FASTFRAME 1300

HARDWARE MANUAL

SPEED AND FLEXIBILITY

30002-00187

COPYRIGHT NOTICE

Copyright ã 2001 by Alacron Inc.

Document Name: FastFrame 1300 Hardware Manual

Document Number: 30002-00187

Revision History: 2.0 April 22, 2003

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

MANUAL FIGURES AND TABLES

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.

- 30002-00146 FastImage and FastFrame HW Installation for PCI Systems
- 30002-00148 ALFAST Runtime Software Programmer's Guide & Reference
- 30002-00150 FastSeries Library User's Manual
- 30002-00153 Fast I/O Hardware User's Guide
- 30002-00155 FastMem Hardware User's Manual
- 30002-00169 ALRT Runtime Software Programmer's Guide & Reference
- 30002-00170 ALRT, ALFAST & FASTLIB Software Installation Manual for Linux
- 30002-00171 ALRT, ALFAST, & FASTLIB Software Installation for Windows NT
- 30002-00172 FastImage 1300 Hardware User's Guide
- 30002-00173 FastMem Programmer's Guide & Reference
- 30002-00174 FastMem Hardware Installation Manual
- 30002-00180 Fast4 1300 Hardware User's Guide
- 30002-00184 FastSeries Getting Started Manual
- 30002-00185 FastVision Hardware Installation Manual
- 30002-00186 FastVision Software Installation Manual
- 30002-00188 FOIL – **F**astSeries **O**bject **I**maging **L**ibrary

SYSTEM REQUIREMENTS

- Windows™ NT with service pack 6 or Windows™ 2000 with service pack 2 operating systems fully installed.
- Minimum 128MB memory installed.
- Software Development Environment (SDE)
- WinZip software.
- Acrobat Reader Software
- FastFrame 1300 runtime software is available for Linux, Solaris™ and Windows™ NT and Windows™ NT 2000 operating systems.
- Air circulation of at least 200 LFM is required for the Alacron FastFrame 1300 boards.
- The operating temperature range of the FastFrame 1300 boards is 0° Celsius to 40° Celsius.

I. FASTFRAME 1300 FEATURES

The Alacron FastFrame 1300 is for original equipment manufacturers and end users who anticipate a demand for diverse I/O requirements and high bandwidth. Available in both analog/digital and digital-only configurations, the FastFrame 1300 with optional Philips TriMedia microprocessor provides for complex image and digital signal processing.

A. FastFrame 1300 Features

- PMC and PCI Form-Factors available.
- Optional on-board TriMedia TM1300 180MHz processor or higher with 16 or 32MB of SDRAM.
- Will support future TM1310/20 processors with 64MB.
- Capable of over 720 MFLOPS of computational power.
- 64 or 32-bit 33MHz PCI interface.
- Input/Output via dual (stacked) 68-pin VHDCI connector.
- Collects data from up to four (4) simultaneous CVBS or Y/C (S-Video) analog inputs
- Collects data from up to 32-bits of LVDS digital input
- Collects data from two (2) Camera-Link (Channel Link) inputs.
- Two (2) RS-232C serial port interfaces.
- Four (4) general-purpose inputs. Four (4) general-purpose outputs.
- Two (2) trigger inputs.
- Four (4) strobe outputs.
- Four (4) clock outputs.
- Selectable termination: Allows daisy-chaining of boards.
- On-board image-buffering and formatting FPGA, with optional 16 to 128MB of frame memory SDRAM operates at 160MB/s in full-duplex mode or operates at 320MB/s in half-duplex mode.
- DMA Engine allows direct-to-host image capture on FastFrame 1300 without TriMedia processor.
- 32-bit Fast Channel interface link to Alacron FastImage 1300 or Alacron FastVision capable of 320MB/s peak throughput.
- Twelve (12) general purpose LEDs—FPGA controlled (One (1) TriMedia controlled)
- Philips SAA7111A Enhanced Video Input Processor

B. Optional Features

- Optional local TriMedia boot for embedded systems using 2 – 4MB of on-board FLASH accessible via the TriMedia's X-I/O interface
- TriMedia as PCI bus master
- Optional stereo Audio Codec up to 20-bit format up to 48K samples/sec
- Optional color Video Encoder (PAL or NTSC) from CCIR-656 data

C. PCI-Version Options

- Optional PMC Site 32Bit/33MHz PCI 32-bit Alacron Fast Channel
- Optional LVDS or TTL Alacron Fast Channel headers

II. INTRODUCTION TO FASTFRAME 1300

Alacron's FastFrame-1300 is available in both analog/digital and digital-only configurations for considerable flexibility at a reasonable cost. Depending upon your specific needs and your anticipated applications, your FastFrame 1300 has been configured either with or without a Philips TriMedia TM1300-series microprocessor—an option designed to keep your costs down while providing you with the tools you need either for complex image and signal processing or for frame storage and buffering.

The FastFrame 1300 design enables Alacron to offer either the half-length PCI board with TM1300 processor or a simplified setup wherein the board components include four analog inputs, one FPGA, and no processor. In this latter form the analog front end is configured by the host system.

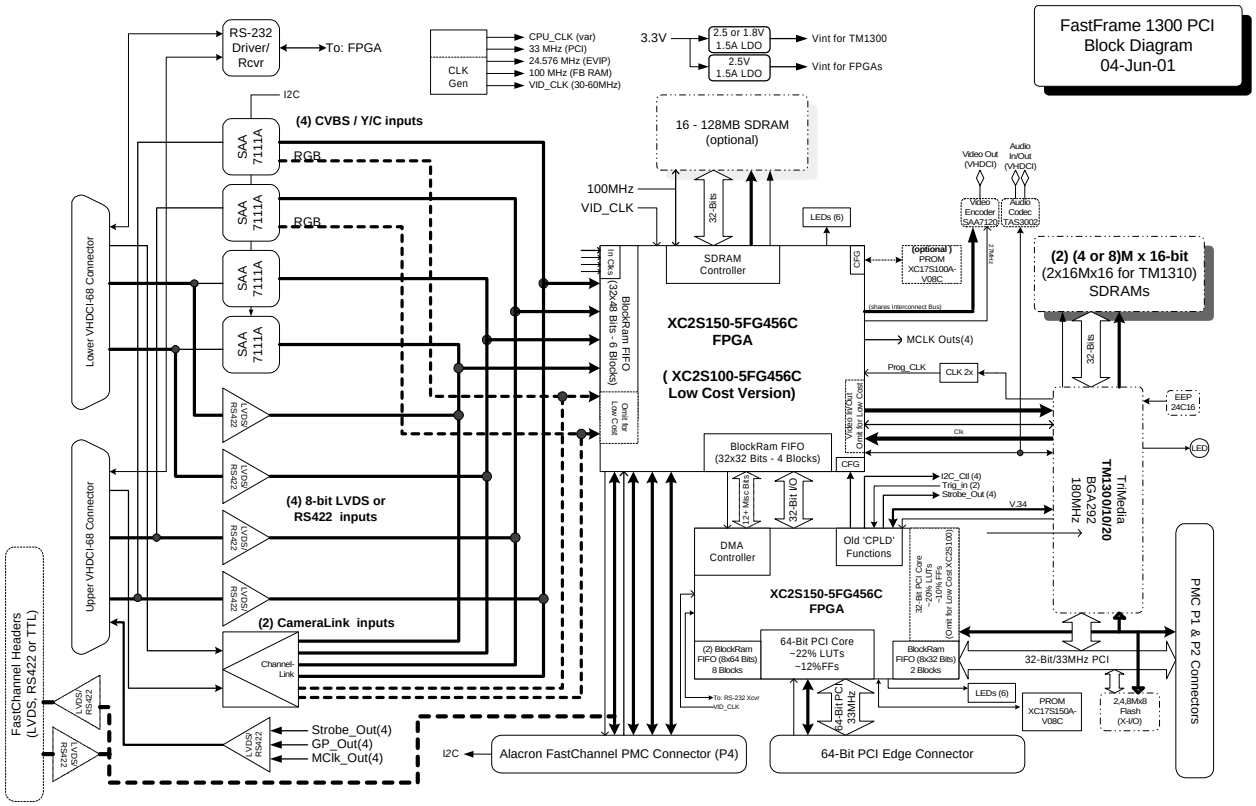


Figure 1 – PCI Block Diagram

In addition to the image-processing capabilities of the optional on-board TriMedia TM1300 processor, the FastFrame 1300 provides image data-capture from up to four analog (CVBS or Y/C) sources or from digital sources (Camera-Link or LVDS) of up to 32 bits.

The on-board FPGA provides data buffering, formatting, and steering. Camera strobes can be generated from software or from external inputs to the board. The analog front-end is configured using I^2C—either by the on-board TriMedia processor or from the host via the J4 connector. The data-path FPGA is configured either directly from the PCI interface or by the TriMedia processor

Input/output to the board in either configuration, with processor or without processor, uses Alacron's FastSeries 68-pin VHDCI system and a Camera-Link cable designed specifically for use with the FastFrame 1300.

The FastFrame 1300 provides two on-board UARTs, implemented in an FPGA and used to communicate with external devices, including CameraLink cameras. There is no handshaking support. The board can also be configured to allow external RS-232 control of a Camera-Link device which is usually connected to the host PC to allow PC-based applications to control and configure the camera.

A. PMC Design

The PMC design is simplified—a configuration without a TriMedia processor in which the only components on the board are the four analog inputs and one FPGA. In this configuration, the analog front-end is configured by the host system.

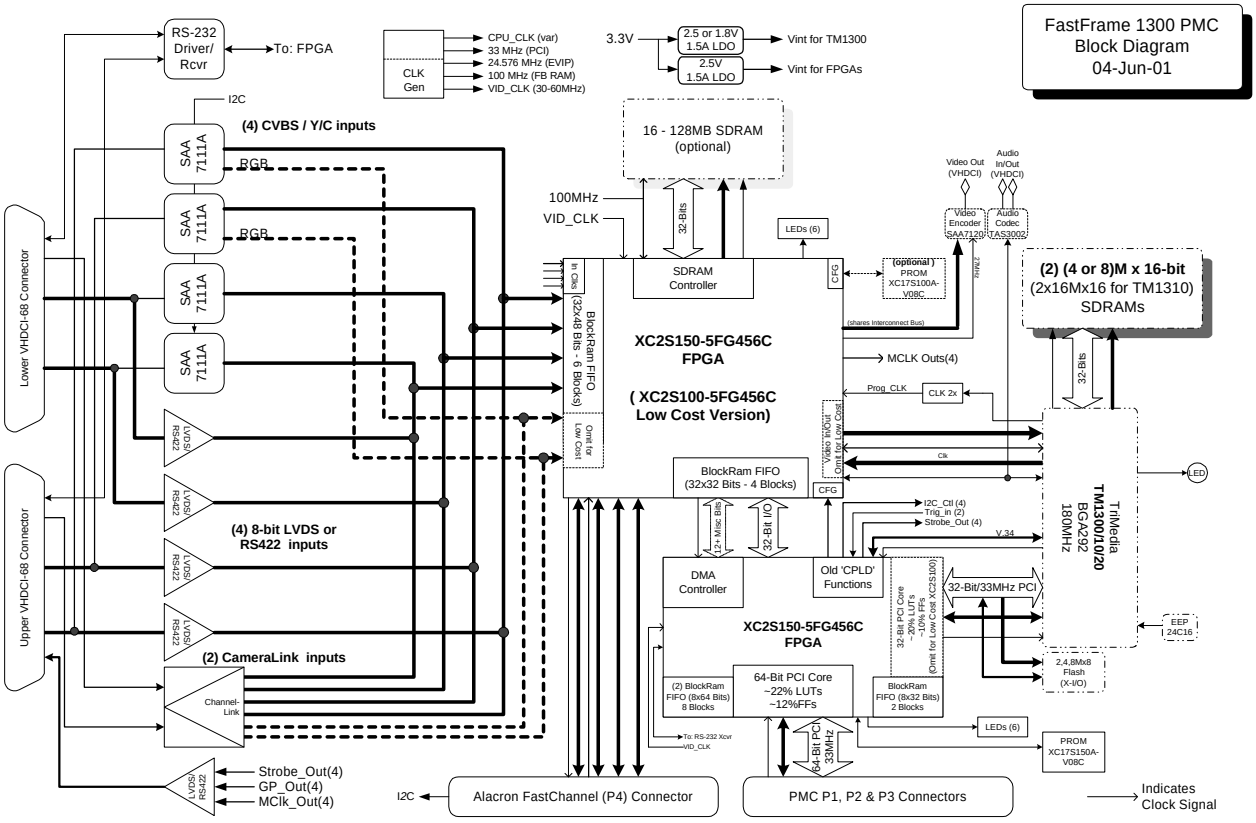


Figure 2 – PMC Block Diagram

III. DATA ACCESS/CAPTURE

The FastFrame1300 supports two configurations of data access:

- Direct DMA-based Frame-Grabber
- TriMedia-based image processing.

The DMA-based mode is used in the simplified version of the FastFrame 1300, and is intended for host-based image processing.

The TriMedia-based mode is more flexible, allowing some data manipulation before the data goes off-board. Because the TriMedia processor and any device attached to the PMC site is behind a non-transparent bridge, the host's PCI scan does not see devices behind the bridge (including video-display adapters, etc). Alacron writes device drivers specifically to initialize and communicate with these devices. Contact Alacron for assistance from the engineering staff if your application(s) call for additional specific device drivers.

A. TriMedia-Based Capture

In TriMedia-based capture, video data is buffered by the TriMedia processor, which can access images in random-access fashion. Data can be transformed or compacted before going off-board: Via the PCI bus, to the PMC site via PCI or Fast Channel, or, in the PCI version of the board, via an optional LVDS Fast Channel connector. The TriMedia-based version of FastFrame 1300 supports a 32-bit PCI bus interface.

B. PCI Bridging Logic

In the TriMedia-based FastFrame 1300, both the processor and an optional device found on the PMC site are accessible from the host PCI bus. This accessibility is provided by PCI bridging logic in the PCI FPGA. The on-board PCI bridge is designed to function as a non-transparent PCI bridge. Thus it does NOT respond to Type-1 configuration cycles. It only responds to Type-0 cycles to the bridge device itself.

The Fast-Frame 1300 device appears to the host as a single PCI device with memory windows. These windows must be configured to be large enough to allow host access to all devices behind the FastFrame 1300 PCI bridge.

Downstream configuration cycles to PCI devices on the backside of this bridge are supported via configuration registers. The PCI Bridge-supported features are listed in Section V.

C. DMA-Based Capture

In Direct Memory Access-based capture, the captured video data is simply buffered, or possibly packed, and made available to a PCI host-based processing engine. This version of the FastFrame 1300 supports 32 and 64-bit PCI interfaces (33MHz). Captured data is sent via DMA to a buffer in host memory, where it is processed by the host processor. The FastFrame 1300 provides DMA master functions.

On-board SDRAM buffers allow buffer storage up to 128MB. DMA block lengths of up to 16MB are supported, and individual PCI transactions can burst up to 256 bytes. The DMA controller has two sets of control registers and thus can "double buffer" transfers, interrupting the host when a programmed DMA transfer have been completed, and immediately switching to the other DMA buffer.

Alternatively, the host can poll a register to interrogate the DMA status.

Additional features include:

- PCI master DMA engine with two (2) buffer pointers allows uninterrupted data flow
- DMA transfer sizes from 16 bytes to 4MB (must be 4-byte aligned)

The video data on the FastFrame 1300 is available only in streaming (serial) format, not in a random-access fashion. The data can go off-board via the host PCI bus or, in the PCI version, via an optional LVDS Fast-Channel connector.

In the DMA-based capture configuration, only the FastFrame 1300 DMA registers are available from the PCI bus. The PMC site option is not supported.

IV. INPUT/OUTPUT FORMATS

The FastFrame1300 is usually configured for your particular application(s) to support one of the input types listed below, although special applications requiring a mix of these inputs may be supported after consultation with the Alacron engineering staff.

A. Analog CVBS or Y/C

The FastFrame1300 board supports simultaneous capture of up to four (4) analog CVBS or Y/C inputs. Each of these input channels can come from one of four (4) CVBS or two (2) Y/C input sources, for a total of sixteen (16) possible analog inputs. The inputs are terminated into 75 ohms and A/C coupled to a Philips SAA7111A video decoder.

All four SAA7111A decoders on the FastFrame 1300 provide 8-bit CCIR 656-style data, or two of them can be configured to generate RGB format data, which is multiplexed into a 16-bit data bus. This bus can then be de-multiplexed into three (3) 8-bit channels by the front-end FPGA.

The SAA7111As are configured by I^2C from the (PMC) J4 connector directly, or from the on-board TriMedia processor. Since the SAA7111As respond to a single I^2C address only, the on-board FPGA is used to select which SAA7111A device the TriMedia communicates with.

Alacron analog-input cable #10024-00162 is used with this type of input setup.

B. Digital RS-422

The FastFrame1300 board supports up to 32 bits of RS-422-format digital-input data. The data can be in the form of four (4) individual 8-bit TAPs, two (2) 16-bit TAPs, one (1) 32-bit TAP or a suitable combination. The data can then be assembled into the desired format by the front-end FPGA. Each of the possible TAPs has individual clock, frame-valid and line-valid inputs. On-board 100ohm termination is provided for RS-422 inputs.

Alacron digital-input cable P/N 10024-00161 or 10024-00224 is used with this type of input setup.

C. Digital LVDS (RS-644)

The FastFrame1300 board supports up to 32 bits of RS-644-format (LVDS) digital-input data. The data can be in the form of four (4) individual 8-bit TAPs, two (2) 16-bit TAPs, one (1) 32-Bit TAP or any combination. The data can then be assembled into the desired format by the front-end FPGA. Each of the TAPs has individual clock, frame-valid and line-valid inputs. On-board 100ohm termination is provided for LVDS inputs.

Alacron digital-input cable P/N 10024-00161 or 10024-00224 is used with this type of input setup.

D. Camera Link

The FastFrame 1300 board supports up to two (2) Camera-Link interfaces, each of which represents the base configuration described in the October 2000 Camera Link specifications. Each interface can run at up to 66MHz.

Each Camera-Link interface supports 24 bits of data, and four bits of control, as well as the bi-directional serial communications interface and CC1 through CC4 signals. Termination for all Camera-Link signals is provided on board.

Alacron supplies a special Camera-Link cable to be used only with the FastFrame 1300. The FastFrame 1300 Camera-Link cable can be used for either of the board's Camera-Link interfaces. It connects to a standard 26-pin 3M MDR connector.

Alacron Camera-Link cable #10024-00250 is used with the Camera-Link setup.

E. Other Options

1. Optional Video Encoder

The FastFrame1300 board supports an optional NTSC/PAL Video Encoder, which will generate a CVBS or Y/C (S-video) output stream from a CCIR 656 data stream. The encoder chip can also generate NABTS Teletext-encoded data on the video signal.

The video-output signal replaces the secondary video input of the last CVBS input channel on the VHDCI connector. This design uses a Philips **SAA7121**, and is configured via I^2C at address **0x88**

2. Optional Audio Codec

The FastFrame1300 board supports an optional 24-bit stereo Audio CODEC to allow analog audio signals to be encoded or decoded into an I2-S format serial stream for processing by the TriMedia or- by the on-board FPGA.

The CODEC can handle sampling at up to 48KHz. The audio channels appear on the VHDCI connector. The design uses a Texas Instruments **TAS3002** device, and is configured via I^2C at address **0x6A**.

3. Optional FastChannel Header

The FastFrame1300 board optionally supports up to 32-bits of LVDS or TTL Fast Channel data via dual 50-pin headers. The direction of this Fast-Channel interface is selectable on a byte (8-bit) boundary via I^2C at address **0x40 & 0x42**.

This option prevents normal operation of the PMC Fast Channel, although the PMC PCI operation is not affected.

V. PCI BRIDGE/TRIMEDIA-BASED CONFIGURATION

A. Bridge Features

1. Supported Features

The PCI Bridge logic on the Fast-Frame 1300 implements a 32-bit 33MHz non-transparent PCI bridge, supporting the following features:

- 3.3V operation with 5.0V tolerant I/O
- Non-transparent bridge. Appears as a single Type-0 Device, allowing address translation between primary and secondary side devices.
- Type-0 configuration registers for each direction
- 3 **B**ase **A**ddress **R**egisters (**BAR0** through **BAR2**) on both primary and secondary interface
- **BAR0**: Used by the bridge as the 1K memory-mapped CSR space.
- **BAR1**: Can be configured as memory or I/O window.
- **BAR2**: Memory window
- Direct Offset address translation for memory and I/O in both directions
- Memory block size adjustable from 256K bytes to 2G bytes.
- I/O window size adjustable from 64 to 256 bytes (PCI-spec maximum for Type 0)
- 256 bytes of Posted Write data buffering in both directions for one transaction.
- 256 bytes of Read data buffering in both directions for one transaction.
- Supports *single* delayed transactions (I/O and configuration) in both directions.
- Can generate Type-0 and Type-1 configuration cycles from either interface via CSR accesses.
- BAR sizes/types stored in TriMedia Boot EEPROM for configuration flexibility

2. Features Not Supported

The FastFrame-1300 Bridge does not support the following:

- Subtractive Decoding, which is not needed for a non-transparent bridge
- Exclusive (LOCK#) access is not supported in the FastFrame 1300
- Multiple delayed transactions. Only one active transaction per side; others are retried.
- Dual-address cycles. Ignores as target and does not initiate as master.
- Fast back-to-back cycles.
- Concatenation or merging of separate contiguous data transactions into one transaction or burst.
- Direct VGA Support. Does not support VGA Palette snooping or I/O, which must be explicitly mapped through a window.

B. Secondary Bus Arbiter

The PCI FPGA also provides a Secondary Bus Arbiter. This logic arbitrates between the three devices on the secondary side of the bridge – the TriMedia, the PMC, and the bridge itself.

The arbiter operates in a round-robin fashion, parking the secondary PCI bus with the last master to use the bus.

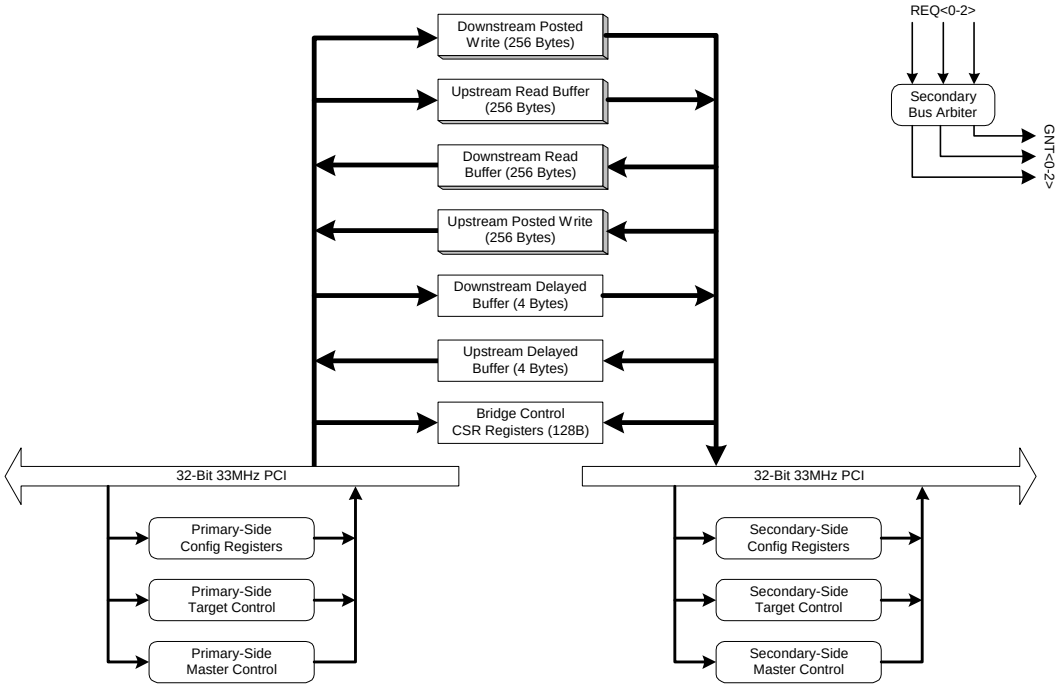


Figure 3 -- FastFrame 1300 PCI Bridge Data Flow

C. PCI Bridge Transactions

The Fast-Frame PCI Bridge handles four types of PCI transactions. The behavior of the bridge during each of these transactions is described on the following pages. Transactions are performed in the following order:

- Posted Writes **must complete** before any other transactions (other than Delayed Write Completion) are accepted.
- Since the bridge has only single transaction pipelines, transactions on either side are accepted in the order they are presented, with the exception of Posted Writes.

1. Posted (memory) Writes

- The bridge accepts burst-write data into a 256-byte (64 data phases) buffer, without wait states, until the buffer is full or a cache-line boundary is crossed (i.e. 4, 8, 16 or 32 D-words per configuration space).
- The bridge responds with Target Disconnect if the buffer is filled. The bridge accepts only ONE posted write transaction (up to 256 bytes) at a time.
- Once all, or at least the "CacheLineSize" bytes, have been written, the target write may begin.
- If the target exceeds the max number of retries (2^{24}), the transaction is terminated and **SERR#** is asserted (if enabled) on the initiating bus. If the target aborts the transfer, **SERR#** is also asserted.
- Memory Write and Invalidate (MWI) commands are treated the same as MW commands.

2. Delayed Writes

- The bridge treats I/O writes and CSR-generated configuration writes as single-cycle "Delayed Writes." The sequence of a Delayed Write is as follows.
- The bridge accepts the write address, C/BEs and data, and then terminates the initiator cycle as a Retry.
- The bridge attempts to complete the cycle on the target bus, and responds to the initiator with "Retry."
- The bridge responds to initiator retry (of identical address, C/BE and Data) with a Cycle Complete.
- The response to the initiator depends on the outcome of the target transfer. A normal **TRDY**# response indicates a normal transfer. Target Abort and Master Abort are reported as "Target Abort" to the initiator.
- If the initiator does not retry within 2^{15} clocks, the posted write is discarded and any written data is lost, although the target write may have been performed. In this case **SERR#** is asserted, if enabled.

3. Delayed Reads

- The bridge treats I/O reads and CSR-generated configuration writes as single-cycle "Delayed Reads." The sequence of a Delayed Read is as follows:

  The bridge accepts the read address and C/BEs, and then terminates the initiator cycle as a Retry.

The bridge attempts to complete the cycle on the target bus. Responds to initiator with “Retry.”

The bridge responds to initiator retry (of identical address and C/BE) with a Cycle Complete.

- The response to the initiator depends on the outcome of the target transfer. A normal **TRDY**# response indicates a normal transfer. Target Abort and Master Abort are reported as “Target Abort” to the initiator.
- If the initiator does not retry within 2^{15} clocks, the posted read is discarded and any data read from the target is lost.

4. Prefetchable Reads

- The following types of transactions are considered prefetchable:

  Memory Read

  Read-Line

  Read-Multiple to prefetchable regions.

- I/O and configuration transactions, or reads from non-prefetchable regions will NOT be prefetched.
- The bridge will perform limited speculative reads to prefetchable regions. It will fill up to a cache-line boundary and then terminate the target transfer.
- If the bridge is still delivering data to the initiator at the same time it is accepting data from the target, the bridge enters Flow-Through mode. In this mode the bridge will continue to transfer data until the buffer is empty for more than 7 cycles, or until a 4KB address alignment boundary is reached.

D. Local Registers

In addition to the configuration registers required by the PCI Bridge specification, the following local registers are defined. They have the same designation as in the DMA configuration.

BAR 0 Offset:	Register
0x0000	DMA_ID_STATUS
0x0010	FE_CONFIG (download)
0x0020	FE_CONTROL
0x0030	FE_SDRAM_SEG
0x0040	UART_I2C_CTL
0x0050	GPIO_REG
0x0060	STB_CTL
0x0070	STB_PRE_SW
0x0090	STRB12_START
0x00A0	STRB12_ENDPD
0x00B0	STRB34_START
0x00C0	STRB34_ENDPD

Table 1 – Local PCI Registers

VI. DMA-BASED CONFIGURATION

In the FastFrame 1300 DMA-based configuration there is no TriMedia or PMC secondary-PCI bus. Captured data is sent via DMA to host memory for host processing. In this configuration the primary PCI interface supports 64 or 32-Bit 33MHz PCIs to allow the highest average PCI throughput to the host.

A. DMA Features

- Four (4) PCI Master DMA engines to support up to four independent video sources.
- Each DMA engine supports double-buffering, using two buffer pointers for uninterrupted data flow.
- DMA transfer sizes: 16 bytes to 64MB (must be 4-byte aligned)
- "Message-Passing-Mode"-like option for frame-per-buffer capability

B. DMA Operation

1. Double Buffering

The FastFrame 1300 DMA engine uses a double-buffering scheme to allow uninterrupted data flow to the host. Initially, the host programs the first buffer pointer for the first transfer, and the second to continue where the first buffer ends. When the DMA completes the count specified in the first buffer, it (optionally) sends an interrupt to the host, and switches (ping-pong fashion) to the second buffer. This allows the host to have a full buffer transfer-time to set up the first pointer in time for the next transfer.

When the second buffer DMA completes, an (optional) interrupt is again sent to the host and the DMA engine switches back to buffer number one. If at any time the buffer is not valid (i.e. not initialized after use) the DMA transfer stops.

2. Front-end FPGA

Data from separate front-end TAPs can be assembled by the front-end FPGA for the DMA engines to stream to the host. The DMA engines indicate which of buffer- memory regions they are reading from. The front-end FPGA translates these region-selects into SDRAM address regions, allowing the buffer architecture to be flexible.

The four DMA engines share the PCI interface via a round-robin arbitration scheme. Arbitration is performed after each PCI burst. The burst length is specified in the **BRST_LEN** field in each DMA length Register. This field specifies the number of 4-byte transfers that the initiator will attempt when capture data is available (4 through 64 bytes supported). In general it should match the destination "CacheLineSize" configuration register value. A Zero (0) value is treated as "16."

C. DMA-Engine Block Diagram

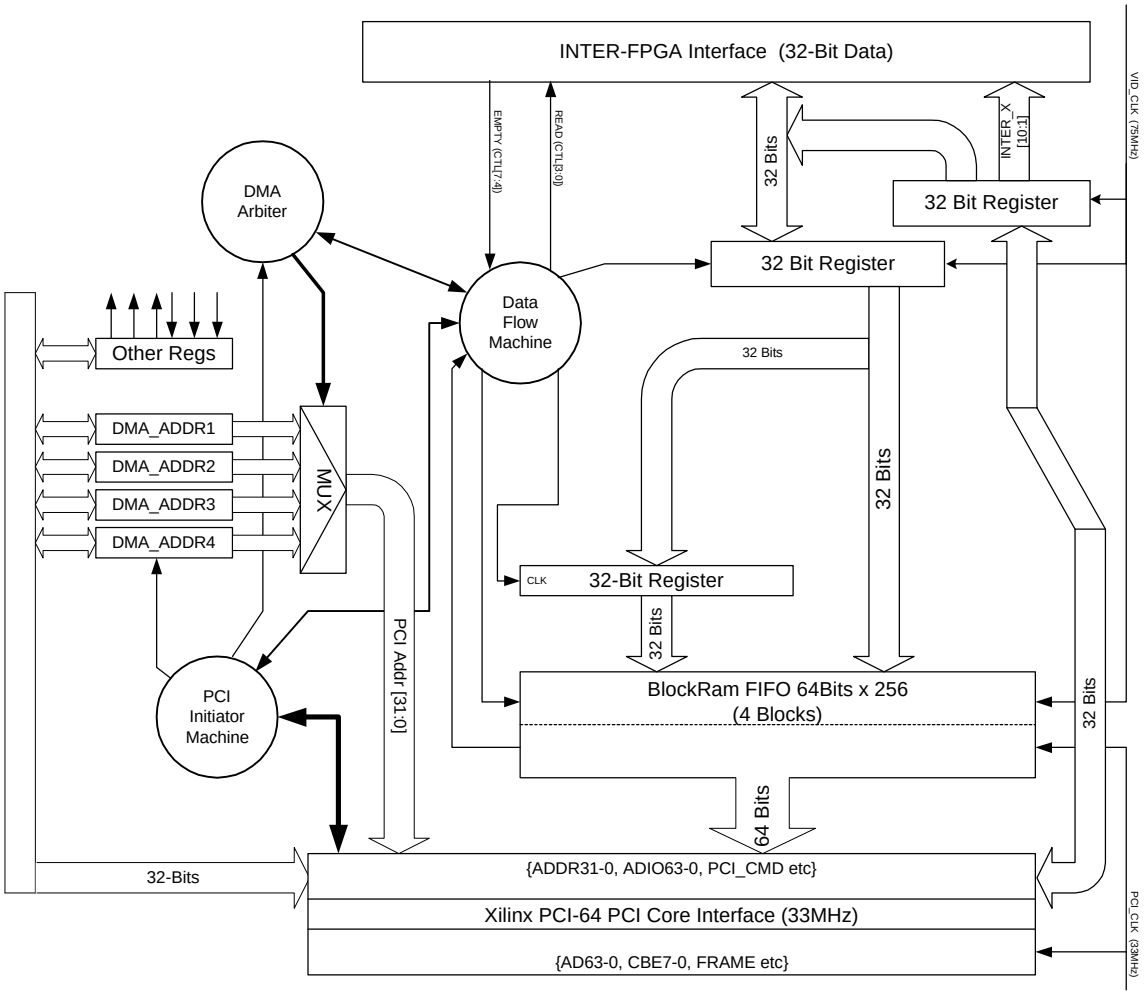


Figure 4 -- DMA Engine Block Diagram

D. DMA-Engine Registers

Each of the four DMA engines has **two** target-address and DMA-length/control registers (for double-buffering). The length/control registers provide status bits (Error, Done) and the count remaining in the transfer.

The Error field will indicate the type of PCI error that was encountered, if any.

DMA_ID_STATUS						
RSVD			ID[7:0]	ANY ERROR	ANY BUSY	ANY DONE
+	CLK SEL[2:0]	BRST_LEN [3:0]	LED[5:0]	INT ENA	Int 1	Int 0

FE_CONFIG (download)					
RSVD	DONE (R/O)	INIT (R/O)	CS_EN (R/W)	CCLK (R/W)	PROG (R/W)

FE_CONTROL						
RSVD	<*TBD*>	MSG_PASS	CAPTURE_ON	REG_R/W	REG_ADDR(4)	REG_DATA(16)

FE_SDRAM_SEG		
RSVD	SDRAM SEGMENT (15 Bits)	0 0

UART_I2C_CTL								
RSVD	DO UART	UART R/W	UART ADDR(4)	UART DATA(8)	DO I^2C	I^2C ADDR(7)	I^2C_R/W	I^2C DATA(8)

GPIO_REG		
RSVD	GPIN[3:0] (R/O)	GPOUT[3:0] (R/W)

STB_CTL							
RSVD	STB3 FUNC	STB4ENA	STB4POL	STB3ENA	STB3POL	EDGE3&4	TRG_SRC(4,3)
+	STB1 FUNC	STB2ENA	STB2POL	STB1ENA	STB1POL	EDGE1&2	TRG_SRC(2,1)

STB_PRE_SW			
RSVD	SW_TRIG3&4	SW_TRIG1&2	PRESCALE (11 Bits)

STRB12_START	
ST2_START (16 Bits)	ST1_END (16 Bits)

STRB12_ENDPD	
ST12_PERIOD (16 Bits)	ST2_END (16 Bits)

STRB34_START	
ST4_START (16 Bits)	ST3_END (16 Bits)

STRB34_ENDPD	
ST34_PERIOD (16 Bits)	ST4_END (16 Bits)

DMA_TGT_ADDR (1-4, A/B)	
PCI Target Address (30 Bits)	***0 0***

DMA_TGT_CNTRL (1-4, A/B)					
RSVD	***PCI ERROS***	***DONE***	***GO***	***DMA Length (24 Bits)***	***0 0***

Table 2 – DMA-Engine Register Definitions

Host Interrupts can be selected or disabled (Interrupt A through Interrupt D). When interrupts are disabled, the host may poll the DONE bit to determine when a transfer has completed.

Note that polling cycles reduce effective DMA bandwidth accordingly.

The ID is a field used by driver software to identify customized FPGA design variations. The LED field can be used to interrogate or set LEDs on the board. Other application-specific registers may be added as required.

Details of the PCI Initiator Address Register are in the diagram below.

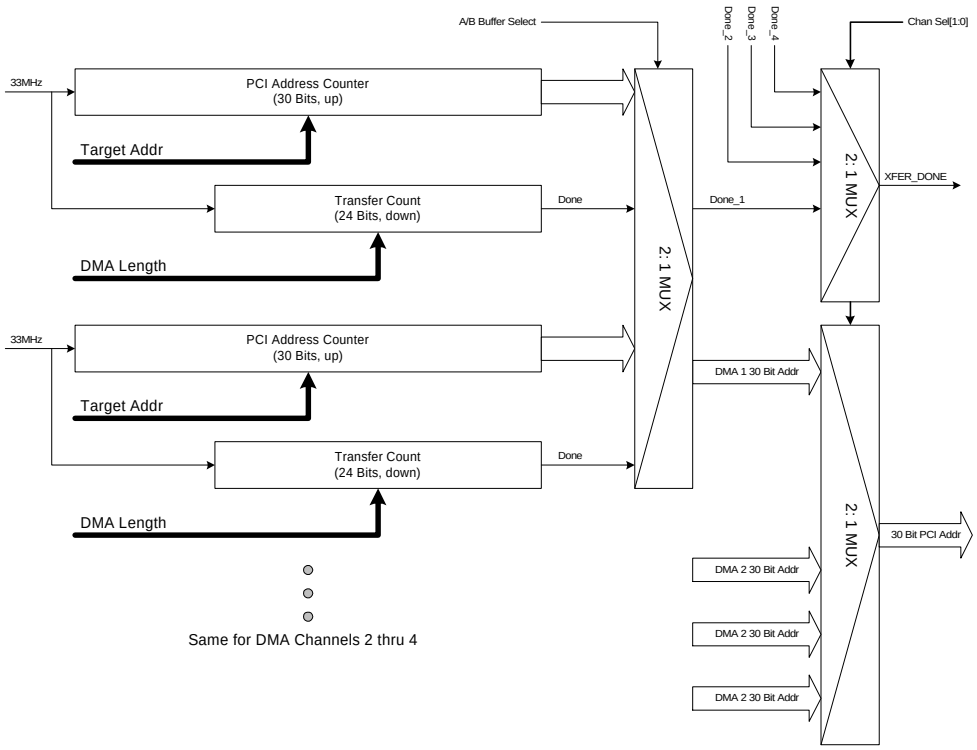


Figure 5 – DMA Address Counter Registers

The DMA interface is accessible from the PCI interface through PCI Base Address Register windows. The control registers are located in **BAR 0** (0x10) which is 32-bit memory-mapped and non-prefetchable.

Data is mastered (PCI initiator) from the DMA controller, and thus does not require a Base Address allocation.

Base Address Register 1 (**BAR 1**, 0x14) allows optional access to frame-buffer memory. It is 32-bit memory-mapped and non-prefetchable. This access is limited to 4KB windows. A segment register must be set to select the desired 4KB region for random access.

The following table shows the memory offsets of the DMA registers at **BAR 0**. Other application-specific registers may be added as required in the space from 0x004 thru 0x00FC. These specific designs can be identified via the 8-bit "ID" register.

BAR 0 Offset:	**Register**
0x0000	DMA_ID_STATUS
0x0010	FE_CONFIG (download)
0x0020	FE_CONTROL
0x0030	FE_SDRAM_SEG
0x0040	UART_I2C_CTL
0x0050	GPIO_REG
0x0060	STB1_4CTL
0x0070	STB_PRE_SW
0x0090	STRB12_START
0x00A0	STRB12_ENDPD
0x00B0	STRB34_START
0x00C0	STRB34_ENDPD
. . .	. . .
0x0100	DMA_TGT_ADDR1A
0x0110	DMA_TGT_CTL1A
0x0120	DMA_TGT_ADDR2A
0x0130	DMA_TGT_CTL2A
0x0140	DMA_TGT_ADDR3A
0x0150	DMA_TGT_CTL3A
0x0160	DMA_TGT_ADDR4A
0x0170	DMA_TGT_CTL4A
0x0200	DMA_TGT_ADDR1B
0x0210	DMA_TGT_CTL1B
0x0220	DMA_TGT_ADDR2B
0x0230	DMA_TGT_CTL2B
0x0240	DMA_TGT_ADDR3B
0x0250	DMA_TGT_CTL3B
0x0260	DMA_TGT_ADDR4B
0x0270	DMA_TGT_CTL4B
. . .	. . .

Table 3 -- DMA Engine Register Locations (BAR 0)

VII. POWER-UP SEQUENCE

The FastFrame 1300 software-controlled procedure for power-up and module initialization follows this sequence:

- 3.3V and 5.0V Power Applied
- 2.5/1.8V power stabilizes. The time this takes depends on the 3.3V ramp-rate.
- PCI FPGA configures from EPROM.
- TM1300 reads EEP configuration data
- Host processor performs BIOS scan and configures TM1300 PCI resources, if present.
- Host processor loads TriMedia program into TriMedia RAM, and runs.
- TriMedia initialization code configures front-end FPGA (optional EEPROM configuration).
- Software initializes on-board registers via TriMedia I^2C controller.
- Data capture may begin.

A. Power Dissipation

The FastFrame 1300 will dissipate power approximately as shown below:

Version	+5V Power	+3.3V Power	+/-12V
PCI	16W	11W	+/- 30mA
PMC	5W	10W	+/- 30mA

Table 4 - Approximate Power Dissipation

Both the PCI and PMC versions of FastFrame 1300 have auxiliary power connectors. The PMC auxiliary power connector is for use in systems that cannot supply 3.3V via the PMC connector. The PCI auxiliary power connector is used to provide supplemental power when the PMC site is used.

See the Installation Manual, #30002-00185 for installation assistance.

B. FastFrame 1300 Cables

The appropriate cables from the list below have been included in your FastFrame 1300 shipment. If you need additional cables, contact Alacron Sales or Technical Support.

Alacron Part Number	Cable Use
100024-00160	DC Power Cable
100024-00161	Digital Input Cable
100024-00162	Analog Input Cable
100024-00224	Digital-In Adapter Cable
100024-00250	Camera-Link Cable

Table 5 – FastFrame 1300 Cables

VIII. CLOCK SCHEME

The FastFrame 1300 design takes advantage of the high data rates possible with the TM1300 processor, while enabling tight control of the timing on high-speed buses, especially when utilizing the Alacron Fast Channel connector. The clocks are selected and generated through the FPGAs.

The on-board clock-generator chips (Cypress CY2292) generate the following clocks:

Clock Signal	Frequency	Used by:
SDRAM Clk	100.00MHz	Buffer SDRAM, FPGA
CPU_CLK	47.666MHz, Selectable	TM1300
EVIP_CLK	24.576MHz	SAA7111A's
V34_CLK	20.000MHz	TM1300, PCI_FPGA
VID_CLK (2)	40-80MHz, Selectable	FPGAs, TM1300, FastChan, VideoOut
PCI_CLK	33MHz	PCI_FPGA, TriMedia, PMC Site
UART_CLK	6.144MHz	PCI_FPGA

The following clocks are also available on the FastFrame 1300 board:

Clock Signal	Freq	Sourced by => Used By:
TM_VOCLK	1-80MHz S/W Select	TM1300 => FE_FPGA, FastChan, VideoOut
TM_AIOSCK	1-40MHz S/W Select	TM1300 => FE_FPGA,
TM_AOSCK	1-40MHz S/W Select	TM1300 => FE_FPGA, AUDIO_CODEC
FECLK1	*1-75MHZ (per Conf)*	*(ChanLnk1 / PixClk1 –or- FchanClkA) => FE_FPGA*****
FECLK2	*1-75MHZ (per Conf)*	*(ChanLnk2 / PixClk4 –or- VID_CLK) => FE_FPGA*****
PIX_CLK(2,3)	1-75MHz	(Analog / Dig Vid In) => FPGA_I/O**
FchanClk(B,C)	1-75MHz	FastChan => FPGA_I/O**

Table 6 – Clock Signals

Clocks above with one asterisk (*) are multiplexed (CMOS switch); selectable by the FPGA. Only one of the two selections is available at any given time. Channel Link (Camera Link), differential and analog inputs are mutually exclusive population options.

Clocks above with two asterisks (**) connect to FPGA I/O pins, not to dedicated global clocks, so care must be taken to prevent problems due to clock skew in the FPGA logic using these clocks.

The Front-End FPGA has the following clock inputs available:

- PIX_CLK1 thru PIX_CLK4 (from video inputs)
- SDRAM Clock (approx 100MHz) for Frame Buffer
- TriMedia VO_CLK
- FastChannel Clks (From J4)

VID_CLK – Selectable Video Clock from CLK Generator The PCI FPGA has the following clock inputs available:

- Inter-FPGA CLK (from F.E. FPGA)
- V.34 CLK (20MHz from ClkGen)

- VID_CLK – Selectable Video Clock from CLK Generator
- PCI_CLK (33MHz for secondary PCI interface or Primary Master)
- UART_CLK (6.144 MHz for UART timing)

IX. INTERRUPTS

The interrupts from the various devices on the FastFrame 1300 board are connected as shown below.

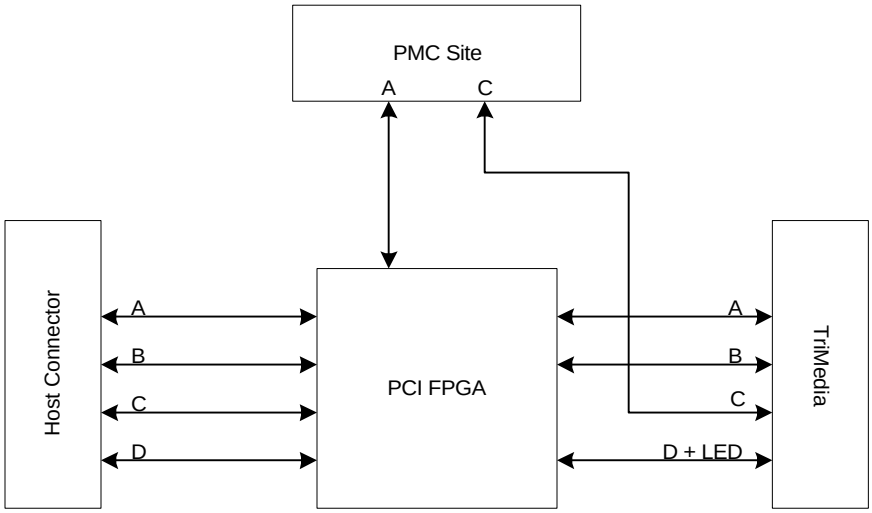


Figure 6 – Interrupt Connections

- This configuration allows flexible connection between host, PMC and TriMedia interrupt sources (open drain), and handlers via the PCI FPGA.
- The TriMedia interrupt “D” pin is also connected to an LED for debugging purposes.

X. FASTCHANNEL CONTROL

Alacron's Fast Channel is a mechanism for Alacron's FastSeries components, like those found on the FastFrame 1300, to communicate with the processing environment without impacting the PCI bus.

When implemented within a high-performance DSP or imaging application, the bus form of data transfer can be subject to bus saturation and contention. However point-to-point communication directly from the data source to the data sink, with each interconnection supporting its own data transfer—isolated from any other data transfer—can prevent such contention and bus saturation.

Fast Channel is Alacron's implementation of a point-to-point, or point-to-many-points, connection between a data source and its data sinks.

Fast Channel is a cost-effective interconnects solution. It is configurable at startup, and no special hardware is needed for source and destination addresses, arbitration, or control of the channel. Simple software protocols can be implemented because the only operation allowed is the transfer of one "word" of data. The Fast Channel is composed of a parallel data path from 1 to 32 bits, a clock, and a data valid. Specifications will vary with environment and usage.

An input data rate of 320 MB, or anything over 133 MB, is too high for transfer over the PCI bus, but it can be transferred over the Fast Channel at 80 MHz—with the additional benefits of isolating the transfer from the host CPU and without impacting PCI activity. In data-intensive applications, Fast Channel's higher bandwidth will provide significant user savings over using the PCI bus, even if additional CPUs and cameras are installed to achieve the higher rates the Fast Channel is designed to handle.

The FastFrame 1300 configuration allows 8-bit granularity in data-path direction. The byte lane and control signals are configured via I^2C (address **0x40** and **0x42**) using a Philips PCF8575 device. These registers also control which clock signals are connected to the Fast Channel.

The control registers are implemented in two I^2C devices. Each device provides 16 control bits that can be written or read via I^2C. All register bits power up as "1," effectively disabling all (active low) signals. The control bits are defined starting on the next page.

Writes to either device must be done as a single multi-byte write. Reads can be either single or multi-byte depending on whether the first 8 or all 16 bits are desired.

A. I²C Address 0 x 40

1. Byte 1:

Bit #	Signal Name	Meaning:
7	~FC3_IN	When '0' connects FC data bits D7 thru D0 to FPGA inputs 31-24
6	~FC2_IN	When '0' connects FC data bits D7 thru D0 to FPGA inputs 23-16
5	~FC1_IN	When '0' connects FC data bits D7 thru D0 to FPGA inputs 15-08
4	~FC0_IN	When '0' connects FC data bits D7 thru D0 to FPGA inputs 07-00
3	~FC3_OUT	When '0' connects FPGA bits 31-24 to FC data bits D7 thru D0
2	~FC2_OUT	When '0' connects FPGA bits 23-16 to FC data bits C7 thru C0
1	~FC1_OUT	When '0' connects FPGA bits 15-08 to FC data bits B7 thru B0
0	~FC0_OUT	When '0' connects FPGA bits 07-00 to FC data bits A7 thru A0

2. Byte 2:

Bit #	Signal Name	Meaning:
7	~FC_CKC7	When '0' connects VID_CLK to FC_CLKD (input OR output)
6	~FC_CKC6	When '0' connects TM_VOCLK to FC_CLKC (input OR output)
5	~FC_CKC5	When '0' connects VID_CLK to FC_CLKB (input OR output)
4	~FC_CKC4	When '0' connects TM_VOCLK to FC_CLKA (input OR output)
3	~FC_CKC3	When '0' connects FCHAN_CKD to FC_CLKD (input OR output)
2	~FC_CKC2	When '0' connects FCHAN_CKC to FC_CLKC (input OR output)
1	~FC_CKC1	When '0' connects FCHAN_CKB to FC_CLKB (input OR output)
0	~FC_CKC0	When '0' connects FCHAN_CKA to FC_CLKA (input OR output)

B. I²C Address 0 x 42

1. Byte 1:

Bit #	Signal Name	Meaning:
7	~FC_TRM_ENA 1	When '0' enables LVDS termination of FCHAN B7 thru B0
6	~FC_TRM_ENA 0	When '0' enables LVDS termination of FCHAN A7 thru A0
5	~FX5_OUT	When '0' enables C03 and C07 outputs from FPGA
4	~FX4_OUT	When '0' enables C02 and C06 outputs from FPGA
3	~FX3_OUT	When '0' enables PIXCK4 and PIXCK2 outputs
2	~FX2_OUT	When '0' enables C01 and C05 outputs from FPGA
1	~FX1_OUT	When '0' enables C00 and C04 outputs from FPGA
0	~FX0_OUT	When '0' enables PIXCK3 and PIXCK1 outputs

2. Byte 2:

Bit #	Signal Name	Meaning:
7	~FC_TRM_ENA 3	When '0' enables LVDS termination of FCHAN D7 thru D0
6	~FC_TRM_ENA 2	When '0' enables LVDS termination of FCHAN C7 thru C0
5	~FX5_IN	When '0' enables C03 and C07 inputs to FPGA
4	~FX4_IN	When '0' enables C02 and C06 inputs to FPGA
3	~FX3_IN	When '0' enables PIXCK4 and PIXCK2 inputs to FPGA
2	~FX2_IN	When '0' enables C01 and C05 inputs to FPGA
1	~FX1_IN	When '0' enables C00 and C04 inputs to FPGA
0	~FX0_IN	When '0' enables PIXCK3 and PIXCK1 inputs to FPGA

XI. CONNECTOR PIN-OUTS

The pin-out of the FastFrame 1300 dual 68-pin VHDCI connector retains the same pin assignments for digital inputs (4 TAPs), including GPINs and GPOUTs, Strobes and Trig_Ins, as are found on Alacron's FastImage 1300 board. Analog inputs retain the original Alacron FastSeries pin-out for the CVBS input, extended to four positions.

The biggest difference between the FastFrame 1300 and Alacron's FastImage 1300 connectors is in the Camera-Link pin-out. Rather than having a separate J2 connector, Camera-link signals replace digital-input signals on the FastFrame 1300 J1A/B connector.

The FastFrame 1300 has a new, straightforward pin-out configuration. The RS-232 interfaces are optional, and replace GPIN pins when they are used.

The LVDS/TTL Fast Channel connectors use the same connectors and pin-out as Alacron's FastImage 1300. However control of their direction and enables have changed.

A. Headers and Jumpers

Conn	Use	Pin 1	Pin 2	Pin 3	Pin 4	Pin 5	Pin 6
P1	*Aux Power*	3.3VDC in	GND	GND	5.0VDC in	N/A	
P2	*LED Test Points*	LED01	LED03	LED04	LED09	LED10	GND
P3	*Xilinx JTAG*	TRST	TCK	TMS	TDI	TDO	GND
P4	*TM1300 EEProm*	Jumper 1-2 for Write Enable.		N/C		Jumper 5-6 for Default (no EEProm)	
P5	*TM1300 JTAG*	TRST	TCK	TMS	TDI	TDO	GND
P6	*J1A/B Term Ena*	Jumper 1-2 to terminate Taps 1 & 2		N/C		Jumper 5-6 to terminate Taps 3 & 4	

Table 8 – FastFrame 1300 Headers and Jumpers

B. VHDCI Connectors

The pin-out of the FastFrame 1300 dual 68-pin VHDCI connector is shown in Table 9 on the next page.

J1A Pinout

Pin #	Dig In Pin	Analog Pin	Cam-Link Pin		Pin #	Dig In Pin	Analog Pin	Cam-Link Pin
1	T1_LVAL+				35	STROBE1+		
2	T1_LVAL				36	STROBE1-		
3	T1_FVAL+				37	STROBE2+		*CC2A+*
4	T1_FVAL				38	STROBE2-		*CC2A-*
5	T1_PXCK+		XACLK+		39	MSTR_CK1+		*CC2A+*
6	T1_PXCK		XACLK-		40	MSTR_CK1-		*CC2A-*
7	GPIN1+				41	MSTR_CK2+		*CC3A+*
8	GPIN1-				42	MSTR_CK2-		*CC3A-*
9	**GND**	**GND**	**GND**		43	**GND**	**GND**	**GND**
10	T1_D0+	YIN1A+			44	T2_D7+		
11	T1_D0-	YIN1A-			45	T2_D7-		
12	T1_D1+	YIN1B+			46	T2_D6+		
13	T1_D1-	YIN1B-			47	T2_D6-		
14	T1_D2+	CIN1A+			48	T2_D5+		
15	T1_D2-	CIN1A-			49	T2_D5-		
16	T1_D3+	CIN1B+			50	T2_D4+		
17	T1_D3-	CIN1B-			51	T2_D4-		
18	T1_D4+		XA0+		52	T2_D3+	CIN2B+	
19	T1_D4-		XA0-		53	T2_D3-	CIN2B-	
20	T1_D5+		XA1+		54	T2_D2+	CIN2A+	
21	T1_D5-		XA1-		55	T2_D2-	CIN2A-	
22	T1_D6+		XA2+		56	T2_D1+	YIN2B+	
23	T1_D6-		XA2-		57	T2_D1-	YIN2B-	
24	T1_D7+		XA3+		58	T2_D0+	YIN2A+	
25	T1_D7-		XA3-		59	T2_D0-	YIN2A-	
26	**GND**	**GND**	**GND**		60	**GND**	**GND**	**GND**
27	X_TRIG1+				61	GPIN2+		RS-232A_RXD
28	X_TRIG1-				62	GPIN2-		RS-232A_TXD
29	GPIN5+		SERTFGA+		63	T2_LVAL+		
30	GPIN5-		SERTFGA-		64	T2_LVAL		
31	GPOUT1+		SERTCA+		65	T2_FVAL+		
32	GPOUT1-		SERTCA-		66	T2_FVAL		
33	GPOUT2+		CC1A+		67	T2_PXCK+		
34	GPOUT2-		CC1A-		68	T2_PXCK		

J1B Pinout

Pin #	Dig In Pin	Analog Pin	Cam-Link Pin		Pin #	Dig In Pin	Analog Pin	Cam-Link Pin
1	T3_LVAL+				35	STROBE3+		
2	T3_LVAL				36	STROBE3-		
3	T3_FVAL+				37	STROBE4+		*CC2B+*
4	T3_FVAL				38	STROBE4-		*CC2B-*
5	T3_PXCK+		*XBCLK+*		39	MSTR_CK3+		*CC2B+*
6	T3_PXCK		*XBCLK-*		40	MSTR_CK3-		*CC2B-*
7	GPIN3+				41	MSTR_CK4+		*CC3B+*
8	GPIN3-				42	MSTR_CK4-		*CC3B-*
9	**GND**	**GND**	**GND**		43	**GND**	**GND**	**GND**
10	T3_D0+	YIN3A+			44	T4_D7+	*AINL*	
11	T3_D0-	YIN3A-			45	T4_D7-	AGND	
12	T3_D1+	YIN3B+			46	T4_D6+	*AINR*	
13	T3_D1-	YIN3B-			47	T4_D6-	AGND	
14	T3_D2+	CIN3A+			48	T4_D5+	*AOUTL*	
15	T3_D2-	CIN3A-			49	T4_D5-	AGND	
16	T3_D3+	CIN3B/CVBS_OUT+			50	T4_D4+	*AOUTR*	
17	T3_D3-	CIN3B/CVBS_OUT-			51	T4_D4-	AGND	
18	T3_D4+		*XB0+*		52	T4_D3+	*CIN/OUT4B+*	
19	T3_D4-		*XB0-*		53	T4_D3-	*CIN/OUT4B-*	
20	T3_D5+		*XB1+*		54	T4_D2+	CIN4A+	
21	T3_D5-		*XB1-*		55	T4_D2-	CIN4A-	
22	T3_D6+		*XB2+*		56	T4_D1+	*YIN/OUT4B+*	
23	T3_D6-		*XB2-*		57	T4_D1-	*YIN/OUT4B-*	
24	T3_D7+		*XB3+*		58	T4_D0+	YIN4A+	
25	T3_D7-		*XB3-*		59	T4_D0-	YIN4A-	
26	**GND**	**GND**	**GND**		60	**GND**	**GND**	**GND**
27	X_TRIG2+				61	GPIN4+		RS-232B_RXD
28	X_TRIG2-				62	GPIN4-		RS-232B_TXD
29	GPIN6+		*SERTFGB+*		63	T4_LVAL+		RS-232C_RXD
30	GPIN6-		*SERTFGB-*		64	T4_LVAL		RS-232C_TXD
31	GPOUT3+		*SERTCB+*		65	T4_FVAL+		RS-232D_RXD
32	GPOUT3-		*SERTCB-*		66	T4_FVAL		RS-232E_RXD
33	GPOUT4+		*CC1B+*		67	T4_PXCK+		
34	GPOUT4-		*CC1B-*		68	T4_PXCK		

Table 9 – Pinout of J1A/J1B VHDCI Connector

C. LVDS/TTL Connectors

Fast-Channel Connector Pin-Out

J2 Pin	Signal	J2 Pin	Signal
1	FC_DAT0P	26	FC_DAT11N
2	FC_DAT0N	27	FC_DAT12P
3	FC_DAT1P	28	FC_DAT12N
4	FC_DAT1N	29	FC_DAT13P
5	FC_DAT2P	30	FC_DAT13N
6	FC_DAT2N	31	FC_DAT14P
7	FC_DAT3P	32	FC_DAT14N
8	FC_DAT3N	33	FC_DAT15P
9	FC_DAT4P	34	FC_DAT15N
10	FC_DAT4N	35	GND
11	FC_DAT5P	36	GND
12	FC_DAT5N	37	FC_CTL0P
13	FC_DAT6P	38	FC_CTL0N
14	FC_DAT6N	39	FC_CTL1P
15	FC_DAT7P	40	FC_CTL1N
16	FC_DAT7N	41	FC_CTL2P
17	*GND*	42	FC_CTL2N
18	*GND*	43	FC_CTL3P
19	FC_DAT8P	44	FC_CTL3N
20	FC_DAT8N	45	FC_PXCK1P
21	FC_DAT9P	46	FC_PXCK1N
22	FC_DAT9N	47	FC_PXCK2P
23	FC_DAT10P	48	FC_PXCK2N
24	FC_DAT10N	49	*GND*
25	FC_DAT11P	50	*GND*

J3 Pin	Signal	J3 Pin	Signal
1	FC_DAT16P	26	FC_DAT27N
2	FC_DAT16N	27	FC_DAT28P
3	FC_DAT17P	28	FC_DAT28N
4	FC_DAT17N	29	FC_DAT29P
5	FC_DAT18P	30	FC_DAT29N
6	FC_DAT18N	31	FC_DAT30P
7	FC_DAT19P	32	FC_DAT30N
8	FC_DAT19N	33	FC_DAT31P
9	FC_DAT20P	34	FC_DAT31N
10	FC_DAT20N	35	GND
11	FC_DAT21P	36	GND
12	FC_DAT21N	37	FC_CTL4P
13	FC_DAT22P	38	FC_CTL4N
14	FC_DAT22N	39	FC_CTL5P
15	FC_DAT23N	40	FC_CTL5N
16	FC_DAT23P	41	FC_CTL6P
17	*GND*	42	FC_CTL6N
18	*GND*	43	FC_CTL7P
19	FC_DAT24P	44	FC_CTL7N
20	FC_DAT24N	45	FC_PXCK3P
21	FC_DAT25P	46	FC_PXCK3N
22	FC_DAT25N	47	FC_PXCK3P
23	FC_DAT26P	48	FC_PXCK3N
24	FC_DAT26N	49	*GND*
25	FC_DAT27P	50	*GND*

Table 10 – Pinout of LVDS FastChannel Connectors

XII. COMPONENTS REPLACEMENTS

A. PCI Components Placements

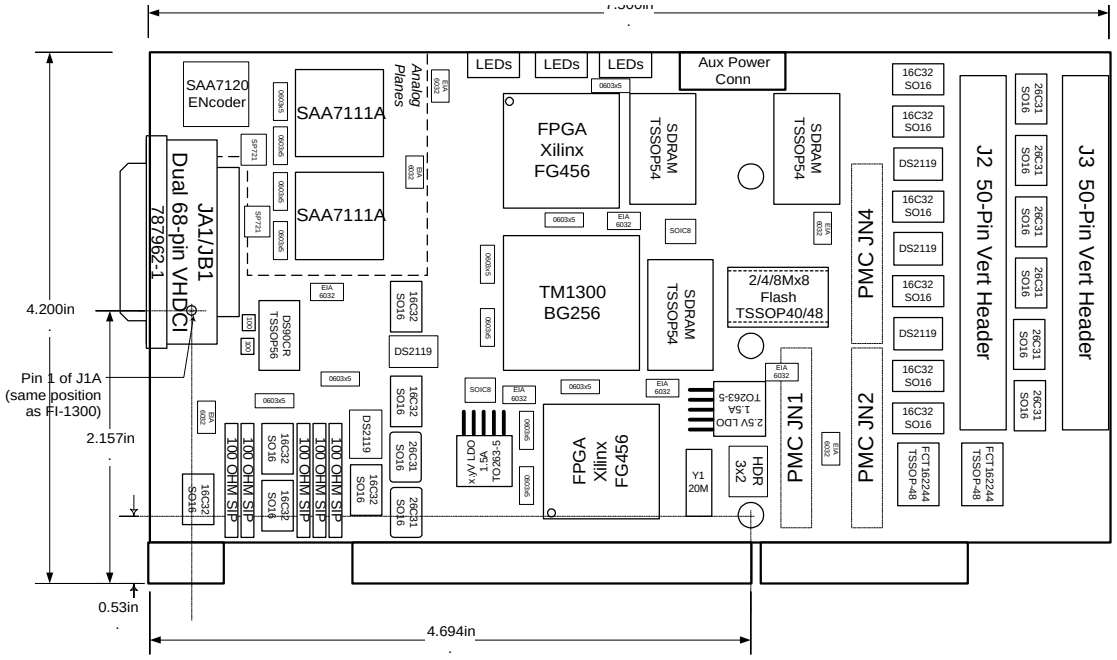
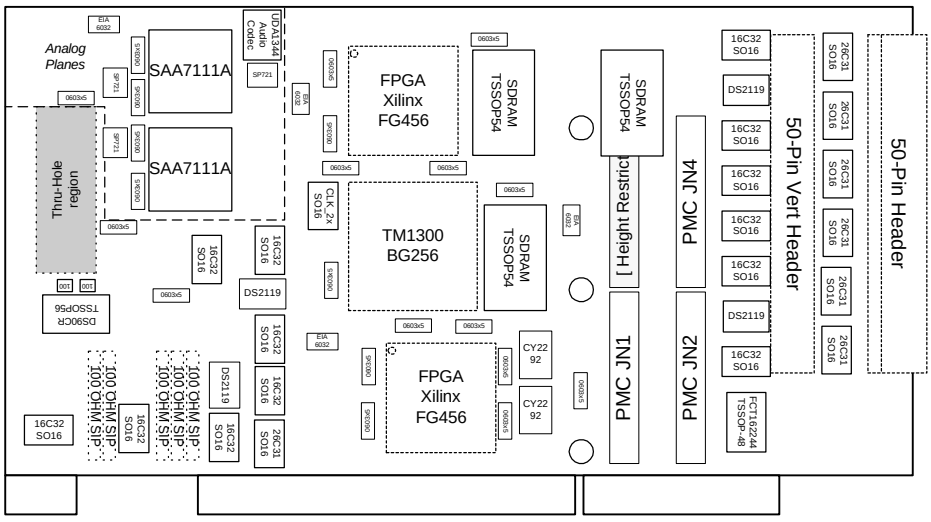


Figure 7 PCI Component Placements



B. PMC Component Placements

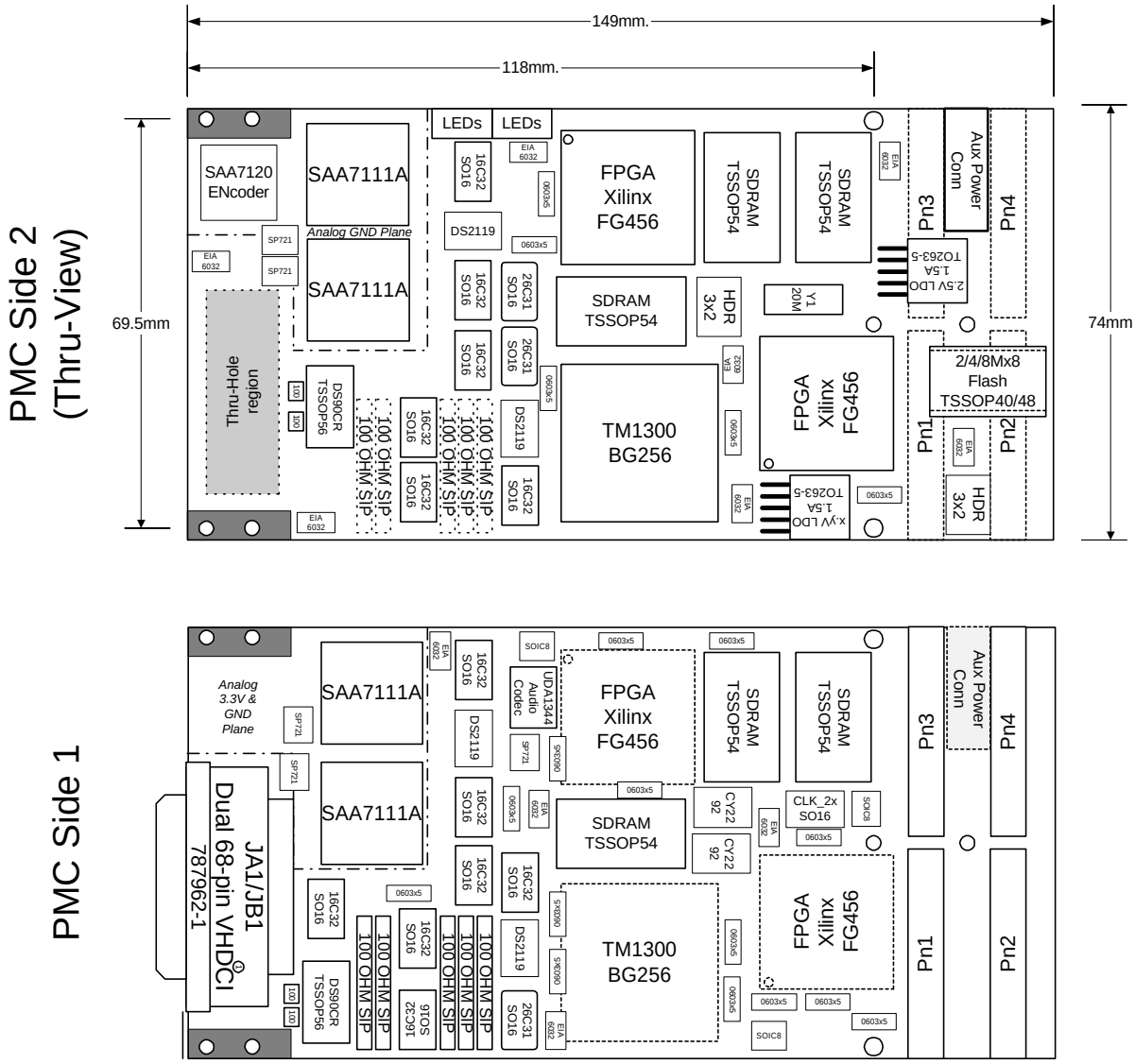


Figure 8 PMC Components Placements

XIII. 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.

XIV. 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.

A. 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, ALFAST, or FASTLIB 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 98.

_____ 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.

B. ***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.

C. 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