Coreco Imaging, Inc

PC-CamLink™

User's Manual

405-00009-00

Rev 01

July 30, 2001

CORECO iMAGING

PC-CamLink™ User's Manual
Document Number 405-00009-00
Revision 01; July 30, 2001

Coreco Imaging, Inc
55 Middlesex Turnpike
Bedford, MA 01730-1421
+1.781.275.2700
FAX: +1.781.275–9590
Email: info@imaging.com
http://www.imaging.com

EMC/EMI Precautions

This product has been tested and found to comply with the FCC limits for a class A device. The intended use of this product is in a commercial, industrial work environment. Operation of this product within a domestic environment may cause radio interference, in which case the user may be required to take adequate measures.

This product has been tested and found to comply with the European Community Directive for EMC, 89/336/EEC. In order to maintain compliance with the EMC Directive this product must be installed in an EEC EMC compliant chassis that has a CE marking. Additionally, all cables used to connect the computer and peripherals must be shielded, grounded and securely fastened.

Table of Contents

Part 1. Introduction

Part 2. Hardware

List of Tables

List of Figures

Part 1
Introduction

Welcome to the PC-CamLink User's Guide

The purpose of this manual is to help you obtain the maximum performance from the PC-CamLink. This guide explores how to use key hardware features and software tools which make This manual covers the installation, setup and use of the PC-CamLink hardware and IFC-SDK (software development kit). This manual is meant to explain most of what you need to know about PC-CamLink to take advantage of the features designed into it.

After installing your hardware and software, ***please take the time to read the Software Release Notes and any "readme" files*** for the latest information available. The Software Release Notes are available in electronic form on the CD-ROM, and on your system after you have installed the ITEX software.

Introduction and Overview

PC-CamLink is a PCI-bus "plug-n-play" board that provides image capture for Camera Link™ digital camera applications. PC-CamLink offers efficient Camera Link™ cabling for camera input, and general-purpose I/O for controlling or monitoring external events. PC-CamLink supports the ***Base Mode*** configuration of the Camera Link interface. This means there is one Camera Link connector supporting one camera input up to 24 bits. The PC-CamLink supports 8, 10, 12, 14, and 16 bit monochrome and 24-bit RGB color input.

One of the most important features of PC-CamLink is the ability to reformat multi-tap data into normal raster scan images in real time.

Installation and Setup

This section describes how to quickly install PC-CamLink board and IFC software, run the diagnostic software, connect a camera and start acquiring images.

NOTE *Please install the IFC Software before installing the PC-CamLink board.*

System Requirements for using PC-CamLink

- PCI-bus computer with Pentium processor or equivalent, and at least 32MB
- Windows 95, 98, ME, 2000 or Windows NT 4.0
- An open half length 32-bit PCI-bus expansion slot.

Optional: To support hardware overlays, the VGA video card should have one of the following chipsets: S3 Virge, Intel 740 or ATI 3D RAGE / RAGE PRO.

Software Installation

IMPORTANT *Please read BEFORE installing the IFC-SDK:*

- *Uninstall any previous versions of the IFC-SDK. To do this, run the Windows "Add/Remove Program Utility" located in the Control Panel folder on your system. The Control Panel folder can be found by selecting:* Start - Settings - Control Panel *from the Windows task bar.*
- *Installing the IFC-SDK under Windows NT requires Administrative Privileges.*
- *Make sure that the Microsoft C/C++ or Visual Basic compiler, version 5.0 or later, has already been installed, before installing the IFC-SDK, so that IFC extension help will be installed and enabled properly.*

1. Before running the Setup program, make sure all other Windows applications have been closed. To install the ITEX SDK software, place the CD in the appropriate drive to start the installation automatically. If Auto-Start is not enabled, manually start the install program by selecting "Run" from the Windows Start Menu and type:

   <*CD drive* >**:\SETUP.EXE**

2. Click-on "Install Software".
3. Select the appropriate Operating System (this should automatically be selected) and software (IFC/Win32) then click "Start Install" to begin installation.
4. At the "Welcome" screen, click "Next". Please take a moment to read the Software License Agreement, click "Yes" to continue.
5. Enter User Information, then click "Next" to continue to the Destination screen. Choose a location on the local system hard drive to install the IFC. Note: This destination is referred to as the "*IFCroot*" directory. Click "Next" to continue.
6. Select Installation Type. "Typical" is recommended and will install all files. To install software on a Run-time system (loads only the DLLs, Device Driver, Diagnostics and Camera Configurator), select "Compact". Click "Next" to continue.
7. Select the product documentation and examples you wish to install, and click "Next".
8. Enter a Program Folder name, and click "Next".
9. You have the option of installing the Acrobat Reader (for viewing and printing documentation supplied in PDF format), choose "Yes" or "No".
   For Windows 95 users, DirectX Media Components are installed at this point. Setup is done. Click "Finish" to restart your computer. Shut down your computer and install the PCVision hardware.

   For Windows 98 users, DirectX Media Components are installed at this point.
10. When "Setup" has finished copying files, you will need to reboot your computer to complete the installation. Click "Finish" to restart your computer.

Environment Changes

The following additions have been made to your system:

- Windows 98 driver location: c:\windows\system\itipci.vxd
- Windows NT driver location: c:\winnt\system32\drivers\mvcntp.sys
- The IFC configuration file path has been added to your computer environment: IFCCNF=C:*IFCROOT*\config.

  Under Windows 98 this is in your Autoexec.bat file. Under Windows NT, "Right Click" on "My Computer" then select "Properties - Environment".
- The IFC Program Folder now contains all the example programs, IFC utilities, documentation, and release notes.

NOTE *IFCROOT refers to the destination directory where IFC was installed for example c:\IFC52.*

- Help file versions of the PC-CamLink Manuals are installed on your hard disk. Help files for other manuals are available on the CD-ROM in the DOC directory. Entire chapters can be printed from the Content tab.
- Adobe Acrobat versions of the PC-CamLink and other manuals are available on the CD-ROM in the DOC\PDFDOC directory for printing hard copy.

NOTE *If you update or reinstall your C++ compiler, the link to extension help is lost. Uninstall and reinstall the IFC software to reestablish this link. This is faster than copying and editing the IFCSW.CNT file.*

Configuration Jumpers

The jumpers on the PC-CamLink board control the power-up values for the outputs, and select the input resistor for the Opto-coupled inputs. Leaving the outputs undefined during power-up can adversely affect external equipment connected to the OPTO-22 parallel port, or the the TTL, Differential and Opto-coupled outputs. You should set these jumpers for the preferred power-up conditions for any equipment connected to the PC-CamLink, to prevent damaging or hazardous conditions in the attached equipment.

The following table defines the functions of all jumpers. Figure 1 illustrates the jumper positions 1–2 and 2–3. A square pad identifies pin 1. The numbers 1,2,3 are not on the board.

Jumper	Pins Controlled	Position	Function
JP1	OPTO_IN(1–0)	1–2	Opto Input 0 Input Resistor 475 ohms (5 Volt operation)
		2–3	Opto Input 0 Input Resistor 2.7 K ohms (24 Volt operation)
JP2	DIFF_OUT(1–0) and CC4	1–2	Differential Outputs and CC4 Tri-Stated (disabled) on power-up
		2–3	Differential Outputs and CC4 Enabled on power-up
JP3	OPTO_IN(1–0)	1–2	Opto Input 1 Input Resistor 475 ohms (5 Volt operation)
		2–3	Opto Input 1 Input Resistor 2.7 K ohms (24 Volt operation)
JP4	TTL_OUT(2–0)	1–2	TTL Outputs high (1) on power-up
		2–3	TTL Outputs low (0) on power-up
JP5	OUT(7–0)	1–2	Parallel Outputs high (1) on power-up
		2–3	Parallel Outputs low (0) on power-up
JP6	OUT(7–0)	1–2	Parallel Outputs tri-stated (disabled) on power-up
		2–3	Parallel Output enabled on power-up
JP7	TTL_OUT(2–0)	1–2	TTL Outputs tri-stated (disabled) on power-up
		2–3	TTL Outputs enabled on power-up
JP8	OPTO_OUT(1–0)	1–2	Opto Outputs high (1) on power-up
		2–3	Opto outputs low (0) on power-up
JP9	DIFF_OUT(1–0)	1–2	Differential Outputs high (1) on power-up
		2–3	Differential Outputs low (0) on power-up

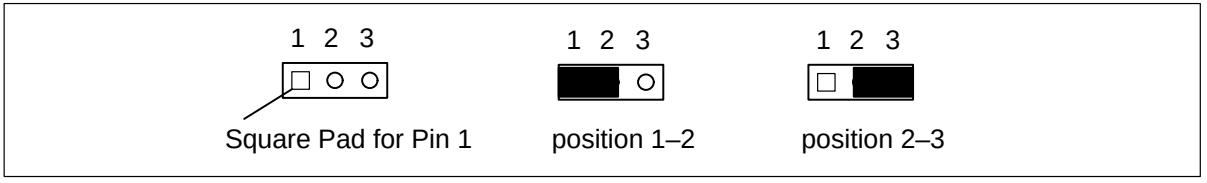


Figure 1. Jumper Positions

Factory Default Configuration

The following table gives the factory default settings for all jumpers on the PC-CamLink board. All outputs are disabled or tri-stated (except the Opto outputs) during power-up, and set to default to 1 if enabled. The 5 Volt 475 ohm resistor is chosen for the two Opto-coupled inputs. Figure 2 shows this configuration.

Jumper	Position	Pins Controlled	Function
JP1	1–2	OPTO_IN(1–0)	Opto Input 0 Input Resistor 475 ohms (5 Volt operation)
JP2	1–2	DIFF_OUT(1–0)	Differential Outputs and CC4 Tri-Stated (disabled) on power-up
JP3	1–2	OPTO_IN(1–0)	Opto Input 1 Input Resistor 475 ohms (5 Volt operation)
JP4	1–2	TTL_OUT(2–0)	TTL Outputs high (1) on power-up
JP5	1–2	OUT(7–0)	Parallel Outputs high (1) on power-up
JP6	1–2	OUT(7–0)	Parallel Outputs tri-stated (disabled) on power-up
JP7	1–2	TTL_OUT(2–0)	TTL Outputs tri-stated (disabled) on power-up
JP8	1–2	OPTO_OUT(1–0)	Opto Outputs high (1) on power-up
JP9	1–2	DIFF_OUT(1–0)	Differential Outputs high (1) on power-up

Use the worksheet at the end of this manual to document your changes to the factory configuration.

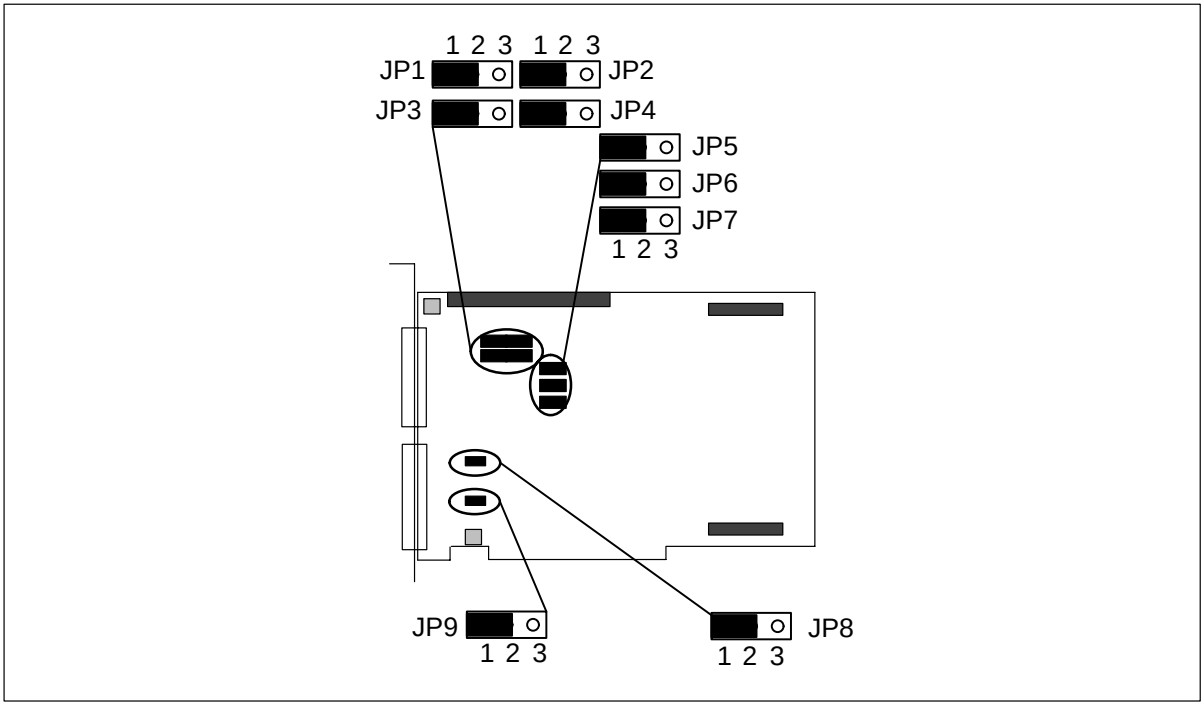


Figure 2. PC-CamLink Factory Default Configuration

Hardware Installation

Before installing the board, turn off power to the computer and peripherals. Discharge static electricity from your body by touching a metal part of the computer chassis (or use a static protection strap if available).

NOTE *Hardware Jumpers on the PC-CamLink board set the default power-up state of the Parallel I/O port (PIO) and the TTL, Differential, Opto-coupled outputs on the I/O connector. Refer to Appendix B for information on these jumpers.*

1. Remove the computer chassis cover. Locate an empty PCI-bus slot. PC-CamLink may fit into a partially obstructed slot. Remove the back-panel slot cover if necessary, for the PC-CamLink connector to pass through.

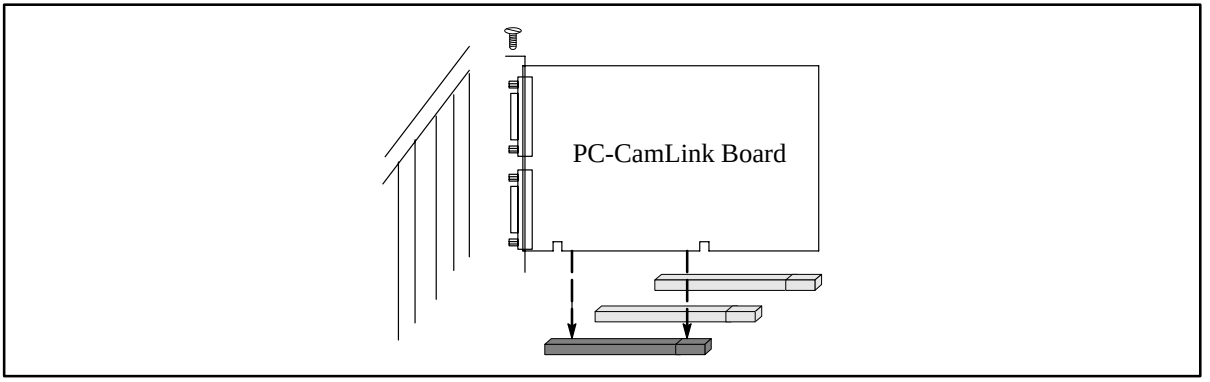


Figure 3. Install the PC-CamLink in the Computer

2. Slide the PC-CamLink board carefully into the slot so the end-bracket slips properly into the filler bracket. Make sure the board is pressed firmly into the connector, and does not touch other boards, components, or heat sinks. Install the end-bracket screw.
3. If you have the I/O ribbon cable, attach it to the 50-pin header connector. *Replace the chassis cover, but do not plug the computer in yet.*
4. Connect all other peripherals to your computer. *Do not connect the camera and cable yet. The Diagnostic software must be run with no camera connected.* Power up and boot your system.

- *Under Windows 95 or 98, the Plug-and-Play Configuration Manager detects new hardware after installing PC-CamLink. If you're prompted for a "manufacturer's CD or diskette" to install the IFC device driver, place the CD-ROM in the appropriate drive and follow the instructions for installing the driver. The driver is located in the root directory on the CD. Windows NT should boot normally.*

Run the Diagnostic Utility

After installing the PC-CamLink for the first time, it is a good idea to run the diagnostic utility to verify the installation is correct and the board is functioning properly. The diagnostic tests must be run without a camera connected.

This program will auto-detect the number of boards you have plugged into the system, initializes them, then performs some functional and memory tests.

NOTE *Before running this tool, shut down any other Windows applications that are running to prevent "timeouts" from occurring causing the program to generate false failures.*

NOTE *Run the diagnostic test with no camera connected. The program tests the internal timebase, and a camera that supplies timing could cause false failures.*
Do not connect or disconnect cameras with the computer power on. Because the connectors provide 12 Volt power, the camera or PCVision could be damaged by "hot plugging".

1. Double-click on the Diagnostics icon in the IFC program group.
2. The program will report back any hardware failures it finds for each module, in a ".DLG" file called "DIAG" (located in the C:\IFCroot\BIN directory).

If there is a failure, contact your local field applications engineer to verify the problem, or contact Customer Support at Coreco Imaging, Inc.

Connect a Camera and Acquire an Image

- Use the Camera Configurator® Utility to match your camera or video source to the correct video port, initialize the system, and acquire a picture.

You are now ready to connect and configure a camera, and acquire an image. First, shut down the computer.

NOTE *Do not connect or disconnect cameras with the computer power on. The camera or PC-CamLink could be damaged by "hot plugging".*

1. Turn power off if you have not already done so.
2. Connect your video source to the 26-pin Camera Link connector. Refer to Figure 4.

NOTE *The Camera Link cable does not supply power to the camera. Use an external supply, or the separate PC-CamLink Power Cable 509-00078-00 shown in Appendix A of this manual.*

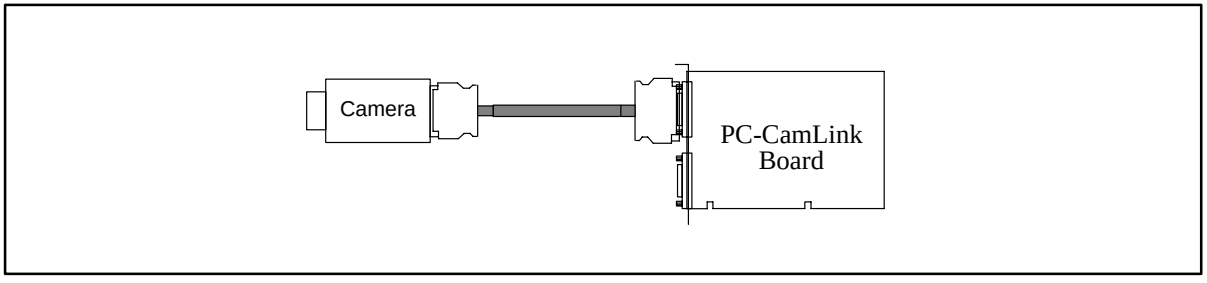


Figure 4. Camera Connection

Configurator Utility

The Camera Configurator® utility makes it very easy to set up and configure each video input to a camera. Additional information about cameras occurs later in this manual. At this time we are just interested in initializing and acquiring an image.

A camera must be connected before power is turned on. Connecting or disconnecting cameras with power on can damage the camera or PC-CamLink.

1. Double-click on the Configurator icon in the IFC program group. The utility will take a moment to detect and initialize the hardware.

When you start the Camera Configurator (or select File|New), the program detects all the Coreco Imaging, Inc. hardware (supported by IFC) installed, and creates a Hardware Data Structure that defines all the installed frame grabber boards.

2. Now you should see the screen shown in Figure 5, the Configurator window, with an Image Display window inside.

   The Camera Configurator opens with three main viewing panes:

   - **Config View** - hardware configuration file and hardware parameters.
   - **Camera View** - camera definitions and image display. (does not open if no hardware is installed)
   - **Log View** - command log and error log.

   Any changes you make to these panes (change size, close, hide) will be applied the next time the program opens.

3. First click on a port in the Config File (Config View), then click on a camera name in the List of Cameras (Camera View). The display ("display context") changes immediately.

   a. If "Single Click Assignment" is checked, that camera gets assigned to the port highlighted in the Config View.

   b. If "Single Click Assignment" is not checked, double-click on the camera name, right-click on the camera name, or drag and drop the camera name onto the highlighted port.

Refer to the IFC Camera Configurator User's Manual for additional information on using the Camera Configurator.

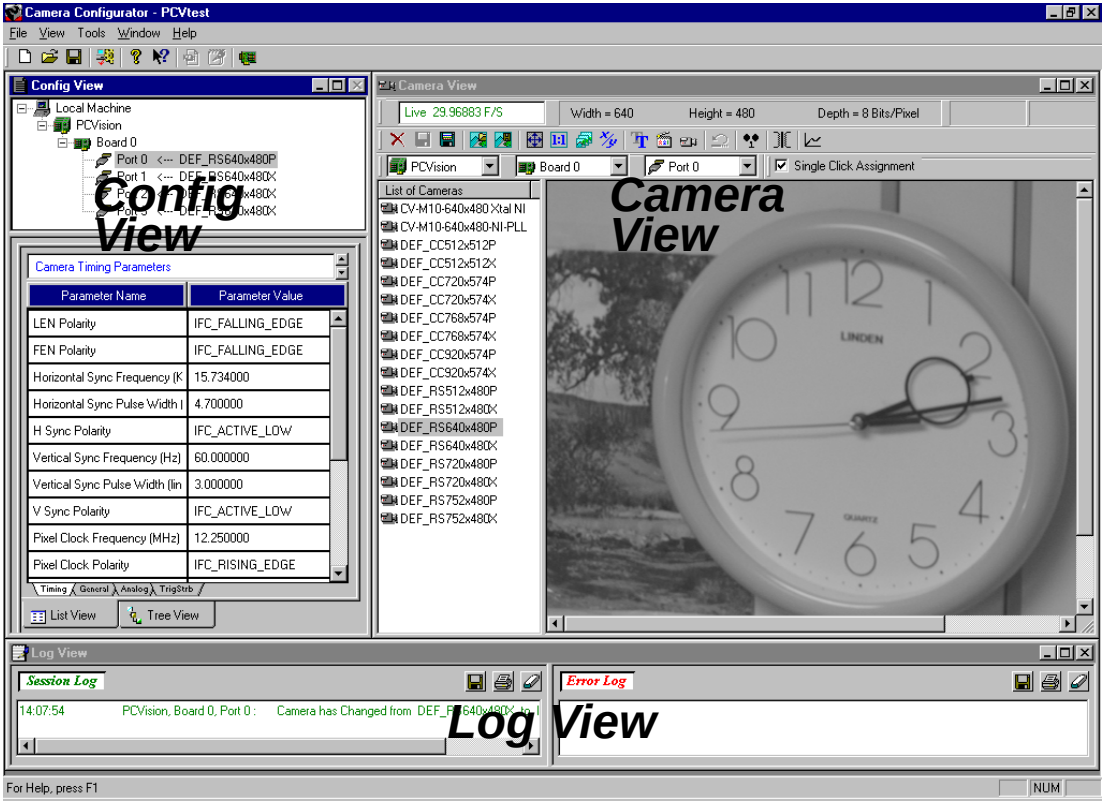


Figure 5. Configurator Startup Screen

Save Files

You can save three different file types using the Camera Configurator. All files are saved as text (*.txt).

You can save the frame grabber and camera port assignments to a Configuration File. Configuration files do not contain the camera parameters or your changes to the parameter values.

You can save your parameter changes to a new or existing Camera Definition file in the camera database. This updates only the one definition file. All configuration files that point to that definition file in the camera database will use the updated information.

You can save your parameter changes to a Portable Configuration File. This updates only the one Configuration File, and not the camera database. Only the current Portable Configuration File receives the updated information.

- A **Camera Definition File** defines one camera's parameter values, without the frame grabber or port assignment. This file is used in the camera database. All configuration files that call out this camera definition name use the updated information. Your files are saved to the Config\User directory. You cannot overwrite the original camera definitions. All User and original definitions are displayed in the List of Cameras.
- A **Configuration File** defines the installed frame grabbers, and the camera port assignments. The configuration file calls out the camera file names used in the camera database. The configuration file does not contain the camera parameter values.
- A **Portable Configuration File** defines the frame grabbers, port assignments, and camera parameter values. The portable configuration file does not call out camera definitions in the camera database.

File Names

All IFC examples try to load a default configuration file. The name of the default file changes with the frame grabber board used. The PC-CamLink examples on the CDROM look for the file name **lnktest.txt**.

Part 2
Hardware

- The PC-CamLink is an image capture board that uses the Camera Link™ interface to acquire images from digital output or "digitizing" cameras up to 24-bits.
- The PC-CamLink provides fully programmable timing and a variety of trigger and I/O, options that are ideal for many machine vision applications.
- The PCI-bus interface incorporates a hardware "scatter gather" table for highly efficient, fully automated and simultaneous image transfers from the image memory.
- PC-CamLink uses the high data transfer rates of the PCI-bus to eliminate the need for on-board processing or display circuitry. Image display and processing is handled by the host computer resources. The linear format image memory allows acquisition of a variety of image sizes. The image memory behaves as a temporary buffer between the camera interface and the host PCI-bus system.
- The PC-CamLink is capable of bus mastering image data directly to a destination memory within the PCI-bus system, such as system memory or VGA memory. Transfer rates in excess of 100MB/s can be sustained, depending upon the host capabilities. Images can be transferred to host memory in a fraction of the time that they were acquired. By minimizing the PCI-bus transfer time and CPU overhead, more bandwidth is available for processing or other system resources.

FEATURES

Sensor Interface

- Camera Link, Base Mode, up to 24-bit differential digital data input: from 20 MHz up to 62.5 MHz
- 8, 10, 12, 14, 16, or 24-bit single tap
- 8, 10, or 12-bit dual tap
- 24-bit RGB color (8-bits per color)

Image Buffering and Output

- 16MB high bandwidth SDRAM supports acquisition rates up to 187.5MB/s (24 bits at 62.5 MHz)
- Provides ROI (cropped) transfer to reduce data volume
- On-the-fly re-sequencing of image data for multi-tap cameras
- Data re-formatting to allow non-destructive overlays. Transfer 8-bit or 16-bit images in normal or 4:2:2 format
- Color plane separation during transfer possible
- Output clipping of Windows colors during bus master to eliminate conflicts with Windows reserved colors

External Trigger, Timing

- Advanced multi-trigger acquisition modes
- 2 External Trigger available as differential, TTL or Opto-coupled
- Software triggering for host controlled acquisition
- Trigger divider allows precise control over incoming trigger events

Programmable Window Generator

- Enables acquisition of subset of camera image
- Maximum window size: 64K by 64K, resolution of 1 pixel and 1 line
- Programmable independent X/Y decimation (by 2, 4, 8, 16) performed during image acquisition
- Color images stored using packed 24'–bit or 16-bit formats, or sequential color planes

Camera Timing and Control

- Camera Link timing inputs: Clock (PCLK), Line valid (LVAL), Frame valid (FVAL), Data valid (DVAL), and spare input (SPR)
- External Sync and Programmable Integration outputs
- 4 Static control lines for selecting camera modes
- Serial port interface between camera and PC-CamLink frame grabber

Look Up Tables (LUTs)

- Dual 16 by 16 LUTs for pixel transformations
- Arithmetic operations on or between 8-bit image data

Multi-Tap Data Resequencing

- Reformats multi–tap data on the fly
- Provides line/row resequencing to output most any multi-tap image in normal raster scan format
- Color plane separation during transfer possible

On Board Digital I/O

- OPTO-22 compatible 8-bits in 8-bits out
- 3 TTL outputs, 2 Differential outputs, and 2 Opto-coupled outputs

PCI Interface

- Pixel replication on transfer to host by factors of 2 or 4 independently in horizontal and vertical axis; performed during bus master transfer to host
- Scatter gather hardware feature for highly efficient image transfers
- Bus master of image data at 120MB/s sustained
- Target access (random R/W) to onboard registers, image memory and LUTs
- Multiple interrupt sources

Display - Windows

- Display Resolution as per installed video display card
- Non-destructive hardware enabled overlay support
- DirectX compatible display card (SVGA) required for real time display

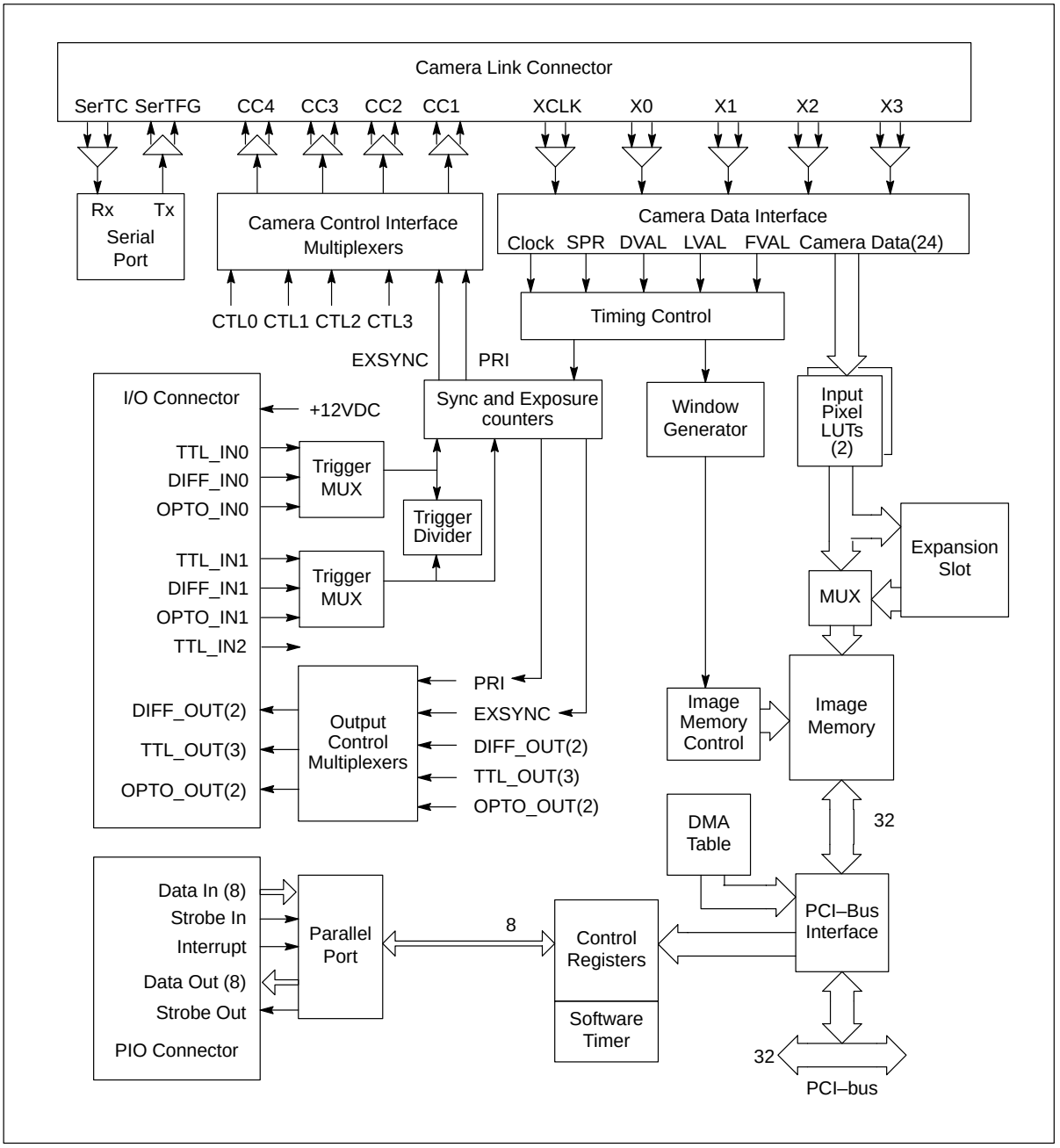


Figure 6. PC-CamLink Block Diagram

Acquisition Mode

The PC-CamLink slaves to the camera through the Camera Link connector. The timing signals are input directly to the PC-CamLink timing control. The PC-CamLink supports area scan and line scan cameras and sensors.

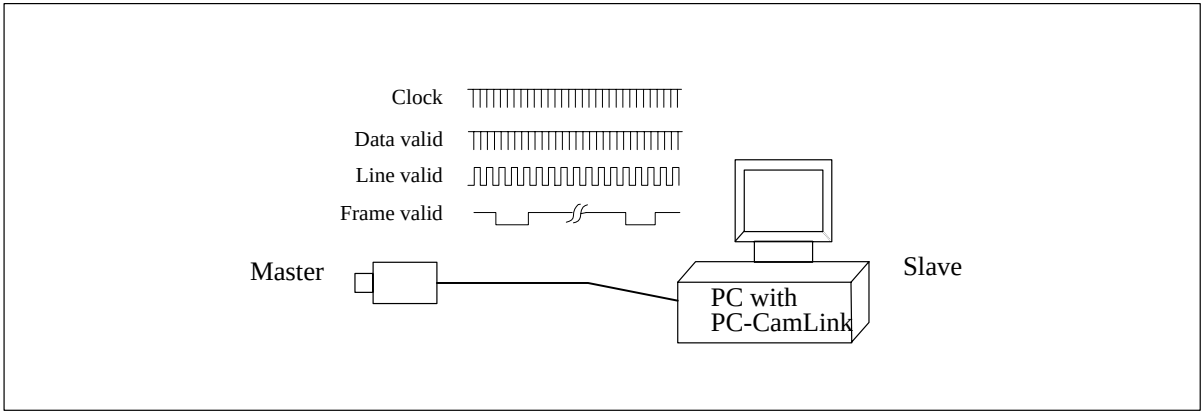


Figure 7. Acquisition and Timing

Normal Acquisition

The beginning and ending of a normal acquisition is based on the framing signal FVAL (frame enable) in area scan mode, or LVAL (line enable) in line scan mode.

Line Scan: In Line Scan mode, The Vertical Active (VACT) defines the number of lines acquired to create a frame image. Line scan is a non-interlaced acquire. The starting field select bits must be programmed to next field. A snap command performs a single frame acquire beginning at the next LVAL (rising edge) and acquires VACT number of lines. A grab command performs continuous acquisition beginning on the next LVAL (rising edge). A freeze command will stop a grab when the VACT count is reached. The End of Acquire trigger event stops a grab at the next falling edge of LVAL (line enable not valid).

Area Scan: A snap command performs a single frame acquire beginning at the next FVAL (rising edge). A grab command performs continuous acquisition beginning on the next FVAL (rising edge). A freeze command will stop a grab at the next falling edge of FVAL. When acquiring non-interlaced images, a snap lasts for one field (1 frame).

When acquiring interlaced images, a snap lasts for one field (one half frame). You must program the frame counter to snap twice the number frames you need. Field select bits determine which field of the interlaced image will be acquired first. Setting these bits for next field allows either field, even or odd, to be acquired next.

External Trigger Acquisition

External Trigger synchronizes image acquisition to external events. When acquiring an image in external trigger mode, the acquisition will not start until PC-CamLink receives a trigger signal. When performing external trigger acquisition, be sure the starting field bits of the Acquire Control register are set for next field.

Line Scan: After the trigger occurs, the next rising edge of LVAL begins the acquisition. The Vertical Active (VACT) defines the number of lines acquired, to create a frame. Alternatively, a stop trigger can be used to end acquisition at the end of the current line.

Area Scan: After the trigger occurs, the next rising edge of FVAL begins the acquisition. The trigger input must be re-armed to acquire subsequent frames.

Multiple Frame Acquire Mode

PC-CamLink has the ability to acquire and store up to eight images sequentially into image memory with the execution of one snap command. Program the Frame Count (FCNT or number of frames argument) to the desired number of frames per snap command (1 to 8). For non-interlaced acquisitions the multiple acquire mode count (FCNT) always refers to full frames. For interlaced acquisitions the multiple acquire mode count (FCNT) always refers to fields, and must be programmed to twice the number of full frames.

In Normal mode, the frame count is set for 1 to 8 images. When a Snap command is executed, the programmed number of images is acquired sequentially into a memory frame. In Grab mode, the set number of images is acquired and then the acquire address wraps back to the start of acquire address, overwriting the previous images. The image size must be taken into account when acquiring multiple images. The image memory is 16MB and wraps around if the number of incoming pixels fills the buffer. With frame count programmed to "acquire infinite frames" the grab continues to acquire images, without resetting the address. "Acquire infinite frames" is only used with the grab command.

In Triggered mode (snap or grab) each incoming external trigger will acquire one frame and advance the frame count. In triggered snap mode, each incoming trigger acquires a frame until the frame count is reached and the snap operation ends. In triggered grab mode, each incoming trigger acquires a frame until the frame count is reached, the frame count resets and the acquire address resets to its original start point (usually address zero), and the triggered acquire continues until a freeze command is issued. "Acquire infinite frames" can be used in triggered grab mode.

Using multiple frames with bus mastering allows simultaneous acquire and transfer without loss of video. A typical application would define two image "frames" in memory and program a grab acquire. At the end of the first image, a request is sent to the bus master controller which processes the request and transfers the first image. Simultaneously, the second image is acquired into the next "frame" location in memory. No data gets over-written as long as the transfer of the first image is completed prior to the end of the second image acquire. The operation repeats when the second image finishes; a second request for transfer is sent to the bus master controller. The second image is transferred and the acquire operation switches back to the first frame. This sets up a "ping-pong" acquire buffer.

Programmable Window Generator

A programmable window generator (PWG) defines how much of the incoming image data in each line gets acquired to the image buffer, and how many lines of the image, up to 64K pixels per line or 64K lines. Data occurring outside the programmed "window" is not stored. This capability does not affect the speed of the acquisition, and is similar to the concept of a sub-region or ROI (region of interest). The PWG can be programmed through the Camera Configurator, or through IFC Camera and Capture Module functions. Image memory bandwidth is maximized when acquired data is a multiple of 256 bytes.

External Triggering

External Trigger synchronizes image acquisition to external events. The trigger inputs can be used to start acquire and stop image acquisition. When acquiring an image in external trigger mode, the acquisition will not start until PC-CamLink receives a trigger signal. When performing external trigger acquisition, be sure the starting field bits of the Acquire Control register are set for next field.

Line Scan: After the trigger occurs, the next rising edge of LVAL begins the acquisition. The vertical active register defines the number of lines acquired, to create a frame. Alternatively, a stop trigger can be used to end acquisition at the end of the current line.

Area Scan: After the trigger occurs, the next rising edge of FVAL begins the acquisition. A stop trigger can be used to end acquisition at the end of the current frame.

PC-CamLink Triggers

PC-CamLink has two active trigger inputs: Trigger0 and Trigger1. Both trigger inputs support three different signal types: differential, TTL, and Opto-isolated. Refer to the Block Diagram in Figure 6. These are the Hardware Trigger Sources, input on the 26-pin D-Sub I/O Connector. The differential input is EIA-644 LVDS. Both trigger circuits have a glitch detector circuit, enabled or disabled by software. There are also two software triggers. The triggers can be programmed to start acquisition, end acquisition, start the external sync counter, or cause an interrupt. The trigger input must be re-armed to acquire subsequent frames.

All trigger features can be enabled and configured in the Camera Configurator® utility, and saved in a camera file. You can also use the IFC Parameter Access functions in the Class CICamera to change the trigger parameters. All PC-CamLink parameters are listed in an Appendix of the *IFC-SDK Software Manual*. Pay special attention to Hardware Specific Classes and functions for the PC-CamLink in Class **CPCLink**, Class **CICapmod**, and the Acquisition Functions in Class **CICamera**.

NOTE *Each frame grabber supported by IFC has different trigger set up requirements and functions. Pay special attention to notes in the Acquisition functions that refer to specific hardware supported by specific functions. For example, many acquisition functions with "trig" in the function name apply only to the PCVision frame grabber.*

Trigger Input Divider

The PC-CamLink incorporates a trigger divider. The divider may be used to rate-divide one of the trigger inputs (Trigger0 or Trigger1) under software control. The divider provides a programmable division of 1 to 256. This feature is particularly useful in line scan applications where the camera scan rates can be set up to operate at rates less than the trigger rate being received from a shaft encoder.

Using the Opto-Coupled I/O

The PC-CamLink has two opto-coupled inputs and two opto-coupled outputs. The plus and minus input connections are the two sides of the LED (light emitting diode). The "OC" output connection is the "open collector" of an NPN transistor. The VCC and GND (power and ground) of this device are also provided on the PC-CamLink connector. Figure 8 shows a single opto-isolator.

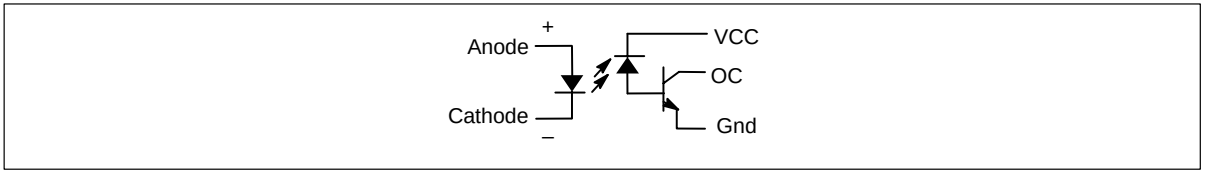


Figure 8. Basic Opto-Isolator

The PC-CamLink uses the HP HPCL-0531 dual-channel isolator. Do not exceed the absolute maximum ratings of the diode and transistor in these devices.

Opto-Coupled Input Circuits

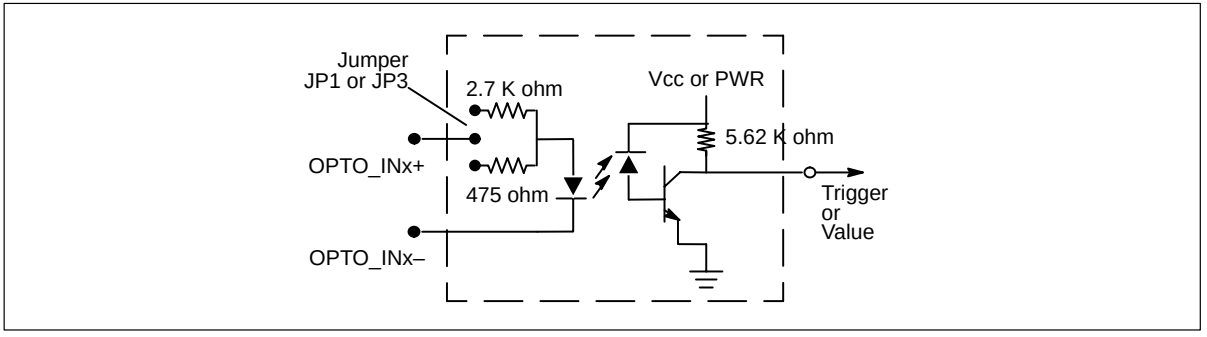


Figure 9. Opto-Coupled Input Circuit

Figure 9 shows the internal circuitry in the Opto-Coupled inputs. When the diode is excited, the transmitter is on, and the internal Value is zero. When the diode is off, the transmitter is off, and the internal Value is one. Jumper JP1 selects the input resistor for the OPTO_IN0+ input. Jumper JP3 selects the input resistor for the OPTO_IN1+ input.

Figure 10 shows a few practical examples for using the input. The positive input is connected to your logic or signal, and the negative input is connected to ground through a 330 ohm 1/8 Watt resistor. The resistor limits current through the diode. The internal 2.7K ohm or 475 ohm resistor limits the voltage across the diode. When the external input goes high the diode turns on, and the internal value becomes zero. When the external input goes low the diode turns off, and the internal Value becomes one. The diode and resistors are the Load connected to a sensor output.

The component values depend upon the voltage and current of your circuits. Do not exceed the Absolute Maximum Ratings of the input diode.

- Diode Absolute Maximum Ratings

If (avg)	25 mA	Maximum Average Forward Current
If (pk)	50 mA	Maximum Peak Forward Current
Vr	5 V	Maximum Reverse Voltage
Pi	45 mW	Maximum Input Power Dissipation

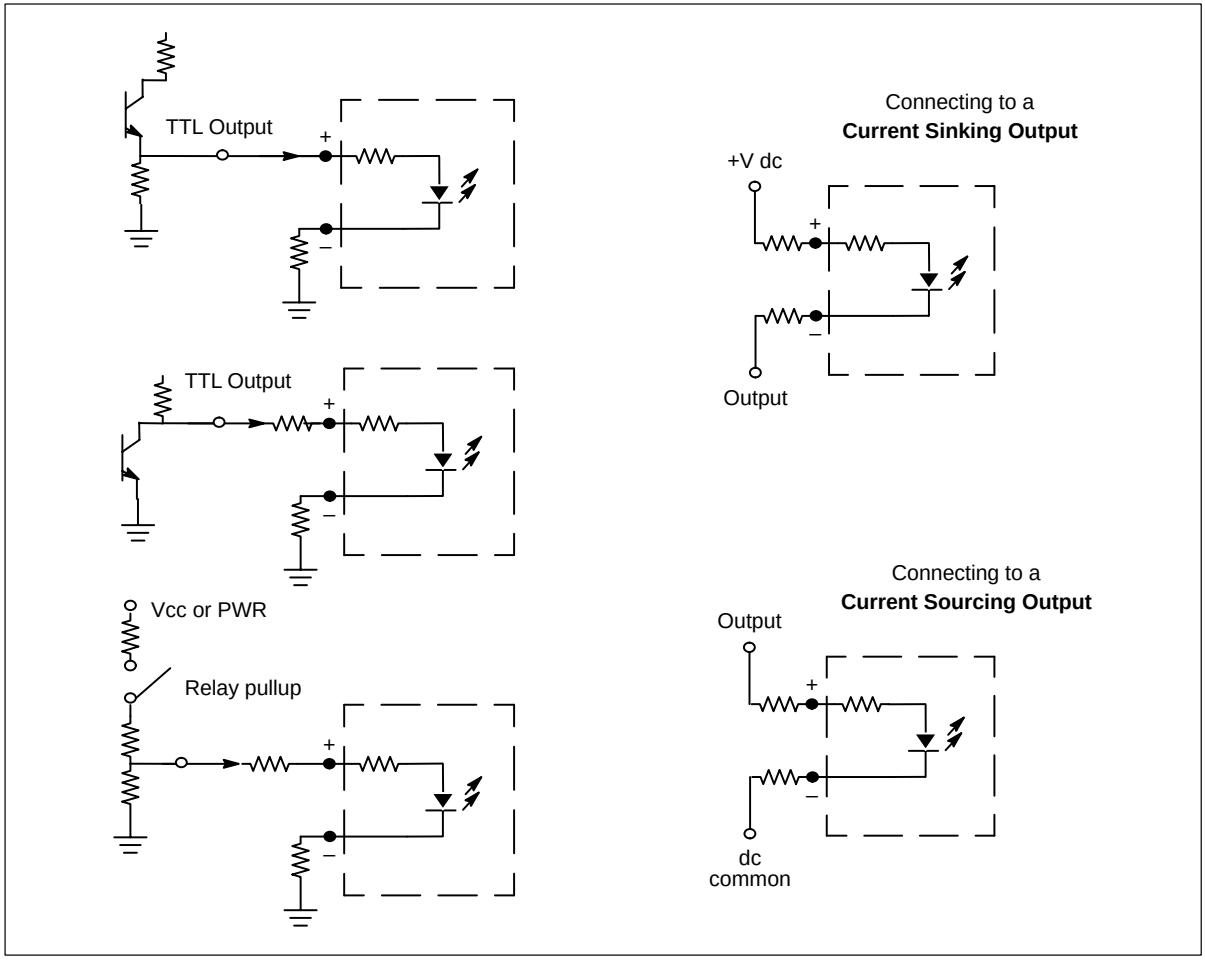


Figure 10. Input to Opto-Coupled Circuit

Opto-Coupled Output Circuits

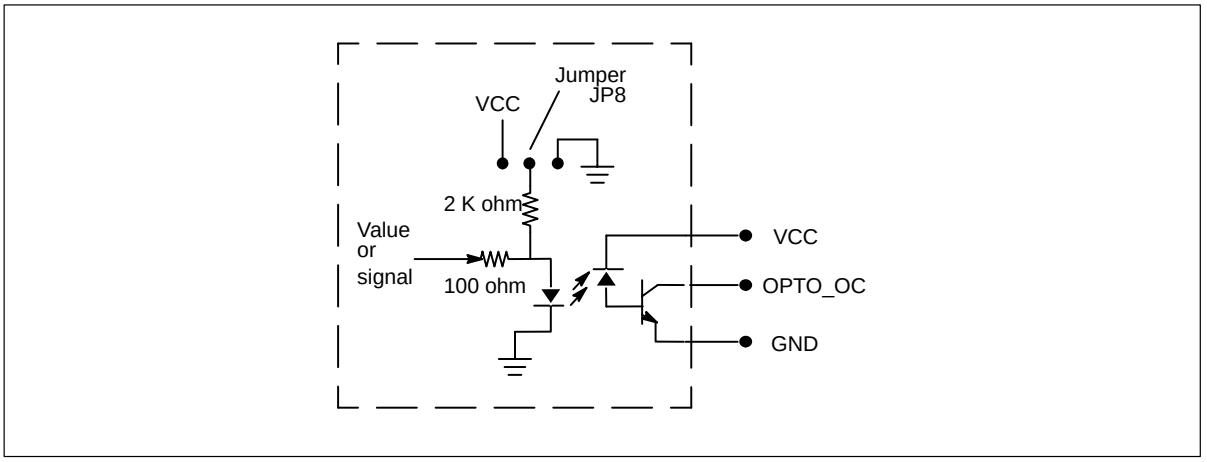


Figure 11. Opto-Coupled Output Circuit

Figure 11 shows the internal circuitry in the Opto-Coupled outputs. When the internal logic is one, the diode and transmitter are on, and the "open collector" connection OPTO_OC is pulled low. When the internal logic is zero, the diode and transmitter are off, and the OPTO_OC is tri-stated or pulled high (depending on the external circuit). Jumper JP8 selects the state of the two opto-coupled outputs only during power up, until the FPGA files are loaded and the PC-CamLink initialized.

Figure 12 shows two circuits for using the output. When the internal logic goes high, the diode and transistor are turned on and the OPTO_OC output goes low. The output transistor can be connected as a Current Sinking Output.

The component values depend upon the voltage and current of your circuits. Do not exceed the Absolute Maximum Ratings of the output transistor.

- Transistor Absolute Maximum Ratings

Vcc	30 V	Maximum Supply Voltage
Vo	20 V	Maximum Output Voltage
Io (avg)	5 mA	Maximum Average Output Current
Io (pk)	50 mA	Maximum Peak Output Current
Po	35 mW	Maximum Output Power Dissipation

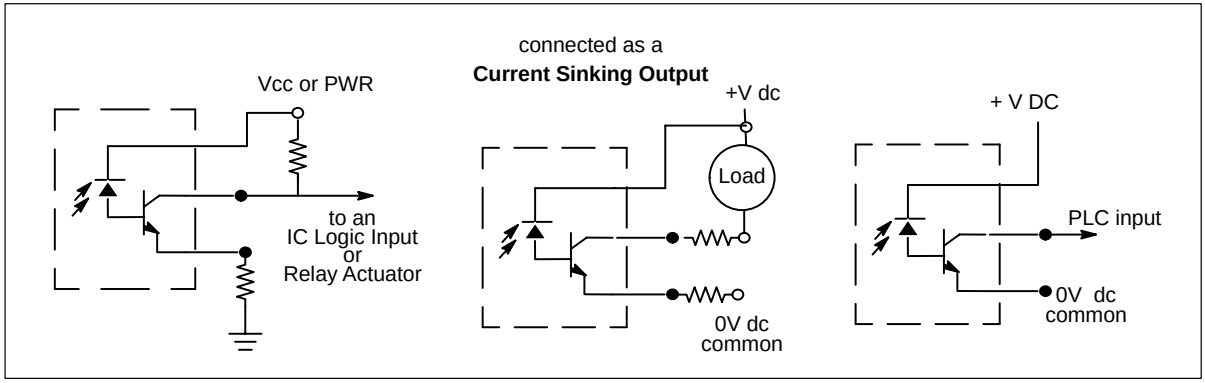


Figure 12. Output from Opto-Coupled Circuit

Sync Output

The external synchronization output (EXSYNC) can control camera timing and integration. EXSYNC is a programmable one-shot pulse output with a period ranging from 210 ns to 18.75 s and a pulse width ranging from 140 ns to 572 us. The polarity of the EXSYNC output is programmable. The EXSYNC counters can be disabled and can be reset using control register writes.

The EXSYNC signal can be selected as output on any of the four camera control lines on the Camera Link connector. The EXSYNC signal can be selected as output on any of the general outputs on the separate I/O connector: three TTL, two Opto-coupled, and two Differential outputs.

EXSYNC can operate in four different modes: free run mode, external trigger mode, vertical blanking mode, or software trigger mode. In Free-Run mode, EXSYNC continuously outputs a pulse at a programmable interval (retriggering itself). In External Trigger mode, the EXSYNC pulse is asserted for the programmed period when an external trigger occurs. In Vertical Blanking mode the EXSYNC pulse is asserted for the programmed period when the PC-CamLink enters the "vertical blank" period between valid frames. In software trigger mode the EXSYNC pulse is asserted whenever the host writes to the Software Trigger register. Software can disable the EXSYNC counter and use software trigger mode to hold the EXSYNC output low or high, generating integration or exposure times longer than supported by the EXSYNC counter. EXSYNC can be a frame reset pulse output.

EXSYNC can be used as the "exposure" input to the camera, when using the Xillix, Kodak MEGAPLUS, and Hammamatsu cameras. EXSYNC is used as the line transfer input to the Dalsa cameras.

All External Sync features can be enabled and configured in the Camera Configurator® utility, and saved in a camera file. You can also use the IFC Parameter Access functions in the Class CICamera to change the EXSYNC parameters. All PC-CamLink parameters are listed in an Appendix of the *IFC-SDK Software Manual*.

Programmable Integration

The PRI output signal is used with Dalsa cameras to control integration time. PRI is a programmable one-shot pulse output with a period ranging from 210 ns to 18.75 s and a pulse width ranging from 140 ns to 572 us. The polarity of the PRI output is programmable. The PRI counters can be disabled and can be reset using control register writes.

The PRI signal can be selected as output on any of the four camera control lines on the Camera Link connector. The PRI signal can be selected as output on any of the general outputs on the separate I/O connector: three TTL, two Opto-coupled, and two Differential outputs.

PRI operate in the same mode selected for the EXSYNC signal: free run mode, external trigger mode, vertical blanking mode, or software trigger mode.

PRI and EXSYNC are used together to synchronize the Dalsa line scan cameras to a varying external trigger pulse, while maintaining constant exposure time.

All Programmable Integration features can be enabled and configured in the Camera Configurator® utility, and saved in a camera file. You can also use the IFC Parameter Access functions in the Class CICamera to change the PRI parameters. All PC-CamLink parameters are listed in an Appendix of the *IFC-SDK Software Manual*.

Input LUTs

By default the IFC library programs the input LUTs to normalize 10-bit, 12-bit and 14-bit data to 16 bits. For example, the most significant bit (msb) of 12-bit data is left-shifted four places, creating a 16-bit data value with zeros in the four least significant bits (lsbs). This feature can be defeated by programming the pixel size to 16 bits. Do not defeat this feature in 10-bit and 12-bit modes. The input multiplexers are hard-wired for the correct data format in these modes.

The input LUTs are two 16-bit-in 16-bit-out Look-Up Tables. Addressing and page size changes with the programmed pixel size. LUT data paths are programmable in single tap modes, but are hard-wired for dual tap modes to format the three Camera Link data ports into two 10-bit or 12-bit fields.

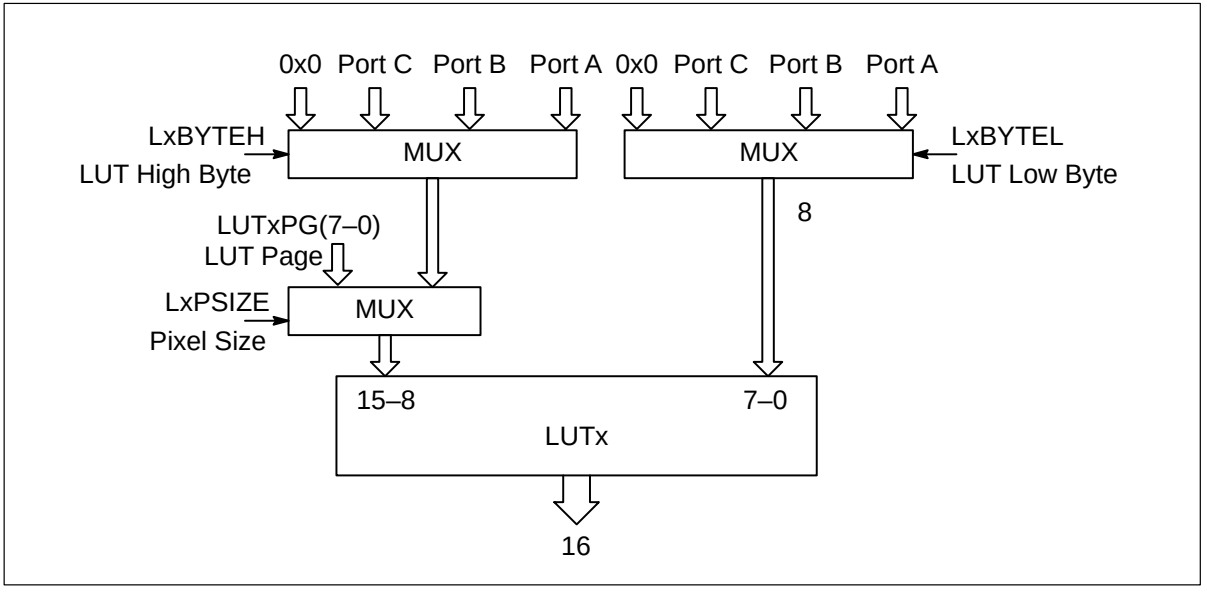


Figure 13. Input LUT

Figure 13 illustrates one of the two Input LUTs. The two LUTs are identical. The low byte and high byte multiplexers select from four 8-bit inputs. The pixel size (L0PSIZE and L1PSIZE) controls how much of the high byte is passed to the LUT, and how much is masked off and substituted with page select bits (LUT0PG and LUT1PG).

LUT Input Modes

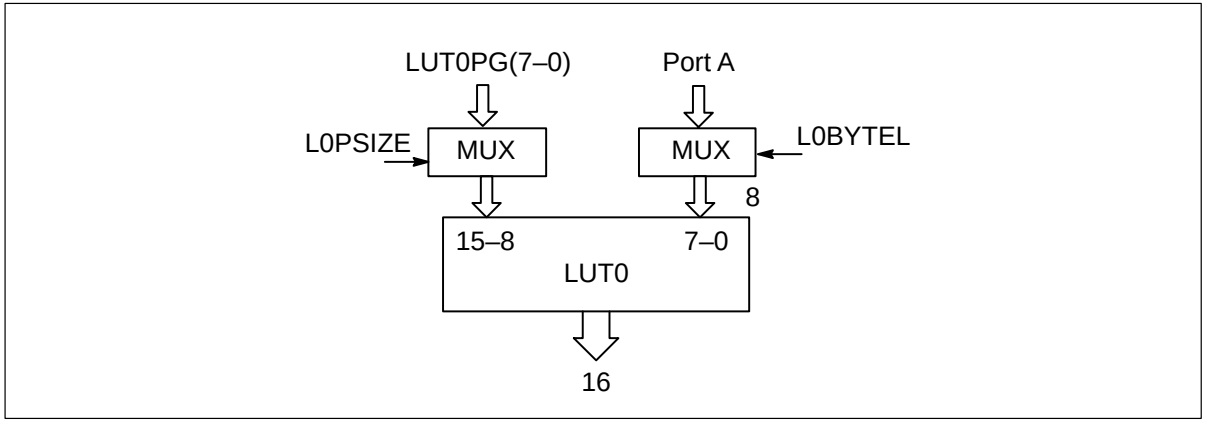


Figure 14. Single Tap 8-bit Input

For 8-bit single tap input, the program the Pixel Size to 8-bits (L0PSIZE), select the 8-bit data (usually Port A) as the Low Byte (L0BYTEL), and use the eight Page Select bits to change the LUT Page (LUT0PG(7–0)) for up to 256 different LUT transforms. Only the lower 8 bits of the output are stored in the image memory.

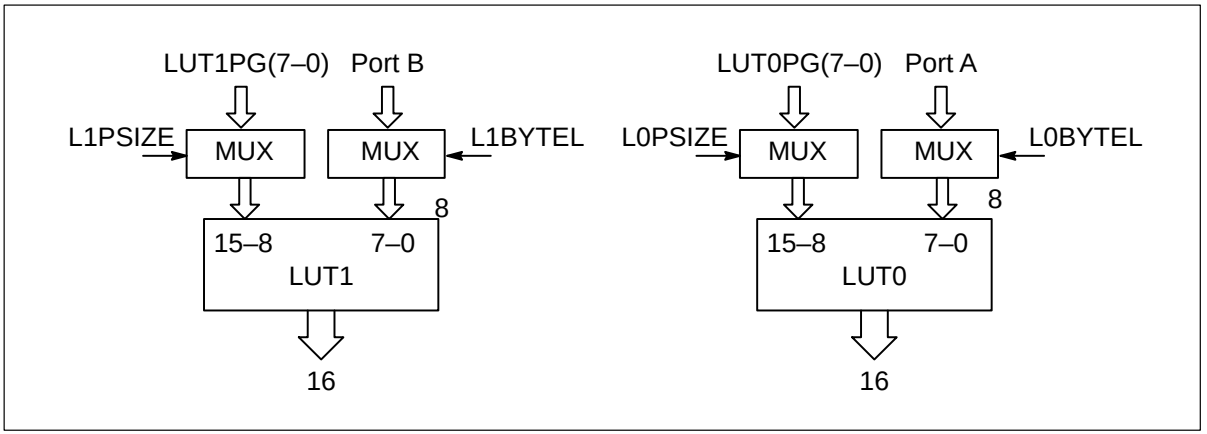


Figure 15. 8-bit two tap input

For 8-bit two tap input, program both LUTs for Pixel Size of 8-bits (L0PSIZE and L1PSIZE), select Port A as the Low Byte for LUT0 (L0BYTEL) and Port B as the Low Byte for LUT1 (L1BYTEL), and use the eight Page Select bits to change the LUT Page (LUT0PG and LUT1PG) for independent selection of up to 256 different LUT transforms for each tap. Only the lower 8 bits of each LUT output are stored in the image memory.

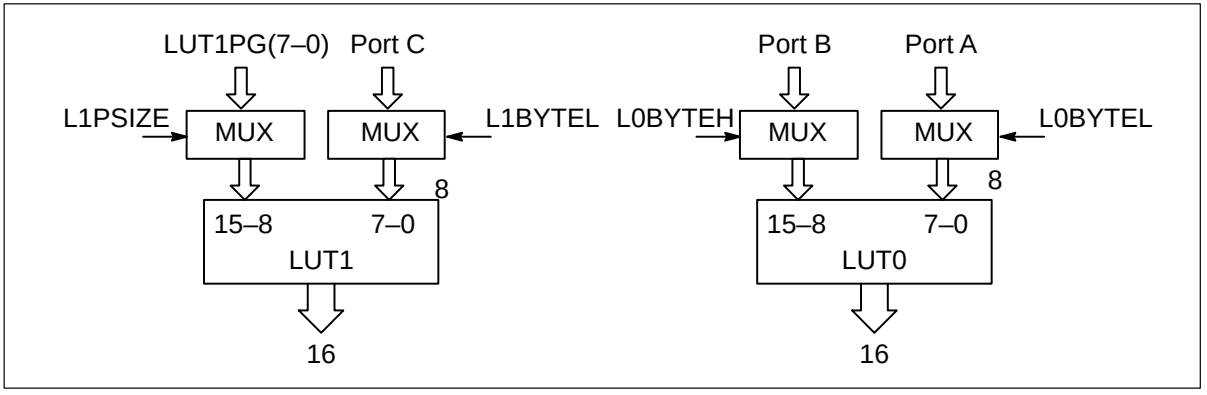


Figure 16. 8-bit three tap input

For 8-bit three tap input, program LUT0 for Pixel Size of 16-bits (L0PSIZE) and LUT1 for Pixel Size of 8-bits (L1PSIZE). Select Port A as the Low Byte for LUT0 (L0BYTEL), Port B as the High Byte for LUT0 (L0BYTEH) and Port C as the Low Byte for LUT1 (L1BYTEL). LUT0 has no available page select bits. LUT1 has eight Page Select bits (LUT1PG). The 16-bit output of LUT0 and the lower 8 bits of LUT1 output are stored in the image memory.

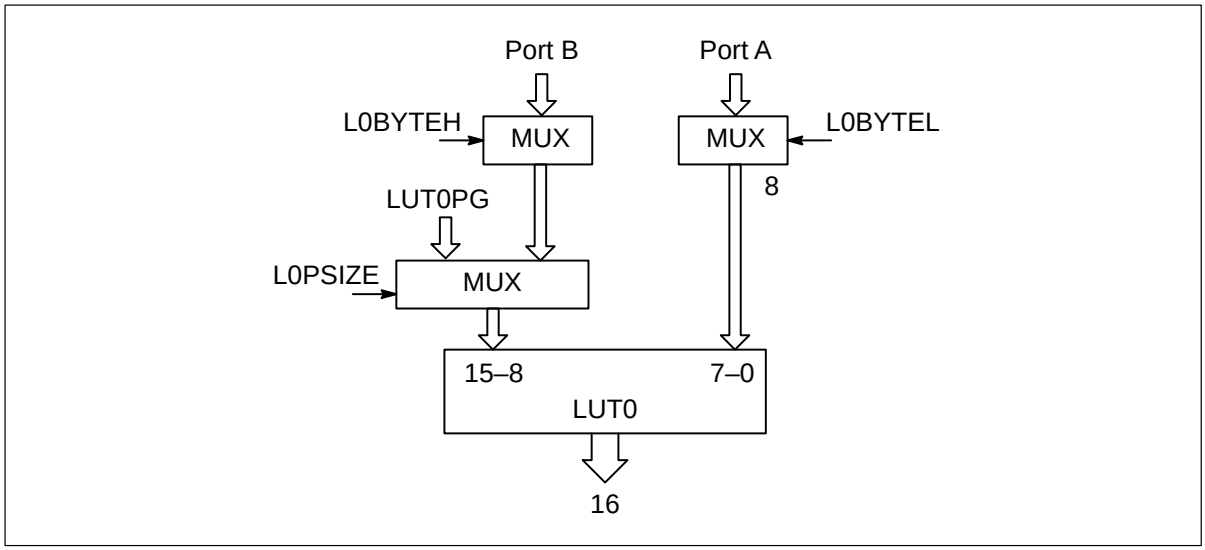


Figure 17. Single Tap 10 to 16 bit input

For 10-bit, 12-bit, 14-bit or 16-bit single tap input, program the correct Pixel Size (L0PSIZE), select Port A for the Low Byte (L0BYTEL) and port B for the High Byte (L0BYTEH), and use the available Page Select bits (LUT0PG) to change the LUT Page. The IFC library programs the LUTs to left-shift 10-bit, 12-bit and 14-bit data to 16-bit data with zeros in the least significant bits.

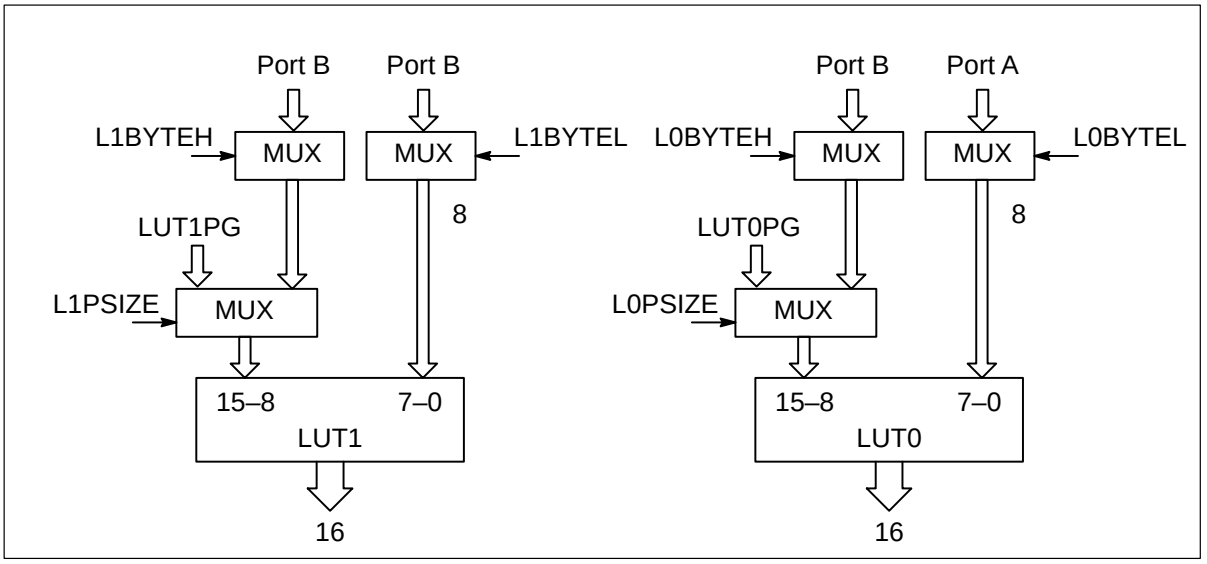


Figure 18. Two Tap 10 or 12 bit input

For 10-bit or 12-bit two tap input, program the correct Pixel Size (L0PSIZE and L1PSIZE), select Port A for the LUT0 Low Byte (L0BYTEL) and Port B for the LUT0 High Byte (L0BYTEH), select Port C for the LUT1 Low Byte (L1BYTEL) and Port B for the LUT1 High Byte (L1BYTEH). The input MUXs have been hard-wired to deliver the correct bits from Port B to each LUT to make up the 10-bit or 12-bit pixels. Use the available Page Select bits (LUT0PG and LUT1PG) to change the LUT Page. The IFC library programs the LUTs to left-shift 10-bit and 12-bit to 16 bits, data with zeros in the least significant bits.

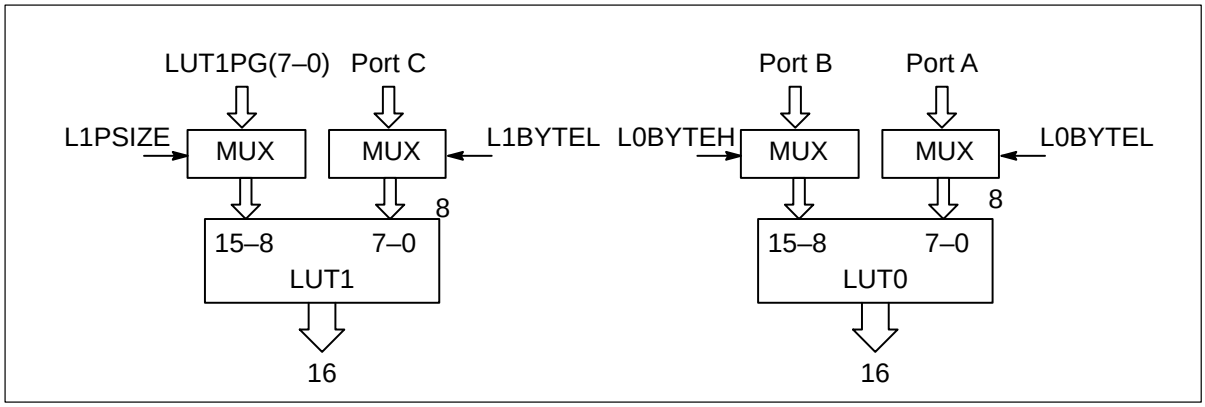


Figure 19. 24-bit RGB input

For 24-bit RGB data, program LUT0 Pixel Size for 16-bits (L0PSIZE) and LUT1 Pixel Size for 8-bits (L1PSIZE). Select Port A as the Low Byte for LUT0 (L0BYTEL), Port B as the High Byte for LUT0 (L0BYTEH) and Port C as the Low Byte for LUT1 (L1BYTEL). LUT0 has no available page select bits. LUT1 has eight Page Select bits (LUT1PG). The 16-bit output of LUT0 and the lower 8 bits of LUT1 output are stored in the image memory.

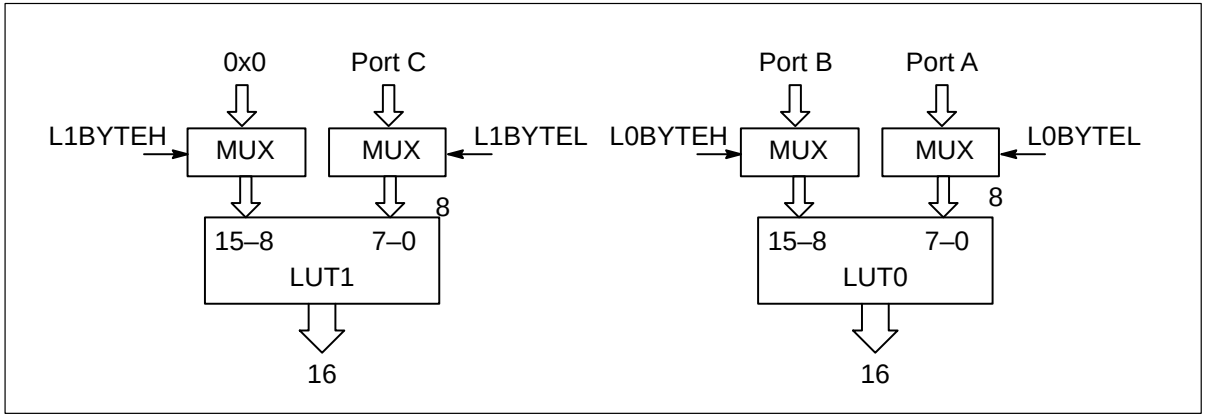


Figure 20. 32-bit RGBa

For 32-bit RGBa data from a 24-bit RGB input, program LUT0 and LUT1 Page Size for 16-bit input (L0PSIZE and L1PSIZE). Select Port A as the Low Byte for LUT0 (L0BYTEL) and Port B as the High Byte for LUT0 (L0BY-TEH). Select Port C as the Low Byte for LUT1 (L1BYTEL) and zero as the High Byte for LUT1 (L1BYTEH). The 16-bit output of LUT0 and LUT1 are stored in the image memory.

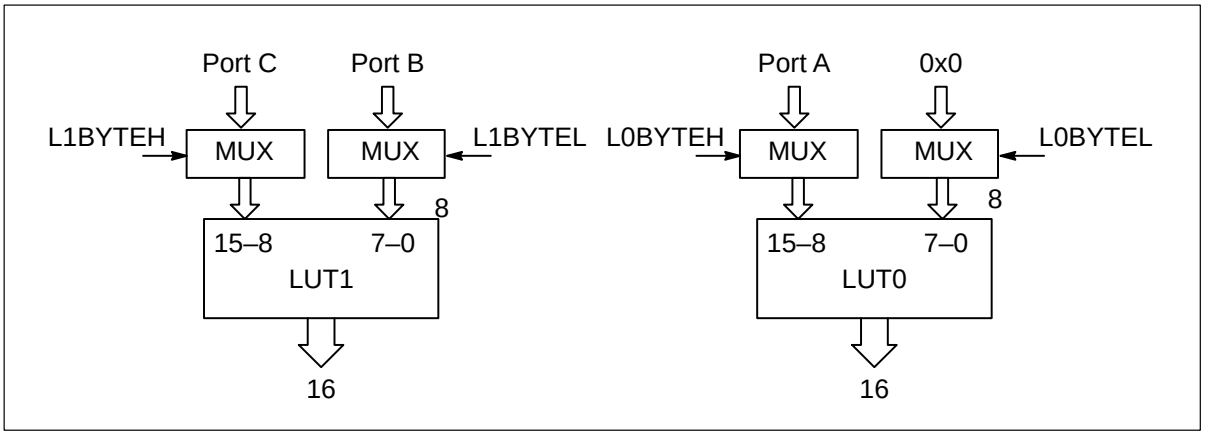


Figure 21. 32-bit aRGB

For 32-bit aRGB data from a 24-bit input, program LUT0 and LUT1 Pixel Size for 16-bit input (L0PSIZE and L1PSIZE). Select zero as the Low Byte for LUT0 (L0BYTEL) and Port A as the High Byte for LUT0 (L0BYTEH). Select Port B as the Low Byte for LUT1 (L1BYTEL) and Port C as the High Byte for LUT1 (L1BYTEH). The 16-bit output of LUT0 and LUT1 are stored in the image memory.

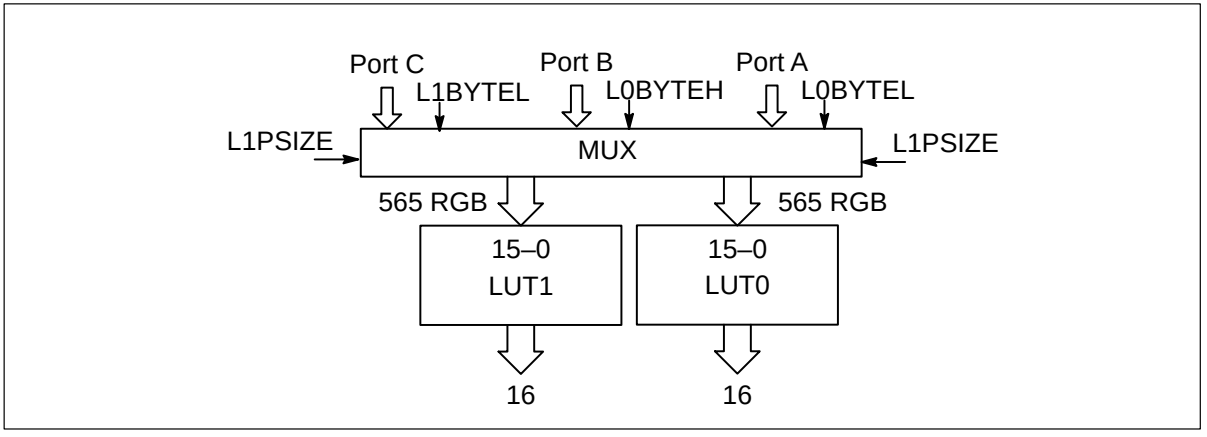


Figure 22. 16-bit 565 RGB

For 16-bit 565 RGB data from a 24-bit input, program LUT0 and LUT1 Pixel Size for 16-bit 565 RGB mode (L0PSIZE and L1PSIZE). Select Port A as the Low Byte for LUT0 (L0BYTEL), Port B as the High Byte for LUT0 (L0BYTEH), and Port C as the Low Byte for LUT1 (L1BYTEL). The input multiplexers are hard-wired to present the same 565 RGB data to both LUTs.

Parallel I/O

The 16-bit digital I/O port has 8 TTL inputs and 8 TTL/CMOS outputs. The pin-out of the 26-pin connector is found in Appendix A. The directions (input/output) cannot be changed.

The I/O port includes support for the following:

- Interrupt generating input
- Input latching
- Read status of latched data or current states at input pins
- Output strobe for latching an external device

The I/O port inputs can be used in "pass through" mode or "latched" mode. Figure 23 illustrates the two paths. In latched mode, a strobe input latches the state of the inputs into a buffer. Reading the input data actually reads the latch buffer. In pass through mode, reading the input data actually reads the pass-through buffer, or the current "live" state of the input.

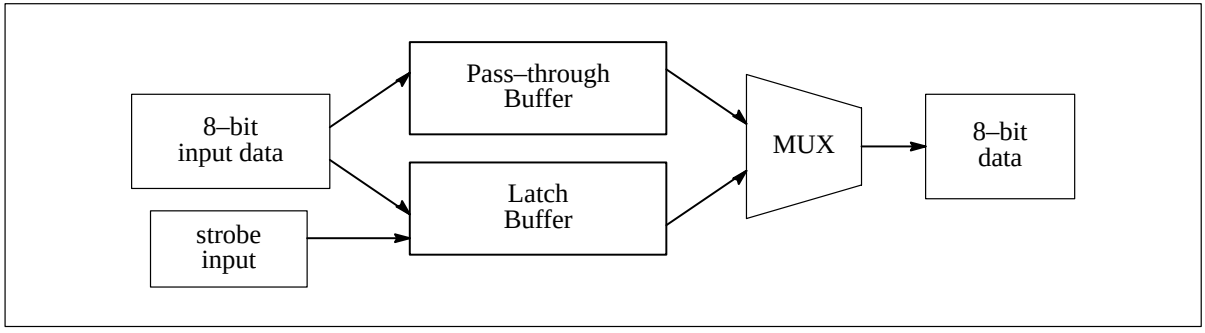


Figure 23. Digital Input Buffers

MULTI-TAP FORMATS

Multi-tap area scan and line scan cameras output image data in a variety of pixel sequences. An important PC-CamLink feature is the ability to transfer images to the host in the normal raster-scan format for processing or display. PC-CamLink provides the necessary line and row resequencing to output virtually any image from a multi-tap camera in the normal raster-scan format. Resequencing is performed in real-time without host processor intervention. Typical multi-tap image resequencing consists of reordering odd and even pixel sequences, odd and even lines, or opposite scans.

The figures in this section illustrate a variety of multi-tap pixel data sequences from line scan and area scan cameras. PC-CamLink can resequence all cases shown.

Single Tap

One channel presents the pixel data in sequential order. The acquired sequence represents the original image, with no need for sorting or reformatting. Figure 24 shows where the data appears in the original image. PC-CamLink can support this format for pixel depths from 8-bits up to 24-bits. The IFC Software Library calls this "One Channel Left to Right" or LNK_1CHAN_L2R.

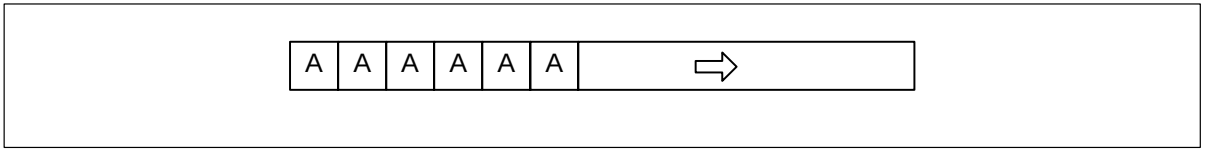


Figure 24. Single Tap

Odd-Even Pixels

One channel (A) carries only even pixels and the other channel (B) carries only odd pixels. Data is presented simultaneously on both channels. The pixels are "interleaved" in the original. Figure 25 shows where the data appears in a line from the original image. PC-CamLink can support this format for pixel depths from 8-bits up to 12-bits. The IFC Software Library calls this "Two Channel Interleaved" or LNK_2CHAN_INTERLEAVED.

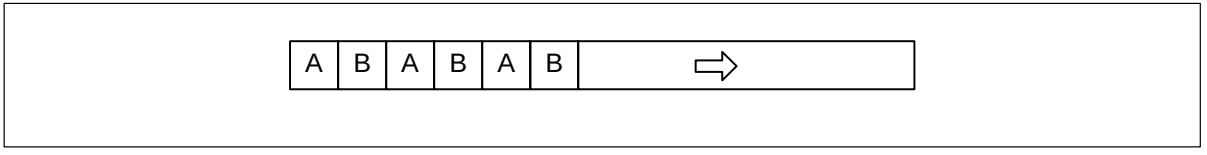


Figure 25. Odd-Even Pixels

Dual Tap with Line Segments

One channel (A) carries the left half of the line or frame, and the other channel (B) carries the right half of the line or frame. Data is presented simultaneously on both channels. Figure 26 shows where the data appears in a line from the original image. PC-CamLink can support this format for pixel depths from 8-bits up to 12-bits. The IFC Software Library calls this "Two Channel Separate Tap Left to Right" or LNK_2CHAN_SEP_TAP_L2R.

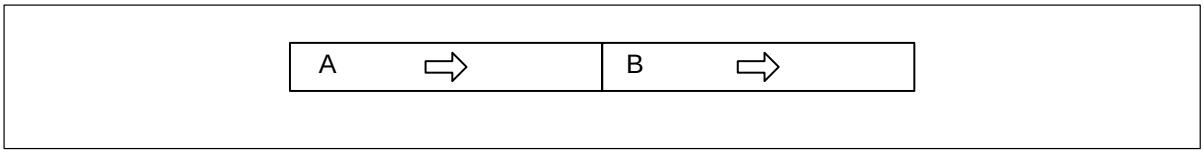


Figure 26. Two Line Segments

Dual Tap with Reversed Line Segments

One channel (A) carries the left half of the line or image, and the other channel (B) carries the right half of the line or image. The direction is right to left, or reversed from what we consider the "normal" left to right scan. Data is presented simultaneously on both channels. Figure 27 shows where the data appears in a line from the original image. PC-CamLink can support this format for pixel depths from 8-bits up to 12-bits. The IFC Software Library calls this "Two Channel Separate Tap Right to Left" or LNK_2CHAN_SEP_TAP_R2L.

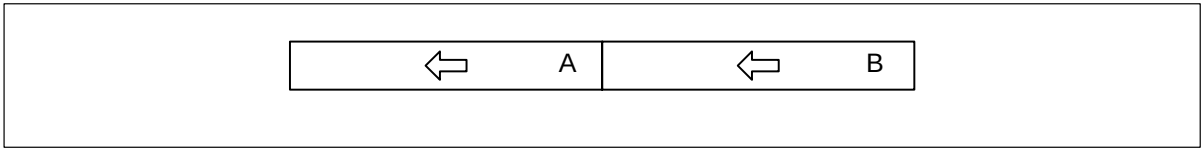


Figure 27. Two Reverse Line Segments

Dual Tap with Converging Line Segments

One channel (A) carries the left half of the line or frame, and the other channel (B) carries the right half of the line or frame. Channel A scans left to right, and channel B scans right to left. Data is presented simultaneously on both channels. Figure 28 shows where the data appears in a line from the original image. PC-CamLink can support this format for pixel depths from 8-bits up to 12-bits. The IFC Software Library calls this "Two Channel Separate Tap Converging" or LNK_2CHAN_SEP_TAP_CONVERGE.

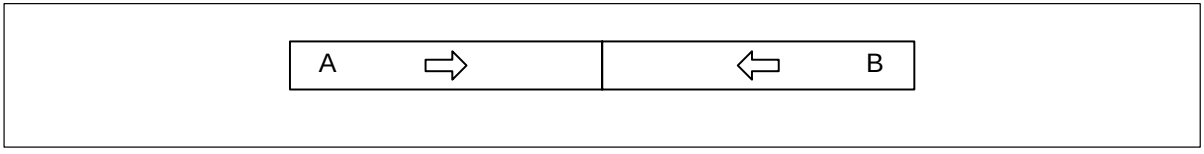


Figure 28. Two Converging Line Segments

Dual Tap Interline

One channel carries the even lines, and the other channel carries the odd lines. The lines are "interlaced" in the original image. Data is presented simultaneously on both channels. Figure 29 shows where the data appear in the original image. There are two options, as shown. Tap A can present the even lines 0,2,4,6,8, or the odd lines 1,3,5,7,9. The PC-CamLink can support this format for pixel depths from 8-bits up to 12-bits. The IFC Software Library calls this "Two Tap Interline A Even" and "Two Tap Interline B Even" or LNK_2TAP_INTERLINE_A_EVEN and LNK_2TAP_INTERLINE_B_EVEN.

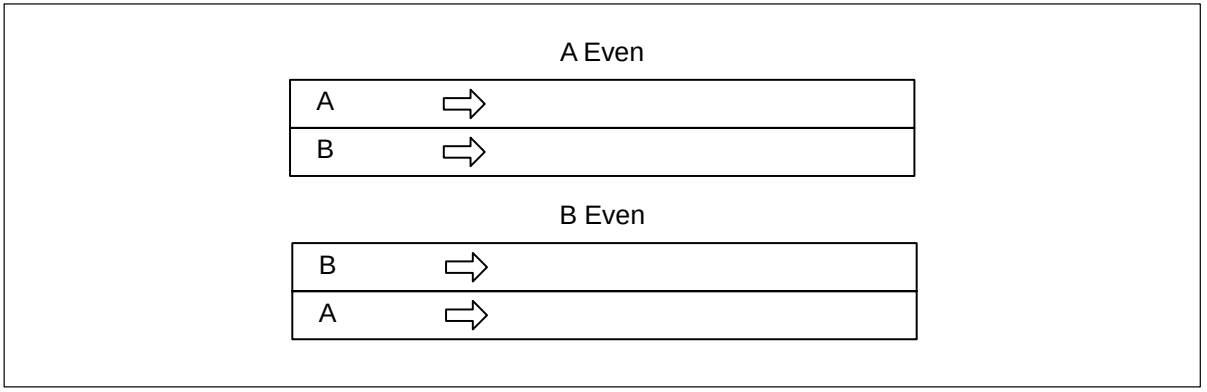


Figure 29. Interline

Three Tap with Line Segments

A line is divided into three equal segments. Data is presented simultaneously on all three channels. Figure 30 shows where the data appears in a line from the original image. The PC-CamLink can support this format for 8-bit pixels. The IFC Software Library calls this "Three Channel Separate Tap Left 2to Right" or LNK_3CHAN_SEP_TAP_L2R.

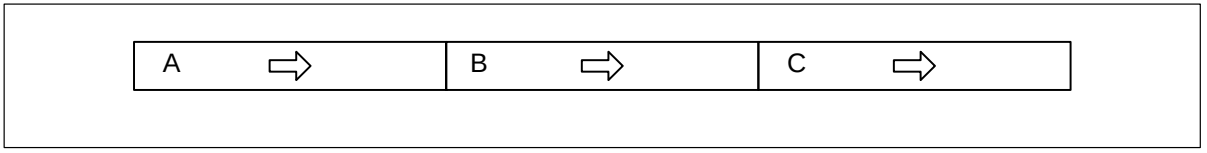


Figure 30. Three Line Segments

Part 3
Software

Overview

The latest information pertaining to the software release can be found in the software release notes included on the shipping media and installed in the root directory that you have chosen, such as "C:\IFC52". The CD-ROM contains the Imaging Studio CD-ROM contains the IFC SDK environment (and ITEX SDK) which includes all of the libraries, header files, and runtime DLLs necessary for software development. In addition, there is on-line help, electronic documentation, and example programs. The example programs illustrate the proper use of functions provided. These examples are not intended to be complete applications with all possible error checking and bullet-proofing usually associated with a robust application. They are intended to illustrate the basic concepts of programming with IFC.

The IFC libraries have been developed to provide the highest performance image acquisition possible, with the least amount of processor and developmental overhead. The functions take advantage of interrupts, multi-threading, multi-processors (if available) and PCI-bus mastering.

The IFC programming language supports the following:

- C++ Using MFC – the IFC App Wizard creates the basic framework
- Using VisualBasic – illustrated by the example \IFCxx\Examples\VB\VBExamp
- Using ANSI C – illustrated by the example *IFCxx*\Examples\CGrab

Visual C++ Programming with MFC

Using Microsoft Visual C++ and the Microsoft Foundation Classes to design your application gives you a great amount of flexibility in addressing the Windows Operating system at a root level. You can use the Applications Wizard to create your program in the standard doc/View/App structure, but there are a few items that you need to be aware of in the program design. Each is addressed here.

Multiple Instances

When locating the initialization code for the PC-CamLink board in your application, keep in mind that multiple applications **cannot** access the hardware at the same time. Because of this, you must develop your application to provide one interface to the hardware. You can also have multiple threads react to events, but make hardware access mutually exclusive.

Multi-Threading

Keeping the above section in mind, the same can also be said about the thread safety of your application when operating on concurrent processes. Each of your threads should not change the hardware state of the PC-CamLink board without signaling the other threads through the use of Semaphores or another similar method.

Keeping your hardware interface code in the Application module accomplishes this. Multiple documents can exist when they have a common IFC application interface. Quick Config is an example of the use of multiple documents. If each document had initialized the hardware, and managed the transfer into their own display, there would be multiple access to the same board, creating arbitrary hardware states for all concurrent documents. The following figure illustrates the differences in modeling your application.

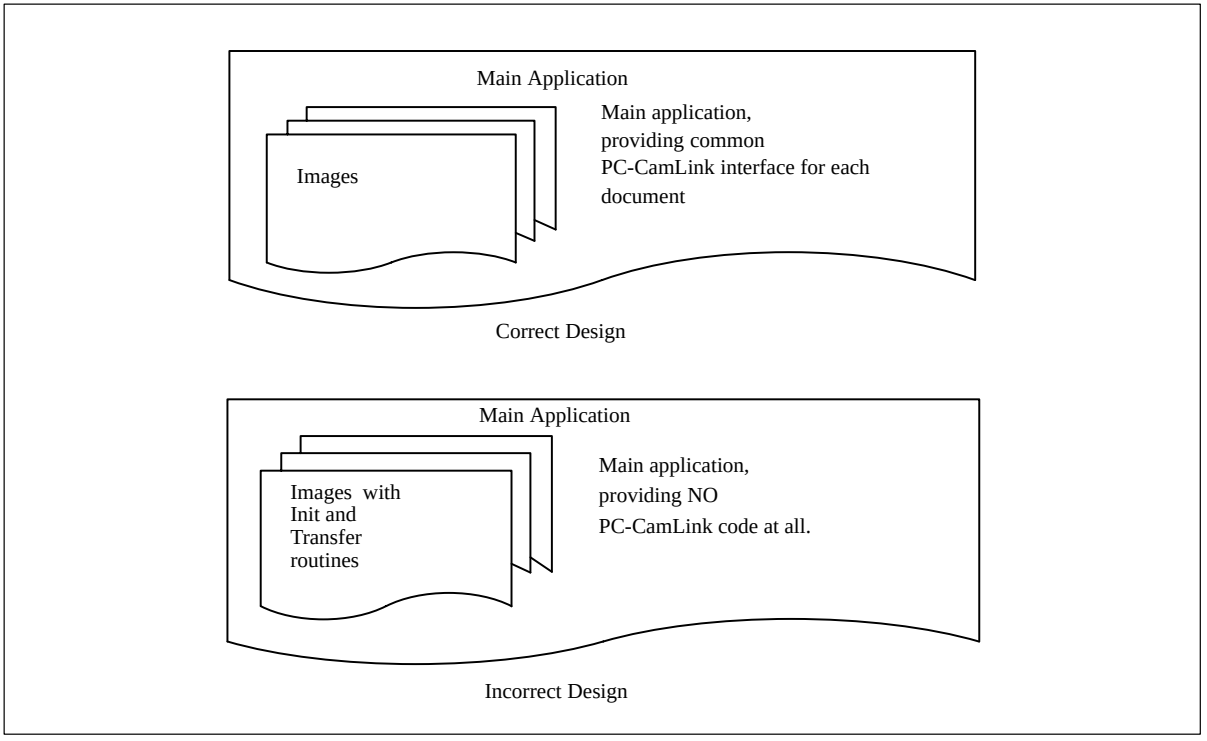


Figure 31. Programming Design

Creating a Project File

Using the IFC Application Wizard in Microsoft C++, we can create the entire framework necessary for a robust windows application. A project can very easily be created using standard project options with the following options specific to IFC:

- Extra Include Directory: *IFCxx*\include
- Extra Library Path: *IFCxx*\lib

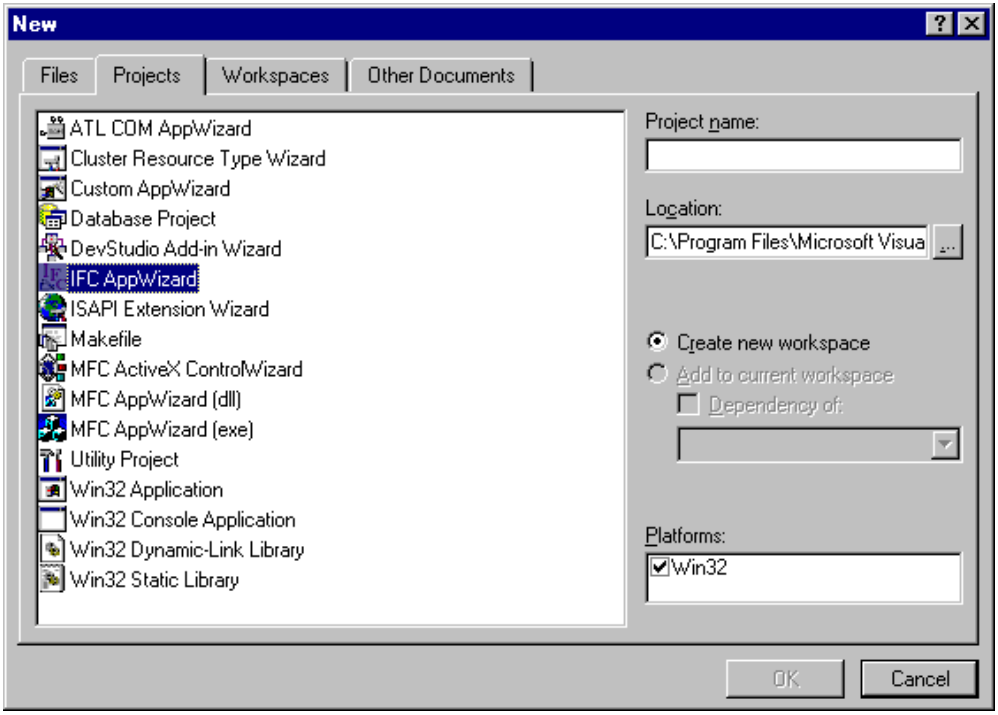
The following steps lead you through creating a project file. The same steps apply to building your own Win32 applications.

Step 1: Start the Visual Studio, pull down the File menu and select New. Select the Projects tab.

Step 2: Select "IFC AppWizard" and fill in the project name. Use the defaults, or point to *IFCxx*\Examples directory, or to your application file directory.

The AppWizard steps you through eight steps to setting up an application. The first six steps are the same as using the MFC AppWizard. The last two steps select video display, and interrupt handling threads and events. Use the defaults for setps 1 through 7. In step 8 select the first interrupt, ICAP_INTR_VB.

Build and then run the application. With a camera attached to the PC-CamLink you can open up to four image windows by selecting WIndow | New Window.



Part 4
Examples

The example programs are intended to illustrate the proper use of the IFC functions. These examples are for your reference only, not meant to be complete applications with all possible error checking and bullet-proofing usually associated with a robust application.

The following examples are provided in the IFC*xx*\Examples directory when you install the IFC software. The directories for all applications are given here, with brief descriptions. The IFC portions of these files are commented in the source files.

Quick Config

Directory: \ifcxx\examples\Qconfig

Quick Config is an example showing how the IFC parameter functions can be used to configure a custom camera. It shows a list of currently available camera files. If your camera is in the list, you can select and display an image. Any of the camera parameters can be changed. This program is a simplified alternative to the Camera Configurator.

Grab Example

Directory: \ifcxx\examples\CGrab

This example illustrates a C wrapper around the IFC library. This example initializes the board and software, and acquires an image into a buffer in host PC memory. This example does not include a display of the captured image.

Interrupt Example

Distributed executable: \ifcxx\bin\seqsnap.exe

Directory: \ifcxx\examples\PC-CamLink\intrex

This example illustrates the use of Triggers and Interrupts, for PC-CamLink. Both hardware and software triggers are accessible. The example displays menus for trigger selection and interrupt monitoring (counters). This example does not include a display of the captured image.

Sequence Acquire

Distributed executable: \ifcxx\bin\seqsnap.exe

Directory: \ifcxx\examples\PC-CamLink\seqsnap

This example illustrates the multiple frame acquire capability of the PC-CamLink board and software. The example also presents options for camera selection, triggering setup, save and read images, overlay drawing, parallel port input and output, LUT display, hardware overlay, window generator (cropping) and hardware zoom (decimation). The captured images are displayed on the PC monitor.

Video & Overlay

Distributed executable: \ifcxx\bin\overlay2.exe

Directory: \ifcxx\examples\overlay

This example illustrates the video display capabilities of PC-CamLink. There are a number of different techniques, discussed in Appendix B of this manual, and each is covered in this example. This example displays the captured image on the PC monitor. You can zoom or stretch the image, and draw rectangles, ellipses and free hand in the overlay.

Large Format Acquire

Distributed executable: ifcxx\bin\largeacq.exe

Directory: examples\PC-CamLink\LargAcq

This example illustrates using a "ring buffer" in host memory and a ping pong buffer in PC-CamLink frame memory. This method is used to acquire very large format images. The snap and grab both use the same ring buffer in host memory. Playing the buffer contents after a grab may show a sequence that jumps backwards, because the ring buffer may not have filled and restarted completely when the grab terminated. The snap sequence always starts and ends cleanly within the ring buffer.

Multi-Threaded Processing Application

Distributed Executable: \ifcxx\bin\imgproc.exe

Directory: \ifcxx\examples\ImProc

This example illustrates processing using threads and interrupt services. This example displays the captured image on the PC monitor. You can open a second image window, and select some basic processing: invert LUT, normal (linear) LUT, mirror, and highlight motion or difference between frames (delta).

Visual Basic Example

Directory: \ifcxx\examples\VB\Vbexamp

This example illustrates a VisualBasic wrapper on the IFC library. Double-click on the VBExamp.vbp file to open the project. Compile and run the example. The captured image is displayed on the PC monitor, with some controls for acquire and LUT processing.

Appendix A
Connectors and Cables

CONNECTORS

PC-CamLink provides three connectors for camera and I/O Ports. In addition, the board has two connectors for the expansion module.

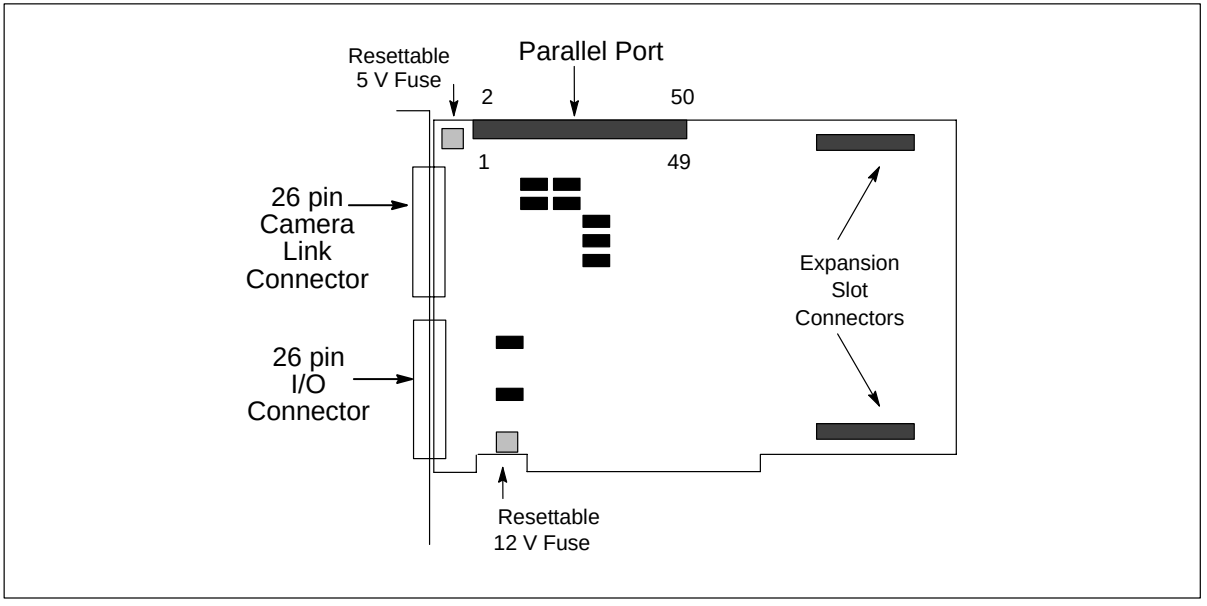


Figure 32. PC-CamLink Board and Connectors

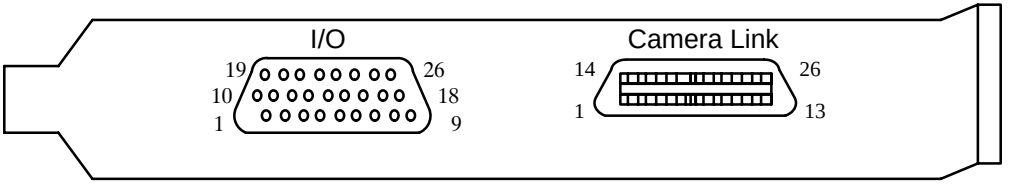


Figure 33. Connector Pin Orientation

Connector Part Numbers

The PC-CamLink Camera Connector is a standardized 26-pin "Mini D Ribbon" (MDR) connector. Standard 26-pin Camera Link cables are available from Coreco Imaging, Inc or from 3M. Contact 3M Interconnect Solutions Division, 6801 River Place Blvd. Austin, TX 78726-9000 800-255-5373

Camera Connector

The 26-pin MDR camera connector conforms to the Camera Link specification. This connector supports camera data, timing, four camera control signals, and serial I/O. All signals are LVDS. This connector supports the Camera Link Base Operation, up to 24–bit data, Ports A, B, and C.

Table 1. Camera Link Connector

Camera	*PC-CamLink*	*Signal name*	*Description*
1	1	SHIELD	Inner cable shield
14	14	SHIELD	Inner cable shield
2	25	X0–	Data Channel 0 neg.
15	12	X0+	Data Channel 0 pos.
3	24	X1–	Data Channel 1 neg.
16	11	X1+	Data Channel 1 pos.
4	23	X2–	Data Channel 2 neg.
17	10	X2+	Data Channel 2 pos.
5	22	XCLK–	Data Clock neg.
18	9	XCLK+	Data Clock pos.
6	21	X3–	Data Channel 3 neg.
19	8	X3+	Data Channel 3 pos.
7	20	SerTC+	Camera Serial Transmit pos. (Cam Tx)
20	7	SerTC–	Camera Serial Transmit neg.
8	19	SerTFG–	Frame Grabber Serial Transmit neg. (Cam Rx)
21	6	SerTFG+	Frame Grabber Serial Transmit pos.
9	18	CC1–	Camera Control 1 neg. (CTL0)
22	5	CC1+	Camera Control 1 pos.
10	17	CC2+	Camera Control 2 pos. (CTL1)
23	4	CC2–	Camera Control 2 neg.
11	16	CC3–	Camera Control 3 neg. (CTL2)
24	3	CC3+	Camera Control 3 pos.
12	15	CC4+	Camera Control 4 pos. (CTL4)
25	2	CC4–	Camera Control 4 neg.
13	13	SHIELD	Inner cable shield
26	26	SHIELD	Inner cable shield

I/O Connector

The 26-pin D-Sub connector contains TTL, Differential, and Opto-coupled programmable inputs and outputs, and 12 Volt power. Jumpers on the PC-CamLink board set the power-up state of the outputs.

Table 2. I/O Connector

Pin #	*Signal name*	*Description*
1	+12_OUT	+12 Volt Power output
2	DIFF_IN0+	Differential input DIFF_IN0 positive
3	DIFF_IN0–	Differential input DIFF_IN0 negative
4	DIFF_IN1+	Differential input DIFF_IN1 positive
5	DIFF_IN1–	Differential input DIFF_IN1 negative
6	XTTL_OUT0	TTL output TTL_OUT0
7	XTTL_OUT1	TTL output TTL_OUT1
8	XTTL_IN0	TTL input TTL_IN0
9	XTTL_IN1	TTL Input TTL_IN1
10	+12_RET	12 Volt return
11	DIFF_OUT0+	Differential output DIFF_OUT0 positive
12	DIFF_OUT0–	Differential output DIFF_OUT0 negative
13	DIFF_OUT1+	Differential output DIFF_OUT1 positive
14	DIFF_OUT1–	Differential output DIFF_OUT1 negative
15	GND	Digital ground
16	XTTL_OUT2	TTL output TTL_OUT2
17	XTTL_IN2	TTL input TTL_IN2
18	GND	Digital ground
19	OPTO_VCC_IN	Opto-coupled outputs VCC input (base pull-up)
20	OPTO_OC_1	Opto-coupled output OPTO_OUT1 (collector)
21	OPTO_OC_0	Opto-coupled output OPTO_OUT0 (collector)
22	OPTO_GND_IN	Opto-coupled outputs Ground input (emitters)
23	OPTO_IN1+	Opto-coupled input OPTO_IN1 positive (anode)
24	OPTO_IN1–	Opto-coupled input OPTO_IN1 negative (cathode)
25	OPTO_IN0+	Opto-coupled input OPTO_IN0 positive (cathode)
26	OPTO_IN0–	Opto-coupled input OPTO_IN0 negative (anode)

Parallel I/O Connector

The Digital I/O is available on a 50-pin dual row post header connector. This provides parallel I/O capability of 8 input and 8 output, high-drive TTL channels. This port allows controlling or monitoring external events. The pin-out is OPTO-22 compatible. All even numbered pins are connected to Digital Ground. Jumpers on the PC-CamLink board set the power-up state of the outputs.

Table 3. 50-Pin Dual-Row Header

Pin #	*Signal name*	*Description*
1	+5V	+5 V power output
3, 5	–	no connection
7, 9	–	no connection
11	STROBE_O	Strobe Output / Output data latch
13	STROBE_I	Strobe Input / Input data latch
15	I/O_INT	Interrupt Input
17	IN7	Digital Input pin 7
19	IN6	Digital Input pin 6
21	IN5	Digital Input pin 5
23	IN4	Digital Input pin 4
25	OUT7	Digital Output pin 7
27	OUT6	Digital Output pin 6
29	OUT5	Digital Output pin 5
31	OUT4	Digital Output pin 4
33	IN3	Digital Input pin 3
35	IN2	Digital Input pin 2
37	IN1	Digital Input pin 1
39	IN0	Digital Input pin 0
41	OUT3	Digital Output pin 3
43	OUT2	Digital Output pin 2
45	OUT1	Digital Output pin 1
47	OUT0	Digital Output pin 0
49	+5V	+5 V power output
2, 4, 6, 8, 10, 12, 14,		Digital Ground
16, 18, 20, 22, 24, 26,		Digital Ground
28, 30, 32, 34, 36, 38,		Digital Ground
40, 42, 44, 46, 48, 50		Digital Ground

CAMERA CABLES

The Camera Link connector is compatible with 3M Cable Assembly 14***x***26–SZLB–***xxx***–0LC.

x is the Shell option: – B= Thumbscrew Shell Kit, T= Thumbscrew Overmold Shell.

xxx is cable length: 100=1 meter, 200=2 meters, 300=3 meters, 400=4.5 meters, 500=5 meters, 700=7 meters, A00=10 meters.

The 3 meter cable 14B26–SZLB–300–OLC is available from Coreco Imaging.

I/O CABLES

The following I/O Cables are currently offered for the PC-CamLink 26-pin I/O Connector.

Table 4. I/O Cables

Cable	*Part Number*	*Description*
Power Cable	509-00078-00	12-pin Hirose connector for +12 Volt power, 1 BNC for TTL_IN0 Trigger 0

The pin-out of the Hirose connector is not guaranteed to work with all cameras. Make sure your camera requires power on pin 2 and ground on pin 1.

CAUTION *DO NOT use the PCVision or PCVisionplus break-out cables with the PC-CamLink. Damage to the PC-CamLink will result.*

Table 5. Power Cable Pin-Out

12-pin Hirose	*Signal Description*	*26-pin I/O*
1	12 Volt return	10
2	+12 Volts DC	1
3–12	no connection	—
BNC connector		
conductor	TTL_IN0 TTL input 0	8
shield	DGND Digital Ground	18

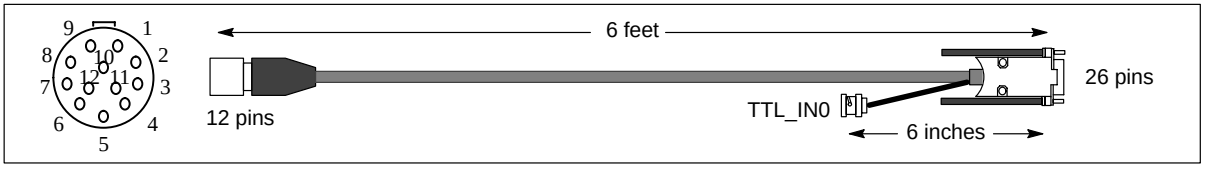


Figure 34. PC-CamLink Power Cable

Appendix B
Video Overlay

Overlay is graphical data, possibly alphanumeric or other symbology, that appears on top of video data on a video display. There are several different techniques for achieving an overlay on video data acquired on Coreco Imaging, Inc.'s image capture products. All techniques described are for "video in a window" on a single monitor driven by the system VGA, as the image capture products have no on-board display. The techniques, up until recently, have provided a destructive overlay, which has disadvantages. These techniques will be discussed for completeness. However, there are now techniques which can provide a non-destructive overlay, which is preferable.

Destructive Overlay

There are two techniques for creating destructive overlays. "Destructive" means that video information and the overlay information reside in the same area of memory. The overlay information is written directly on the video information "destroying" it. The drawback with destructive overlay is, if you want to view live video at 30 fps, 25 fps, or any speed other than static, the video information is also overwriting the overlay information and destroying it.

In the two destructive overlay techniques discussed, there are no inherent limitations on acquisition modules that can be used on the image capture products.

Destructive Mix in VGA Memory

The first technique for implementing destructive overlay is to use Direct Draw to transfer video information to the VGA card. The host then writes graphical data on the video data on the VGA card to achieve the overlay. The graphical data has to be constantly rewritten by the host because new video data is being written over it, usually at 25 or 30 fps. This requires a certain amount of host involvement. Another disadvantage is that the overlay flickers on the display. The video is being written at 30 Hz, the host is writing the overlay however often it has the time to, and the VGA is running at its own speed, probably 72 Hz. This inherent lack of synchronization causes flicker.

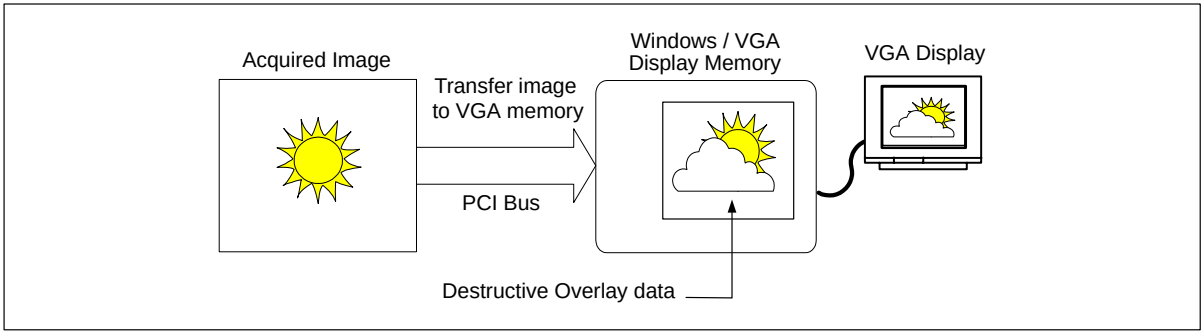


Figure 35. Destructive Mix in VGA Memory

Destructive Mix in Host Memory

The second destructive overlay technique is to use DIB (Device Independent Bit-maps) to transfer video information to the VGA card. A DIB transfer starts by performing a bus master transfer to host memory. The host then writes graphical overlay data on the video data, and transfers the entire frame to the VGA card. This method removes flicker, but requires much more host involvement in transferring and formatting the video data.

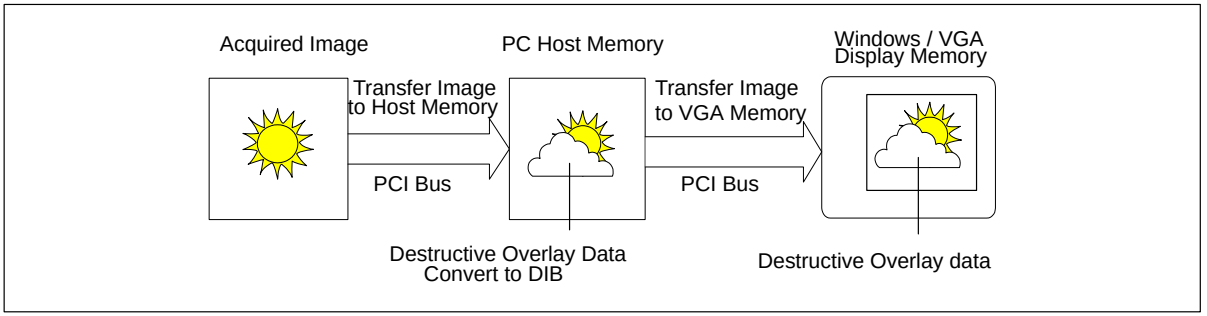


Figure 36. Destructive Mix in Host Memory

Non-Destructive Overlay

The non-destructive techniques use a feature of certain newer VGA chips called "streams processing." This feature allows a VGA chip to parse the memory on the VGA card into two parts called the "primary surface" and the "secondary surface." These surfaces are individually addressable. These two surfaces can be mixed together by the VGA chip for display.

The secondary surface can only be defined as a 16 bit surface. There are several different 16-bit formats to choose from, but 16-bit is the only choice. We use two different 16-bit formats in the two non-destructive overlay techniques. The first is RGB (5,5,5,1); there are five bits each of red, green, and blue information with an extra bit which serves as a kind of mask. If that bit is one, then that pixel will be mixed in for overlay. If it is zero, it will not be. This format is inherently poor at displaying monochrome data since the effective bit depth of an image is 5 bits or 64 gray-scales. It is, however, fine for displaying color as you have a palette of 215 colors. The second format is YCrCb (4:2:2). In this format "Y" is the luminance (monochrome) component and the "Cr" and "Cb" are color components. The term 4:2:2 is a representation of how much of each kind of data (Y, Cr, and Cb respectively) is present in this format. In each 16-bit pixel there are 8 bits of luminance data and 8 bits of color data. An unsophisticated compression is achieved by having adjacent pixels share the same chroma information. Therefore the data effectively looks like: Y0Cr0, Y1Cb0, Y2Cr2, Y3Cb2. All of the luminance data is retained while some of the color data is averaged or shared. This loss of color data actually has little effect on the image as viewed, and therefore is a good format for displaying data. This is a very popular format in graphics and multimedia.

Bus Master to the Primary Surface

The first non-destructive technique is to use Direct Draw to provide video data to the primary surface and put the overlay in the secondary surface. In this method, the secondary surface must be a 16 bit RGB surface. The Direct

Draw of the video data to the primary surface is a familiar technique to achieve a live display. Now, however, it is possible to have overlay data present in the secondary surface and have the two surfaces mixed together by the VGA hardware for display. There is no speed penalty for performing this hardware mix. The overlay data in the secondary surface is never overwritten, therefore it does not have to be constantly updated by the host.

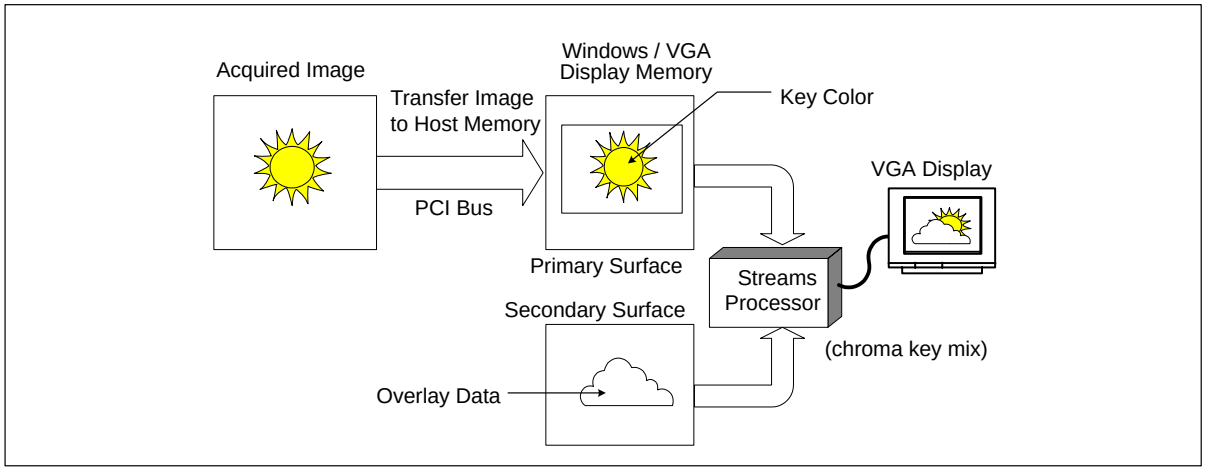


Figure 37. Bus Master to the Primary Surface

Limitations

There are some limitations with this technique. First, there are the normal restrictions incurred with Direct Draw of video data to the VGA card: VGA bit-depth matching and "color dots." As always, we have to match the pixel bit-depth of the output of the frame grabber with the VGA display. If 8-bit monochrome data is being acquired, the VGA card primary surface must be set at eight bits to bus master to it. This equates to a computer display of only 256 colors to be shared by Windows and all of the gray-scale values of your video (which ideally would be 256 just by themselves). Windows will demand a minimum of 20 colors (usually numbers 0-9 and 246-255 in the palette). If video data is bus mastered to the VGA memory with those values reserved for Windows colors, the pixels with those values will take on these colors instead of the gray-scale values that were intended. We call these "color dots" and they can be distracting or unacceptable. The technique used to remove color dots is to clip the data with a Look-Up Table before it is transferred. This leaves only 236 colors left for gray-scale video which reduces the contrast of the video somewhat. It also may be said that limiting Windows to its minimum 20 colors makes for an unattractive Windows display. If 24-bit color data is being acquired, then the VGA primary surface must be set at 24 bits. This presents fewer "color dot" problems. The color video data can be clipped by a look-up-table to leave room for the 20 Windows colors (and more if desired) with negligible degradation to the video quality (20 colors out of 224). There is a restriction on the overlay as well. The overlay is in the secondary surface of the VGA card and is not being managed by Windows. Windows keeps track of all of the objects in the primary surface so that they all remain "good Windows neighbors." Pull down menus, dialog boxes, tool bars, main windows, video in a window (bus mastered data) etc.,

are all managed so that the proper components are on top, in the background, in the foreground, and displaying properly. Windows does not keep track of the secondary surface where the overlay data resides, a problem arises if any Windows graphics such as a dialog box intrudes on the same area of the screen as the overlay. The overlay will be turned off by the VGA card if the graphics encroaches on its area, even if just by a few pixels. This can cause problems with the application's GUI.

Bus-Master to the Secondary Surface

The second non-destructive technique is to bus master video data to the secondary surface and put the overlay in the primary surface. In this method the secondary surface must be a 16-bit YCrCb 4:2:2 surface. The frame grabber must have 16-bit YCrCb data to bus master to this surface. Coreco Imaging, Inc.'s image capture products can provide data in this format. This 16-bit data can be either color data from an NTSC or PAL camera, or monochrome data from an RS-170 or CCIR camera.

The overlay is accomplished by mixing in video data from the secondary surface into the video window of the application GUI in the primary surface using a chroma key. A specific chroma value is designated as the hardware chroma key and wherever this value appears in the primary surface, the video data from the secondary surface is substituted for it in the display. The application GUI, overlay, and all windows graphics reside in the primary surface and the video is mixed in wherever it is needed by placing the chroma value. There are no problems with "color dots" using this technique. There are also no problems with the overlay being "turned off" as in the previous technique since Windows is managing all of the graphics while video is just being quietly mixed in by the chroma key. The primary surface can be run at whatever bit-depth is best for the application.

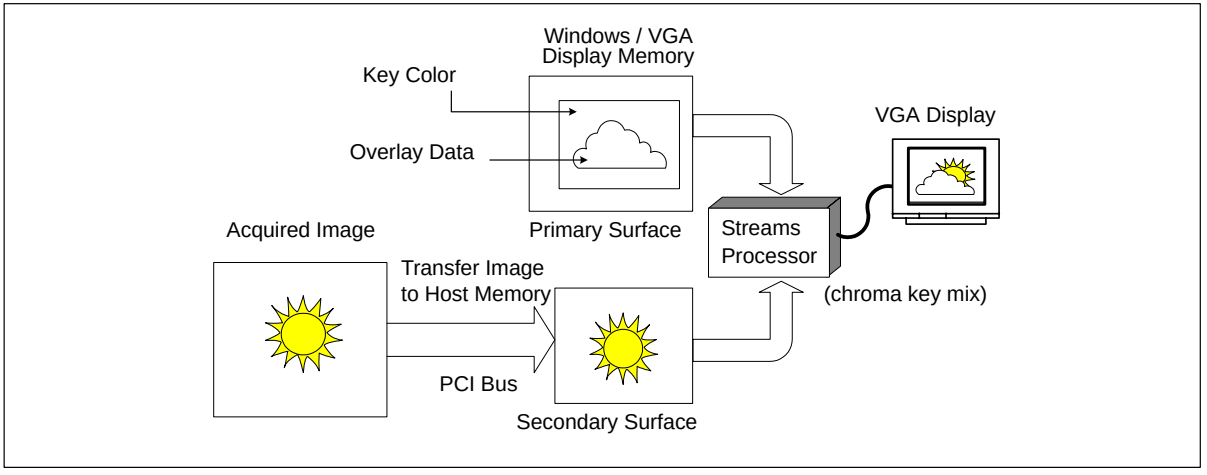


Figure 38. Bus Master to Secondary Surface

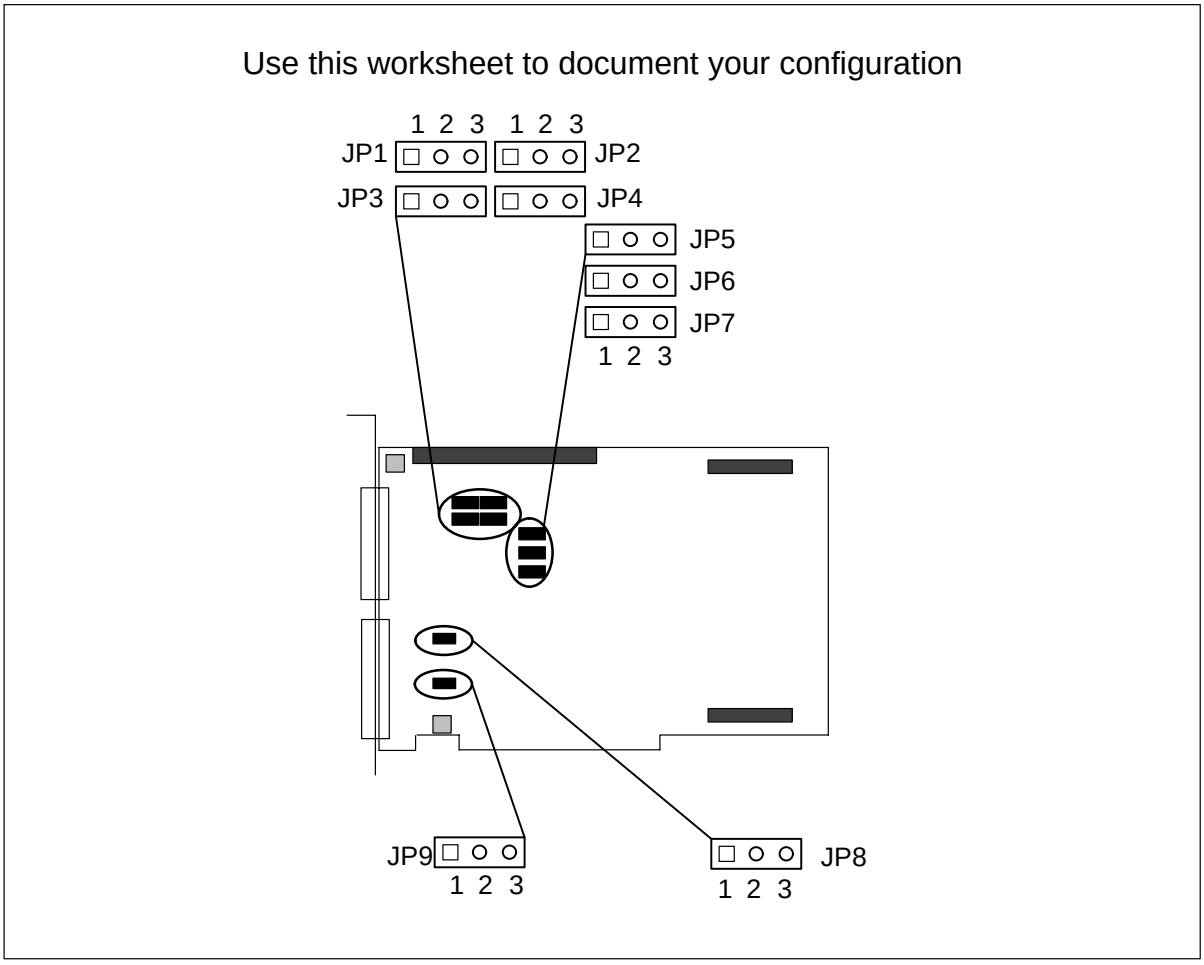


Figure 39. PC-CamLink Configuration Worksheet