_GRABBER module. The negative edge of the CLOCK_IN signal is used because it falls squarely in both the high and low segments of the PIXEL_READY signal from the PIXEL_GRABBER module. If the NEW_LINE input signal is high, the line_status flag is checked to see if it is low which would indicates this is the first valid data for a new NEW_LINE signal. If this is the case, then the line_status flag is set, the pixel location index value is set to a pre-rollover value, and the PIXEL_ORG output is redundantly recleared. Next, the NEW_PIXEL value is checked. If the value is high it indicates new pixel data is available. The new pixel value is passed through to the PIXEL_VAL output bus, the current pixel location index value is output on the PIXEL_LOC output bus, and the PIXEL_ORG toggle line is pulled high. If the NEW_PIXEL value is low, new pixel data is not currently available so the PIXEL_READY toggle is cleared and the pixel location index value is incremented for the next available pixel value. If the NEW_LINE input signal is low, then the line status flag is considered. If the line status flag is high this indicates the NEW_LINE input signal has just fallen. For this circumstance the line status flag is cleared, the pixel organizer toggle is cleared, and the latched pixel location value is also cleared.

module PIXEL_ORGANIZER(
    input wire CLOCK_IN, //event clock
    input wire NEW_LINE,  //line valid input
    input wire NEW_PIXEL  //new pixel value ready input
);
input    wire[11:0]   PIXEL_VALUE;   //latched pixel value bus
output   wire        CLOCK_OUT;    //pass through copy of event clock
output   reg         PIXEL_ORG;    //pixel organizer ready toggle
output   reg [11:0]  PIXEL_VAL;    //current pixel value
output   reg [8:0]   PIXEL_LOC;    //current pixel location
);
    reg [8:0]   i;
    reg         line_status;        //position index

initial
begin
    line_status = 1'b0;           //line valid signal status
    PIXEL_ORG = 1'b0;            //clear organizer pixel ready toggle
    i = 9'b0000000000;          //set pixel location index value
end
assign CLOCK_OUT = CLOCK_IN;    //assign event clock pass through

always @(negedge CLOCK_IN)
begin
    if (NEW_LINE == 1)       //if camera line valid is high...
        begin
            if (line_status == 1'b0)  //if this is a new line valid...
                begin
                    line_status = 1'b1;    //set line status flag as true
                    i = 9'b111111111;      //set pixel location to roll-over
                    PIXEL_ORG = 1'b0;     //clear organizer pixel ready toggle
                end
            if (NEW_PIXEL == 1'b1)   //if new pixel data is available...
                begin
                    PIXEL_VAL <= PIXEL_VALUE;  //pass through latched pixel value
                    PIXEL_LOC <= i;        //latch current pixel location
                    PIXEL_ORG = 1'b1;      //set pixel organizer toggle
                end
            else
                begin
                    i = i + 1'b1;          //increment pixel location index
                    PIXEL_ORG = 1'b0;     //clear pixel organizer toggle
                end
        end
    else
        begin
            if (line_status == 1'b1)  //if line valid signal is not high...
                begin
                    //if line valid signal just ended...
                end
        end
end
begin
    line_status = 1'b0;  // clear camera line valid flag
    PIXEL_ORG = 1'b0;   // clear pixel organizer toggle
    PIXEL_LOC = 9'b000000000;  // clear pixel location latch
end
<table>
<thead>
<tr>
<th>NO. OF COPIES</th>
<th>ORGANIZATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DEFENSE TECHNICAL INFORMATION CTR (PDF only) DTIC OCA 8725 JOHN J KINGMAN RD STE 0944 FORT BELVOIR VA 22060-6218</td>
</tr>
<tr>
<td>1</td>
<td>DIRECTOR US ARMY RESEARCH LAB IMAL HRA 2800 POWDER MILL RD ADELPHI MD 20783-1197</td>
</tr>
<tr>
<td>1</td>
<td>DIRECTOR US ARMY RESEARCH LAB RDRL CIO LL 2800 POWDER MILL RD ADELPHI MD 20783-1197</td>
</tr>
</tbody>
</table>
ABERDEEN PROVING GROUND

31  DIR USARL
(30 HC RDRL WML A
1 CD)  B BREECH (CD only)
RDRL WMP A
  J BALL
  P BERNING
  I BILOIU
  M COPPINGER
  J FLENIKEN (5 CPS)
  C HUMMER
  T KOTTK (5 CPS)
  A PORWITZKY
  J POWELL
  B RINGERS
  G THOMSON
  W UHLIG
  T VALENZUELA
  C WOLFE
RDRL WMP B
  C HOPPEL
RDRL WMP C
  T BJERKE
  B LEAVY
RDRL WMP D
  J RUNYEON
  R MUDD
RDRL WMP E
  P SWOBODA
RDRL WMP F
  N GNIAZDOWSKI
RDRL WMP G
  N ELDREDGE

28