ICS-500R

OPERATING MANUAL

Interactive Circuits and Systems Ltd.
January 2005

Extra copies of this manual are available from the factory.

ICS Ltd
5430 Canotek Road
Gloucester, Ontario
K1J 9G2 Canada

http://www.ics-ltd.com
support@ics-ltd.com

Tel: (613) 749-9241
USA: (800) 267-9794
Fax: (613) 749-9461

E10651 Rev. C

TABLE OF CONTENTS

1 INTRODUCTION

The ICS-500R is a FPDP II to PCI Bus Buffer in PCI Mezzanine Card (PMC) format. The FPDP II interface provides receive capability (FPDP/RM and FPDP/R) only. The FPDP II interface reverts to ANSI/VITA 17 FPDP (FPDP I) operation when connected to one or more FPDP interfaces. The product is also available as FPDP transmit version (ICS-500RT). The PCI Bus interface is PCI 2.2 (64-bit, 66 MHz) compatible. The design has been optimized for applications that require bulk data transfer at very high speed between these interfaces.

The main features of ICS-500R FPDP II PMC Buffer are:

- Compatible with FPDP II
- Backward-compatible with ANSI/VITA 17 Front Panel Data Port (FPDP)
- Conforms to IEEE1386.1-2001 PCI Mezzanine Card (PMC) Specification
- Conforms to electrical specification of PCI Local Bus Specification Revision 2.2, with 64-bit, 66 MHz PCI interface
- 32-bit FPDP II interface supports data transfer at sustainable rates of up to 400 MBytes/s
- Linked-list DMA controllers on PCI bus
- 64-bit DMA on PCI interface, PCI burst at up to 528 MBytes/s
- FPDP function software selectable as FPDP Receive Master (FPDP/RM) or FPDP Receive (FPDP/R)
- Swing buffer with selectable buffer size up 2 MBytes (standard version), or 8 MBytes (extended memory version)
- Continuous and one-shot operating modes, software selectable
- Software-selectable channel selection and corner turning features
- Supports both packed and unpacked data
- Allows insertion of sequence numbers in data
- Windows and VxWorks software device drivers

All references to FPDP in this manual should be taken to refer to either FPDP II or FPDP operation of the product unless otherwise stated.

ICS offers software drivers for the ICS-500R for a number of platforms (including Windows NT, Windows 98, Windows 2000, Windows Me, and VxWorks). These drivers greatly simplify control and operation of the ICS-500R, and are strongly recommended. Using one of these drivers will generally save the programmer much time since he/she is relieved of the requirement to understand the complexities of the ICS-500R hardware model and of the PCI interface device. Contact ICS for further details.

1.1 References

1. ICS-500R Windows Software Development Kit User's Manual, Document No. E10694, Interactive Circuits and Systems Ltd.

2. QL5064 Production Information, QuickLogic Inc., www.quicklogic.com

3. ANSI/VITA 17-1998, American National Standard for Front Panel Data Port Specifications, VMEbus Industry Trade Association, 1999.

4. IEEE 1386.1-2001, IEEE Standard Physical and Environmental Layers for PCI Mezzanine Cards: PMC, IEEE, 2001

5. PCI Local Bus Specification, Revision 2.2, PCI Special Interest Group, December 18, 1998

1.2 Glossary

Quadword	A 64 bit word.

2 GENERAL DESCRIPTION

A simplified block diagram of the ICS-500R FPDP receiver is shown in Figure 1. The design includes two interfaces, an FPDP II / FPDP receive-only interface and a 64-bit, 66 MHz PCI 2.2 interface. Data provided to the FPDP input port may be in any of the data frame formats listed in the ANSI/VITA 17 FPDP standard (see Reference [3]). The data received at the FPDP interface is transferred to the PCI interface by way of channel selection and corner-turning logic and a 2 MByte swing buffer (optionally 8 MBytes). The swing buffer is implemented using high speed synchronous SRAM. The channel selection logic allows a sub-set of the input data to be selected, while the remainder is rejected. The corner-turning logic performs reordering of the data from channel sequence to time series, as required by most Digital Signal Processors (DSPs). Sequence numbers may be automatically inserted into the data stream to allow the user to verify data integrity.

The maximum data transfer speed for the FPDP interface when operating in FPDP II mode is 400 MBytes/s. The interface is backward compatible with the ANSI/VITA 17-1998 FPDP interface. When operating with an FPDP data source, the maximum transfer rate is in excess of 160 MBytes/s. The interface supports both packed and unpacked data, i.e. sampled data of less than 32-bit resolution may be packed with multiple samples in each 32-bit FPDP word.

The PCI interface uses the QuickLogic QuickPCI™ QL5064 interface chip. The QL5064 provides a PCI core and a programmable logic part. The PCI core has a 64-bit/66 MHz Master/Target PCI controller and a powerful DMA engine. The ICS-500R uses the QL5064 to support the PCI protocol and uses the programmable logic part to implement a 64-bit bi-directional ICS Internal Local Bus. With this arrangement, the ICS-500R PCI interface can provide up to the full 528 MBytes/s PCI 2.2 burst data transfer rate.

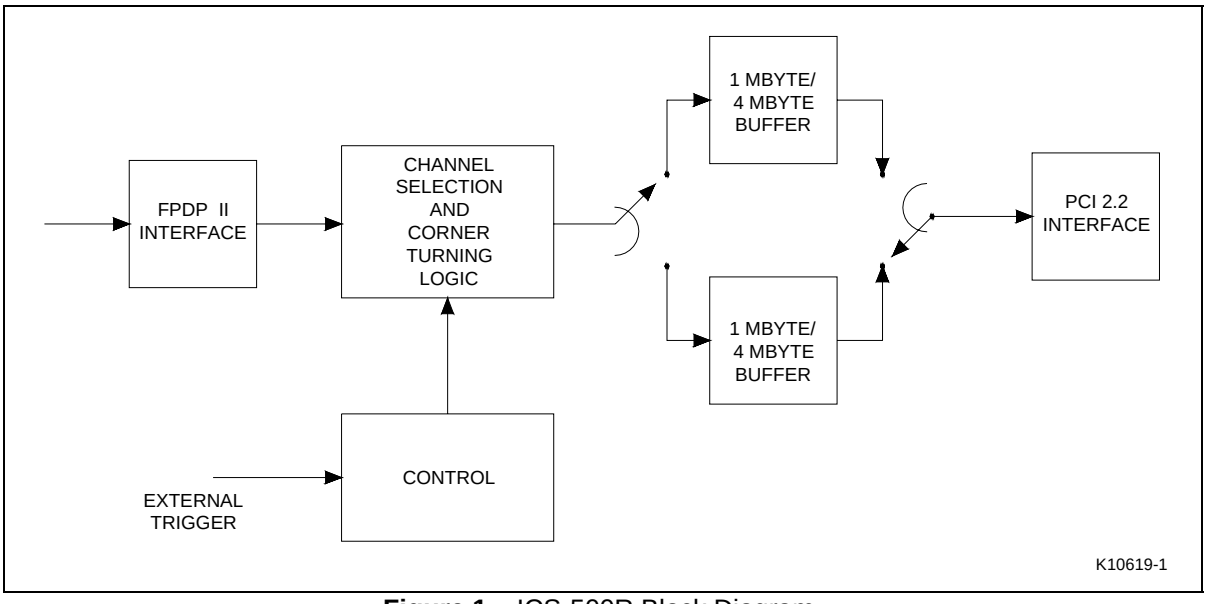


Figure 1 – ICS-500R Block Diagram

2.1 Board Specifications

General

Buffer Memory	2 x 1 MByte or 2 x 4 MByte
Trigger	Internal or External edge triggered or level triggered

FPDP Interface

Type	FPDP II Maximum strobe frequency 50 MHz. Maximum throughput 400 MBytes/s. All ANSI/VITA 17 data frame formats supported Reverts to ANSI/VITA 17-1998 FPDP operation with non-FPDP II capable interfaces

PCI Bus Interface

Type	PCI 2.2 compliant 64-bit, 66 MHz Master/Slave Capable of revertion to 64-bit, 33 MHz or 32-bit, 33 MHz "Univeral" type (accepts either 3.3V or 5V signalling)

Physical

Format:	Single width IEEE 1386.1 PCI Mezzanine Card (PMC) 74.0 x 149.0 mm
Environmental:	
Operating Temp:	0°- 50°C (at entry point of forced air)
Storage Temp:	-40° - +85°C
Humidity:	95% non-condensing
Cooling:	Approximately 490 LFM
Power:	
2 MByte memory version	1.6A at +3.3V 0.6A at +5V Total power consumption < 9W
8 MByte memory version	1.7A at +3.3V 1.0A at +5V Total power consumption < 11W

Specifications are subject to change without notice

3 DETAILED DESCRIPTION

3.1 FPDP / FPDP II Interface

The Front Panel Data Port (FPDP, see Reference [3] above) is an industry standard interconnection for board to board or system to system data transfer. This interface standard has gained acceptance in the industry for use in a broad range of signal processing applications. It is a high performance 32-bit parallel interface that uses ribbon cable to connect boards or systems together, providing a data transfer rate of up to 160 MBytes/s. The simple and well-defined physical and electrical interface provides the basis for integrating many different types of boards and sub-systems. The ICS-500R implements the FPDP II interface. This enhanced version of FPDP provides a maximum data transfer rate more than double that of FPDP, at 400 MBytes/s. An attractive feature of FPDP II is that it is backward compatible with FPDP; the FPDP II interface will sense the presence of one or more FPDP interfaces connected on the cable, and will revert to using FPDP protocol.

The P1 connector on the front panel of the ICS-500R connects to the FPDP / FPDP II interface. The ICS-500R implements an FPDP Receive Master or Receive interface (FPDP/RM or FPDP/R). When a single ICS-500R is present on the cable, it must be configured to be an FPDP/RM type. When multiple ICS-500R boards are present, only the board at the end of the cable should be configured as FPDP/TM; the remaining boards must be configured as FPDP/R. The selection of the Receive Master or Receive functionality is made by software control.

A Strobe signal is generated by the FPDP/TM (FPDP Transmit Master) interface that is the source of the data; this signal is used by the ICS-500R to clock the FPDP data. The ICS-500R accepts either the TTL or PECL Strobe (clock) signal. Strobe signal selection is made under software control in bit CR<9>. The PECL Strobe should be used when the required Strobe frequency is greater than 20 MHz, or when a long FPDP cable is used. The FPDP/TM interface that feeds data to the ICS-500R must be configured to use the same Strobe type as the ICS-500R. When using the PECL Strobe, the ICS-500R can operate at up to 50 MHz.

The ICS-500R will not accept data from the FPDP port until it has been triggered (see Section 3.4 below).

3.1.1 FPDP Data Frame Format

The ICS-500R supports all data frame formats defined in ANSI/VITA 17 FPDP Specifications (see Reference [3] above) - unframed data, single frame data, fixed size repeating frame data and dynamic size repeating frame data. Channel selection and corner turning are only available for fixed size repeating frame data mode.

The user should establish which modes of operation of FPDP are supported by the FPDP/TM interface that will communicate with the ICS-500R and select the appropriate available mode.

Before the ICS-500R is triggered to receive FPDP data, the Control register must be correctly configured to establish the frame format to be used (see Section 5.7.14).

3.1.2 Channel Selection

The channel selection feature allows the user to select a sub-set of each input data frame, i.e. a specific range of channels, when using Fixed Size Repeating Frame Data mode (see Section 3.1.1 above). Only data for these channels will be written to buffer memory. The data from other channels are ignored (see Figure 2). The feature allows only a consecutive range of channels to be selected; they are specified by the starting channel number and the number of channels to be selected.

The channel selection feature is always enabled when using Repeating Frame Data modes. If the starting channel is programmed as 0 and the selected channels is programmed as the frame length, then the entire data frame will be selected.

The channel selection feature is used as follows:

- Write the start channel number to the Start Channel register (CHAN_START) in the range 0 - 1027;
- Write the number of channels to be selected to the Selected Channels register (SEL_CHAN) in the range 1 -2048. In the special case in which the frame length is 1, Dynamic Size Repeating Frame Data mode should be selected in CR<14:13>.

The maximum number of channels allowed in the frame (the frame length) is 2048.

3.2 Corner Turning

The corner turning feature is used to reorganize the data from channel ordering into a time series, when using Fixed Size Repeating Frame Data mode (see Section 3.1.1 above). This is often required when the multiple channel sensor data is to be processed by a Digital Signal Processor (DSP). The FPDP data stream consists of frames of data – each frame contains a single data sample for each channel in the system. When corner turning is enabled, the data is reorganized in the buffer memory into multiple blocks of data in which each block contains a time series for a single channel. Figure 3 shows the effect of the corner turning function for unpacked FPDP data. Figure 4 relates to packed FPDP data.

To use the corner turning feature, the procedure is as follows:

- Enable corner turning by writing a '1' to CR<0> (see Section 5.7.1 below);
- Write the required block size to the Block Size register. The block size is the number of 64-bit words used to store the data from a single channel in one side of the Swing Buffer. The same block size will apply to each channel selected. For ICS-500-R2, the minimum legal value for the block size is 1024 (0x400). The Block Size must be a multiple of 1024. For example, 1024, 2048, 3072, …… For ICS-500-R8, the minimum legal value for the block size is 4096 (0x1000). The Block Size must be a multiple of 4096. Refer to Section 5.10 Block Size Register (BSR) and Section 5.11 Buffer Length Register (BLR) for detailed descriptions about how to choose the Block Size.
- Determine the buffer length required to accommodate the number of channels selected x the block size. Program this value to the Buffer Length register. Refer to 5.11 Buffer Length. Buffer Length Register (BLR) to calculate Buffer Length.

When using corner turning, the maximum number of channels that may be selected is sixty-four.

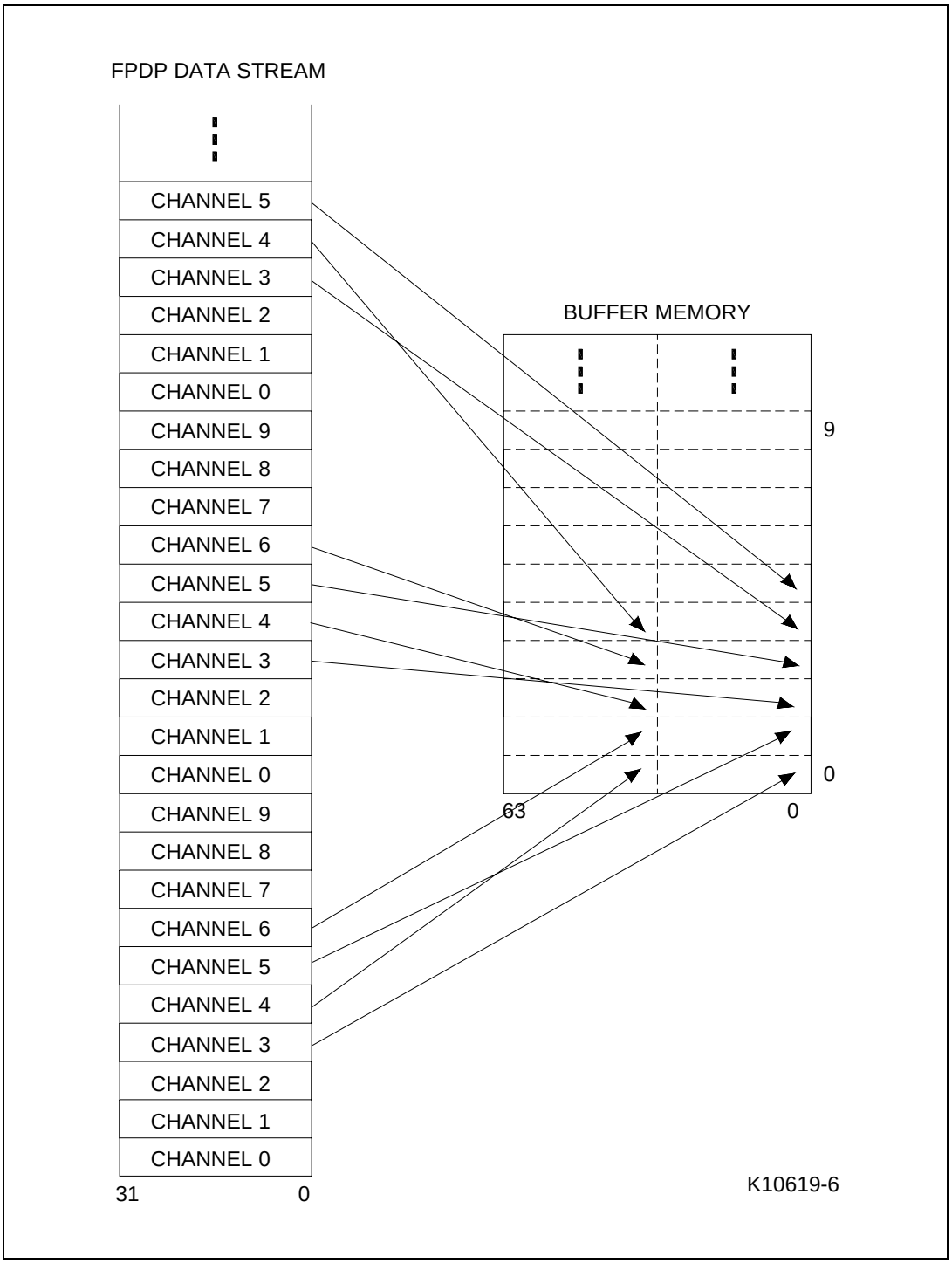


Figure 2 – Channel Selection (start channel = 3, selected channels = 4, no corner turning)

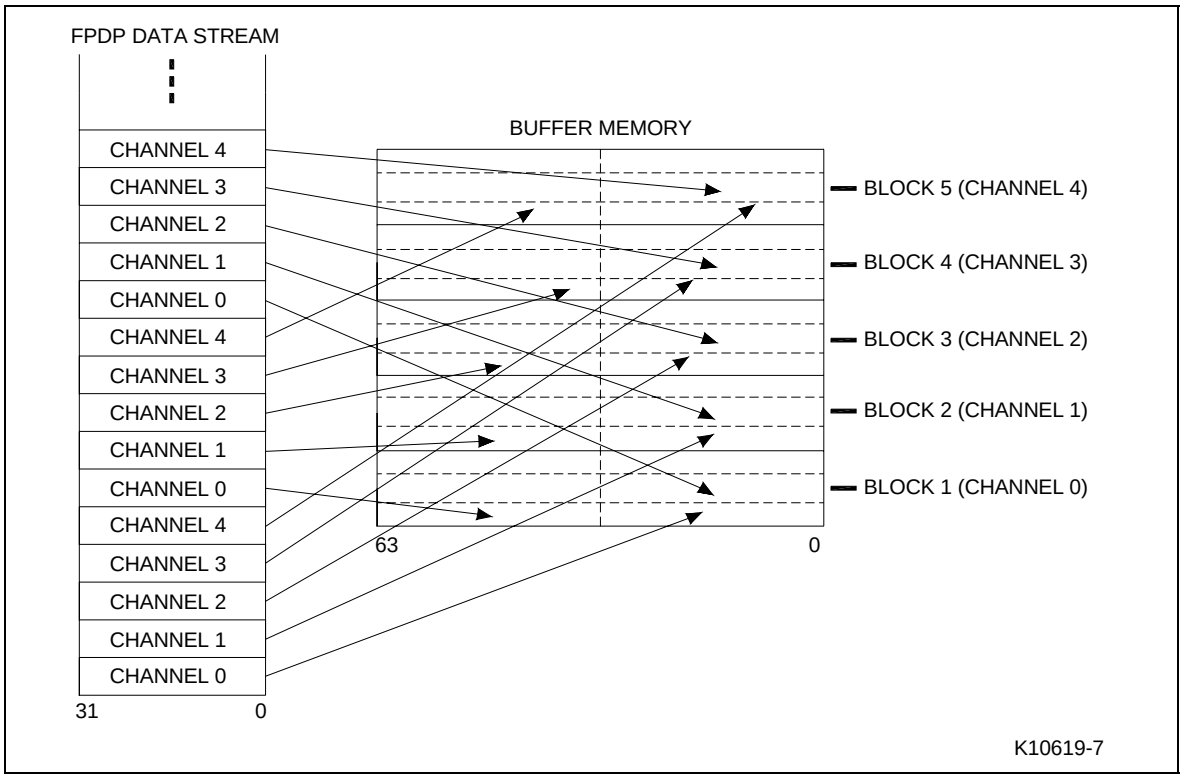


Figure 3 – Corner Turning for Unpacked FPDP Data

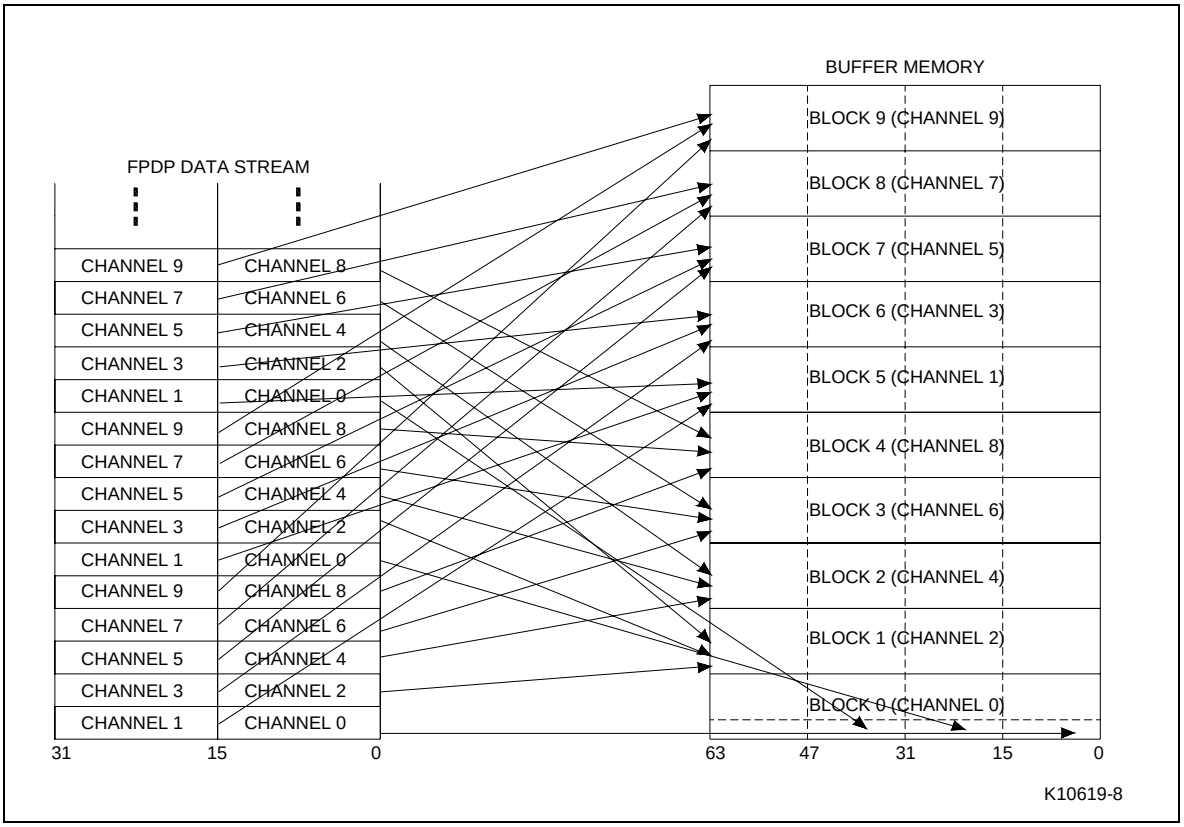


Figure 4 - Corner Turning for packed FPDP data

3.3 Trigger

The ICS-500R must be triggered before it will receive FPDP data, even though the FPDP bus may be active. When not triggered, the interface will not assert the FPDP Suspend signal – the interface will simply ignore all data transmitted on the bus.

The trigger may be either internal (software generated) or external, by means of a TTL-level signal applied to the PIO1 FPDP signal (pin 19) or to Jumper block J1 pin 1 on the board. The choice of internal/external trigger is selected by Control register bit CR<6>. If Internal Trigger is selected, the board is triggered by setting CR<12>.

The external trigger input may be selected to be either edge-sensitive or level-sensitive (selected by CR<11>). The edge-sensitive trigger causes triggering after the rising edge of the external trigger input. The signal must remain high for at least two periods of the FPDP Strobe signal. The level-sensitive trigger causes triggering while the signal remains high. The board will not process the input data stream when the external trigger is in the low state. When using the corner turning feature, only the edge-sensitive trigger may be used. When using the level-sensitive trigger, the trigger may be repeatedly enabled and disabled without reconfiguring the board.

When a Repeating Framed Data mode of operation is used (Fixed Length or Dynamic Length), data acqusition will always start at the beginning of a frame and end at the end of a frame, even though the edge-sensitive or level-sensitive trigger may be asynchronous with the data frames.

3.4 Operating Modes

The ICS-500R has two modes of operation – Continuous and One-shot modes.

3.4.1 Continuous Mode

When the ICS-500R is enabled and operating in Continuous mode, the ICS-500R continuously receives data from the FPDP interface once triggered by either an internal or external trigger. FPDP data is continuously transferred to the PCI Bus. After a programmable number of quadwords (equal to the Buffer Length) have been written to one side of the swing buffer, the buffer will swap, and data will be written to the other side of the buffer. A PCI bus interrupt will be generated each time the swing buffer swaps. There are several different trigger methods in continuous mode:

1. Internal trigger
   When using internal trigger, the board will begin to receive FPDP data only after the internal trigger bit on the Control Register (CR<12>) is set to '1'. The board continues receiving FPDP data until one of the following events occur:
   - The board is disabled (CR<4> = 0);
   - The internal trigger bit is cleared (CR<12> = 0);
   - Issue a Buffer Reset (refer to Section 5.12 Buffer Reset Register (BRR)).
2. Edge-sensitive external trigger
   When using edge-sensitive external trigger, the board will receive FPDP data after the rising edge of the external trigger signal. The board will continue receiving FPDP data until one of the following events occur:
   - The board is disabled (CR<4> = 0);
   - A Buffer Reset is issued (refer to Section 5.12).
3. Level-sensitive external trigger
   When using the level-sensitive external trigger, the board receives FPDP data only when the external trigger signal remains high. If the external trigger goes low, the ICS-500R ceases receiving FPDP data (at the end of the current frame for Framed data formats). When the external trigger signal goes high again, ICS-500R resumes receiving data. The board will stop receiving FPDP data when one of the following events occur:
   - External trigger signal is in the low state.
   - The board is disabled (CR<4> = 0);

- A Buffer Reset is issued (refer to Section 5.12).

After reconfiguring the board for a new FPDP data acquisition cycle, a Buffer Reset must be applied (refer to Section 5.12).

3.4.2 One-shot Mode

When the ICS-500R is enabled and operating in one-shot mode, the ICS-500R will acquire a number of quadwords equal to the value programmed to the Buffer Length register. When this has been completed, a PCI Bus interrupt will be generated and data acquisition will cease.

Trigger behavior is similar to that of Continuous mode. For all trigger types, when the total number of data words read from the FPDP interface is equal to the programmed buffer length, acquisition will cease and an interrupt will occur, if enabled.

After reconfiguring the board for a new FPDP data acquisition cycle, a Buffer Reset must be applied (refer to Section 5.12).

3.5 Sequence Number

A sequence number may be inserted in the acquired data as the first word in each side of the swing buffer. A 32-bit counter is used to count the number of swing buffers that have been read by the host machine. When the sequence number counter is enabled by setting the CR<7> Sequence Number Enable bit (see Section 5.7.8), the current value of the counter is written in the first 64-bit word of each buffer.

3.6 Single Board / Multiple Board Operation

The ICS-500R may be operated either in single or multiple board configurations. This provides for situations where it is desirable to feed all or part of the input data to multiple host CPU or DSP boards. The channel selection feature enables each board to process a sub-set of the input data stream. The ICS-500R FPDP / FPDP II interface may be configured either as FPDP Receive Master (FPDP/RM) or as FPDP Receiver (FPDP/R), as defined in Reference [3]. FPDP/RM operation is selected by setting CR<3>; FPDP/R operation is selected by clearing CR<3>. In a single board configuration, when the ICS-500R is the only FPDP receive device connected to the FPDP cable, it must always be configured as FPDP/RM, in order to ensure that the bus is correctly terminated. In a multiple board configuration, when more than one ICS-500R boards is attached to the same cable, only one board should be configured as FPDP/RM; this board must be located at one end of the cable. All other boards should be configured as FPDP/R.

3.7 PCI Interface

The ICS-500R implements a 64-bit, 66 MHz PCI bus Master/Slave interface compliant with the PCI 2.2 specification. It uses the QuickLogic QuickPCI™ QL5064 PCI Bus interface chip. The QL5064 includes a PCI core and a field-programmable gate array (FPGA). The PCI core implements the PCI protocol while the FPGA contains an ICS board level Local Bus Controller in the FPGA part. The local bus controller converts the signals from PCI core to the ICS Internal Local Bus Signals. With this mechanism, both PCI bus and the internal local bus could operate at up to full PCI bandwidth (528 MBytes/s). When the data in Buffer Memory are read out through PCI bus, ICS-500R uses the chain mode DMA Channel in QL5064.

ICS-500R will operate in 64-bit mode and at 66 MHz clock speed on busses that support these options.

The ICS-500R accepts either 3.3V or 5V signalling and is keyed in accordance with Reference [4].

3.8 Light-Emitting Diodes

The ICS-500R is fitted with a set of light-emitting diodes (LEDs) which provide information about board operation. The LEDs are provided for diagnosis of major operational problems of PCI bus and the internal local bus.

The diodes are located on side 2 of the PMC board (outside of the module when it is mounted on the host module). There are three LEDs installed in a horizontal line. Table 3.1 describes the ICS-500R LEDs starting the leftmost LED when the board is installed in normal vertical orientation (see Figure 5).

Note that the intensity of illumination of each LED will depend on the frequency of the access to that interface. It may not be possible to detect infrequent accesses to an interface.

Table 3.1 Light Emitting Diodes

	LED Function	Color	Description
LED1	Local bus busy	Green	Illuminated when the local bus is busy
LED2	DMA State	Green	Illuminated when DMA is in progress
LED3	Local bus interrupt request	Green	Illuminated when there are PCI interrupts requests from local bus

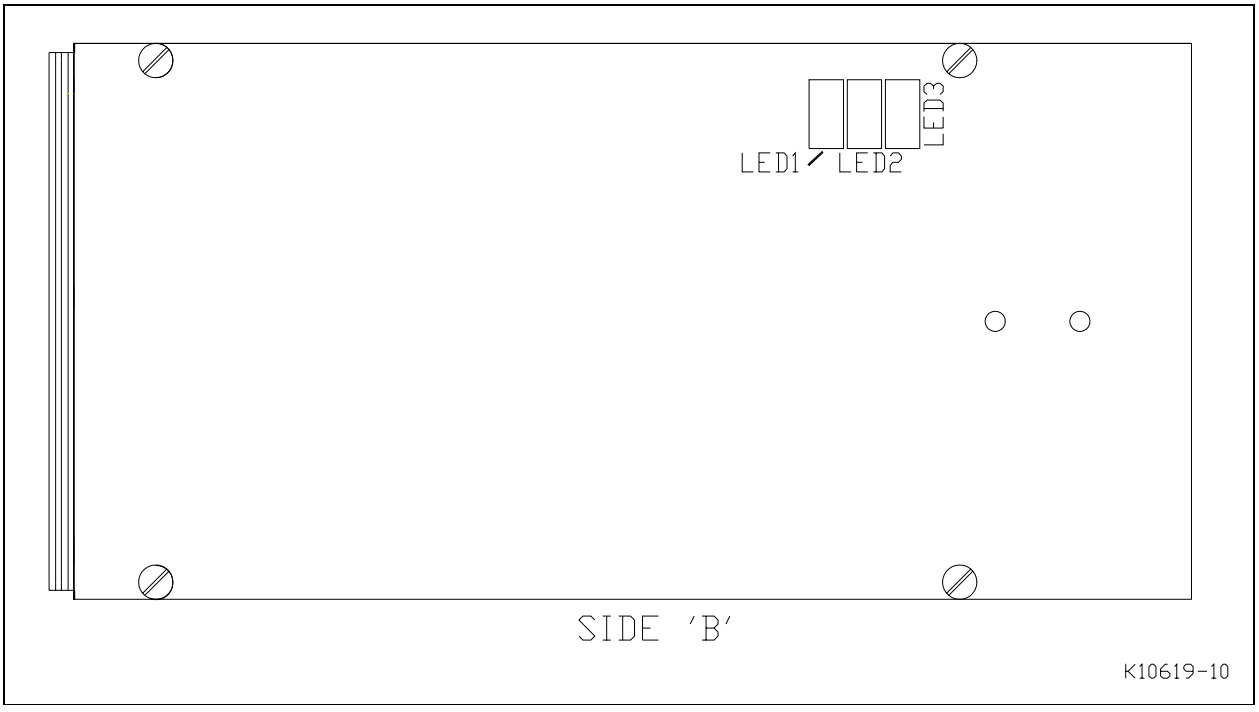


Figure 5 – Position of LED indicators on ICS-500R

4 HARDWARE CONFIGURATION

This section provides information necessary for hardware preparation of the ICS-500R.

When an external trigger is used, it may be provided either on the FPDP PIO1 signal, or by means of a connection to pin 1 of the J1 jumper block located on side A of the module (see Figure 6). When using an external trigger, the J1 jumper (wire link) must be configured according to the source of the trigger.

When using the internal (software-controlled) trigger, the jumper should be left in the factory default position, bridging pins 2 and 4 of J1.

When using an external trigger with the trigger signal supplied as a TTL-level signal, the jumper should be left in the factory default position, and the trigger signal should be connected to J1 pin 1. Pin 3 is the ground connection.

When using an external trigger with the trigger signal supplied as a TTL-level signal on the FPDP PIO1 signal wire, the jumper must be moved to the position bridging J1 pins 1 and 2.

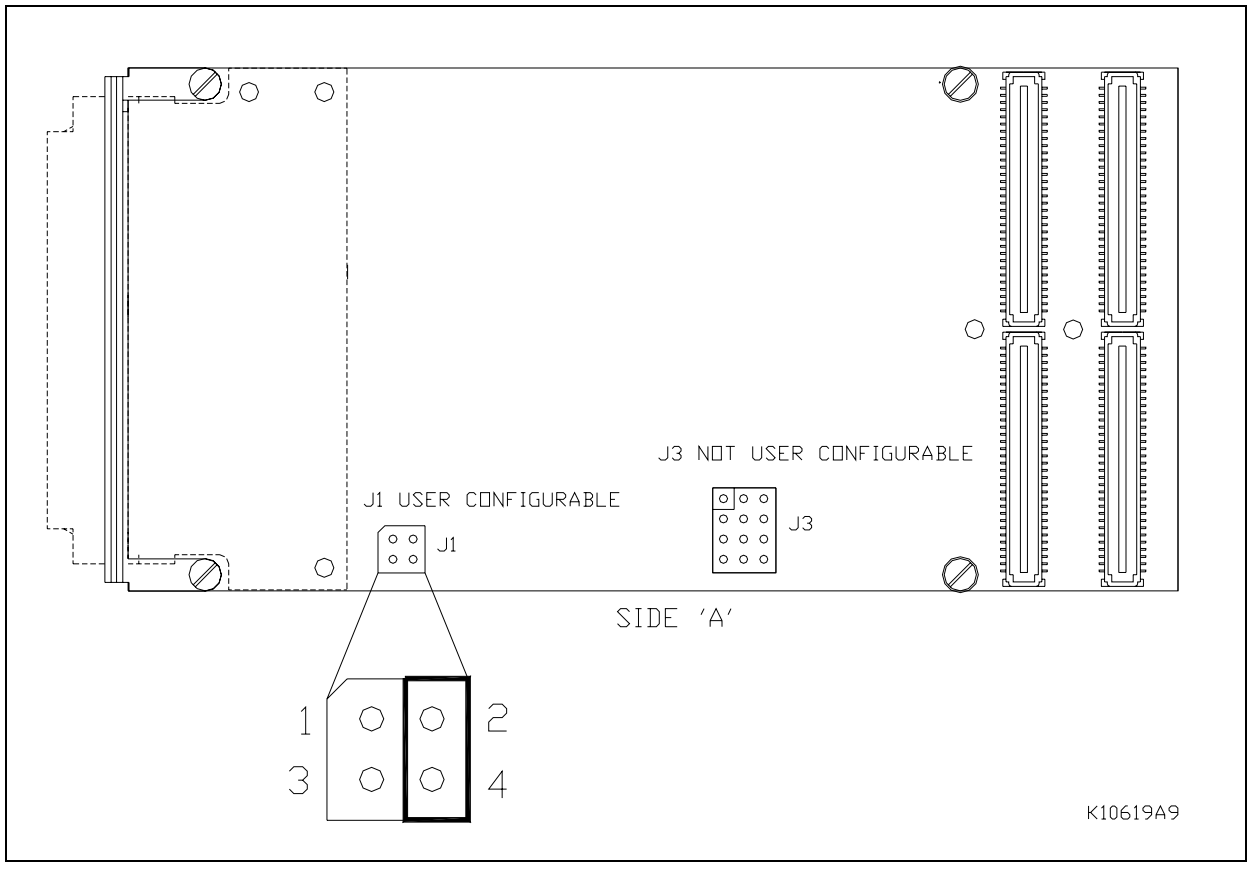


Figure 6 – Position of J1 Jumper Block on ICS-500R

5 PROGRAMMING MODEL

The ICS-500R is controlled by means of memory-mapped registers accessed through the PCI interface. The PCI address map is summarized in Table 5.1. The ICS-500R register set is listed in Table 5.2. Data should be read from the PCI Bus interface using the chain mode DMA controller of the QL5064, as described in Section 5.3 below.

A detailed description of the QL5064 register and bit allocations is given in Reference [2].

A detailed list of ICS-500R registers is given in Table 5.2. Register bit allocations are given in Figure 7.

Table 5.1 PCI Address Map (Summary)

Name	PCI Base Address	Description	Size
QL5064 Internal registers	PCI_BASE0		4K Bytes
QL5064 Configuration Space 1	PCI_BASE0+ 0x00-0xFF	QL5064 control resisters Memory space, 32-bit addressing	256 Bytes
QL5064 Configuration Space 2	PCI_BASE0+ 0x1000-0x10FF	QL5064 control resisters Memory space, 32-bit addressing	256 Bytes
ICS-500R Register Set	PCI_BASE1	ICS-500R register access	4K Bytes
ICS-500R Memory Space	PCI_BASE2	ICS-500R buffer memory access	16 MBytes

Table 5.2 ICS-500R Address Map

Register	**PCI Bus Address**	**Type**
ICS-500R Register Set	**PCI_BASE1**	
Status register	PCI_BASE1 + 0x0000	Read only
Interrupt Mask register	PCI_BASE1 + 0x0008	Read/Write
Control register	PCI_BASE1 + 0x0010	Read/Write
Channel Start register	PCI_BASE1 + 0x0018	Read/Write
Select Channel register	PCI_BASE1 + 0x0020	Read/Write
Buffer Length register	PCI_BASE1 + 0x0028	Read/Write
Block Size register	PCI_BASE1 + 0x0030	Read/Write
Buffer Reset register	PCI_BASE1 + 0x0050	Write Only

5.1 General Notes

All programming and control of the ICS-500R is accomplished through the PCI Bus interface using 64-bit or 32-bit transfers. All control register bits that are not defined have no effect on the operation of the ICS-500R. All undefined bits may be read as zero or one.

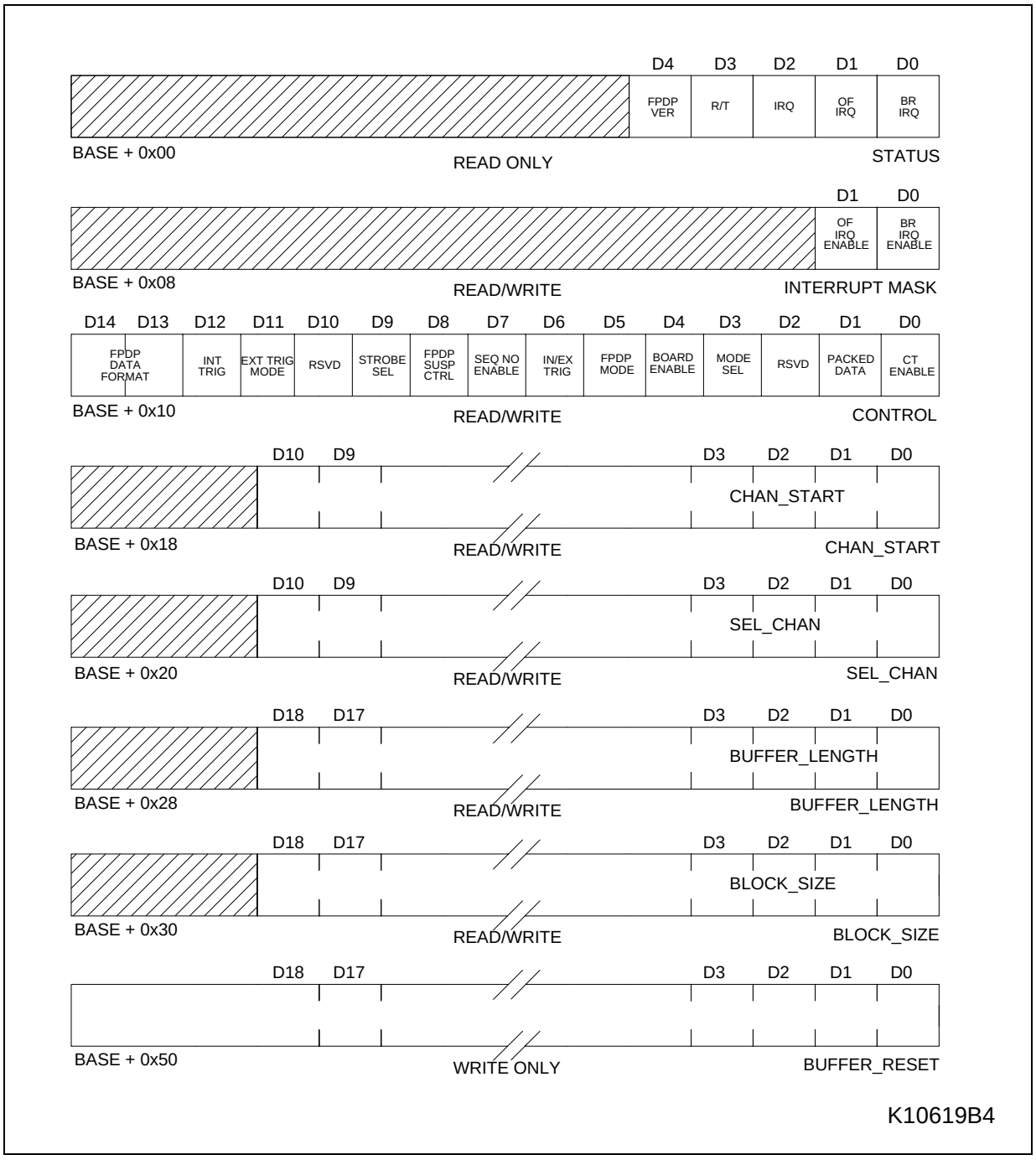


Figure 7 – ICS-500R Register Map

5.2 QL5064 Registers

The ICS-500R uses the QuickLogic QL5064 PCI Bus interface chip. The QL5064 register set is complex, and a complete description is not given here. Details of the QL5064 may be found in the QL5064 data sheet (see Reference [2]).

As with any PCI Bus device, the QL5064 has a unique 256-byte region called the Configuration Header Space, or just Configuration Space (see Reference [2] and Reference [5]). It is mandatory that portions of this configuration header are addressable in order for a PCI agent to be in full compliance with the PCI specification. In order for ICS-500R device registers to be addressable, it is necessary that information in the configuration space is read. Base address values and lengths must then be written to registers also in configuration space. This is normally done by the operating system.

The ICS-500R uses three of the available base address registers defined by the PCI specification for access the module. These base address registers, and the lengths assigned to the corresponding mapped areas of PCI address space are given in Table 5.2. The space addressed by PCI_BASE1 is split into two areas of 256 bytes, with an unused area between them.

The QL5064 registers are described in Table 5.3.

The vendor and sub-vendor information contained in the PCI configuration space is as follows:

Vendor ID: 0x11E3
Device ID: 0x0006
Sub-vendor ID: 0x1464
Sub-system ID: 0x0500

5.3 Endian Conversion

Endian ordering refers to the significance with which data is ordered within a computer system. The ICS-500R implements “Little-endian” ordering of data within larger data widths. When operating the ICS-500R within a PC- or Intel-based system, no endian conversion will be required, since these are Little-endian systems. In Little-endian systems, the highest address is applied to the most significant byte (or word) of data of a given size; in Big-endian systems (e.g. most Unix-based systems such as Sparc and PowerPC), the lowest address is applied to the most significant bytes (or word) of data in an element of data of a given size.

The QL5064 interface provides “endian” conversion for Big-endian systems for a variety of data types. This feature results in 8-bit, 16-bit or 32-bit word order being reversed within larger word sizes (16-bit, 32-bit or 64-bit). The selection of endian function is made in the QL5064 Control register, address offset 0x1008 (see Table 5.3). Table 5.4 describes the effects of the various settings of this register.

The Endian ordering should be set according to the type of system in which the ICS-500R is installed and the bus width employed (32-bit or 64-bit). The most commonly used settings are shown in bold type.

Note that when using chain-mode DMA (see Section 5.4 below), the linked list data must be reordered by the user before being written to host memory, when using a Big-endian host. This is because the data is read directly by the DMA controller without being passed through the endian conversion hardware.

Table 5.3 QL5064 Control Registers

Register	Offset (PCI_BASE0)	Type	Width (bits)	Description
Master Write Address 0	0x00	RW	64	PCI address for write to PCI target
Master Write Transfer Count 0	0x08	RW	32	Write transfer count in bytes
Master Read Address 0 / Chain Descriptor Start Address	0x48	RW	64	PCI address for read from PCI target / pointer to first chain descriptor PCI start address
Master Read Transfer Count 0	0x50	RW	32	Read transfer count in bytes
FIFO Level Controls	0x68	RW	64	Almost full and almost empty levels for 6 FIFOs Recommended setting is 0x0008.0020.0020.001B
DMA Control	0x90	RW	64	Set Bit 16 to start Transmit 0 DMA transfer Set Bit 18 to start Receive 0 DMA transfer Set Bit 20 to start Chain Mode DMA transfer
Board Reset	0xE8	W	1 (bit 0)	Write 1 - perform a Host Total Reset Write 0 - no effect
Control Register	0x1008	R/W	8	[01:00] must be 00
LOC_XMT0_CNT_ADDR	0x1010	RW	64	[31:00] Local Start Address for DMA Transmit 0 [63:32] Transfer count in bytes
LOC_RCV0_CNT_ADDR	0x1020	RW	64	[31:00] Local Start Address for DMA Receive 0 [63:32] Transfer count in bytes
LOC_DMA_CANCEL	0x1030	W	8	Write 1 to this register to cancel unfinished DMA

Table 5.4 Endian Conversion when transferring data from ICS-500R Local Bus to PCI Bus

PCI Bus		**AD[63:56]**	**AD[55:48]**	**AD[47:40]**	**AD[39:32]**	**AD[31:24]**	**AD[23:16]**	**AD[15:8]**	**AD[7:0]**
	CR<5:2>								
ICS Local Data Bus	**0000**	**LD[63:56]**	**LD[55:48]**	**LD[47:40]**	**LD[39:32]**	**LD[31:24]**	**LD[23:16]**	**LD[15:8]**	**LD[7:0]**
	0001	**LD[31:24]**	**LD[23:16]**	**LD[15:8]**	**LD[7:0]**	**LD[63:56]**	**LD[55:48]**	**LD[47:40]**	**LD[39:32]**
	0010	LD[15:8]	LD[7:0]	LD[63:56]	LD[55:48]	LD[47:40]	LD[39:32]	LD[31:24]	LD[23:16]
	0011	LD[23:16]	LD[15:8]	LD[7:0]	LD[63:56]	LD[55:48]	LD[47:40]	LD[39:32]	LD[31:24]
	0100	LD[7:0]	LD[63:56]	LD[55:48]	LD[47:40]	LD[39:32]	LD[31:24]	LD[23:16]	LD[15:8]
	0101	LD[39:32]	LD[31:24]	LD[23:16]	LD[15:8]	LD[7:0]	LD[63:56]	LD[55:48]	LD[47:40]
	0110	LD[47:40]	LD[39:32]	LD[31:24]	LD[23:16]	LD[15:8]	LD[7:0]	LD[63:56]	LD[55:48]
	0111	LD[55:48]	LD[47:40]	LD[39:32]	LD[31:24]	LD[23:16]	LD[15:8]	LD[7:0]	LD[63:56]
	1000	**LD[7:0]**	**LD[15:8]**	**LD[23:16]**	**LD[31:24]**	**LD[39:32]**	**LD[47:40]**	**LD[55:48]**	**LD[63:56]**
	1001	**LD[39:32]**	**LD[47:40]**	**LD[55:48]**	**LD[63:56]**	**LD[7:0]**	**LD[15:8]**	**LD[23:16]**	**LD[31:24]**

5.4 Performing DMA Transfers

When reading data from the ICS-500R through the PCI interface, the DMA controller in the QL5064 should be used. The controller can be used with or without chain mode (scatter-gather) operation. The QL5064 should normally be the PCI bus master (initiator) for transfers; this will result in improved transfer efficiency. When using chain mode DMA, the DMA controller requires the chain mode descriptor tables to be located in PCI addressable memory. This is usually located in the host processor.

The QL5064 PCI core uses its Transmit 0 DMA channel to move data from the ICS-500R to PCI Bus and its Receive 0 DMA channel to move data from PCI Bus to the ICS-500R. The ICS-500R does not use the Receive DMA function.

5.4.1 Non-Chain Mode DMA operation

To use DMA without scatter-gather, the QL5064 DMA controller must be programmed with the PCI Bus starting address for the destination data and the length of the transfer.

The steps required to start Transmit 0 DMA operation for writing data from the ICS-500R to a PCI Bus target address are as follows:

1. Write the starting PCI target address to QL5064 control register "Master Write Address 0" (offset 0x00).
2. Write the transfer count (in bytes) to QL5064 control register "Master Write Transfer Count 0" (offset 0x08, bit[31:0]. The minimum transfer count is 8 bytes. The maximum transfer count is 2^{32}-8 bytes. Transfers must always be in multiples of 8 bytes. The transfer must be quadword (8 byte) aligned.
3. Write the transfer count and starting Local address to the ICS Local Bus control register LOC_XMT0_CNT_ADDR (offset 0x1010). The transfer count written to this register must be the same as that written to the Master Write Transfer Count 0 register (offset 0x08). The starting local address must be quadword (8 byte) aligned.
4. To start the DMA transfer, write a "1" to the QL5064 control register DMA Control register Start/Done bit (offset 0x90, bit 16).
5. The transfer will continue until the transfer count expires or is cancelled.

5.4.2 Chain Mode DMA operation

The following steps are required to use chain mode DMA:

1. Write the chain mode DMA descriptor tables to PCI memory. The format of the Chain Mode DMA descriptor tables is given in Figure 8. The requirements for Local Starting Address and Transfer Count in the chain descriptors are the same as for non-chain mode DMA. If the host is a “Big-endian” machine, it will be necessary to reorder the descriptor data, as described in Section 5.3 above. The usual reordering required is byte order within 64-bit words. However, this may vary depending on the details of the system implementation.
2. Write the PCI address of the first descriptor pointer to QL5064 control register "Master Read Address/Chain Descriptor Start Address" (PCI offset 0x48).
3. Start execution of the instructions in the descriptor tables by writing "1" to "Chain Enable" (PCI offset 0x90, bit[20]).
4. The chain mode DMA will run until the end-of-chain flag is encountered or until the DMA transfer is cancelled.

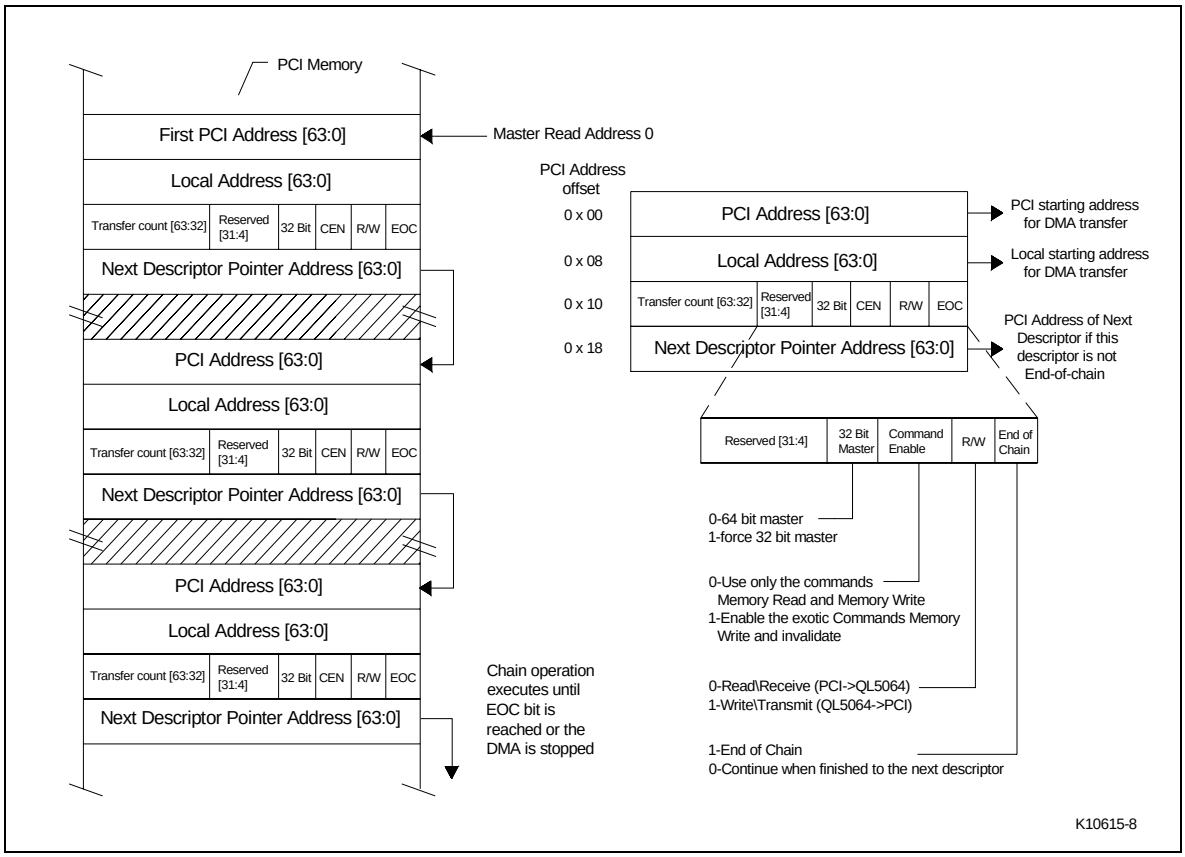


Figure 8 – DMA Chain Mode Descriptors

5.4.3 Cancelling a DMA Transfer

It may become necessary to cancel a DMA transfer that is in progress, for example if a time out has occurred. The procedure for doing this is as follows:

1. Cancel DMA in QL5064 core (see Reference [2]).
2. Write 1 to the LOC_DMA_CANCEL register (see Table 5.3 above).

The order of operations shown above must be observed.

5.5 Status Register (SR)

Read Only

The Status Register contains information about the state of ICS-500R. It is used to determine the ICS-500R local bus interrupt status, and the source of a pending interrupt.

5.5.1 SR<0> - Buffer Read IRQ

The bit reflects the status of the buffer access control unit. When the swing buffer swaps (continuous mode) or a data write completes (one-shot mode), the buffer access control unit will assert this flag, indicating that new data is available to be read from the buffer memory. If the Buffer Read interrupt request is enabled in the IMR register when this bit is asserted, an interrupt request will occur.

SR<0>	Buffer Read IRQ
READ ONLY	
0	Buffer access control unit is not asserting interrupt request
1	Buffer access control unit is asserting interrupt request

5.5.2 SR<1> - Buffer Overflow IRQ

This bit indicates that the swing buffer is full. Once the swing buffer is full, the buffer access control unit asserts the SUSPEND# signal of the FPDP interface to suspend the FPDP data stream (see Table C.2).

SR<1>	Buffer Overflow IRQ
READ ONLY	
0	FPDP data is not suspended
1	FPDP data steam is suspended and Buffer Overflow interrupt request is asserted.

5.5.3 SR<2> - IRQ (Interrupt Request)

This bit indicates that the ICS-500R is asserting a PCI bus interrupt from either of the two available sources (i.e. Buffer Read IRQ or Buffer Overflow IRQ).

SR<1>	Interrupt Request
READ ONLY	
0	PCI Bus interrupt is not asserted
1	PCI Bus interrupt is asserted

5.5.4 SR<3> - R/T

This bit indicates whether the board is a FPDP transmitter (ICS-500T) or receiver (ICS-500R). The value read will always be 0 for the ICS-500R.

SR<3>	R/T
READ ONLY	
0	Value read is always 0 (ICS-500R).

5.5.5 SR<4> - FPDP Version

This bit indicates whether the ICS-500R FPDP interface is receiving FPDP II or FPDP data protocal.

SR<4>	FPDP Version
READ ONLY	
0	FPDP
1	FPDP II

5.6 Interrupt Mask Register (IMR)

Read/Write

This register is used to enable PCI bus interrupts for the Buffer Read interrupt request and for the Buffer Overflow interrupt request. Masked interrupts are still seen in the corresponding bits of the status register, but do not cause an interrupt to occur.

5.6.1 IMR<0> - Buffer Read IRQ Enable

This bit is used to enable the Buffer Read IRQ. The interrupt handler must clear this bit in order to prevent the possibility of a further interrupt before the swing buffer has been read to the programmed length.

IMR<0>	Buffer Read IRQ Enable
READ/WRITE	
0	Buffer Read IRQ is disabled
1	Buffer Read IRQ is enabled

5.6.2 IMR<1> - Buffer Overflow IRQ Enable

This bit is used to enable the Buffer Overflow IRQ. The interrupt handler must clear this bit in order to prevent the possibility of a further interrupt.

IMR<0>	Buffer Overflow IRQ Enable
READ/WRITE	
0	Buffer Overflow IRQ is disabled
1	Buffer Overflow IRQ is enabled

5.7 Control Register (CR)

Read/Write

The control register allows the user to configure the following operating parameters:

- Enable Conner Turning
- FPDP data format (packed/un-packed)
- Enable Channel Selecting
- Operation Mode (Continuous/One shot)
- Enable the board
- FPDP RM/R
- FPDP Termination
- Sequence number enable
- Internal/External trigger
- Internal Trigger
- Select FPDP clock (TTL or PECL)

5.7.1 CR<0> - Corner Turning Enable

This bit controls the selection of the Corner Turning function of the board

CR<0>	**Enable Corner Turning**
READ/WRITE	
0	Corner Turning is disabled
1	Corner Turning is enabled

5.7.2 CR<1> - FPDP data format (packed/un-packed)

This bit indicates the data format of FPDP data stream (Packed or Un-packed). When unpacked data is selected and Corner Turning is enabled, input data is assumed to be 32-bit for purposes of reorganization into the buffer. When packed data is selected and Corner Turning is enabled, input data is assumed to be 16-bit.

CR<1>	FPDP Data Format
READ/WRITE	
0	FPDP Data is unpacked
1	FPDP Data is packed

5.7.3 CR<2> - Reserved

Reserved. Must always be set to '0'.

5.7.4 CR<3> - Operating Mode

This bit controls the selection of FPDP receiver operation mode.

CR<3>	Operating Mode
READ/WRITE	
0	Continuous mode
1	One-shot mode

5.7.5 CR<4> - Board Enable

This bit is used to enable the board. This must be done after the configuration has been programmed to the board.

CR<4>	Board Enable
READ/WRITE	
0	Board is disabled
1	Board is enabled

5.7.6 CR<5> - FPDP Mode

This bit is used to control the selection of FPDP Mode as either Receive Master (FPDP/RM) or Receiver (FPDP/R). See Section 3.7 above for details.

CR<5>	FPDP Mode
READ/WRITE	
0	FPDP/R (Receive)
1	FPDP/RM (Receive Master)

5.7.7 CR<6> - Internal/External Trigger

This bit selects internal or external triggering. When internal triggering is selected, the board must be triggered under software control by writing a 1 to CR<12>. When external triggering is selected, the board is triggered using a TTL signal applied to the EXT_TRIG pin of J1 or Pin 19 (PIO1) of P1 (FPDP connector). The trigger mode may be selected as either edge-sensitive or level-sensitive by means of the appropriate setting in CR<11>. When edge triggering is used, triggering occurs following the rising edge of the external trigger signal. The signal must remain high for at least two cycles of the FPDP Strobe (clock). When level-sensitive triggering is used, the board acquires data only while the trigger signal is in the high state. When using the channel section and/or corner turning features with the external trigger, only the edge-sensitive trigger may be used.

CR<6>	**Internal/External Trigger**
READ/WRITE	
0	Internal Trigger
1	External Trigger

5.7.8 CR<7> - Sequence Number Enable

This bit controls the Sequence Number feature (see Section 3.5 above). When enabled, the first word written to swing buffer after each buffer swap is the current value of the sequence number counter. When disabled, no sequence number is written to swing buffer.

CR<7>	**Sequence Number Enable**
READ/WRITE	
0	Sequence number disabled
1	Sequence number enabled

5.7.9 CR<8> - FPDP Suspend Control

This bit selects whether the FPDP Suspend signal is active. When enabled, Suspend will be asserted by the ICS-500R when the board is unable to accept more data. As soon as the board is able to accept more data it will be negated. When disabled, the Suspend bit will never be asserted even if the ICS-500R is unable to accept further data. In this case, data may be lost if the data source does not limit the rate at which data is transmitted.

CR<8>	**FPDP Suspend Control**
READ/WRITE	
0	FPDP Suspend enabled
1	FPDP Suspend disabled

5.7.10 CR<9> - FPDP Strobe Selection

This bit is used to select TTL or PECL FPDP strobe. The PECL Strobe should be used, if supported by the transmitting interface, when Strobe frequencies higher than 20 MHz are used.

CR<9>	FPDP Strobe Selection
READ/WRITE	
0	TTL Strobe
1	PECL Strobe

5.7.11 CR<10> - Reserved

Reserved. Must always be set to '0'.

5.7.12 CR<11> - External Trigger Mode

This bit is used to select the external trigger as either edge or level sensitive. When using the corner turning feature (see CR<0>) with the external trigger, only the edge-sensitive trigger may be used.

CR<11>	External Trigger Mode
READ/WRITE	
0	External trigger is edge-sensitive
1	External trigger is level-sensitive

5.7.13 CR<12> - Internal Trigger

When internal trigger is selected by CR<6>, this bit is used to trigger the board.

CR<12>	Internal Trigger
READ/WRITE	
0	No effect
1	Trigger board

5.7.14 CR<14:13> - FPDP Data Frame Format

These bits select the FPDP data frame format of the data to be received from the FPDP bus.

CR<14:13>	FPDP Data Frame Format
READ/WRITE	
00	Fixed Size Repeating Frame Data
01	Dynamic Size Repeating Frame Data
10	Single Frame Data
11	Unframed Data

5.8 Channel Start Register (Chan_Start)

Read/Write

This register is used to select the starting word number in the FPDP data stream for acquisition of data from each frame, when using Fixed Size Repeating Frame Data mode (see Section 3.1.1 and Section 5.7.14 above). In other modes, it does not need to be programmed. The channel selector counts the 32-bit FPDP words in the data frame, with number 0 representing the first word in the frame. The maximum number that can be set in this register is 2047. To select all channels in the frame for processing, this register should be programmed with the value 0 and the Selected Channel register should be programmed with the number of 32-bit words in the frame. The channel selection feature is described in more details in Section 3.2 above.

5.9 Selected Channel Register (Sel_Chan)

Read/Write

This register programs the number of 32-bit words to be acquired by the ICS-500R from each FPDP data frame, when using Fixed Size Repeating Frame Data mode (see Section 3.1.1 and Section 5.7.14 above). In other modes, it does not need to be programmed. The range of valid numbers that may be programmed to this register is 1 - 2048. To select all channels in the frame for processing, this register should therefore be programmed with the number of 32-bit words in the data frame. The channel selection feature is described in more details in Section 3.2 above. When the Corner Turning feature is used, the maximum value that may be programmed is 64.

To select the whole frame for processing in the special case of Repeating Frame Data mode (Fixed Size or Dynamic Size) with data frames containing a single channel, Dynamic Size Repeating Frame Data should be selected in CR<14:13>. When this mode is selected, the entire range of available input channels is selected under all circumstances.

5.10 Block Size Register (BSR)

Read/Write

This register is used to set the memory block size in 64-bit words. This value determines the size of memory block used for storage of data from each channel when the Corner Turning feature is enabled (see Section 3.3 above). This feature is only available when using Fixed Size Repeating Frame Data mode (see Section 3.1.1 and Section 5.7.14 above). In other modes, it does not need to be programmed. For ICS-500-R2, the legal values that may be programmed are multiples of 1K (1024) 64-bit words, ranging from 1K to the maximum buffer size. For ICS-500-R8, the legal

values that may be programmed are multiples of 4K (4096) 64-bit words, ranging from 4K to the maximum buffer size.

5.11 Buffer Length Register (BLR)

Read/Write

The Buffer Length register is used to determine the number of 64-bit words of data stored in the ICS-500R swing buffer between each buffer swap, or before acquisition ceases, when one-shot mode is used. The register is programmed with a value one less than the buffer size required.

Some operating systems have relatively long interrupt latency times (e.g. Microsoft Windows). For these operating systems, it is desirable to use longer buffer lengths in order to minimize the chance that data may be lost due to delayed interrupt response.

The ICS-500R is available with two memory sizes – 2 MBytes and 8 MBytes. For the 2 MBytes memory size, the maximum number can be programmed to this register is 131071 (128K -1). When the memory size is 8 MBytes, the maximum number can be set for the buffer length is 524287 (512K - 1).

When Corner Turning is enabled, the values programmed to the Buffer Length, Block Size and Selected Channel registers have a fixed relation that must be observed. This is described in the table below, where Buffer Length Register is BLR and Block Size Register is BSR.

	Unpacked FPDP Data	Packed FPDP Data
Sequence Number Disabled	BLR = (SEL_CHAN x BSR) – 1	BLR = (2 x SEL_CHAN x BSR) - 1
Sequence Number Enabled	BLR = (SEL_CHAN x (BSR + 1)) - 1	BLR = (2 x SEL_CHAN x (BSR + 1)) – 1

Please note that the channel selection logic for packed FPDP input data allows only group channel selection. The capability to select individual channels is not currently available. Data is acquired in 32-bit words, and then streamed to the selected channel group. For example, to select channels 1 and 2 (channels numbered 0 – N), you must select a group including channels 0 – 3 inclusive, which will create four output blocks of packed FPDP data. For unpacked FPDP input data, you can select individual channel; therefore, selection of channels 1 and 2 would be possible.

5.12 Buffer Reset Register (BRR)

Write Only

A write to this register performs a buffer reset, causing buffer pointers to be initialized. It is advisable to do this after the board is configured since the existing configuration of the board may be uncertain. It may also be necessary after the board has been disabled. In cases where the board has been disabled following normal operation, the FPDP Suspend signal may be asserted. In order to clear the Suspend, a write to this register should be done.

APPENDICES

APPENDIX A TYPICAL ORDER OF OPERATIONS FOR RECEIVING FPDP DATA STREAMS

The following procedure describes the sequence of activities required to program the ICS-500R. A single Receive Master (FPDP/RM) board is assumed. When using a software driver supplied by ICS, it will not be necessary to configure the QL5064 since this is done by the driver.

1. Configure QL5064 as appropriate.
2. Set Control Register for required mode of operation.
3. Set Start Channel Word register.
4. Set Selected Channel Word register.
5. Set Buffer Length register.
6. Set Block Size register if Corner Turning is enabled..
7. Enable the board.
8. Trigger the board (Internal or External).
9. Wait for buffer read interrupt
10. Read data to buffer length
11. Return to step 9 if operating in Continuous mode and if not finished
12. Return to step 8 if in One-shot mode and need more data.
13. Disable board.

APPENDIX B EXTERNAL TRIGGER CONNECTOR

When using an external trigger signal, it may be supplied either on the PIO1 FPDP signal wire (FPDP pin 19) or at pin 1 (ground on pin 3) of the J1 jumper block on the board. Refer to Section 3.4 and Section 4 above for more details. Refer to Appendix C below for FPDP connector details.

When using external trigger input, the J1 jumper block must be configured as described in Section 4.

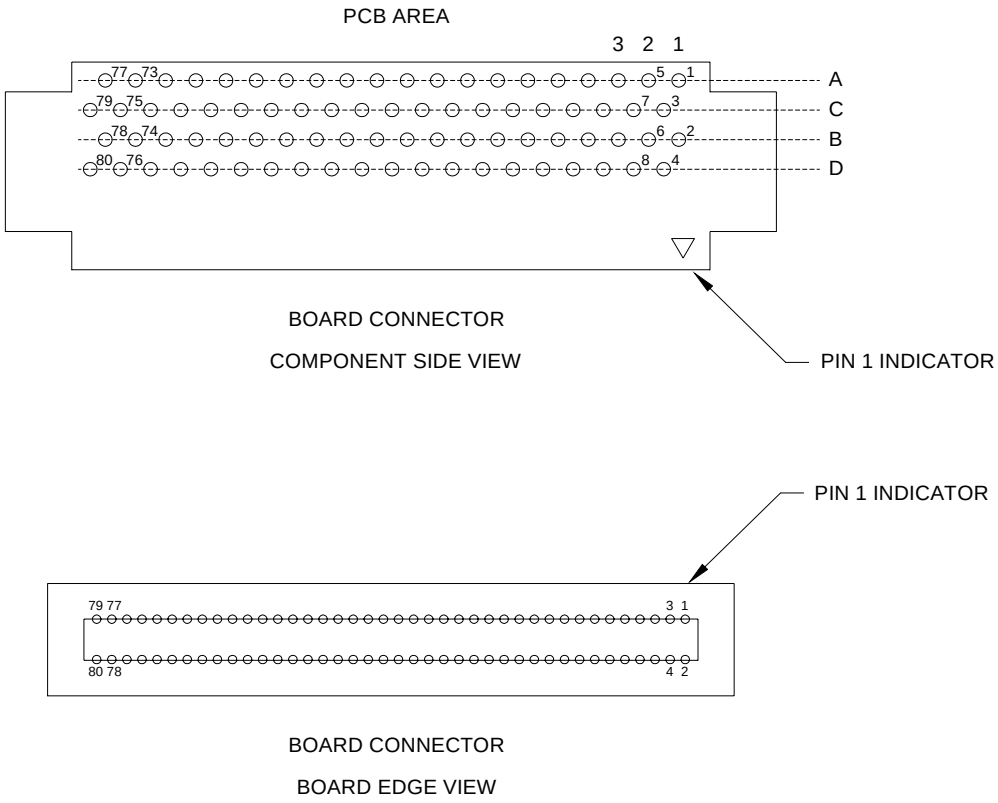
APPENDIX C FPDP CONNECTOR DETAILS

Connector on board:	8831E-080-170L (KEL Corporation)
	P50E-080P1-SR1-TG (Robinson-Nugent)
Mating connector:	8825E-080-175 KEL (with strain relief)
	8825R-080-175 KEL (without strain relief)
	P25E-080S-TG Robinson-Nugent
Manufacturers:	KEL Corporation, (408)720-9044
	Robinson-Nugent, (812)945-0211

The pinout list given in Table C.1 below corresponds to the wire numbers on the ribbon cable, starting from the connector pin 1 index mark (see Figure 7). Previous ICS documentation referred to the board connector row and column numbers. These are also shown in the figure to allow for cross-referencing.

When connecting an ICS-500 board to another FPDP interface, the user should be aware that some FPDP implementations, particularly daughter card implementations (but not the ICS-500), have the FPDP connector inverted. These implementations also have inverted signal/pin allocations and are referred to in the ANSI/VITA 17 FPDP specification as “inverted” FPDP implementations. The standard signal/pin allocation is known as the “non-inverted” case. For the ICS-500 and all non-inverted implementations, the connector index mark appears at the bottom right hand side of the connector when the board is mounted vertically in a chassis. The inversion of both connector and signal/pin allocations avoids the need to fold the FPDP ribbon cable when connecting between inverted and non-inverted connectors; however, the cable used must be made with the connector at one end inverted. The pinouts for inverted connectors are therefore reversed, compared to the ones given in the table. In other words, for inverted connectors, pin 1 connects to pin 80 at the ICS-500, while pin 2 connects to pin 79 at the ICS-500.

The FPDP interface connector is an 80-pin high-density connector available from KEL and Robinson-Nugent. The connector on the board is a KEL 8831E-080-170L or R-N P50E-080P1-SR1-TG and is shown in Figure 7. The mating cable connector is a KEL 8825E-080-175S or R-N P25E-080S-TG, and takes a single ribbon cable of 80 conductors on 0.025 inch pitch. Table C.1 defines the connector pin assignments. Table C.2 defines the signal names and functions.



Connector row and column designations are shown outside connector outline.

Cable numbers are shown inside connector outline.

FPDP BOARD-MOUNTED CONNECTOR
KEL 8831E-080-170L
or
Robinson-Nugent P50E-080P1-SR1-TG

K10128B2

Figure 9 – FPDP Connector

Table C.1 - FPDP Connector Pin Assignments (Non-Inverted)

PIN	SIGNAL	PIN	SIGNAL	PIN	SIGNAL	PIN	SIGNAL
1	GND	2	STROBE	3	GND	4	GND
5	GND	6	GND	7	NRDY*	8	GND
9	DIR*	10	GND	11	RESERVED	12	GND
13	SUSPEND*	14	GND	15	II	16	GND
17	PIO2	18	GND	19	PIO1	20	GND
21	RESERVED	22	GND	23	RESERVED	24	GND
25	PSTROBE	26	GND	27	PSTROBE*	28	GND
29	SYNC*	30	GND	31	DVALID*	32	GND
33	D31	34	D30	35	GND	36	D29
37	D28	38	GND	39	D27	40	D26
41	GND	42	D25	43	D24	44	GND
45	D23	46	D22	47	GND	48	D21
49	D20	50	GND	51	D19	52	D18
53	GND	54	D17	55	D16	56	GND
57	D15	58	D14	59	GND	60	D13
61	D12	62	GND	63	D11	64	D10
65	GND	66	D09	67	D08	68	GND
69	D07	70	D06	71	GND	72	D05
73	D04	74	GND	75	D03	76	D02
77	GND	78	D01	79	D00	80	GND

C.1 FPDP Signals

A description of FPDP signals is given in Table C.2.

Further details concerning the FPDP design are given in ANSI/VITA 17, Front Panel Data Port Specification, available from the VMEbus Industry Trade Association (www.vita.com) or in ICS INPUT Technical Note No. 15, which is available from the factory or from the ICS web site at:

www.ics-ltd.com/tech.html

C.2 FPDP Data Synchronization

There are two data synchronization mechanisms available on the ICS-500-R board for versions R2 and R8. The first allows for synchronization of data after disconnecting the FPDP cable. The second allows for synchronization of data after an overflow occurs.

C.2.1 FPDP Cable Disconnect and FPDP Data Synchronization

FPDP signal DIR* (see Table C.1 and Table C.2) is used to detect if the cable is disconnected. When the FPDP cable is disconnected, DIR* will be pulled up on the ICS-500R. After the cable is re-connected, ICS-500R will automatically synchronize with the FPDP data.

Note: the DIR* signal must be asserted by the FPDP transmitter about one second before the ICS-500R board receives FPDP data. In most applications, after the FPDP cable is connected, the FPDP transmitter DIR* signal is always driven low. But if the FPDP transmitters uses Software to control the driver of the DIR* signal, the software should be setup to drive DIR* to low one second before the ICS-500R begins to receive FPDP data.

C.2.2 FIFO Overflow and FPDP Data Synchronization

When the ICS-500R Control Register FPDP Suspend bit is disabled (CR<8>=1), and a SUSPEND* is generated by the FPDP receiver, followed by a FIFO overflow condition, the ICS-500R will automatically re-synchronize with the FPDP data stream.

Note: to use this function, the CR<8> FPDP Suspend bit must be disabled.

Table C.2 - FPDP Signal Descriptions

Signal/s	Signal Name	Description
D31:00	Data Bus	32 bit data bus driven by the data source.
DIR*	Data Direction	The data source asserts DIR* low.
DVALID*	Data Valid	When asserted, DVALID* indicates that the data bus has valid data. This signal is generated by the data source with each data sample.
STROB	Data Strobe	STROB is a free running clock supplied by the data source. The receiving end should clock the data with the rising edge of STROB.
NRDY*	Not Ready	NRDY* is asserted by the receiver when it is not ready to receive data. The data source must sample this signal until the receiver brings it high, at which time the transfer can commence. Since NRDY* is asynchronous to STROB, the data source should double-synchronize to it before sampling its state; this avoids metastability problems.
PIO1,PIO2	Prog. I/O	The PIO signals are programmable I/O lines for user-defined functions. They can be configured as inputs or outputs.
PSTROBE	+ PECL Data Strobe	This signal along with PSTROBE* are generated by the data source as an optional differential PECL (Positive Emitter-Coupled Logic) data strobe. PSTROBE is the positive version of the differential clock and has the same polarity as STROB. For high data rate applications, the differential PECL data strobe should be used instead of STROB. The user must configure the board appropriately; see Section 5.
PSTROBE*	- PECL Data Strobe	This signal is the negative version of the differential PECL data strobe.
RESERVED		Do not connect to reserved signals.
SUSPEND*	Suspend Data	SUSPEND* is generated by the receiver to inform the data source of a pending FIFO overflow condition. The data source is allowed as many as 16 cycles before suspending the transfer. Since SUSPEND* is asynchronous to STROB, the data source should be double-synchronized to it before sampling its state; this avoids metastability problems.
SYNC*	Sync Pulse	The data source can provide a sync pulse to the receiver to synchronize data transfers. The receiver waits for the sync pulse before accepting data. The receiver starts accepting data on the first Data Valid period following the sync pulse.
II	FPDP II mode	The data source asserts II High.