ICS-500T, ICS-500AT

OPERATING MANUAL

Interactive Circuits And Systems Ltd.
June 2002

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

E10702 Rev. A

TABLE OF CONTENTS

1 INTRODUCTION

The ICS-500 is a FPDP II / PCI Bus Buffer in PCI Mezzanine Card (PMC) format. The product is available in both FPDP Receive (ICS-500R) and FPDP Transmit (ICS-500T, ICS-500AT) versions. The product supports the original ANSI/VITA 17 Front Panel Data Port (FPDP) 32-bit parallel cable interface, as well as the new FPDP II interface. The PCI Bus interface is PCI v2.2 (64-bit, 66MHz) 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-500T FPDP II PMC Buffer are:

- Compatible with FPDP II (backward-compatible with ANSI/VITA 17 FPDP)
- PCI v2.2 compliant PMC module with 64-bit, 66MHz PCI interface
- Linked-list DMA controllers on PCI bus
- 64-bit DMA on PCI interface, PCI burst at up to 528 MB/s
- FPDP II (FPDP) Transmit Master interface, supports data transfer at sustainable rates of up to 400MB/s
- Programmable FPDP strobe frequency from 10MHz to 50 MHz. Provides both TTL and PECL strobes
- Software selectable to transmit FPDP II or FPDP data
- Continuous, loop and one-shot (with or without reloading of data) operating modes, software selectable
- FPDP frame rate control
- Software selectable internal trigger or external trigger (edge or level sensitive)
- Swing buffer with selectable buffer size up 2Mbytes (standard version), or 8Mbytes (extended memory version)
- Windows, Linux 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-500T for a number of platforms (including Windows NT, Windows 98, Windows 2000, Windows Me, Windows XP Professional, Linux and VxWorks). These drivers greatly simplify control and operation of the ICS-500T, and their use is 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-500T hardware model and of the PCI interface device. Contact ICS for further details.

1.1 References

1. ICS-500 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-500T FPDP transmitter is shown in Fig. 1. The design includes two interfaces, an FPDP II/FPDP transmit-only interface and a 64-bit, 66MHz PCI 2.2 interface. The data received at the PCI interface is transferred to the FPDP interface by way of a 2 MByte swing buffer (optionally 8 MBytes). The swing buffer is implemented using high speed synchronous SRAM.

The maximum data transfer speed for the FPDP II interface is 400 Mbytes/s. The interface is backward compatible with the ANSI/VITA 17-1998 FPDP interface. When operating with an FPDP receiver, the maximum transfer rate is in excess of 160Mbytes/s. FPDP or FPDP II operating mode is software selectable. The interface supports both packed and unpacked data.

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/66MHz Master/Target PCI controller and a powerful DMA engine. The ICS-500T 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-500T PCI interface can provide up to the 528Mbytes/s burst data transfer rate.

The ICS-500AT product uses a 25 MHz crystal oscillator instead of the 14.31818 MHz part used by the ICS-500T. This affects the calculation of FPDP strobe frequency (see section 6.2), but no other aspect of the board.

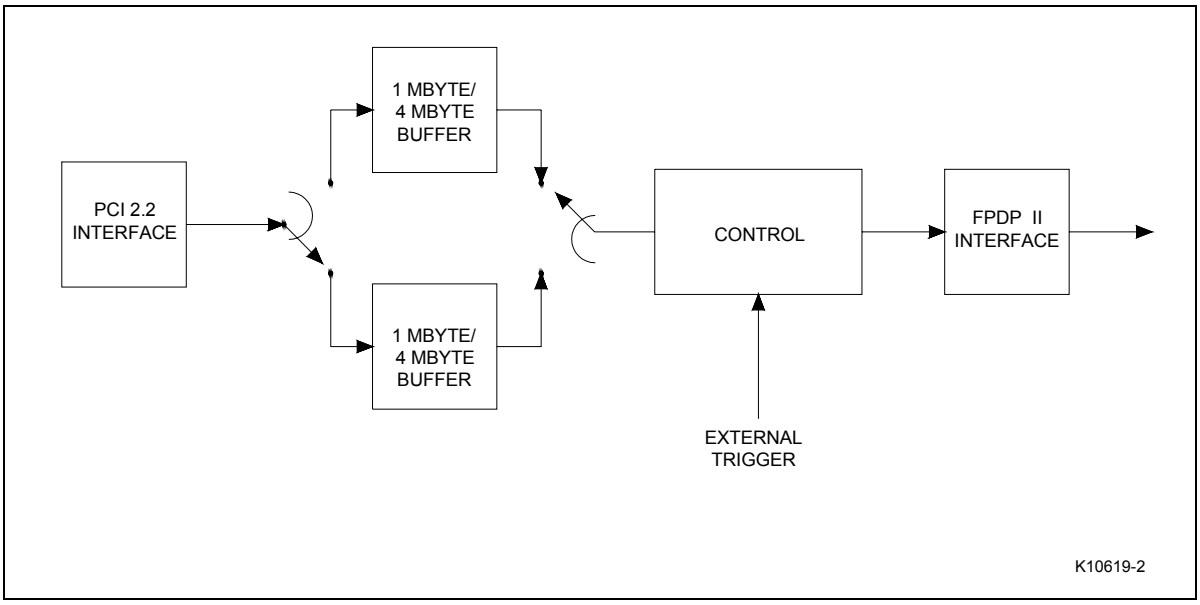


Figure 1 – ICS-500T Block Diagram

2.1 Board Specifications

ICS-500T Specifications	
General	
Buffer Memory	2 x 1Mbyte or 2 x 4Mbyte
Trigger	Internal or External edge triggered or level triggered
FPDP Interface	
Type	FPDP II Maximum strobe frequency 50MHz Minimum strobe frequency 10MHz for FPDP, 20MHz for FPDP II Maximum throughput 400Mbytes/s All ANSI/VITA 17 data frame formats supported except Dynamic Size Repeating Frame Data mode Reverts to ANSI/VITA 17-1998 FPDP operation with non-FPDP II capable interfaces
PCI Bus Interface	
Type	PCI 2.2 compliant 3.3V 64-bit, 66MHz Master/Slave Capable of revertion to 64-bit, 33MHz or 32-bit, 33MHz "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:	
2MB memory version	1.6A at +3.3V 0.6A at +5V Total power consumption < 9W
8MB 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 **F**ront **P**anel **D**ata **P**ort (FPDP, see Ref. 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 160Mbytes/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-500T implements the FPDP II interface. This enhanced version of FPDP provides a data transfer rate more than double that of FPDP, at 400 MB/s.

The P1 connector on the front panel of the ICS-500T connects to the FPDP / FPDP II interface. The ICS-500T implements an FPDP / FPDP II Transmitter Master interface (FPDP/TM).

An attractive feature of FPDP II is that it is backward compatible with FPDP. The user may select FPDP II or FPDP operation. However, if the ICS-500T detects the presensce of an FPDP interface on the cable, it will automatically revert to FPDP operation, even if FPDP II operation has been selected.

A strobe signal is generated by the ICS-500T; this signal is used by the FPDP receive interface (FPDP/R or FPDP/RM) to clock the FPDP data. The ICS-500T provides both TTL- and PECL-level strobe signals (STROB and PSTROBE+/-); the receive interface may therefore be configured to use either of these, depending on the circumstances (cable length, strobe frequency, etc.). The frequency of the FPDP strobe signals is programmable from 10 to 50 MHz. The PECL Strobe should be used when the required strobe frequency is greater than 20MHz, or when a long FPDP cable is used.

FPDP data Frame Size (the number of 32-bit words are transmitted in every FPDP data frame) is programmable from 1 to 2k.

The ICS-500R will not transmit data from the FPDP port until it has been triggered (see section 3.3 below).

3.1.1 FPDP Data Frame Format

The ICS-500T supports all data frame formats defined in ANSI/VITA 17 FPDP Specifications (see Ref. 3 above), except for dynamic size repeating frame data. It therefore supports unframed data, single frame data, and fixed size repeating frame data.

The user should establish which modes of operation of FPDP are supported by the FPDP/R- interface that will receive data from the ICS-500T, and select the appropriate available mode.

Before the ICS-500T is triggered to transmit FPDP data, the Control register must be correctly configured to establish the frame format to be used (see section 5.6.14).

3.2 Trigger

The ICS-500T must be triggered before it will transmit FPDP data. 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 transmit output data when the external trigger is in the low state. When using the level-sensitive trigger, the trigger may be repeatedly enabled and disabled without reconfiguring the board.

For One-shot mode with or without reload, only the edge-sensitive trigger may be used.

When the level-sensitive trigger is used, data transmission will stop as soon as the trigger is negated, regardless of whether this occurs in the middle of a frame. When the trigger is asserted again, the transmission will resume from the point of suspension, and the FPDP Sync signal will be correctly asserted at the end of the frame, when operating in Repeating Frame Data mode.

3.3 Operating Modes

The ICS-500T has four modes of operation – Continuous, Loop, One-shot with reload, and One-shot without reload modes.

In Continuous mode, while triggered, the module will continuously transmit data on the FPDP port until disabled under software control. In order that no gaps occur in transmission, the user must supply data to the module over PCI Bus at a rate sufficient to ensure that the module is not starved of data.

In Loop mode, data of length equal to the programmed buffer length is written to the module once only. After this, while triggered, the module will repetitively transmit the same data set until disabled under software control.

In One-shot mode without reload, data of length equal to the programmed buffer length is written to the module once only. After this, while triggered, the module will transmit the data set once only unless disabled under software control. After the data has been completely transmitted, reassertion of the trigger will cause the same data to be retransmitted, starting from the beginning.

One-shot mode with reload provides similar functionality to one-shot mode without reload, except that, after the data pattern has been completely transmitted, the data (of length equal to the buffer length) must be reloaded before transmission can occur again.

In all modes, the rate at which transmission of data occurs is subject to the FPDP Strobe frequency selected on the ICS-500T and the degree to which the FPDP Suspend signal is asserted by the FPDP receiver module/s.

3.3.1 Continuous Mode

When the ICS-500T is operating in Continuous mode, the ICS-500T continuously transmits data over the FPDP interface, after being triggered by either an internal or external trigger. Data written to the module over PCI Bus is continuously transferred to the FPDP interface. 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. Trigger methods are described in section 3.3.5 below.

Proper usage of Continuous Mode is:
1. Configure ICS-500T Registers.
2. Enable ICS-500T (by setting CR<04>).
3. Wait for a Buffer Write interrupt on PCI Bus. Write a block of data to module over PCI Bus, of length = (8 * Buffer_Length) bytes (i.e. one side of the swing buffer).
4. Trigger ICS-500T by one of the methods described in section 3.3.5. Following the trigger, data will be transmitted on the FPDP II / FPDP Interface (if available in buffer).
5. Return to step 3 unless finished.
6. Disable module by clearing CR<04>.

Note 1: The board can be triggered before data is written to the board. In this case, steps 3 and 4 above would be switched; however, data will not be transmitted until written to the swing buffer (step 3).

Note 2: Up to two buffers of data may be written to the module before triggering. In this case, data would be transmitted immediately upon triggering, and there would be a delay before a buffer interrupt occurs. This is the preferred mode of operation.

To stop Continuous Mode operation:
1. Disable ICS-500T (by clearing CR<04>).

Before starting a new FPDP data output cycle, a Buffer Reset must be applied (refer to 5.13 Buffer Reset Register (BRR)).

3.3.2 Loop Mode

When the ICS-500T is operating in Loop mode, the ICS-500T swing buffer is loaded with data from PCI Bus once only, and the module then repetitively transmits the same data over the FPDP interface, after being triggered by either an internal or external trigger. 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. PCI bus buffer swap interrupts are not used in this mode of operation. Trigger methods are described in section 3.3.5 below.

Proper usage of Loop Mode is:
1. Configure ICS-500T Registers.
2. Enable ICS-500T (by setting CR<04>).
3. Wait for a Buffer Write interrupt on PCI Bus. Write a block of data to module over PCI Bus, of length = (8 * Buffer_Length) bytes (i.e. one side of the swing buffer).
4. Trigger ICS-500T by one of the methods described in section 3.3.5. Following the trigger, data will be transmitted on the FPDP II / FPDP Interface (if available in buffer).
5. Wait for a second Buffer Write interrupt on PCI. Write another block of data to ICS-500T.

Note 1: The board can be triggered before data is written to the board. In this case, steps 3 and 4 above would be switched; however, data will not be transmitted until written to the swing buffer (step 3).

Note 2: Up to two buffers of data may be written to the module before triggering. In this case, data would be transmitted immediately upon triggering, and no interrupt would occur. This is the preferred mode of operation.

To stop Loop Mode operation:
1. Disable ICS-500T (by clearing CR<04>).

Before starting a new FPDP data output cycle, a Buffer Reset must be applied (refer to 5.13 Buffer Reset Register (BRR)).

3.3.3 One-shot Without Reload Mode

When the ICS-500T is operating in One-shot Without Reload mode, one side of the ICS-500T swing buffer is loaded with data from PCI Bus once only. The module then transmits the data over the FPDP interface, once only, after being triggered by either an internal or external trigger. A PCI bus Buffer Write interrupt will occur when the sequence has been completely transmitted. After this, the module may be retriggered at any time, and will transmit the same data sequence. Trigger methods are described in section 3.3.5 below.

Proper usage of One-shot Mode Without Reload is:
1. Configure ICS-500T Registers.
2. Enable ICS-500T (by setting CR<04>).
3. Wait for a Buffer Write interrupt on PCI Bus. Write a block of data to module over PCI Bus, of length = (8 * Buffer_Length) bytes (i.e. one side of the swing buffer).
4. Trigger ICS-500T by one of the methods described in section 3.3.5. Following the trigger, data will be transmitted on the FPDP II / FPDP Interface (if available in buffer).
5. If required, retrigger the module by returning to step 4 above.

Note 1: The board can be triggered before data is written to the board. In this case, steps 3 and 4 above would be switched; however, data will not be transmitted until written to the swing buffer (step 3).

To stop One-Shot Mode operation:
1. Disable ICS-500T (by clearing CR<04>).

3.3.4 One-shot With Reload Mode

One-shot with reload mode is very similar to One-shot without reload mode. The difference is that fresh data must be loaded to the board each time before the module is triggered.

Proper usage of One-shot Mode with Reload is:
1. Configure ICS-500T registers.
2. Enable ICS-500T (by setting CR<04>).
3. Wait for a Buffer Write interrupt on PCI Bus. Write a block of data to module over PCI Bus, of length = (8 * Buffer_Length) bytes (i.e. one side of the swing buffer).
4. Trigger ICS-500T by one of the methods described in section 3.3.5. Following the trigger, data will be transmitted on the FPDP II / FPDP Interface (if available in buffer).
5. If required, reload data and retrigger the module by returning to step 3 above.

Note 1: The board can be triggered before data is written to the board. In this case, steps 3 and 4 above would be switched; however, data will not be transmitted until written to the swing buffer (step 3).

Note 2: Up to two buffers of data may be written to the module before triggering. In this case, data would be transmitted immediately upon triggering. Transmission would end when all the data in the first side of the buffer had been transmitted, and another Buffer Swap interrupt would occur then. Transmission of the data in the second side of the buffer would occur as soon as another trigger occurred. The first side of the buffer would be ready for immediate reloading of data, whether transmission of data in the second side had started or not. This is the preferred mode of operation.

To stop One-shot Mode operation:
1. Disable ICS-500T (by clearing CR<04>).

Before starting a new FPDP data output cycle, a Buffer Reset must be applied (refer to 5.13 Buffer Reset Register (BRR)).

3.3.5 Trigger Behavior

The behavior of the trigger is similar in all modes.

1. Internal trigger
   When using internal trigger, the board will begin to transmit FPDP data only after the internal trigger bit on the Control register (CR<12>) is set to '1' and data is available in the buffer. The board continues transmitting FPDP data until one of the following events occurs:
   A. The board is disabled (by clearing CR<4>);
   B. The internal trigger bit is cleared (CR<12>);
   C. No data is available in the buffer memory or no further triggers occur, when using One-shot mode.
   D. A Buffer Reset is issued (refer to 5.13 Buffer Reset Register (BRR)).
2. Edge-sensitive external trigger
   When using the edge-sensitive external trigger, the board will begin to transmit FPDP data after the rising edge of the external trigger signal occurs and data is available in the buffer. The board continues transmitting FPDP data until one of the following events occurs:
   A. The board is disabled (by clearing CR<4>);
   B. No data is available in the buffer memory or no further triggers occur, when using One-shot mode.
   C. A Buffer Reset is issued (refer to 5.13 Buffer Reset Register (BRR)).
3. Level-sensitive External Trigger
   When using the level-sensitive external trigger, the board transmits FPDP data only when the external trigger signal remains in the high state and data is available in the buffer. When the external trigger signal goes low, the ICS-500T ceases transmission until the signal goes high again. The board continues transmitting FPDP data until one of the following events occurs:
   A. The board is disabled (by clearing CR<4>);
   B. No data is available in the buffer memory or no further triggers occur.
   C. A Buffer Reset is issued (refer to 5.13 Buffer Reset Register (BRR)).

Note: For One-shot Modes (with or without reload), only the edge-sensitive trigger can be used.

3.4 Frame Rate Controls

The ICS-500T has a frame rate control feature. This feature is active only when using repeating frame data format. The purpose of this feature is to enable data to be generated on the FPDP interface with an average data rate that is lower than the FPDP Strobe frequency. This is done by inserting a programmable delay between each FPDP frame transmitted. This feature is useful for simulating the actual behavior of some real-time systems.

The actual frame rate is a function of the FPDP strobe frequency, the value programmed to the Frame Size register, and the value programmed to the Frame Pause register. The frame size is specified as the number of 32-bit words in each frame. The frame pause value is the number of cycles of Strobe during which no data is transmitted between each successive frame.

Suppose the FPDP strobe frequency is f_{strobe} and the required frame rate is f_{frame}. For FPDP II data,

$$f_{frame} = 2 \times f_{strobe}/(\text{Frame\_Size} + \text{Frame\_Pause})$$

For FPDP I data,

$$f_{frame} = f_{strobe}/(Frame\_Size + Frame\_Pause)$$

Note that, when operating in Continuous or One-Shot with Reload modes, the frame rate is also influenced by the rate at which the data are written to the ICS-500T memory from PCI bus. If the rate is not fast enough, the frame rate will be less than the calculated value.

If the Frame Pause register is set to zero, there will be no delays generated between the frames. The FPDP interface will generate data in bursts, with pauses dictated by the availability of data in the buffer memory. A burst may contain any number of frames.

The rate at which data is transmitted may also be affected by the assertion of the FPDP Suspend signal by any FPDP receive interface. This may occur if a downstream processor is unable to sustain the required rate of reading the data.

3.5 PCI Interface

The ICS-500T implements a 64-bit, 66MHz 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 buffers the signals between the PCI core and the module's local bus. This mechanism allows both the PCI bus and the local bus to operate at up to full PCI bandwidth (528 MB/s). When writing data to the buffer memory across PCI bus, ICS-500T uses the QL5064 chain-mode DMA channel.

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

The ICS-500T accepts either 3.3V or 5V signalling and is keyed in accordance with Ref. 4.

3.6 Light-Emitting Diodes

The ICS-500T is fitted with a set of light-emitting diodes (LEDs) which indicate 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 B of the PMC board (outside of the PMC board when it is mounted on the carrier board). There are three LEDs installed in a horizontal line. Table 3.1 describes the ICS-500T LEDs starting the leftmost LED when the board is installed in normal vertical orientation (as shown in Fig. 2 below).

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

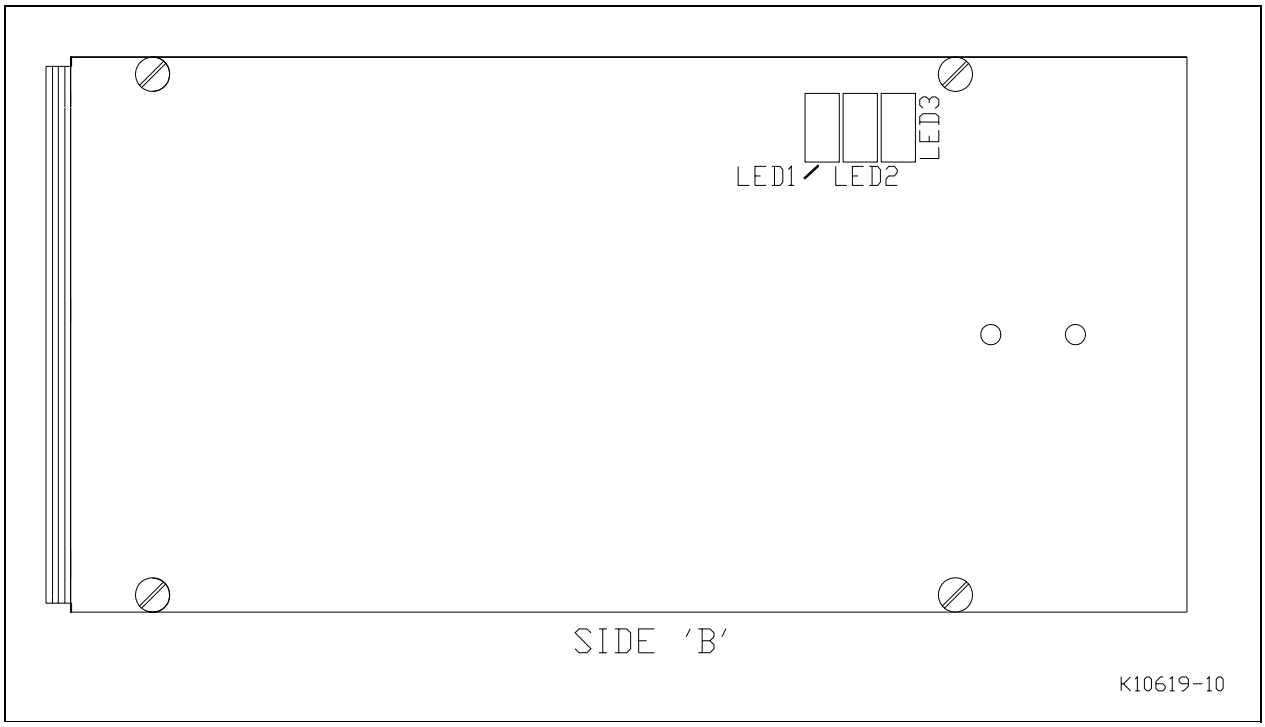


Figure 2– Position of LED indicators on ICS-500T

4 HARDWARE CONFIGURATION

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

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 Fig. 3). 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.

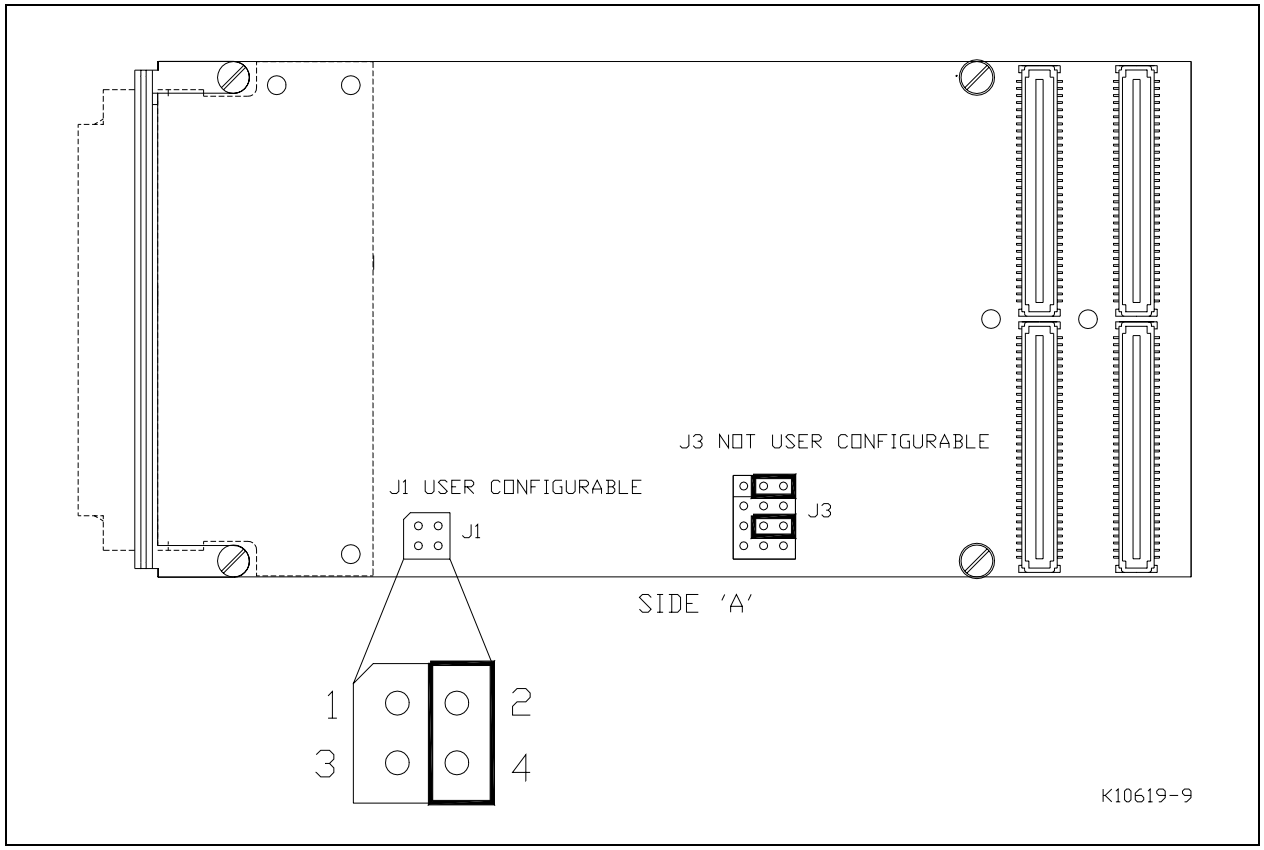


Figure 3 – Position of J1 Jumper Block on ICS-500T

5 PROGRAMMING MODEL

The ICS-500T 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-500T register set is listed in Table 5.2. Data should be written to 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 Ref. 2.

A detailed list of ICS-500T registers is given in Table 5.2. Register bit allocations are given in Fig. 4. Bits marked as RSVD in Fig. 4 are reserved for future use. They may be written as either value.

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-500T Register Set	PCI_BASE1	ICS-500T register access	4K Bytes
ICS-500T Memory Space	PCI_BASE2	ICS-500T buffer memory access	16MBytes

Table 5.2 ICS-500T Address Map

Register	PCI Bus Address	Type
ICS-500T 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
Buffer Length register	PCI_BASE1 + 0x0028	Read/Write
Block Size register	PCI_BASE1 + 0x0030	Read/Write
Frame Size register	PCI_BASE1 + 0x0038	Read/Write
Frame Pause register	PCI_BASE1 + 0x0040	Read/Write
Strobe Frequency register	PCI_BASE1 + 0x0048	Write Only
Buffer Reset register	PCI_BASE1 + 0x0050	Write Only

5.1 General Notes

All programming and control of the ICS-500T 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-500T. All undefined bits may be read as zero or one.

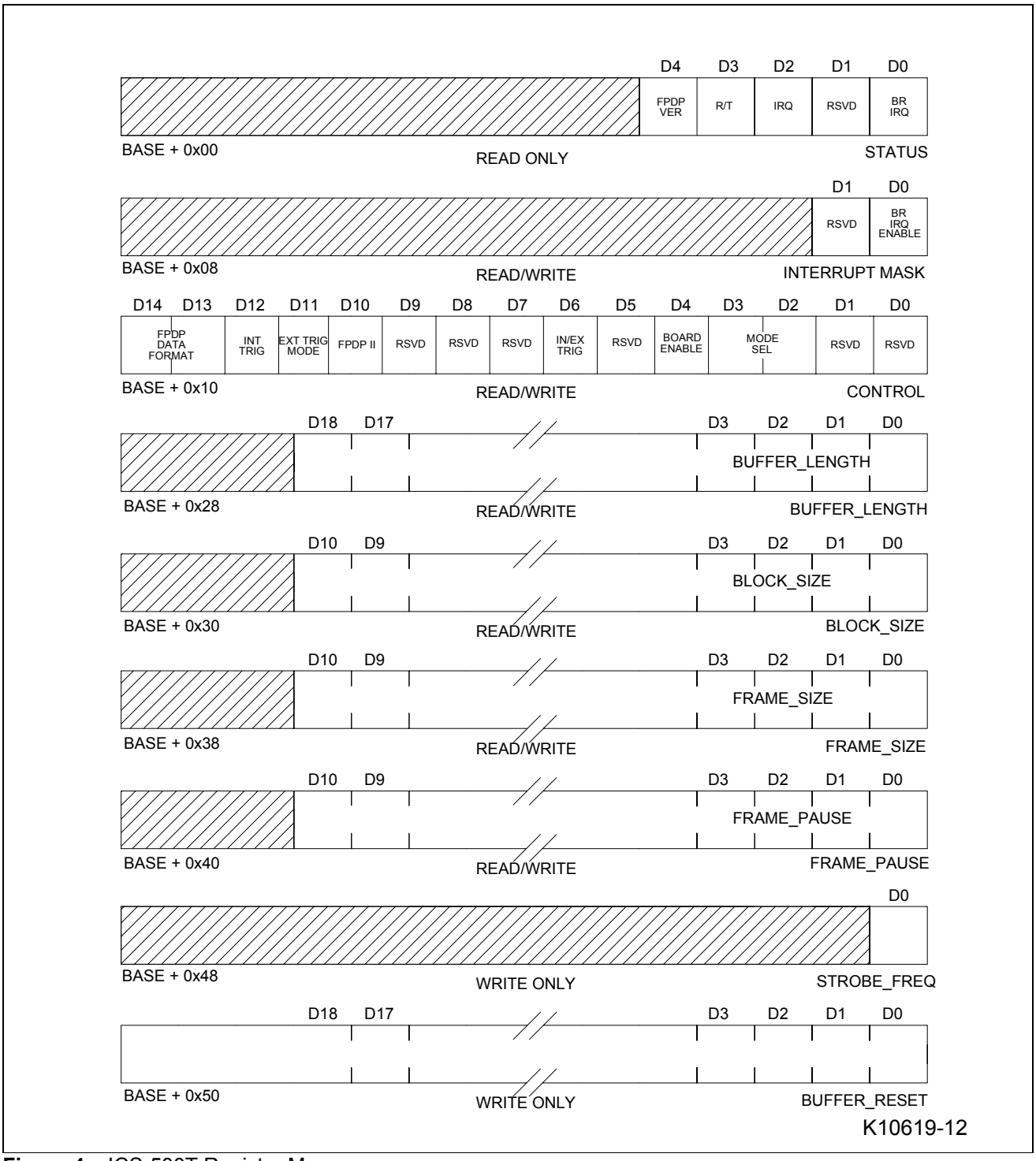


Figure 4 – ICS-500T Register Map

5.2 QL5064 Registers

The ICS-500T 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 Ref. 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 Ref's 2, 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-500T 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-500T 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 5.1. 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-500T implements “Little-endian” ordering of data within larger data widths. When operating the ICS-500T 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 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-500T 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
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 PCI Bus to ICS-500T Local 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 data is being transferred to the ICS-500T across the PCI interface, the DMA controller on the QL5064 may be used. The controller can be used with or without chain mode (scatter-gather) operation. 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 Receive (Read) 0 DMA channel to move data from PCI Bus to the ICS-500T. The ICS-500T does not use the Transmit DMA function, since the direction of data flow is always from PCI Bus to the ICS-500T.

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 source data and the length of the transfer.

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

1. Write the starting PCI source address to QL5064 control register "Master Read Address 0" (offset 0x48).
2. Write the transfer count (in bytes) to QL5064 control register "Master Read Transfer Count 0" (offset 0x50, bits[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_RCV0_CNT_ADDR (offset 0x1020). The transfer count written to this register must be the same as that written to the Master Read Transfer Count 0 register (offset 0x50). 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 18).
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 Fig. 5. 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.

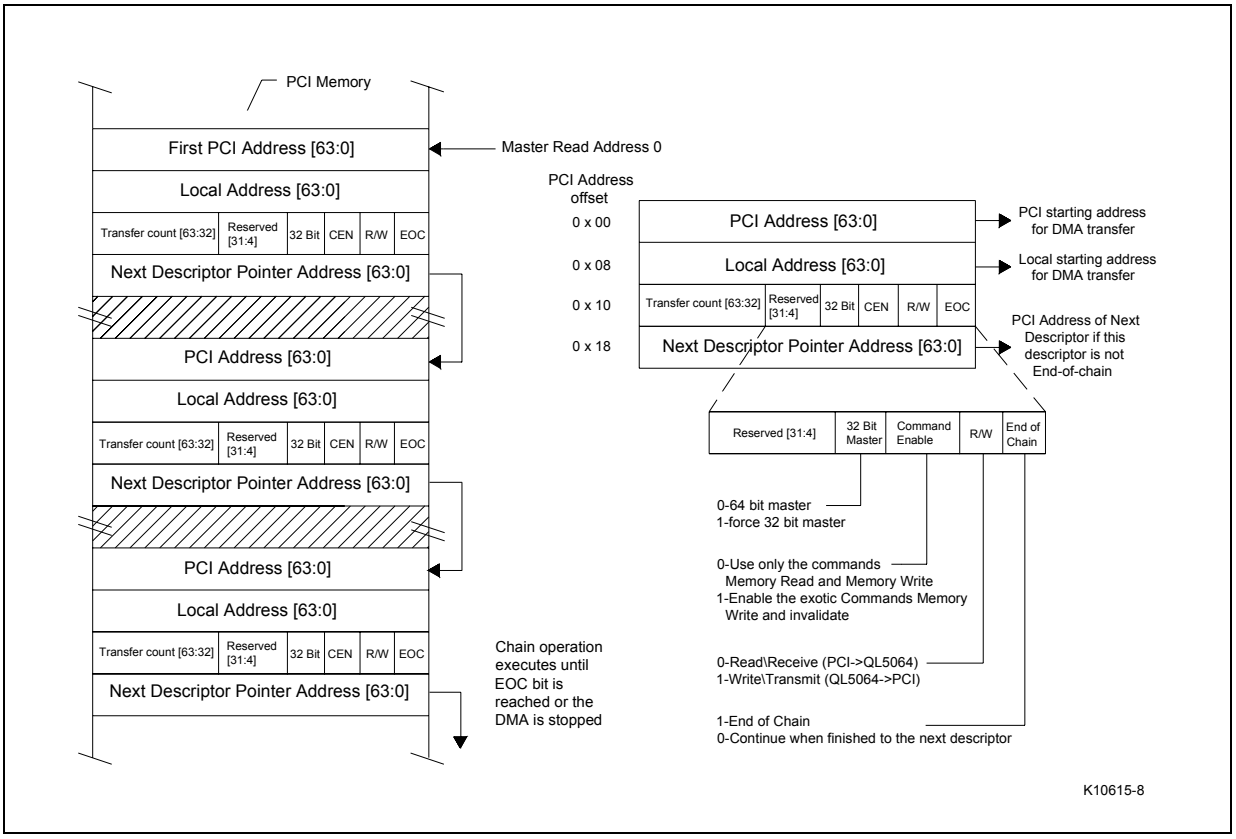


Figure 5 – DMA Chain Mode Descriptors

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 Ref. 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-500T. It is used to determine the ICS-500T local bus interrupt status, and the source of a pending interrupt. It also gives information about whether the FPDP bus is operating in FPDP II or FPDP mode.

5.5.1 SR<0> - Buffer Write IRQ

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

SR<0>	Buffer Write 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> - Reserved

This bit is reserved.

SR<1>	Interrupt Request
READ ONLY	
0	Always 0

5.5.3 SR<2> - IRQ (Interrupt Request)

This bit indicates that the ICS-500T is asserting a PCI bus interrupt. The only source for an interrupt is the Buffer Write IRQ, SR<0>.

SR<2>	Interrupt Request
READ ONLY	
0	PCI Bus interrupt is not asserted
1	PCI Bus interrupt is asserted

5.5.4 SR<3> - FPDP R/T

This bit indicates whether the board is a FPDP transmitter (ICS-500T) or receiver (ICS-500R). The value read will always be 1 for the ICS-500T.

SR<3>	FPDP R/T
READ ONLY	
1	Value read is always 1 (ICS-500T).

5.5.5 SR<4> - FPDP Version

This bit indicates whether the ICS-500T FPDP interface is transmitting 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 Write interrupt request. Masked interrupts are still seen in the corresponding bit of the status register, but do not cause an interrupt to occur.

5.6.1 IMR<0> - Buffer Write IRQ Enable

This bit is used to enable the Buffer Write 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 Write IRQ Enable
READ/WRITE	
0	Buffer Write IRQ is disabled
1	Buffer Write IRQ is enabled

5.7 Control Register (CR)

Read/Write

5.7.1 CR<3:2> - Mode Select

This bits select the operating mode of the module.

CR<3:2>	FPDP Operating Modes
READ/WRITE	
00	Continuous Mode
01	Loop Mode
10	One-Shot Mode without reload
11	One-Shot Mode with reload

5.7.2 CR<4> - Board Enable

This bit is used to enable the board. This should be done after all other registers have been configured. After enabling the module, an external or internal trigger must be applied to start operation, whichever has been selected in CR<6>.

CR<4>	Board Enable
READ/WRITE	
0	Disable module
1	Enable module

5.7.3 CR<6> - Internal/External Trigger

This bit selects internal or external triggering of the board. 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 must be triggered using a TTL signal applied to the EXT_TRIG pin of J1 or Pin C5 (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>.

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

5.7.4 CR<10> - FPDP II Enable

This bit is used to select the FPDP protocol version to be used, i.e. FPDP or FPDP II data protocol. When FPDP II is enabled, the link will revert to FPDP data protocol if the receiving interface/s do not support FPDP II operation.

CR<10>	FPDP II Enable
READ/WRITE	
0	FPDP II is disabled
1	FPDP II is enabled

5.7.5 CR<11> - External Trigger Mode

This bit is used to select the external trigger as either edge or level sensitive. When using One-shot mode with or without reload, only the edge-sensitive external trigger may be used. 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 One-shot with or without reload modes, only the edge-sensitive trigger may be used.

CR<11>	External Trigger Mode
READ/WRITE	
0	External sensitive edge trigger
1	Level sensitive external trigger

5.7.6 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	The board is not triggered
1	The board is triggered

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

These bits select the FPDP data frame format of the data to be transmitted on the FPDP bus.

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

5.8 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-500T swing buffer between each buffer swap, or before transmission 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-500T is available with two memory sizes – 2Mbytes and 8 Mbytes. For the 2Mbytes memory size, the maximum number that can be programmed to this register is 131071 (128K -1). When the memory size is 8Mbytes, the maximum number can be set for the buffer length is 524287 (512K - 1). The value written to Buffer Length register must obey the following relation with the value written to Frame Size register:

Buffer_length = (k * Frame_size)/2 - 1.

Where k is 1, 2, 3, 4, …..

5.9 Block Size Register (Block_Size)

This register is reserved for future use.

5.10 Frame Size Register (Frame_Size)

Read/Write

This register is used to set the FPDP Frame Size. This is the number of 32-bit words in each FPDP frame. This value is required for Repeating Frame Data mode only. The value must be programmed as the required number of words per frame, less one, The maximum number can be programmed in this register is 2047, representing a frame length of 2048 32-bit words.

5.11 Frame Pause Register (Frame_Pause)

This register is used to program the number of FPFP Strobe periods to delay between transmission of each FPDP frame. It is effective only when the FPDP Data Frame Format is Fixed Length Repeating Frame Data. This feature allows the board to be used in systems that required a controlled frame rate. A value of zero will result in no delay between frames, and the data will be transmitted at the maximum rate possible, given the programmed Strobe frequency, whether the receiving interface asserts the FPDP Suspend signal, and the data transfer rate capability of the bus interface and the host system. In data frame formats other than Repeating Frame Data, transmission always occurs at the Strobe Frequency, subject to suspension by the receiving interface, and system limitations, regardless of any value programmed to this register.

5.12 FPDP Strobe Frequency Register

Write Only

This register is used to program the FPDP Strobe (clock) frequency. This is accomplished by writing a 22-bit programming word to the on-board programmable oscillator. The programming word must be computed using an algorithm described in section 6.3. Data is written serially to the oscillator, least significant bit first. One write must occur for each bit of data to be written to the oscillator, with the data to be programmed in bit 0.

Note that the minimum Strobe frequency that may be programmed to the ICS-500T is 10MHz.

5.13 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.

6 APPENDICES

6.1 Typical Order of Operations for Transmitting FPDP Data Streams

The following procedure describes the sequence of activities required to program the ICS-500T. When using a software driver supplied by ICS some activities are simplified by the use of driver functions; it is not be necessary to configure the QL5064 since this is done by the driver. Also programming of the FPDP Strobe frequency oscillator is greatly simplified.

The order of programming shown below may be varied somewhat. For example, it is permissible to wait for a Buffer Write interrupt before writing data to the board for all modes (see section 3.3 above).

6.1.1 One-shot Mode (with or without Reload)

1. Reset board; clears control resisters.
2. Configure QL5064 as appropriate.
3. Program the FPDP Strobe frequency oscillator (Refer to 6.2).
4. Set Control Register.
5. Set Frame Size Register.
6. Set Buffer Length Register.
7. Enable the board.
8. Write to the Buffer Reset register.
9. Write data to the ICS-500T to the programmed buffer length, once or twice, so as to fill one or both sides of buffer.
10. Trigger the board (internal or external, as previously configured). The module will now transmit an FPDP II / FPDP data stream from one side of the buffer only.
11. If in One-shot with Reload mode, return to step 8 if it is required to re-trigger the module (second side of buffer will now be transmitted).
12. If in One-shot with no Reload mode, return to step 10 if it is required to re-trigger the module (second side of buffer will now be transmitted).
13. Else disable the module when finished.

6.1.2 Loop Mode

1. Reset board; clears control resisters.
2. Configure QL5064 as appropriate.
3. Programming the FPDP Strobe frequency oscillator (Refer to 6.2).
4. Set Control Register.
5. Set Frame Size Register.
6. Set Buffer Length Register.
7. Enable the board.
8. Write to the Buffer Reset register.
9. Write data to the ICS-500T to the programmed buffer length, twice, so as to fill both sides of buffer.
10. Trigger the board (internal or external, as previously configured). The module will now transmit an FPDP II / FPDP data stream.
11. Disable the module when finished.

6.1.3 Continuous Mode

1. Reset board; clears control resisters.
2. Configure QL5064 as appropriate.
3. Programming the FPDP Strobe frequency oscillator (Refer to 6.3).
4. Set Control Register.
5. Set Frame Size Register.
6. Set Buffer Length Register.
7. Enable the board.
8. Write to the Buffer Reset register.
9. Write data to the ICS-500T to the programmed buffer length, twice, so as to fill both sides of buffer.
10. Trigger the board (internal or external, as previously configured). The module will now transmit an FPDP II / FPDP data stream.
11. Wait for a Buffer Write interrupt or timeout.
12. When Buffer Write interrupt occurs, write data to the ICS-500T to the programmed buffer length, once, so as to fill one side of buffer.
13. Return to step 22 unless finished.
14. Else disable the module.

6.2 Programming the FPDP Strobe Frequency Oscillator

6.2.1 Introduction

The frequency of the ICS-500T FPDP Strobe frequency oscillator is programmed through the FPDP Strobe Frequency register of the PCI bus memory map (see Fig. 4). The device is programmed not in engineering units, but by means of a complex programming word whose construction is described below; a series of control words must also be written to the device. Data is transferred to the device serially; it is necessary to write the data to the ICS-500T FPDP Strobe Frequency register one bit at a time; bit 0 of the register is the relevant bit. Thus the programming sequence should normally be done by repeatedly writing each control or programming word to the FPDP Strobe Frequency register, shifting the data right by one bit after each write, until all bits of the word have been written.

Automatic calculation of the programming word is provided by the ics500calcFoxWord() function and other functions in the 'C' language function library supplied with the optional ICS-500T software drivers, available for Windows environments. These routines generate a 22-bit formatted programming word for the oscillator equivalent to the nearest possible frequency to that supplied as input by the user (in MHz), and also supply the actual frequency represented by the programming word, as an output.

6.2.2 Programming Summary

The oscillator device contains two registers, the Control Register and the Programming Word Register. The programming sequence is as follows:

1. Write to Control Register to configure device and prepare the device to receive the Programming Word.

2. Write Programming Word.

3. Write to Control Register to load Programming Word data into device.

4. Wait at least 10ms for device Voltage Controlled Oscillator (VCO) frequency to stabilize.

5. Write to Control Register to enable device output of new frequency.

6.2.3 Control Register

When writing data to the Control Register, it is necessary to include a Protocol Field to identify the data as Control Register data.

Protocol Field (6 bits) = 0 1 1 1 1 0

It is important that the sequence of four ones contained in the protocol field never be sent except as part of the Protocol Field. Thus, when writing programming word data to the device, it is necessary to use a technique of inserting an extra zero after a run of three ones; this technique is called "bit stuffing", and is described in more detail in section 7.3.4.

The control register contains eight bits, which are defined as shown in Table 6.1.

Table 6.1

Bit No.	Description	Function	Power-up Default
0	Enable Programming Word register to be written by next data	0 = Program Register Disabled 1 = Program Register Enabled to Receive Data	0
1	Internal Output Disable	0 = Output is VCO or f_{REF} 1 = Output is tri-stated	0
2	Internal Multiplexer	0 = Output is VCO frequency 1 = Output is f_{REF}	1
3	Device pin 7 usage	Set to zero only	0
4 - 7	Reserved	Set to 0	0000

Control register data is written to the device starting with bit 0 of the control word, continuing to bit 7 of the control word, and followed by the 6 bits of the Protocol field, for a total of 14 bits.

Note that the default configuration of the device following power-up results in the output being at the frequency of the reference oscillator which is 25.00000 MHz for ICS-500AT (for ICS-500T, the frequency is 14.31818 MHz).

Note: The examples on this document is using 25.00000 MHz oscillator (for ICS-500AT board). For ICS-500T board, the oscillator frequency is 14.31818 MHz.

6.2.4 Programming Register

The Programming register is written with a 22-bit word describing the required frequency of the output. However, it is essential that programming words do not mimic the Control word Protocol field (see section 6.2.3 above) by containing runs of four consecutive ones. Thus, the device specification requires that a zero must be inserted in the word after each occasion when three one's have been transmitted to the device, regardless of whether the next bit is a 0 or a 1. This procedure is known as "bit stuffing". For this reason, the actual length of a programming word may vary between 22 and 29 bits.

For example, to send this programming data:

Last Bit				**First Bit**
1111	**0101**	**0111**	**1110**	**111111**

Transmit this serial bit stream:

Last Bit				**First Bit**
10111	**0101**	**00111**	**01110**	**01110111**

The fields of the programming word are described in Table 6.2.

Table 6.2

Field	Bits	# of Bits	Notes
P Counter value (P)	<21:15>	7	MSB (Most Significant Bits)
Duty Cycle Adjust Up (D)	<14>	1	Set to logic 0
Mux (M)	<13:11>	3	
Q Counter Value (Q)	<10:4>	7	
Index (I)	<3:0>	4	LSB (Least Significant Bits)

The frequency of the programmable oscillator f_{VCO}, and the output frequency f_{OUT} are determined by these fields as follows:

$$f_{VCO} = 2 * f_{REF} * (P+3)/(Q+2)$$

where, f_{REF} = Reference frequency (i.e. 25.00000 MHz)

$$f_{OUT} = f_{VCO} / 2^{M}$$

The values of the P and Q parameters must be selected so that f_{VCO} remains between 50 MHz and 150 MHz, inclusive. The value programmed to the M field programs a division register to allow sub-multiples of the VCO frequency to be obtained at the output; the maximum divisor possible is 128.

The **Index field (I)** is used to preset the VCO to an appropriate range. The value for this field should be chosen from the following table. (Note that this table is referenced to f_{VCO} rather than to the desired output frequency.)

I	f_{vco} (MHz)
0000	50 – 80
1000	80 - 150

If the desired VCO frequency is exactly 80 MHz, then either index value may be used (since both limits are tested). However, the manufacturer recommends using the setting corresponding to the higher frequency range.

6.2.5 VCO Programming Constraints

There are four primary programming constraints the user must be aware of:

1: 50MHz <= fvco <= 150MHz
2: 1 <= P <= 127
3: 1 <= Q <= 127
4: 200kHz <= f_{REF}/(Q+2) <= 1MHz

The constraints have to do with the trade-offs between optimum speed with lowest noise, VCO stability and factors affecting the loop equation.

6.2.6 Program Register Example

The following is an example of how to calculate a clock programming word:

Derive the proper programming word for 12.8 MHz clock frequency.

Since 12.8 MHz < 50 MHz, quadruple it to 51.2 MHz
Set M to 010_2
Set I to 0000_2

The result:

Fout = 12.8 = (2*14.31818*(P+3)/(Q+2))/2^M

where M = 0, 1, 2, 3, 4, 5, 6, 7
since M = 2: (P+3)/(Q+2) = 1.024000

The two choices of P and Q giving the nearest to the required frequency are:

P	Q	f_{vco}	Error (PPM)
125	123	51.20000	0
82	81	51.20482	94
124	122	51.20968	189

Taking the first set of values, i.e. (P,Q) = (125,123):

P = 125 decimal = 1111101 binary = 11(0)11101

Q = 123 decimal = 1111011 binary = 1(0)111011

NOTE: The presence of three ones in a row in both P and Q values causes zero bit-stuff values to be inserted in each. However, it is necessary to examine the values in the previous (I) field before inserting a zero in the least significant bits of the Q value. If the I field had contained a 1 in the most significant position the inserted zero would have been in a different position.

The full programming word, W is therefore:

W = P, D, M, Q, I = 11011101, 0, 010, 10111011, 0000 = 110111010010101110110000 (DD2BB0 Hex)

6.2.7 Oscillator Programming Example

The oscillator requires three control words plus the programming word in order to program a new output frequency (see programming sequence given in section 6.2.2 above). The following paragraphs provide an overview of how the control words are build along side the programming word.

All data is written to the oscillator serially through the ICS-500T FPDP Strobe Frequency register bit 0. The data is written least significant bit first. An example of programming the oscillator to an initial or new frequency is as follows:

1. Load Control register to enable loading of the Programming word register, enable the output and select the reference frequency for output from the Internal Multiplexer. The Protocol word follows the control word. All data is shifted in LSB (Least Significant Bit) first.

	Last Bit **First Bit**
Control word	0 1 1 1 1 0 0 0 0 0 0 0 1 0 1
	Protocol Word Control Reg. Data

2. Shift in the desired output frequency value via a programming word that is 22 bits in length, plus any required bit stuffs. (Up to 29 bits can be obtained with bit-stuffing). The example programming word for 12.8 MHz, calculated in section 6.2.6 above, is shown below.

	Last Bit **First Bit**
Programming word for 12.8 MHz (0x DD2BB0)	1 1 0 1 1 1 0 1 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 0

3. Load the Control register to disable further loading of the Programming word register. This causes the required output frequency value to be programmed while keeping the output set to the reference frequency for the time being.

	Last Bit **First Bit**
Control word	0 1 1 1 1 0 0 0 0 0 0 1 0 0
	Protocol Word Control Reg. Data

4. Wait at least 10ms for the VCO to settle to the new frequency. The value will be accurate to within 0.1% within this time.

5. Load the Control register to disable the Internal Multiplexer. This will cause the output to immediately swing to the new frequency.

	Last Bit **First Bit**
Control word	0 1 1 1 1 0 0 0 0 0 0 0 0 0 0
	Protocol Word Control Reg. Data

6.3 External Trigger Connectors

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 sections 3.2 and 4 above for more details. Refer to section 6.4 below for FPDP connector details.

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

6.4 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

Note : The pinout list given in Table 6.3 below corresponds to the wire numbers on the ribbon cable, starting from the connector pin 1 index mark (see Fig. 6). Previous ICS documentation referred to the board connector row and column numbers. These are also shown in the figure to allow cross-reference.

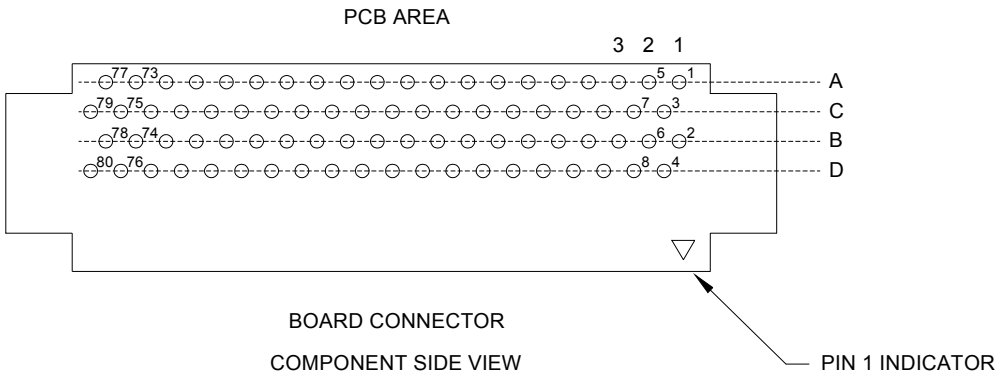
6.4.1 FPDP Connector Pin Assignments

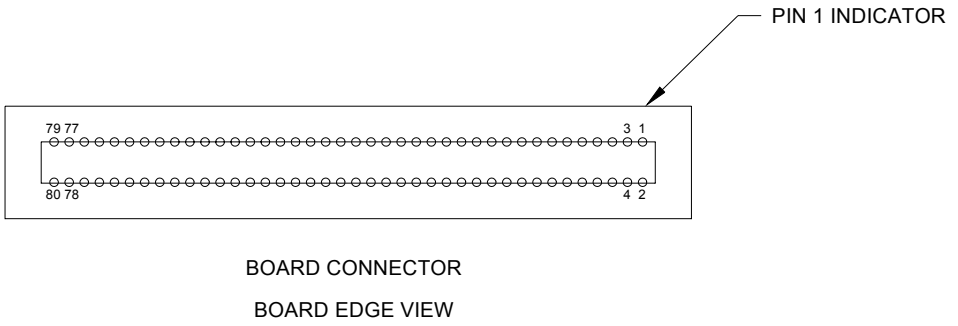
The FPDP interface connector is an 80-pin high-density connector available from KEL and Robinson-Nugent. The connector on the board is a KEL 8831E-080-170L or R-N P50E-080P1-SR1-TG and is shown in Fig. 6. The mating cable connector is a KEL 8825E-080-175S or R-N P25E-080S-TG, and takes a single ribbon cable of 80 conductors on 0.025 inch pitch. Table 6.3 defines the connector pin assignments.

6.4.2 FPDP Signals

A description of FPDP signals is given in Table 6.4.

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).





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 6 - FPDP Interface Connector

Table 6.3 FPDP Connector Pin Assignments

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

Table 6.4 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 avoid 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.

