FAST-X

TECHNICAL PRODUCT DESCRIPTION

30002-00192

COPYRIGHT NOTICE

Copyright ? 2001-2006 by Alacron Inc.

Document Name: Fast-X Technical Product Description

Document Number: 30002-00192

Revision History: 1.0 October 25, 2006

Trademarks:

Alacron Inc.
71 Spit Brook Road, Suite 200
Nashua, NH 03060
USA

Telephone: 603-891-2750
Fax: 603-891-2745

Web Site: http://www.alacron.com

Email: sales@alacron.com or support@alacron.com

TABLE OF CONTENTS

OTHER ALACRON MANUALS

Alacron manuals cover all aspects of FastSeries hardware and software installation and operation. Call Alacron at 603-891-2750 and ask for the appropriate manuals from the list below if they did not come in your FastSeries shipment.

30002-00148 ALFAST Runtime Software Programmer's Guide & Reference

30002-00150 FastSeries Library User's Manual

30002-00169 ALRT Runtime Software Programmer's Guide & Reference

30002-00184 FastSeries Getting Started Manual

30002-00185 FastVision Hardware Installation Manual

30002-00186 FastVision Software Installation Manual

30002-00395 FastMotion User's Manual

INTRODUCTION

The Alacron Fast-X is a highly flexible, programmable and expandable video capture and processing device. Features of this device target the needs of original equipment manufacturers and end users who anticipate a demand for diverse I/O requirements and the need to support a high video data bandwidth.

The Fast-X is available with both Camera Link™ and GigE Vision™ digital video interfaces in various configurations ranging from the Full Camera Link to a combination of Medium, Base, or up to three asynchronous Base Camera Link connections. The four bi-directional GigE Vision interfaces are available to receive video streams from GigE Vision-enabled cameras or distribute video streams to other video processing devices and computers.

The Fast-X data-flow architecture combines parallel bit-processing in the front-end FPGA with the Stretch S5610 software-configurable processor to execute complex image and digital signal processing concurrently with the video capture.

The Fast-X is able to transfer video data either directly to the Host PC system memory or add on-board video processing, object detection or video compression. Such on-board processing could dramatically reduce resource demands on the Host computer system and carry important benefits.

Reduction in computational and data carrying loads on a Host computer enables this video capture system to handle a higher video bandwidth from faster, higher-resolution, or from multiple simultaneous sources of video data. The on-board video processing will prevent the Host computer system from experiencing unpredictable stalls and from losing video frames caused by the memory buffer overload.

It is even more important for success on the emerging GigE Vision™ digital video interface standard. GigE Vision interface makes use of commodity computer network interface and network hardware. It is rightfully expected to give vision application developers an unparalleled flexibility and tremendous potential cost savings by:

- Connecting a GigE Vision camera to any computer with the GigE Interface and appropriate driver,
- Sharing of multiple GigE Vision cameras among computers on the network,
- Video streaming over the GigE network between GigE Vision-capable computer systems for distributed video processing and archiving.

Full utilization of the modest GigE Vision bandwidth and consequently many design opportunities for the GigE Vision devices depend on the support from computer hardware that can compress raw video into the GigE fabric. Only computer-assisted GigE Vision transmission will assure that all frames make it through the network.

The intelligent and programmable data-flow-through architecture of the Fast-X bridges the high-bandwidth of current Camera Link interface and the bright future of the multi-lane GigE Vision fabric.

The fully programmable architecture of Fast-X enables an extensive customization by the end-user or OEM including storage of proprietary or often used application extensions in the on-board Flash memory.

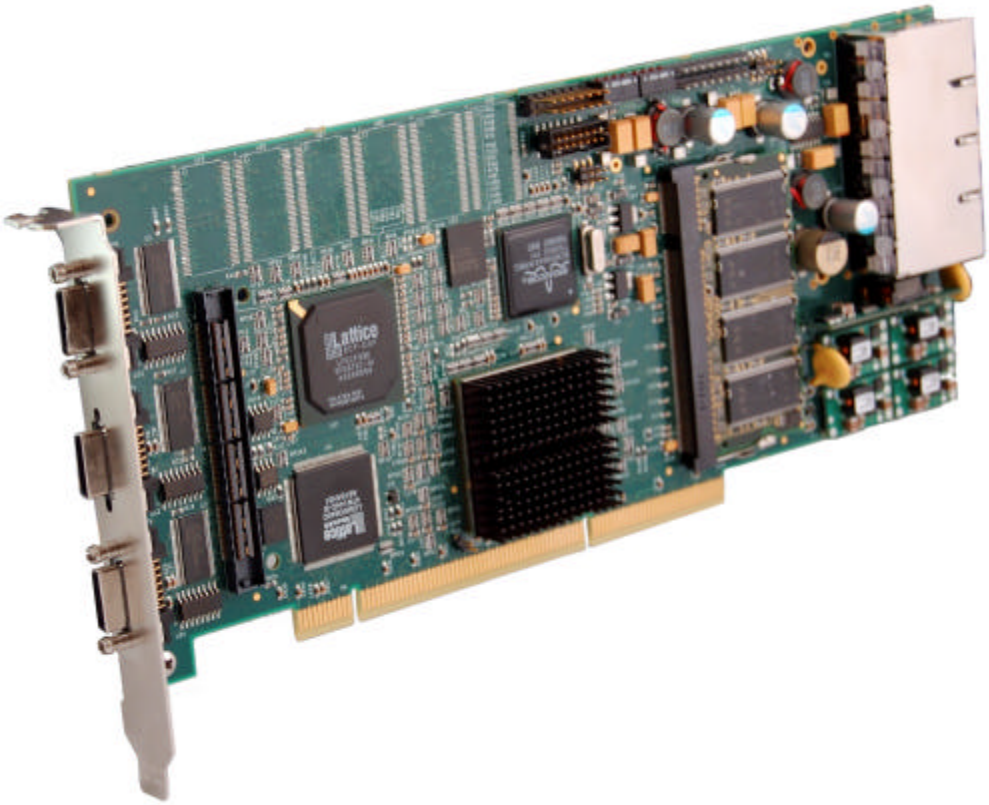

FIGURE 1 - FAST-X PCI-X BOARD

FAST-X FEATURE SUMMARY

Board	Half-length, PCI-X rev.2.3 (64-bit, 133 MHz)
Video I/O Connections	? Three (3) 85MHz Camera Link SDR-26 connectors that can be used with: - 3 Basic Camera Link cameras - 1 Basic and 1 Medium Camera Link cameras - 1 Full Camera Link camera - 1 Trans-standard Camera Link camera with 10 taps by 8bit ? Four (4) Gigabit Ethernet RJ-45 ports will support up to four simultaneous GigE Vision video streams
Video I/O Extender card	Mezzanine expansion card with 152-pin VHDCI connector. Plugging-in a Video I/O Extender daughter

	card replaces standard video connections with a variety of analog and digital video formats
Video Processor	? Stretch S5610 with 2GB, 400MHz DDRAM memory, 300MHz Xtensa CPU core, two reconfigurable Instruction Set Extension Fabric (ISEF) units ? Lattice ECP2 FPGA with 6 banks of 64MB 16-bit, 400MHz DDR memory each for the on-board image-buffering and formatting.
Software Development	? Microsoft Windows XP .NET Framework ? Stretch C Integrated Design Environment (IDE) ? Alacron ALRT run-time libraries, libJPEG, libTIFF, and other video compression and processing libraries
System Requirements	PCI-X compliant system running MS Windows XP SP2 and higher with at least 1 GB system memory
Temperature	0^{o}C (32^{o}F) to 55^{o}C (131^{o}F) Relative Humidity: up to 95% (non-condensing)

OPTIONAL FEATURES

? Video I/O Expansion mezzanine daughter card

? Unassisted by Host Fast-X operations made possible by local application boot of the Stretch S5610 processor from 32MB of on-board FLASH.

HARDWARE OVERVIEW

This chapter provides an overview of Fast-X hardware functional units and describes their operations.

FAST-X BLOCK DIAGRAM

The key to the Fast-X video data handling and video processing capabilities is its hybrid architecture that augments processing power and data handling of Stretch S5610 Software-Configurable DSP Processor (SCP) with the extended bit-handling by the Lattice ECP FPGA.

The front-end Lattice FPGA provides video data buffering, formatting, and steering. It contains multiple state machines needed to handle structured video information in the input streams. The front-end FPGA counts and time-stamps video frames before passing them to the Stretch processor. Triggering and camera strobes can be generated from the on-board software or from external inputs to the board. The data-path FPGA is configured by the Stretch processor from the program file stored in local Flash memory or received from the Host.

The Stretch SCP is a multi-functional device, a true System-On-Chip (SoC). It is a hub of Fast-X that collects, processes and distributes video data helped by its novel design and the fast data interconnect on the chip. Stretch SCP supplies large number of high-speed computer interfaces that move video data between local and external devices. It also provides image-processing capabilities using two FPGA-like programmable fabrics extending its traditional processor architecture. The sizeable, up to 2GB and fast Stretch SDRAM memory has sufficient bandwidth to buffer video data streams during capture and processing.

The Fast-X uses two UARTs in the Stretch processor to communicate with external video devices, including CameraLink cameras. Serial links allow machine vision applications to control and configure cameras and other video devices.

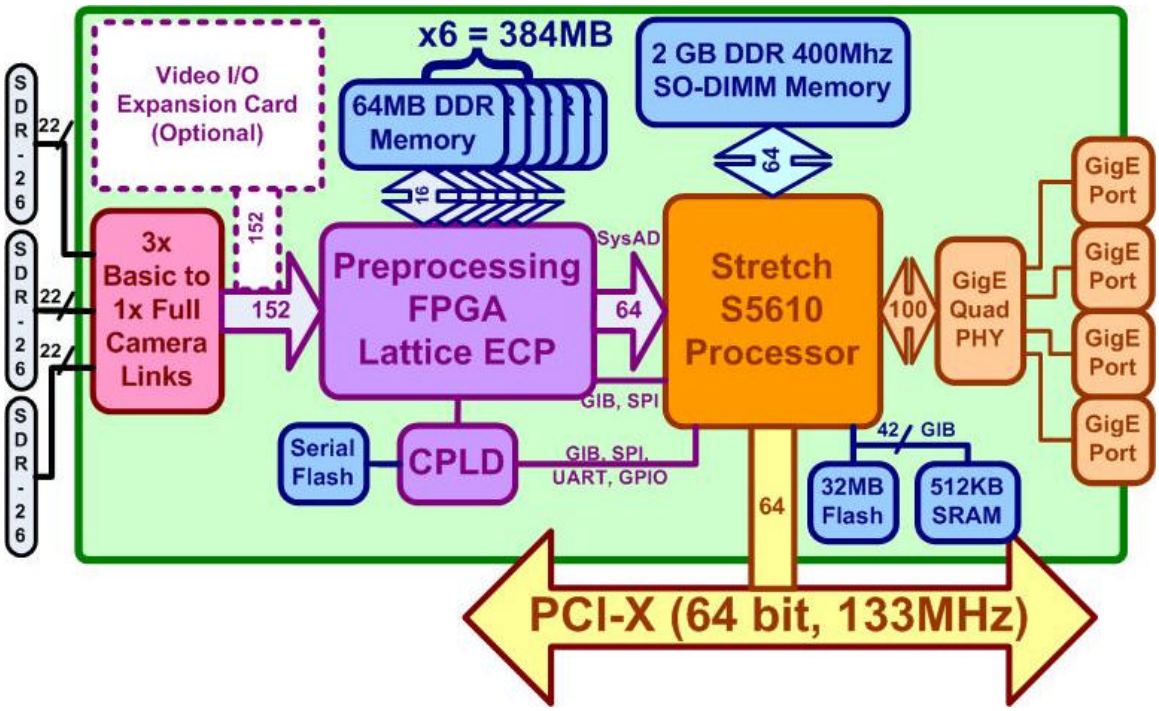

Figure 2 – Fast-X PCI-X Block Diagram

VIDEO DATA INPUTS

The right choice of video data inputs is essential for successful implementation of machine vision applications. It must take into account video data bandwidth, maximum length and flexibility of the cables it supports, its ability to control video device operations and last but not least, its cost. Fast X default configuration makes the choice of multiple Camera Link and GigE Vision interfaces. Alacron's Fast-X video data inputs offer machine vision developer, huge bandwighth with considerable flexibility at a reasonable cost.

Other video input formats provided on the optional Video I/O Extender board connect directly to the FPGA front-end via mezzanine connector. Contact Alacron for assistance from the engineering staff if your application(s) call for additional specific device drivers.

Camera Link Digital Video Interface

Camera Link was developed by a consortium of camera and video device manufacturers to ensure physical compatibility of devices and availability of standard boards, cables and connectors. Each Camera-Link interface supports 24 bits of data, and four bits of control, as well as the bi-directional serial communication interface and CC1 through CC4 signals. See "Specification of the Camera Link Interface Standard for Digital Cameras and Frame Grabbers", Version 1.1, January 2004.

Termination for all Camera-Link signals is provided on the board. The Fast-X board supports up to three (3) Camera-Link interfaces, each of which can run at up to 85 MHz.

.

	J1	J2	J3
1	CL Base	CL Base	CL Base
2	CL Medium		CL Base
3	CL Full		N/A

Table 1 – Fast-x Camera Link Configurations

Alacron can supply standard Camera-Link cables to be used with the Fast-X and Camera Link cameras. The two types of Camera Link cables connect the board's SDR-26 Camera-Link interfaces to the Camera Link devices equipped with either MDR-26 or SDR-26 connector. The actual pinout of the Fast-X Camera Link interfaces is contained in **Appendix A.**

GigE Vision Digital Video Connect

The GigE Vision Standard was developed under sponsorship by the AIA (Automated Imaging Association) since June 2003 by a committee of machine vision equipment manufacturers. It defines communications interface for vision applications based on ubiquitous Ethernet technology. The full description of the current GigE Vision Standard can be found in the GigE Vision Camera Interface Standard for Machine Vision, Draft 1.6, February 28, 2006.

GigE Vision Standard enables a wide spectrum of interconnect topologies between compliant video capture and processing devices. Even if its name refers explicitly to Gigabit Ethernet it can be implemented over other Ethernet modalities and speed grades. The underlying Ethernet technology is compatible with copper-wired, wireless, and fiber-optical interconnects opening the door to many innovative applications of machine vision.

Stretch SCP has four built-in Gigabit Ethernet Media Access Controllers (GMACs) supported by the dedicated DMA controllers. The Stretch GMAC devices are able to transmit data at speeds

near the physical line speed. The tight integration of GMACs with the SCP processor and memory system eliminates the need to make a copy of data before transmission. The Ethernet DHCP and UDP stacks running on the Stretch processor and a physical media interface or PHY chip is all that is required to establish GigE Vision connection.

The Fast-X board carries a block of four standard RJ-45 Ethernet connectors with the built-in LED devices displaying connection status and activity of each GigE line. Standard CAT-5e or CAT-6 cables should be used for the reliable GigE Vision connections.

FPGA FRONT-END VIDEO PROCESSING

Efficient handling of the large amount of video data carried by three Camera Link connectors requires Front-End processing capable of performing the necessary video data parsing and transformations. The Fast-X high-speed video capture board uses large Lattice ECP FPGA as the Front-End processor that massages incoming video data and hands them over to the Stretch processor for further processing and disposition via selected interface.

In its operations the Front-end processor uses the four essential ***enable*** signals carried by the Camera Link interface alongside the data bits. They are:

- FVAL – Frame valid
- LVAL – Line valid
- DVAL – Data valid

Capture state machines running in the Front-End processor use four enable signals to parse video frames and lines for Line Cameras before any processing is done. Additional video capture service functions in the Front-end processor time-stamp and annotate the headers attached to each video frame. Front-end processing can include compiling histogram, finding maxima and minima, and pixel-wise table transformations. Depending on the video bandwidth, the Front-end processor could perform video data alignment, and even morphological operations that simplify user applications. Lattice ECP FPGA family includes pin-compatible devices of varying capacity. Proper device from the Lattice ECP family can be matched to user's requirements of the Front-end processing increasing board's capabilities if desired or keeping it simple for lower equipment costs.

Efficiency on the Front-end processing in the Lattice ECP FPGA is supported by the 6 independent banks of attached fast 400MHz DDR memory. Multiple independent 16-bit memory ports provide huge aggregate data bandwidth delivering preloaded data tables and buffering intermediate results in the attached fast memory. This memory is optional since most of the capture operations are straightforward and structured video data get transferred by the DMA from the Front-end directly into the Stretch SCP memory.

STRETCH APPLICATION PROCESSOR

Overview

Stretch S5610 Software-Configurable Processor (SCP) role in the Fast-X design is to realize a wide class of data-intensive computations without impeding video data flow through the board. SCP does it by integrating on the same chip a DSP-like processor with a variety of high-speed I/O options:

- The two main 64-bit high-bandwidth busses: PCI-X and SysAD move large amounts of data in and out of the chip,
- The high-speed on-chip data interconnect moves data streams between various chip peripherals without mutual blocking,
- The high-speed 64-bit interface for the large DDR400 SDRAM memory,

- Internal blocks of SRAM, Instruction and Data caches serve to increase data reuse and availability,
- Four Independent bi-directional FIFOs or GMACs,
- Long list of low and high speed peripheral devices - UARTs, High-speed Serial Peripheral Interface (SPI), GPIO, Generic Interface Bus (GIB) and others.

Stretch SCP is a novel hybrid microprocessor architecture. It combines a well-defined Xtensa CPU design from the Tensilica, Inc. and two FPGA-like, flexible and re-programmable, wide data processing devices that implement application-specific processing-intensive algorithmic kernels as user-programmed Extended Instructions. By processing data items in parallel these Instruction Set Extension Fabric (ISEF) devices deliver orders of magnitude boost in the data processing speed just where it is necessary.

Similarly to FPGA, the ISEF units are most effective when processing fixed-point data. Floating-point data processing should be done using Xtensa built-in 32-bit Floating-Point Unit.

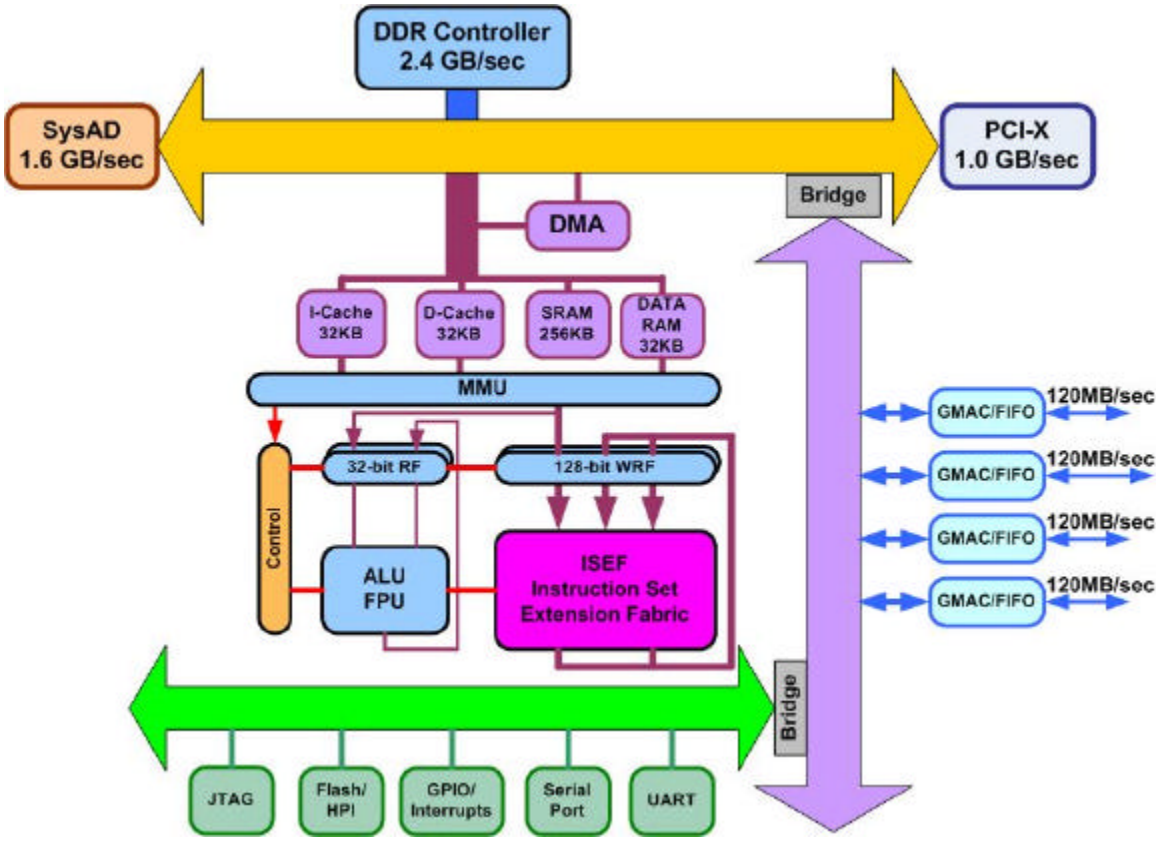

Figure 3 – Stretch S5610 Block Diagram

Xtensa Core

The Xtensa processor core is a modern RISC microprocessor targeted at embedded applications. Its well defined Instruction Set Architecture (ISA) offers industry-leading code density, enables high performance and low-power. Even more important that its name reflects ease with which this ISA can be extended to enable a seamless integration with a tightly coupled co-processor. Xtensa core runs in a lockstep with its co-processors issuing them instructions, providing them with data and picking-up results. It can also change their configuration re-programming them on the fly with a minimal latency to a different set of operations.

Instruction Set Extension Fabric (ISEF)

The Stretch Extension Unit is the key to its DSP-like performance. It is tightly integrated within the chip and consists of the following:

? The Wide Register File – a set of 128-bit wide registers for holding data (three ports for read and one port for write from inside the ISEF fabric)

? The Special Registers – a collection of support registers for a variety of functions

? Two Instruction Set Extension Fabric (ISEF) units – an FPGA-like configurable set of bit-slice computational resources that implement Stretch Extension Instructions.

? The SCP 128-bit internal bus can load and store a wide register each cycle, if data is available from the Data cache or blocks of internal data SRAM, a little bit longer if data is stored in the external SDRAM.

? Each ISEF fabric can be configured to support eight Extended Instructions for a total of 16 per ISEF unit.

? An Extension Instruction can be issued every third cycle in the S5610 to be executed concurrently with other Extended Instructions in progress.

An Extended Instruction can be quite complex and will take multiple cycles to complete instruction depending on its code. Once issued, an Extended Instruction continues its computations concurrently with the base Xtensa ISA processing. The S5610 allows up to 31 clock cycles per Extended Instruction. The number of Extended Instructions in an application is not limited to a single ISEF set since one ISEF fabric can be re-loaded with new Extended Instructions while the other goes on processing.

Results of an Extended Instruction can be stored in the Wide Register File to be used by the following ISEF instructions or stored in one of the SCP memories. The need to feed data to the ISEF unit and fetch results of processing via the SCP 128-bit interconnect is often the real determinant of the ISEF computational performance. Extensive use of the Wide Register File and internal memories for intermediate results can help but will not eliminate the necessity of maintaining the data flow through the SCP at the rate specified by an application and the external video data sources.

User programs the ISEF unit in C programming language by identifying kernels of data-intensive iterative computations and designating kernels to be placed in the ISEF and replaced by an Extension Instruction in the Xtensa code flow. There is a trade-off between amount of ISEF resources used by an Extended Instruction and a number of such instructions directly addressable without reloading. An application must explicitly load ISEF instructions to make use of them.

To maximize execution speed of an Extended Instruction, the programmer can analyze its interlocks with the Pipeline Analyzer tool and make changes in the instruction flow to avoid resource contention. The resource savings can be significant and more than double the speed of execution. Ultimately, the ISEF computational resources are finite and the user-defined Extended Instruction can run out of the available bit-slices necessitating instruction re-design or even splitting in two.

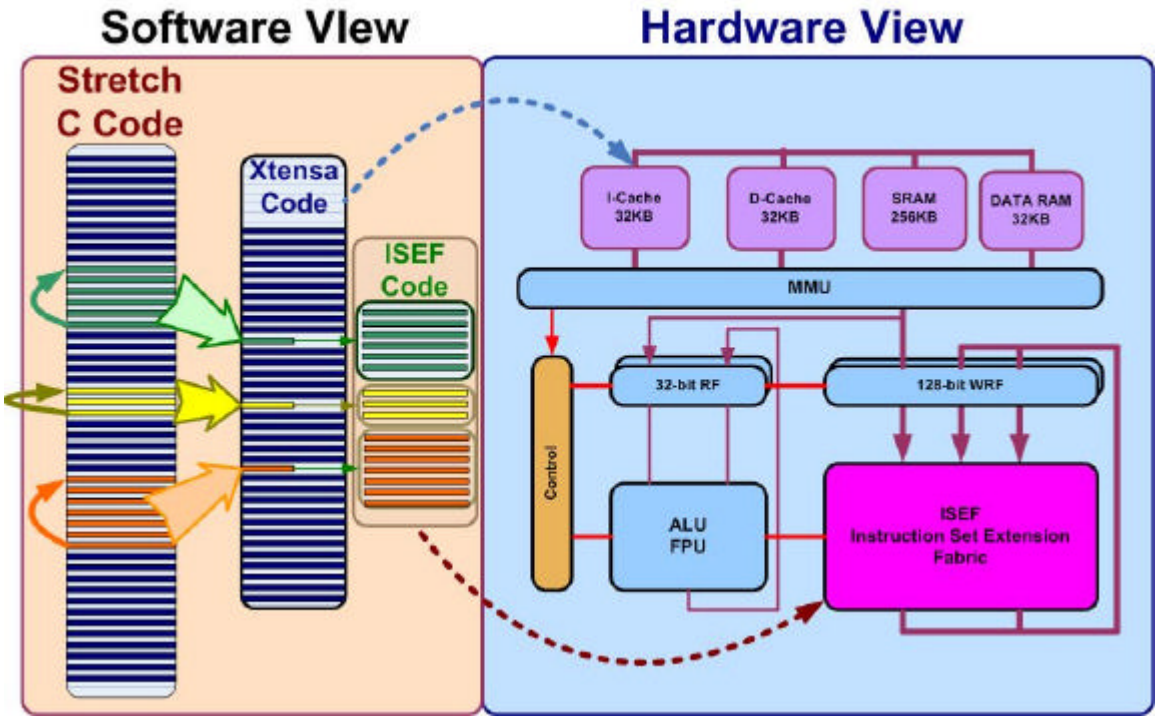

Figure 4 –C kernels transformed into Stretch ISEF Extended Instructions

Using C programming language to implement data-parallel computations in the ISEF unit concurrently with Xtensa ISA is a very useful feature that is very effective in the DSP and ISP-type data-intensive applications.

Memory System

The SCP memory architecture is designed to support multiple simultaneous data transfers to various internal functional devices. It includes:

? 256KB on-chip single-port SRAM

? 32KB dual-port data SRAM

? 32KB Data cache

? 32KB Instruction cache

? DDR SDRAM controller to up to 3GB of external SDRAM

The SCP DDR SDRAM controller interfaces to the various available DDR400 memories. It supports the following features:

? Clock frequencies from 100 to 200 MHz

? Fully pipelined command, read, and write data interface

? Advanced bank look-ahead for high memory throughput

SysAD Interface

The SysAD bus interface allows the SCP to be a co-processor to the front-end FPGA with the FPGA using DMA data transfers directly in the SCP memory. The FPGA operations are

controlled by the SCP via the Stretch-accessible registers in the FPGA. The SysAD bus features include:

? 1+ GB/sec bandwidth – 64-bits at 200 MHz

? Multiplexed address-data bus

? 36-bits pf physical address

? 64-bits of data

? Support for two outstanding read requests, which can be returned out of order

? Simple state machine that uses minimal amount of resources in the FPGA

GigE MACs

The Stretch Gigabit Ethernet MACs connect directly to the chip data interconnect and communicate with an external Quad-PHY chip via the standard Gigabit Media Independent Interface (GMII). Each GMAC has the built-in Transmit and Receive FIFO and two dedicated channels in the SCP DMA controller that moves data in and out of FIFOs.

S5610 Low and Mid-speed Peripherals

Stretch S5610 has a variety of built-in low and mid-speed peripheral devices that simplify construction of a rich embedded system around this chip and enhance its functionality.

The mid-speed peripheral interfaces include one Generic Interface Bus (GIB) that is used on Fast-X board to control setting on programmable parts and two Streaming Serial Ports (SSPort) that are not used.

Among the low-speed interfaces are two Universal Asynchronous Receiver/Transmitter ports that control settings of the CameraLink devices.

Interrupt Controller

For detailed information about the Stretch S5610 SCP, the user is referred to the SCP Architectural Reference, parts 1, 2 and 3 and to the S5000 Peripheral Reference. A full set of Stretch documentation is available with the Stretch IDE toolset.

All on-board interrupts from the Fast-X various devices are forwarded to the Stretch processor and will be handled locally. This configuration allows much faster interrupt processing under the SCP real-time OS environment and protects HOST OS from an excessive amount of interrupts it is not well equipped to handle.

PCI-X INTERFACE TO HOST COMPUTER

The Fast-X interface to the PCI-X Host bus is provided by the Stretch SCP internal PCI-X controller.

The Stretch PCI-X supports the following features

? Compliance with PCI-X Bus Specification Rev. 2.3

? Compliance with PCI-X Addendum Revision 1.0a

? 64-bit PCI-X interface

? Synchronous 0–133 MHz PCI-X-to-application clock frequencies

? Support for up to 2 Master Deferred Read transactions

- Support for up to 8 Target Delayed Read transactions
- Support for up to 16 Target Split Read-Write transactions
- Support for up to 2 Master Split Read-Write transactions
- Programmable PCI Target Burst Read disconnect (from 16 KBytes to 64 KBytes, aligned)
- PCI Target Burst size of up to 2 MBytes, aligned
- PCI and PCI-X Master Burst size of up to 256 KBytes (2 MBytes, aligned)
- PCI-X Target Burst size of up to 256 KBytes (2 MBytes, aligned)
  - PCI-X interface connector is keyed to 3.3V signaling

FAST-X POWER-UP SEQUENCE

The Fast-X software-controlled procedure for power-up and module initialization follows this sequence:

- 3.3V Power Applied
- 2.5/1.8V power stabilizes. The time this takes depends on the 3.3V ramp-rate.
- CPLD configures from the attached serial FLASH.
- STRETCH S5610 boots from the FLASH memory
- Host processor performs BIOS scan and configures STRETCH S5610 PCI-X resources.
- Host processor loads STRETCH program into STRETCH SDRAM, and starts STRETCH application.
- STRETCH initialization code configures Front-End FPGA.
- Software initializes on-board registers via Stretch GPIO controller.
- Data capture may begin.

SOFTWARE OVERVIEW

The Fast-X is delivered fully functional, bundled with the video capture program, ***Fast Motion*** that supports video preview and permits saving captured video stream on disk. ***Fast Motion*** includes an assortment of the camera definition files for the most popular models. Fast Motion allows the user to modify many video capture parameters: frame rate, frame exposure, frame size, position of Region-Of-Interest in a frame, and others. The user can designate which directory or folder to store captured video.

The Fast-X significant on-board computational resources qualify it as an embedded computing system with DSP and ISP capabilities. It offers to the machine vision application developer or OEM a whole new dimension in customization. There are two software components that can be modified by the user: Host-resident and Board-resident software. The two software components communicate and synchronize their activity via the services provided by the Alacron Real-Time operating system (***ALRT***) residing on the Fast-X board with the ***ALRT*** driver running on the HOST computer. A Fast-X application developer writes both software components in the C/C++ development environments for the HOST processor and for the Stretch SCP using corresponding components of the ***ALRT*** Board-Support Package.

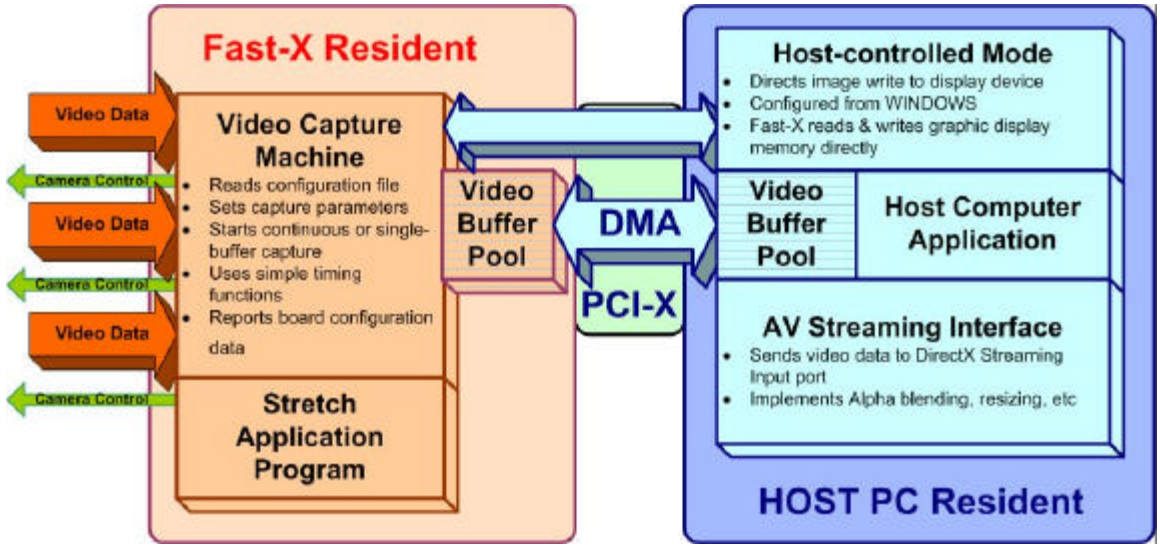

Figure 5 – Fast-X Software Run-Time Configuration

The first version of the Fast-X runtime software is available under Microsoft Windows XP operating systems. Alacron offers Board Support Packages for Microsoft Windows and Linux OS environments.

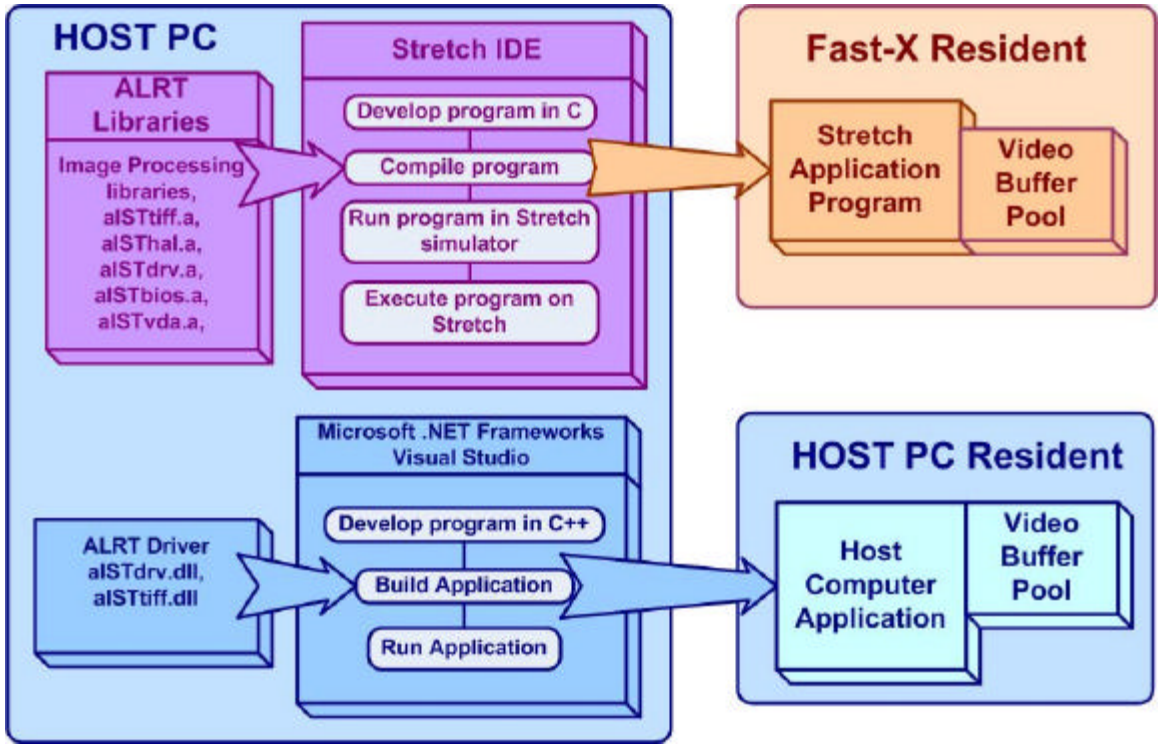

Figure 6 – Fast-X Software Development Environment

HOST SYSTEM REQUIREMENTS

In order to support operations of the Fast-X board and to develop custom applications the following requirements must be fulfilled:

- An Intel or AMD computer system compatible with the Microsoft Windows XP SP2 or later
- An unused PCI-X 64-bit 100 or 133 MHz slot on the motherboard
- Minimum 1GB of DDR or DDR2 memory installed
- Acrobat Reader program to read documentation
- Microsoft .NET Visual Studio Software Development Environment
- The Stretch IDE toolset
- ALRT Fast-X runtime software for Windows? XP operating systems.
- Sufficient air circulation through the enclosure of at least 200 LFM
- The operating temperature range of the Fast-X boards is 0? Celsius to 40? Celsius.

HOST SOFTWARE DEVELOPMENT SUPPORT

The Microsoft Windows .NET Framework delivers a widely used Microsoft Visual Studio Integrated Development Environment. In order to use Visual Studio to develop software that works with the Fast-X the user must import header files with parameters and the ***ALRT*** driver call definitions. The user will need to modify the project settings to include the ***ALRT*** libraries ***alstdrv.dll*** and ***alsttiff.dll.***

STRETCH SOFTWARE DEVELOPMENT

Software development for the Fast-X on-board applications targets the Stretch SCP and uses the Stretch Interactive Development Environment (Stretch IDE) toolkit. The Stretch IDE conforms to established software development practices and exists for the Microsoft Windows XP and Red Hat Linux environments.

The Stretch IDE runs on the HOST PC and supports all the major components necessary for a successful software development:

- Creating and managing software projects
- Managing Debug and Release build targets
- Creating and editing files
- Debugging on the HOST PC with the cycle-accurate instruction set simulation (ISS)
- Debugging on the Fast-X board
- Profiling code and detecting data-intensive execution kernels
- Analyzing the SCP instruction pipeline and optimizing ISEF resource usage

An extensive set of libraries support Stretch operations on the Fast-X board related to video capture and processing. The set includes the ALRT functions managing on-board activities and communications with the HOST PC (***alsthal.a, alstdrv.a, alstvda.a, alstbios.a***), as well as video compression and formatting for the archiving (***alsttiff.a, alstjpeg.a***).

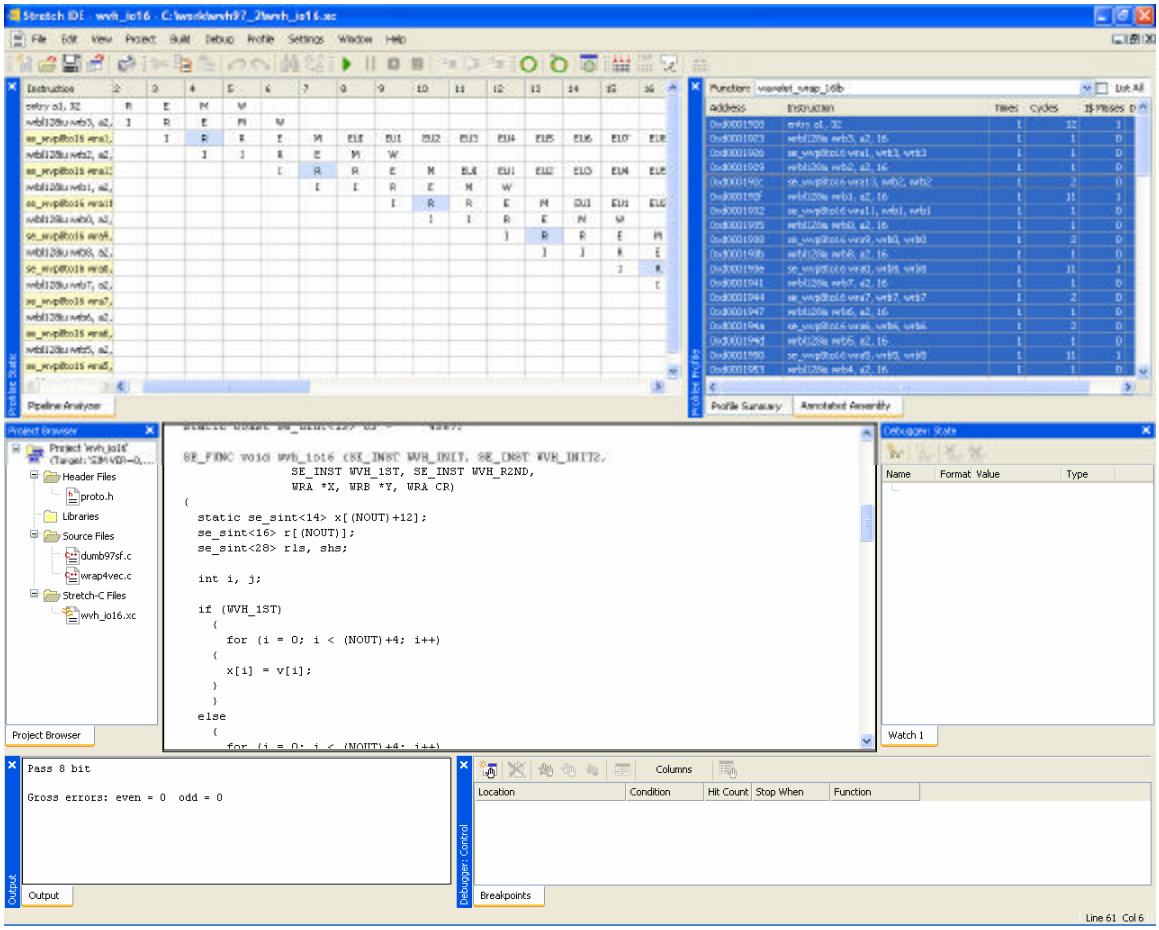

Figure 7 – Fast-X Stretch IDE

The picture in the upper left panel of the Figure 7 shows a detailed view of the SCP instruction pipeline including concurrent execution of Extended Instructions in the ISEF unit.

The typical development flow for the SCP looks like the diagram below.

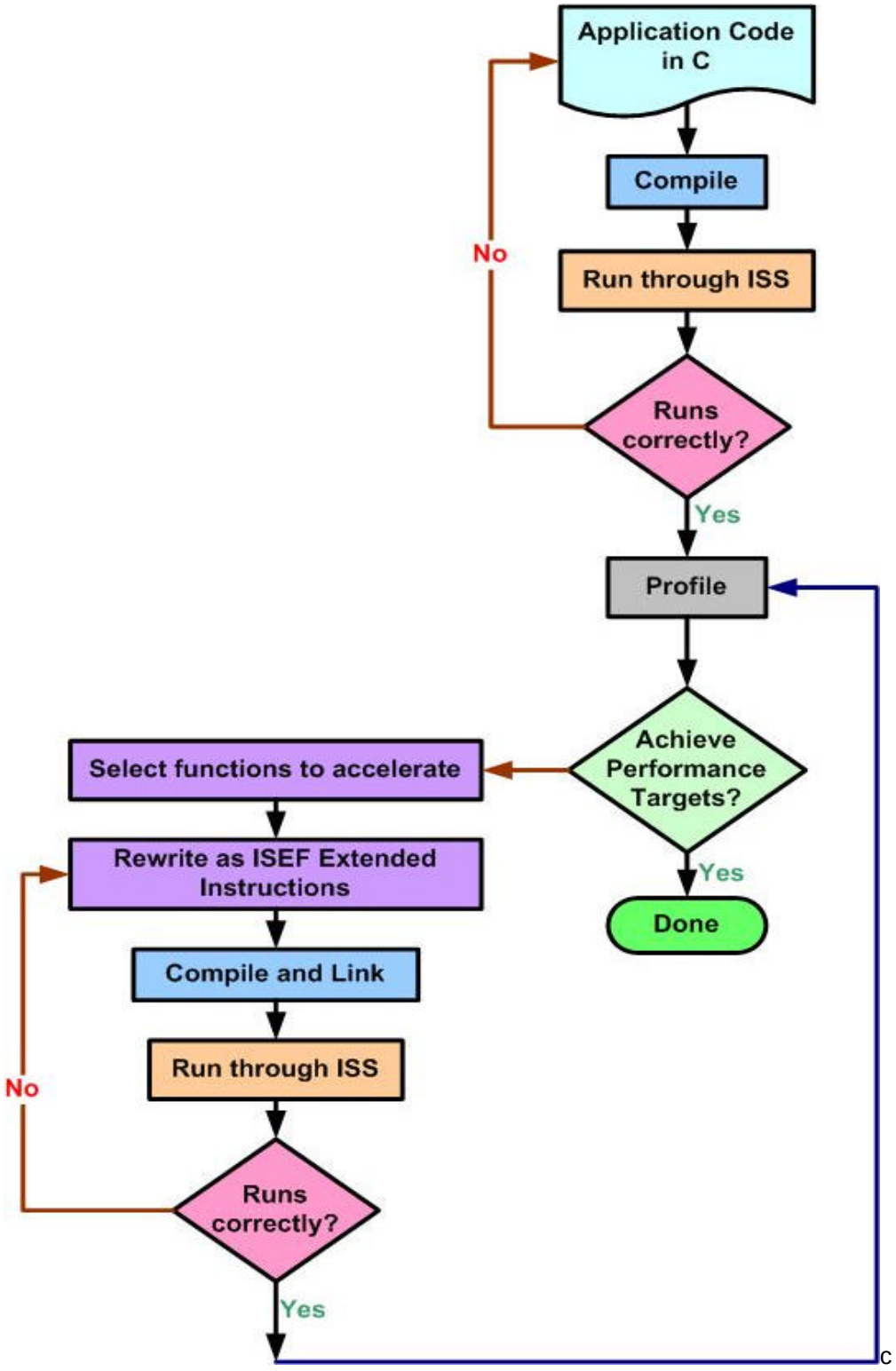

Figure 8 – Fast-X Stretch software development flow

APPENDIX A. VIDEO INPUT CONNECTOR PIN-OUTS

Pos	J1 and J3 Base Configuration		Pos
1	ground	ground	14
2	CC4-	CC4+	15
3	CC3+	CC3-	16
4	CC2-	CC2+	17
5	CC1+	CC1-	18
6	Ser TFG+	Ser TFG-	19
7	Ser TC-	Ser TC+	20
8	RXIN3+	RXIN3-	21
9	RXCLKIN+	RXCLKIN-	22
10	RXIN2+	RXIN2-	23
11	RXIN1+	RXIN1-	24
12	RXIN0+	RXIN0-	25
13	ground	ground	26

Table 2 – Pinout of J1 and J3 CameraLink Connectors

Pos	J2				Pos
	Base Config	Full Configuration	Medium Configuration	Base Config	
1	shield	shield	shield	shield	14
2	CC4-	Z3+	Z3-	CC4+	15
3	CC3+	ZCLK+	ZCLK-	CC3-	16
4	CC2-	Z2+	Z2-	CC2+	17
5	CC1+	Z1+	Z1-	CC1-	18
6	Ser TFG+	Z0+	Z0-	Ser TFG-	19
7	Ser TC-	ZNC+	ZNC-	Ser TC+	20
8	X3+	Y3+	Y3-	X3-	21
9	XCLK+	YCLK+	YCLK-	XCLK-	22
10	X2+	Y2+	Y2-	X2-	23
11	X1+	Y1+	Y1-	X1-	24
12	X0+	Y0+	Y0-	X0-	25
13	shield	shield	shield	shield	26

Table 3 – Pinout of J2 CameraLink Connector

APPENDIX B. VISUAL INDICATORS

LED	Description
1-4	CPLD controlled
5-8	FPGA controlled

Table 4 – Fast-X LEDs

The eight LEDs on the GigE four RJ-45 connector block are controlled directly by the GMAC and by the software that runs it.

APPENDIX C. HEADERS AND JUMPERS

Header	Description
P1	GigE QuadPHY JTAG
P2	Write Protect jumpers for CPLD and Stretch FLASH Memories
P3	Stretch JTAG
P4	SO-DIMM socket-1
P5	SO-DIMM socket-2
P6	FPGA JTAG
P7	CPLD JTAG; LED 3, LED4 CONTROL
P8	RS232 to Stretch

Table 5 – Fast-X Headers and Jumpers

TROUBLESHOOTING

There are several things you can try before you call Alacron Technical Support for help.

_____ Make sure the computer is plugged in. Make sure the power source is on.

_____ Go back over the hardware installation to make sure you didn't miss a page or a section.

_____ Go back over the software installation to make sure you have installed all necessary software.

_____ Run the Installation User Test to verify correct installation of both hardware and software.

_____ Run the user-diagnostics test for your main board to make sure it's working properly.

_____ Insert the Alacron CD-ROM and check the various Release Notes to see if there is any information relevant to the problem you are experiencing.

The release notes are available in the directory: **\usr\alacron\alinfo**

_____ Compile and run the example programs found in the directory: **\usr\alacron\src\examples**

_____ Find the appropriate section of the Programmer's Guide & Reference or the Library User's Manual for the particular library and problem you are experiencing. Go back over the steps in the guide.

_____ Check the programming examples supplied with the runtime software to see if you are using the software according to the examples.

_____ Review the return status from functions and any input arguments.

_____ Simplify the program as much as possible until you can isolate the problem. Turning off any operations not directly related may help isolate the problem.

_____ Finally, first **save your original work**. Then remove any extraneous code that doesn't directly contribute to the problem or failure.

ALACRON TECHNICAL SUPPORT

Alacron offers technical support to any licensed user during the normal business hours of 9 a.m. to 5 p.m. EST. We offer assistance on all aspects of processor board and PMC installation and operation.

CONTACTING TECHNICAL SUPPORT

To speak with a Technical Support Representative on the telephone, call the number below and ask for Technical Support:

Telephone: **603-891-2750**

If you would rather FAX a written description of the problem, make sure you address the FAX to Technical Support and send it to:

Fax: **603-891-2745**

You can email a description of the problem to ***support@alacron.com***

Before you can contact technical support have the following information ready:

_____ Serial numbers and hardware revision numbers of all of your boards. This information is written on the invoice that was shipped with your products.

_____ Also, each board has its serial number and revision number written on either in ink or in bar-code form.

_____ The version of the ALRT, or FastMotion software that you are using.

_____ You can find this information in a file in the directory: **\usr\alfast\alinfo**

_____ The type and version of the host operating system, i.e., Windows XP.

_____ Note the types and numbers of all your software revisions, daughter card libraries, the application library and the compiler

_____ The piece of code that exhibits the problem, if applicable. If you email Alacron the piece of code, our Technical-Support team can try to reproduce the error. It is necessary, though, for all the information listed above to be included, so Technical Support can duplicate your hardware and system environment.

RETURNING PRODUCTS FOR REPAIR OR REPLACEMENT

Our first concern is that you be pleased with your Alacron products.

If, after trying everything you can do yourself, and after contacting Alacron Technical Support, you feel your hardware or software is not functioning properly, you can return the product to Alacron for service or replacement. Service or replacement may be covered by your warranty, depending upon your warranty.

The first step is to call Alacron and request a "Return Materials Authorization" (RMA) number.

This is the number assigned both to your returning product and to all records of your communications with Technical Support. When an Alacron technician receives your returned hardware or software he will match its RMA number to the on-file information you have given us, so he can solve the problem you've cited.

When calling for an RMA number, please have the following information ready:

_____ Serial numbers and descriptions of product(s) being shipped back

_____ A listing including revision numbers for all software, libraries, applications, daughter cards, etc.

_____ A clear and detailed description of the problem and when it occurs

_____ Exact code that will cause the failure

_____ A description of any environmental condition that can cause the problem

All of this information will be logged into the RMA report so it's there for the technician when your product arrives at Alacron.

Put boards inside their anti-static protective bags. Then pack the product(s) securely in the original shipping materials, if possible, and ship to:

Alacron Inc.
71 Spit Brook Road, Suite 200
Nashua, NH 03060
USA

Clearly mark ***the outside of your package:***

Attention **RMA #80XXX**

Remember to include your return address and the name and number of the person who should be contacted if we have questions.

REPORTING BUGS

We at Alacron are continually improving our products to ensure the success of your projects. In addition to ongoing improvements, every Alacron product is put through extensive and varied testing. Even so, occasionally situations can come up in the fields that were not encountered during our testing at Alacron.

If you encounter a software or hardware problem or anomaly, please contact us immediately for assistance. If a fix is not available right away, often we can devise a work-around that allows you to move forward with your project while we continue to work on the problem you've encountered.

It is important that we are able to reproduce your error in an isolated test case. You can help if you create a stand-alone code module that is isolated from your application and yet clearly demonstrates the anomaly or flaw.

Describe the error that occurs with the particular code module and email the file to us at:

support@alacron.com

We will compile and run the module to track down the anomaly you've found.

If you do not have Internet access, or if it is inconvenient for you to get to access, copy the code to a disk, describe the error, and mail the disk to Technical Support at the Alacron address below.

If the code is small enough, you can also:

FAX the code module to us at **603-891-2745**

If you are faxing the code, write everything large and legibly and remember to include your description of the error.

When you are describing a software problem, include revision numbers of all associated software.

For documentation errors, photocopy the passages in question, mark on the page the number and title of the manual, and either FAX or mail the photocopy to Alacron.

Remember to include the name and telephone number of the person we should contact if we have questions.

Alacron Inc.
71 Spit Brook Road, Suite 200
Nashua, NH 03060
USA

Telephone: 603-891-2750
Fax: 603-891-2745
Web site
http://www.alacron.com/
Electronic Mail
sales@alacron.com
support@alacron.com