Tahoma Technology
Model 10115/10117

Hardcopy Interface
for PCI Bus

5 September, 2008

CONTENTS

Model 10115/10117 Hardcopy Interface for PCI Bus

Tahoma Technology

2819 Elliott Avenue, Suite 206
Seattle, WA, USA, 98121

phone: 206.728.6465
fax: 206.728.1633
http://www.tahomatech.com
tahoma@tahomatech.com

Introduction

This document describes Tahoma Technology's Models 10115 and 10117 PCI Hardcopy interface boards. Included are basic hardware and software installation instructions, as well as detailed technical information appropriate to the device driver programmer.

The 10115 and 10117 are high-performance PCI boards capable of driving printers and plotters compatible with the Versatec (VPI) and Centronics (IEEE1284) interface specifications. The 10115 is a full-size PCI board supporting VPI TTL (short line), VPI Differential (long line) and Centronics compatible plotters. The 10117 is a ½ size PCI board that supports VPI Differential and Centronics compatible plotters.

Both boards are PCI bus-master capable with 16K bytes of on-board FIFO buffer memory. Scatter/gather DMA and burst transfers can supply data to the attached plotter at up to 2Mbyes/second (4Mbytes/second in Tahoma Burst Mode).

Multiple operating modes, programmable handshake speeds, and adjustable cable termination provide flexibility to support a wide range of printer/plotters, from small format inkjet and laser printers, to the largest thermal and electrostatic plotters.

About Tahoma Technology

Tahoma - The Chinook jargon name for Mt. Rainier

Tahoma Technology is a new company with a 40+ year history.

The original Ikon Corporation was founded in 1966. In its early years, Ikon Corporation designed and built minicomputer based systems and specialized peripheral equipment. With the advent of personal work stations, Ikon Corporation found a market for high performance interface products - initially for the Multibus. The first product was a Multibus based Versatec (VPI) interface for electrostatic plotters, followed by a DR11-W emulator which allowed supporting VAX related external hardware and interprocessor links on Apollo work stations. VPI was adopted by many printer and plotter manufacturers as the interface of choice. The DR11 became a standard for high-speed 16 bit interfacing. As new bus architectures came on line, these interfaces were ported to new host busses, including, among others, VersaBus, VMEbus, ISA, EISA, MicroChannel, Sbus, and PCI. Along with the bus migration, Centronics compatibility was added to the plotter interfaces, and speeds continued to improve. Today, the PCI plotter interface bursts data from the bus at 40MB/sec, and feeds the plotter at 4-5MB/sec. The PCI DR11-W can sustain 20MB/sec outputs.

In 1997, Ikon Corporation was acquired by IKON Office Solutions (IOS) and became the Ikon Corporation division of IOS.

In August, 2001, the people from the original Ikon Corporation, who had continued to run Ikon Corporation as an IOS division, formed Tahoma Technology and purchased certain of the assets of Ikon Corporation. It has been an enjoyable four years with the IOS, but we are happy and excited to be back on our own.

Tahoma Technology has acquired certain of the assets and liabilities of Ikon Corporation, including warranty support. While the asset purchase was definitely NOT a name change (Tahoma Technology is a new entity; the Ikon Corporation name, marks, and structure remain with IOS), the impact on our customer base should be nil, beyond changing the name in the address book. We are the same people, with the same high performance products, and the same attitude about customer support. We look forward to continuing our 35 year relationship with our customers under the new banner.

In 2002, Tahoma Technology added the 10119, 10120, and 10121 family of IEEE1284 and Ethernet to VPI converters to the product line.

Please note that references to Ikon Corporation and IKON found in this document and in text and software codes found on the Tahoma Technology website are left in place for compatibility and historical reasons. As the code and documentation have previously been placed in the public domain, this does not constitute a use of the IKON Office Solutions (IOS) marks, which are the property of IOS. These documents and programs are maintained by Tahoma Technology.

The following registered trademarks are used in this manual for descriptive purposes to identify compatibility:

PLX Technology:	PLX Technology
PCI Bus:	PCI Special Interest Group
IOS:	IKON Office Solutions

Installation

Installation involves hardware - the board and cable, and software - the device driver and application software, if any. PCI boards, including those from Tahoma Technology, require very little hardware configuration. Parameters such as board address and interrupt level are set by configuration software, not by on-board jumpers or switches.

Note: the board's interface selection ribbon cable MUST be configured to match the attached printer/plotter BEFORE installing the board. See Hardware Installation.

Software installation varies with operating system and software provider. Please note that Tahoma Technology provides device drivers for HPUX, Linux, SunOS, and Solaris (both SPARC and x86) at no charge. Drivers are generally available on diskette, or may be downloaded from the Tahoma Technology website: http://www.tahomatech.com.

For pricing and availability of DOS, Windows, and AIX drivers, please contact Interface Consultants:

website: http://www.ifc2.com
email: wcs@ifc2.com
phone: 713.529.3706
fax: 713.529.3715

Tahoma Technology does not provide plotting application programs such as rasterizers. Application software is available from our OEM customers, and from third party plotting hardware and software vendors. Tahoma Technology does provide some testing and utility programs which are available from our website.

Hardware Installation

Printer/Plotter Interface Selection

The 10115/10117 requires very little hardware configuration. However, it is necessary to set the board's interface protocol - Versatec (VPI) Centronics (IEEE1284) - to match the attached plotter. A mis-configured board will not communicate with the printer/plotter. Error messages may or may not be generated, depending on the software used with the board.

If the plotter uses VPI TTL interfacing (10115 only),connect the board's ribbon cable from the I/O-OUT pins to the V-TTL pins.

If the plotter uses VPI Differential interfacing, connect the ribbon cable from the I/O-OUT pins to the V-DIFF pins.

For Centronics compatible plotters, connect the cable from the I/O-OUT pins to the CENT pins.

Note: the board's interface selection ribbon cable MUST be configured to match the attached printer/plotter BEFORE installing the board.

Termination Network

The 10115 and 10117 use a socketed resistor network to terminate incoming signals from the attached plotter. This network may be changed as necessary to accommodate the plotter's particular requirements. Network changes are useful for Centronics compatible plotters only. The stock network should be used for VPI applications.

Symptoms of an inappropriate termination network include intermittent plotting, no plot output, and a permanent device busy or offline indication.

As shipped, the board has a 220/330 Ohm network installed. This network gives the best possible impedance match with typical board to plotter cabling. Some IEEE1284 compatible printers and plotters, however, are not capable of driving this termination. For those plotters, the included 470 Ohm network should be used. This network should be installed in the socket labeled TERM. The 10115 carries this extra network in a socket labeled SPARE TERM. The 10117 does not have a spare term socket. The extra network is included in the board's shipping container.

Software Installation

Please note that Tahoma Technology provides device drivers for HPUX, Linux, SunOS, and Solaris (both SPARC and x86) at no charge. Drivers are generally available on diskette, or may be downloaded from the Tahoma Technology website: http://www.tahomatech.com. Tahoma's drivers include full source code.

For pricing and availability of DOS, Windows, and AIX drivers, please contact Interface Consultants:

website: http://www.ifc2.com
email: wcs@ifc2.com
phone: 713.529.3706
fax: 713.529.3715

Tahoma Technology does not provide plotting application programs such as rasterizers. Application software is available from our OEM customers, and from third party plotting hardware and software vendors. Tahoma Technology does provide some testing and utility programs which are available from our website. In particular, the ikonex utility may be useful for verifying initial installation, and/or troubleshooting problems.

Software installation varies with operating system and software provider. Device drivers may be provided by Tahoma Technology, or others. The following is a brief description of Tahoma Technology device driver installation. Please refer to vendor instructions for non-Tahoma supplied drivers and application software.

Tahoma Tech's drivers are available from our website. Most drivers are also available on diskette. Driver distribution formats vary; some are collections of discrete files, some are gathered in compressed tarfiles, and others (Solaris) are provided in package format. All drivers are available as collections of individual files as well as in more compressed formats on our website.

Note that all Solaris Hardcopy driver packages are called IKONihcp. The user must ensure that the driver being loaded is appropriate for the host operating system and board being installed. Further, some package files include both the ihcp and idr (for DR11 boards) drivers. Only the Hardcopy driver should be loaded. Attempting to load both drivers will cause an error message. Other than adding unnecessary driver code to the system, this is harmless.

Note also that the board should be installed before the driver is loaded. It is possible to install driver files without the board in place, but the driver install will not complete successfully if the board is not detected.

Drivers on Diskette

Drivers may be provided on diskette as discrete files, compressed tarfiles, or packages (Solaris). The label will indicate the format of the diskette's contents.

HPUX, SunOS, Linux

Drivers on diskette in tar format (typically HPUX, SunOS, and Linux) should be untarred into a directory:

```
tar -xvf /dev/fd0 (or other floppy device name, depending on operating system)
```

Follow the included README file for further installation instructions.

Future releases on diskette may be tarred and compressed prior to tarring to diskette. These drivers should be untarred, uncompressed, and possibly untarred again:

```
tar -xvf /dev/fd0
uncompress <resulting file>
tar -xvf <resulting file>
```

Follow the included README for further installation instructions.

Solaris

Solaris drivers are currently provided on diskette in datastream package format. Datastream format floppies read very slowly during package installation. In the future, Tahoma may deliver Solaris driver packages in compressed tar format which should speed up the installation process.

To install a Solaris driver from a datastream package format diskette (package format on the diskette label), as super user:

For hosts that mount the floppy as a file system:

```
volcheck
pkgadd -d /vol/dev/aliases/floppy0
```

For hosts that do not mount the floppy:

```
pkgadd -d /dev/rdiskette
```

The pkgadd utility will indicate the packages available on the floppy, and request a selection. Choose the IKONihcp package by number, and proceed. (Be patient, it takes a while to install from a diskette in package format!) The package chosen will be installed to /opt/IKONihcp, the driver and .conf files copied to /usr/kernel/drv (and/or /usr/kernel/drv/sparcv9 if a 64 bit driver), /etc/devlink.tab modified, and the driver added to the kernel.

Consult the README file(s) in /opt/IKONihcp for additional driver usage and configuration information.

To install a driver package from a compressed tar package format driver diskette (compressed tar package on the diskette label):

For hosts that mount the floppy as a file system:

```
volcheck
tar -xvf /vol/dev/aliases/floppy0
```

For hosts that do not mount the floppy:

```
tar -xvf /dev/rdiskette
```

Either of the above operations will result in a file named something like pci_sparc.Z. To install the compressed Solaris driver package, as super user:

```
uncompress pci_sparc.Z
pkgadd -d ./pci_sparc
```

The pkgadd utility will indicate the packages available on the floppy, and request a selection. Choose the IKONihcp package by number, and proceed. The package chosen will be installed to /opt/IKONihcp, the driver and .conf files copied to /usr/kernel/drv (and/or /usr/kernel/drv/sparcv9 if a 64 bit driver), /etc/devlink.tab modified, and the driver added to the kernel.

To remove a driver package, as super user:

```
pkgrm IKONihcp
```

All files will be removed, devlink.tab will be restored to its prior state, and the driver removed from the kernel.

Downloaded Drivers

Drivers may be downloaded as discrete files, compressed tarfiles, or compressed packages (Solaris). Follow the website's Downloads link to the driver appropriate to the operating system and Tahoma board being installed.

You may use your browser for http file transfers from www.tahomatech.com or an ftp utility for anonymous ftp file transfers from ftp.tahomatech.com.

Note: use binary file transfer mode to download driver files.

HPUX, SunOS, Linux

If a driver is downloaded as a set of discrete files, follow the included README for installation instructions.

If the driver is downloaded as a compressed tarfile (named something like ihcp.tar.Z) first uncompress and then untar:

```
uncompress ihcp.tar.Z
tar -xvf ./ihcp.tar
```

Follow the included README for further instructions.

Solaris

Solaris drivers are usually available for download in compressed package format (named something like pci_sparc.Z). To install a compressed Solaris driver package, as super user:

```
uncompress pci_sparc.Z
pkgadd -d ./pci_sparc
```

The pkgadd utility will indicate the packages available on the floppy, and request a selection. Choose the IKONihcp package by number, and proceed. The package chosen will be installed to /opt/IKONihcp, the driver and .conf files copied to /usr/kernel/drv (and/or /usr/kernel/drv/sparcv9 if a 64 bit driver), /etc/devlink.tab modified, and the driver added to the kernel.

Consult the README file(s) in /opt/IKONihcp for additional driver usage and configuration information.

To remove a driver package, as super user:

```
pkgrm IKONihcp
```

All files will be removed, devlink.tab will be restored to its prior state, and the driver removed from the kernel.

Post-installation Issues

Successful board and driver installation may be verified in one of several ways. The easiest is to simply use the chosen plotting application to generate a plot. Please note that Tahoma Technology does not provide plotting application software.

It is possible to do some low level testing by sending print or plot files directly to the device driver. In unix terms, to send a text file “textfile” or plot file “plotfile” to the plotter device node:

cat textfile >/dev/ihcp0 or cat plotfile >/dev/ihcp0

It is important to understand that the device driver and board do not know whether a given file is a plot file or a text file. The file’s data is sent to the plotter in the current mode, which may be either print or plot. If a text file is sent to a plotter that is in plot mode, the likely output is blank paper, or rows of random looking dots. If a plot file is sent to a plotter that is in print mode, nonsense text is the result. Tahoma’s drivers can be configured to default to either mode. Consult driver README files for details.

In normal operation, the plotting application software will control the print/plot mode of the board, driver, and plotter.

The ikonex utility available on Tahoma’s website can be used to send existing print and plot files to the plotter in the mode appropriate to the file. Ikonex can manipulate print and plot modes, issue paper motion (and other) commands, send files, and return board an plotter status. Ikonex is not intended for production plotting, but may be useful for testing and troubleshooting. Ikonex is available for Linux, Solaris SPARC, and Solaris x86. Source is included.

Introduction to Detailed Specifications

This section of the manual describes the hardware specifications and programming models (register descriptions) for the Tahoma Technology Model 10115 and Model 10117 Hardcopy interfaces for the PCI bus. The 10115 and 10117 are nearly identical products, with a few significant exceptions. The 10115 is a full-size PCI card. The 10117 is a half size card, and deletes some of the features of the 10115, while maintaining the 10115's high performance. With the obvious exception of the deleted functions on the 10117 - and the device I.D. presented to PCI BIOS - the two boards program and perform identically.

The Tahoma Technology Models 10115 and 10117 Hardcopy interfaces are very high performance interfaces for hardcopy output devices. They can be configured to drive either Centronics or Versatec compatible printers and plotters. Attached Versatec-compatible devices may use either TTL (10115 only), or Differential (10115 and 10117) interfacing.

Both boards are fully compatible with Versatec's "Green Sheet" interface specification. They also include sufficient flexibility to accommodate devices that deviate from the Centronics specification. Support for ECP compatible devices is provided, including high-speed transfers, and reverse channel data transfers.

A single external 37 pin "D" connector, mounted in the backplate is used to connect to the attached device, with Versatec/Centronics and TTL/Differential selection done via an internal jumper cable. The 10117 supports Versatec Differential and Centronics interfaces - it does not provide a Versatec TTL interface.

Data transfers to the hardcopy device may be done via burst Programmed-I/O or DMA transfers. The 10115/10117 transfers 32 bits at a time from the PCI bus to the on-board FIFOs using multiple long-word bursts. The burst rate to FIFO is 30-40Mbytes/second in a system that can support these transfer rates. In X86 type architectures which support string moves to memory-mapped devices, it is entirely possible to achieve these rates without using DMA. DMA may be useful in systems that do not support burst type memory access instructions, although cached memory systems may still provide burst transfers from cache to device, even in the absence of string move instructions. Device side transfer rate is approximately 2Mbytes/second maximum (this may be increased by factory modification if a customer's particular plotter requires it - most don't). A deep FIFO (16Kbytes) allows maintaining this device transfer rate while the bus side transfers are done in bursts, for maximum bus efficiency.

The 10115/10117 fully conforms to the PCI specification. The register set used in the board may be optionally mapped into high memory, low memory (under 1Mbyte), or I/O space. These options are factory selected by the EEPROM installed on the board. The default configuration is memory mapped in high memory. The low memory option may be useful in systems that run in X86 real mode, without access to upper memory. The I/O mapped configuration may make driver writing easier in some situations, but will require DMA for reasonable performance, as the 10115/10117 does not burst when mapped into I/O space.

Outgoing data and control line drivers are high-drive bipolar devices. The internal TTL (10115 only) and Differential Versatec connectors are driven by separate devices to allow optimizing device type for the particular interface. Centronics connector drivers are capable of sourcing and sinking high currents. Cable receivers are true Schmidt triggers with r/c integration for maximum noise rejection. The Centronics connector termination resistors are socket-selectable.

Specifications

Bus Interface - Slave Access

Fully compatible with PCI Specification.

Vendor ID 11D5 (hex).

Device ID 0115 (hex) for 10115
0117 (hex) for 10117

Registers occupy three address ranges: PCI configuration registers are accessible only in configuration space. Bus interface chip run-time registers may be mapped by BIOS into I/O and memory space. "IKON" run-time registers may be mapped by BIOS into I/O or memory space as determined by installed EEPROM. Bus interface chip run-time registers are also accessible in "IKON" register map.

Bus interface chip registers occupy 512 bytes (early versions - the 9060 REV 2 and REV 3 chips - used 256 bytes) of I/O and/or memory space. "IKON" registers occupy 8Kbytes of I/O or memory space.

Register base addresses are assigned by BIOS. I/O or memory map is determined by the 10115/10117 EEPROM.

Bus Interface - DMA

Full bus master operation with chaining DMA capability (very early version of the 10115 using 9060 REV 2 chip did not support DMA, early versions of the 10117 may not support chaining).

DMA chain entries in local or PCI memory for the 10115, PCI memory space for the 10117. (Early versions of the 10115 allow local memory chaining only. First version of 10117 may not support DMA chaining).

32 bit burst transfers used for maximum bus efficiency.

NOTE: Several versions of the PLX Technology 9060/9080 PCI bus interface chip have been used on the 10115/10117. The first one available to Tahoma Technology - the 9060 REV 2 - did not support usable DMA. The second version, the 9060 REV 3, supports DMA on both the 10115 and 10117, with chaining from local memory chain lists on the 10115, and non-chaining DMA on the 10117. The "final" version of the chip is the 9080 REV 3, which supports DMA with chaining from either local memory or PCI host memory on the 10115, and PCI host memory chaining only on the 10117 (the 10117 has no local memory).

Bus Interface - Interrupts

Interrupt level fixed at PCI INT A. (ISA interrupt level equivalent assigned by BIOS).

FIFO depth

Standard FIFO 32 bits wide. Total capacity 16Kbytes. Higher capacities available on a custom basis.

Device Interface - General

"Data Streaming" capability for Tektronix color copiers, and other devices that can use synchronous transfers (no ack handshake) for higher speed.

Directly supports all Versatec "Green Sheet" compatible products in both TTL (10115 only) and Differential (long-line) modes. Long-line capability to 300 meters.

Supports any Centronics-compatible device.

Control bit allows ignoring Centronics BUSY signal for the few devices which don't drive it.

Special handshakes mode can use BUSY instead of ACK for higher transfer rates - with compatible plotters, and for ECP compatibility, and allow a 4-edge handshake.

Full software control of all outputs, including Data Strobe, for compatibility with ECP specification.

Software driven reverse channel data transfer capability.

High-current output drivers can drive any TTL-type termination.

Socketed input terminators allow matching device manufacturer's preferred terminations.

All inputs received by Schmidt-triggers with r/c integration networks for maximum noise rejection.

Four different handshake timing choices allow for exact matching to device's requirements while maintaining maximum transfer rate. Timing is selected by driver software.

Device Interface - Timing

Four timing selections are available in each of three modes : Centronics, Versatec, and 4-edge.

Centronics:

Choice	Data Set-up	Strobe Width	Data Hold
#0	150ns	200ns	150ns
#1	250ns	300ns	200ns
#2	450ns	500ns	300ns
#3	550ns	800ns	450ns

Versatec:

Choice	Data Set-up	Strobe Width	Data Hold
#0	150ns	200ns	150ns*
#1	250ns	300ns	150ns*
#2	350ns	500ns	150ns*
#3	550ns	800ns	150ns*

* Minimum hold time - data held until plotter ready for next transfer.

4-edge:

Choice	Busy on to Strobe off	Busy off to Strobe on	Data to Strobe on	Busy off to Data change
#0	300ns	300ns	150ns	150ns
#1	400ns	400ns	250ns	150ns
#2	500ns	500ns	350ns	150ns
#3	700ns	700ns	550ns	150ns

In 4-edge mode, the strobe is held on until Busy is asserted, and held off until Busy is removed. Strobe length is dependent on Busy from the attached device.

Pulses to attached device - Versatec Clear, Remote Line Terminate, Remote Form Feed, Remote EOT, same width as strobe.

Versatec Reset, and Centronics Input Prime, are driven by a latch - pulse length is determined by software.

Mechanical

Board dimensions:	10115	full size PCI.
	10117	half size PCI

Power

Power Consumption:	1.2A @5VDC

Features and Revision Levels

The 10115 and 10117 offer the same basic features and high performance. There are, however, a few differences between the two models, and within each model - as indicated by revision level. Note that revision level refers to the revision level reported in the board's configuration registers and on the EEPROM label, not to the artwork revision letter of the board itself.

The differences between the 10115 and 10117 are caused by the need to drop some of the features of the 10115 in order to fit the same high-performance core logic on a half-size PCI card. The 10117 does not support the Versatec TTL interface, which was used by some older plotters. The 10117 also eliminates the device test toggle switches of the 10115. Because it has no local memory, the 10117 can't do chaining DMA when assembled with the PLX 9060 REV 2 chip. It will do DMA chaining from host memory when assembled with the later 9080 chip. The FIFOs used are not part of local memory. Both the 10115 and 10117 carry 16Kbytes of FIFO.

The availability of the PLX chips that both boards use has caused differences in features between rev levels within a given model:

10115 REV 0	Uses PLX 9060 REV 2 chip. Has no local memory. This revision has no DMA capability.
10115 REV 1	Uses PLX 9060 REV 3 chip. Has local memory. Has DMA chaining capability using local memory for DMA chain entries. DMA control/status registers must be accessed in the "IKON" register map.
10115 REV 2	Uses PLX 9080 chip. Has local memory. Will do DMA chaining from either local memory or host PCI memory. DMA control/status registers may be accessed in the "IKON" map, or directly in the "PLX" register space.
10117 REV 0	Uses PLX 9060 REV 3 chip. Has no local memory. Will do DMA, but not DMA chaining. DMA control/status registers may be accessed in the "IKON" map, or directly in the "PLX" register space.
10117 REV 1	Uses PLX 9080 chip. Has no local memory. Will do DMA chaining from host PCI memory only. DMA control/status registers may be accessed in the "IKON" map, or directly in the "PLX" register space.
NOTE:	The REV referred to above is the revision level reported in the board's configuration registers and on the EEPROM label, not the board's artwork revision level.

Current designs are based on the PLX 9080 REV 3. Full documentation on these parts is available from PLX Technology:

http://www.plxtech.com

On-Board Registers

The Model 10115/10117 implements three sets of registers: the PCI configuration registers, the bus interface chip run-time registers, and the "working" ("IKON") registers.

The working registers are referred to as “IKON” registers for compatibility with earlier documentation and software.

The PCI configuration registers are used to configure the board's slave addresses and ISA compatible interrupt level. These registers are configured by BIOS, using accesses in configuration space, to assign slave addresses to the register sets used by the 10115/10117. The configuration registers may be read by a device driver - using calls to BIOS PCI services - to determine what the assigned slave addresses are, and whether they are mapped into I/O or memory space.

The bus interface chip's run-time registers (and local configuration registers - which will in general not be used by the device driver) occupy 512 bytes (in the REV 1 release of the 10115, and the REV 0 release of the 10117, they occupy 256 bytes), and may be mapped into I/O and/or memory space. The chip indicates to BIOS that it supports both maps, and, in general, BIOS will allocate space in both maps. Configuration registers indicate where in the maps space has been allocated for the working registers. The PLX chip permits BIOS to map its registers anywhere in 32 bit addressed memory space. Typically, it will be mapped in to high memory, which is not directly accessible by an X86 type processor operating in real mode. In a DOS environment, DOS extenders will be required to access these registers in the memory map. If BIOS also maps the run-time registers into I/O space, simple I/O instructions may be used. Versions of the 10115 beyond REV 0, and all versions of the 10117, also allow access to these registers in a portion of the "IKON" register map, which will allow for simpler access, either via the I/O map, or in low memory, depending on the EEPROM installed in the 10115/10117. Access to the run-time registers is required if interrupts are to be enabled or DMA used.

The Model 10115/10117 "working" registers occupy an 8Kbyte block of addresses which may be located in high memory, low memory, or I/O space. The EEPROM installed on the 10115/10117 indicates to BIOS which map is to be used. The standard factory EEPROM allows mapping anywhere in memory space, which will normally cause BIOS to map the board into very high memory - requiring DOS extenders, or protected mode operation. Optionally, EEPROMs may be installed which will cause the board to be mapped into low memory or I/O space. Mapping into I/O space can simplify a driver, since simple X86 I/O instructions may be used to access the "IKON" registers (and the run-time PLX registers if BIOS has allocated space in the I/O map). Operation in I/O space is not recommended for the REV 0 release of the 10115, or any application of the 10115 or 10117 which makes use of burst programmed writes to FIFO. I/O mappings do not allow burst writes to FIFO. While the board will function in this configuration, it will be much slower than a board mapped into memory space, where X86 string instructions may be used to burst data into the 10115/10117 FIFOs at 30+ Mbytes/second. The DMA functions supported in versions beyond 10115 REV 0 will provide high-speed bus master burst accesses to PCI memory, and will avoid the need for burst writes to FIFO (although this will still be supported).

All on-board registers are organized as 32 bit words. The configuration and run-time registers are full 32 bit registers. All of the "IKON" registers, except the 32 bit Data Out register, and Burst Out range, use only 8 bits - aligned as the low byte, on 32 bit boundaries. The Data Out register and Burst Out range use all 32 bits, allowing four bytes to be written to FIFO with each transfer.

NOTE: If the 10115/10117 is mapped into memory, and caching is enabled for that region of memory, care must be taken after writing to registers, or writing or bursting to FIFO, to make sure that the cache has been written to the 10115/10117 before the next operation is begun.

If two FIFO burst writes are performed back-to-back, the first may be overwritten by the second, if the first is still in cache when the second is begun. There is also a potential problem if an interrupt is enabled before the cache has emptied - a FIFO not half full interrupt could occur immediately if it is enabled before the cache has drained to the on board FIFO.

It may be necessary to disable cache for the regions occupied by the 10115/10117s registers, memory, and FIFO. If that is not possible, doing a register read following a write may force the cache to drain. The approach taken will be determined by the characteristics of the specific system in which the 10115/10117 is installed.

Register Formats

All register addresses are in HEX - indicated by the “h” following the numeral/letter.

Bit 31 is the MSB for all registers during 32 bit access. The run-time registers, the 32 bit Data Out register, and the Burst Out range use all four bytes. All "IKON" registers except the 32 bit Data Out register, and Burst Out range, use only the low byte. For those registers, only the low 8 bits are shown here.

The PLX registers shown here are representative of the registers supported by the PLX 9060, and 9080.

Current designs are based on the PLX 9080-3. Full documentation on these parts is available from PLX Technology:

http://www.plxtech.com

As of this writing:

9080 Revision 3 manual: 9080_105ds.pdf
9080 Revision 3 errata: 9080-3errata-13.pdf

PCI Configuration Registers

PCI Config Address	Register Function			
00h	Device ID		Vendor ID	
04h	Status		Command	
08h	Class Code			Revision ID
0Ch	BIST	Header	Latency	Cache Line
10h	PCI Memory Base for Runtime Registers			
14h	PCI I/O Base for Runtime Registers			
18h	PCI I/O or Memory Base for “IKON” Registers			
1Ch				
20h				
24h				
28h				
2Ch	Subsystem (device) ID		Subsystem Vendor ID	
30h				
34h				
38h				
3Ch	M ax Latency	Min Grant	Int Pin	Int Line

These registers are described in detail in documents available from PLX Technology. For the most part, they are configured according to the parameters established by the 10115/10117 EEPROM, and by BIOS. They may be accessed by a device driver by first asking BIOS if the board is present - by calling BIOS services with the board's Vendor ID (0x11D5) and Device ID (0x0115 or 0x0117). BIOS returns a "handle" which is used in later BIOS calls to read the configuration registers.

The PCI memory base for runtime registers indicates where in the memory map the PLX chip's registers are located. This may be anywhere in memory. The PCI I/O base for runtime registers indicates where in I/O space these registers are located. Most BIOS implementations will assign areas in both spaces to the run-time registers.

The PCI I/O or memory base for "IKON" registers indicates where the "IKON" registers are located, and in which map. The EEPROM tells BIOS whether the registers may be located in I/O, low memory, or high memory space.

The only other register normally of use to the device driver is the Int Line. The 10115/10117 is always connected to PCI INT A. This interrupt may be assigned by BIOS to an ISA-type interrupt level, which is reported in this register, and should be used by the driver when handling interrupts from the 10115/10117.

Local Configuration Registers

The address shown in parentheses is the offset into the "IKON" register set. REV 0 of the 10115 does not map the PLX registers (Local Configuration, Shared Runtime, DMA) into the "IKON" space.

PCI Runtime Offset ("IKON" Offset)	Register Function
00h (80h)	Range for PCI to "IKON" Registers
04h (84h)	Re-map for PCI to "IKON" Registers
08h (88h)	Local Arbitration Register (SD & 9080 only)
0Ch (8Ch)	Big/Little Endian Register (SD & 9080 only)
10h (90h)	Range for Expansion ROM (not used)
14h (94h)	Re-map for ROM (not used) & BREQ Control
18h (98h)	Bus Region Descriptors
1Ch (9Ch)	Range for Direct Master (not used)
20h (A0h)	Local Base for Direct Master (not used)
24h (A4h)	Local Base for Direct Master (not used)
28h (A8h)	PCI Re-map for Direct Master (not used)
2Ch (ACh)	PCI Configuration for Direct Master (not used)

These registers are described in detail in documents available from PLX Technology. They are set to appropriate values by the EEPROM on the 10115/10117, and should not be modified by the device driver.

Shared Run-time Registers

PCI Runtime Offset ("IKON" offset)	Register Function	
40h (C0h)	Mailbox Register 0 (not used)	
44h (C4h)	Mailbox Register 1 (not used)	
48h (C8h)	Mailbox Register 2 (not used)	
4Ch (CCh)	Mailbox Register 3 (not used)	
50h (D0h)	Mailbox Register 4 (not used)*	
54h (D4h)	Mailbox Register 5 (not used)*	
58h (D8h)	Mailbox Register 6 (not used)*	
5Ch (DCh)	Mailbox Register 7 (not used)*	
60h (E0h)	PCI to Local Doorbell Register (not used)	
64h (E4h)	Local to PCI Doorbell Register (not used)	
68h (E8h)	Interrupt Control/Status	
6Ch (ECh)	EEPROM Control & User I/O Bits	
70h (F0h)	Device ID	Vendor ID
74h (F4h)	Unused	Revision ID

* not available in 9060SD

These registers are described in detail in documents available from PLX Technology. The only functions used in these registers will be the Interrupt Control/Status bits, and the User Output bit - which is used in conjunction with a bit in the "IKON" registers when byte "swizzling" is selected.

DMA Registers

PCI Runtime Offset ("IKON" Offset)	Register Function
80h (100h)	DMA Channel 0 Mode
84h (104h)	DMA Channel 0 PCI Address
88h (108h)	DMA Channel 0 Local Address
8Ch (10Ch)	DMA Channel 0 Transfer Count
90h (110h)	DMA Channel 0 Descriptor Pointer
94h (114h)	DMA Channel 1 Mode
98h (118h)	DMA Channel 1 PCI Address
9Ch (11Ch)	DMA Channel 1 Local Address
A0h (120h)	DMA Channel 1 Transfer Count
A4h (124h)	DMA Channel 1 Descriptor Pointer
A8h (128h)	DMA Command/Status Register
ACh (12Ch)	DMA Mode/Arbitration Register 0
B0h (130h)	DMA Threshold Register

These registers are described in detail in documents available from PLX Technology. They will be used by the device driver to set up and initiate DMA (bus master) transfers from PCI memory to the 10115/10117 FIFOs. DMA is not supported in the REV 0 version of the 10115. REV 1 10115, and REV 0 10117 boards will have DMA capability, with the DMA registers accessed as part of the "IKON" register map. Later boards will also allow access to the DMA registers in the run-time register map.

NOTE: The 10115 was designed to use DMA channel 0 of the PLX 9060. Because of some early bugs in the PLX chip (which did not affect Tahoma Technology's designs) some PLX customers elected to use DMA channel 1 for single DMA channel designs. Because of this, PLX decided to use DMA channel 1 as the sole channel in the 9060 SD chip. At Tahoma Technology's urging, PLX has agreed to design the 9060 SD in such a way that it will respond to register accesses to DMA 0 and DMA 1. If PLX succeeds in this design effort, code written for the 9060 REV 3 chip should run unmodified on the 9060 SD, even though it accesses DMA channel 0.

Later 10115 artwork - board REV C and later - and all artwork revisions of the 10117 allow using either DMA channel when the 9060 REV 3 or 9080 REV 3 part is installed. Earlier 10115 board artwork versions do not connect DMA channel 1 to the Tahoma Technology logic.

IKON Registers

All offsets are from the "IKON" memory or I/O space base address.

Offset	Register Function	read/write
00h	Interrupt mask	read/write
04h	Mode	read/write
08h	Device Control	read/write
0Ch	Interface Control	read/write
10h	Interface Status	read only
14h	Device Status	read only
18h	Reverse Data	read only
1Ch	reserved	
20h	Auto LTR Count (low byte)	read/write
30h	Auto LTR Count (high byte)	read/write
40h	8 Bit Data Out	write only
48h	Command Out	write only
50h	32 Bit Data Out	write only
80h-13Fh	PLX Runtime Registers	read/write
140h-FFFh	Local Chain Memory*	read/write
1000h-1FFFh	Burst Out Range	write only

* not available on Model 10117

These are the registers of primary interest to the device driver. All are 8 bits wide aligned as the low byte of 32 bit words, and may be accessed as 8, 16, or 32 bit values, except the 32 Bit Data Out register, the Burst Out Range, and the PLX run-time registers, which are 32 bits wide. The PLX registers may be accessed as 8, 16, or 32 bit values. The 8 Bit Data Out register may be written as 8, 16, or 32 bits, but only the low byte will be used. The 32 Bit Data Out register and the Burst Out range must be written as 32 bit values, as the 10115/10117 assumes all writes to these addresses have four valid bytes.

Interrupt Mask

Bit	Name	Function
07	DIRM	Device & Interface Ready
06	EMTM	FIFO Empty
05	NHFM	FIFO Not Half Full
04	NFLM	FIFO Not Full
03		reserved
02	NPPM	Paper Empty
01	NSLM	Not Selected (Offline)
00	FLTM	Centronics Fault

These latched bits control interrupts from the 10115/10117. It is also necessary to enable interrupts in the PLX chip by setting bit 11 (PCI local interrupt enable) to 1 in the Interrupt Control/Status register, and leaving bit 8 (PCI interrupt enable) at its default state (set).

All interrupts on the 10115/10117 are level type interrupts. Setting a mask true when its associated condition is true will cause the interrupt line to be driven. This is different from the edge style interrupts used on some earlier Tahoma Technology products (as dictated by the "classic" ISA architecture). In general, a particular mask should not be enabled until the operation that will cause an interrupt has been completed -- one would set the not half full interrupt mask after doing the I/O burst that moved data to the FIFO and may have caused it to become more than half full. In the case of DMA, the PLX DMA interrupt would be used.

This register is cleared to all zeros by a bus initialize, MCLR in the Interface Control register, or via bit 30 (PCI software adapter reset) in the PLX EEPROM Control & User Bits register.

DIRM Allows an interrupt when the FIFO is empty, the interface is idle, and the attached device is ready. This can be used as a "master interrupt" for the board, but is not necessarily the best choice for maximum performance. Using the FIFO not half full interrupt allows overlapping I/O or DMA bursts to FIFO with device transfers, and will help to keep the device continuously supplied with data.

EMTM Allows the FIFO empty condition to generate an interrupt.

NHFM This mask enables an interrupt any time that the FIFO is less than half full + 1. This should probably be the "master interrupt" for applications that do not use DMA. After bursting a block of data to the FIFO, the driver can check for less than half full +1, and enable this interrupt if the FIFO is more than half full +1.

NFLM Allows an interrupt any time the FIFO is not full.

NPPM Enables an out-of-paper interrupt.

NSLM Enables an off-line interrupt.

FLTM Enables the Centronics FAULT signal to cause an interrupt.

Mode

Bit	Name	Function
07	4EDG	Four Edge Handshake
06	REVD	Reverse Data
05	UBSY	Use BUSY not ACK
04	IBSY	Ignore BUSY
03	BRST	Tahoma Burst Mode
02	SWIZ	Byte Swizzle
01	SPD1	Handshake Speed Select
00	SPD0	

These latched bits control the mode of communication with the attached device.

This register is cleared to all zeros by a bus initialize, MCLR in the Interface Control register, or via bit 30 (PCI software adapter reset) in the PLX EEPROM Control & User Bits register.

4EDG Setting this bit causes the 10115/10117 to use a full 4-edge handshake when transferring data to the attached device. It will present data, then assert strobe, and wait for the device to assert BUSY or ACK, as selected by the UBSY bit in this register. The 10115/10117 then removes the strobe, and waits for the device to de-assert BUSY. The 10115/10117 then presents new data, and asserts strobe. This mode is provided for ECP compatibility.

The default handshake mode uses a pulsed strobe, and a 2-edge ACK or BUSY handshake. Strobe is pulsed for a fixed duration, and the 10115/10117 waits for the device to go not ready, and then ready again. This may be done by toggling READY if the Versatec interface is selected, or by pulsing ACK, or toggling BUSY (depending on the state of UBSY) in the case of a Centronics interface.

In most cases, the default handshake will give the highest performance, if the device is capable of operating with the 10115/10117's shorter data setup and strobe widths.

REVD Setting REVD causes the 10115/10117 to tri-state its output data lines (Centronics and Versatec TTL modes only), and allows the driver software to read data presented by the device in the Reverse Data register. This will usually be done at the same time the handshake lines are manipulated by software to provide IEEE P1284 type reverse channel capability.

Data transfers in reverse channel mode are entirely under software control - there is no automatic handshake provided by the 10115/10117.

UBSY Setting this bit causes the 10115/10117 to use BUSY, rather than ACK, to provide the acknowledge handshake from the device. This mode will provide higher throughput with some plotters that first toggle BUSY, then pulse ACK, after each byte transferred. If the device is actually ready to receive another byte at the end of BUSY, the time delay added by the width of ACK can be avoided.

This bit must be set when using an ECP mode 4-edge handshake, which uses BUSY rather than ACK.

IBSY Setting IBSY causes the Centronics BUSY input to be ignored.

BRST BRST selects Tahoma Burst Mode transfers. This mode is an enhancement to the VPI handshake that allows transfer rates of up to 5Mbytes/second (4Mbytes/second in the current 10115//10117 implementation). This mode also greatly reduces the effect of cable length on transfer rate.

Tahoma Burst Mode is only usable when connected to a plotter that also supports this enhanced mode of operation. See the Tahoma Burst Mode section of this manual for further information.

SWIZ Byte swizzle mode reverses the order in which the bytes in a 32 bit word are sent to the device. The default causes the low order byte to be sent first. If SWIZ is set, the high order byte is sent first.

To use swizzle mode, the User Output bit in the EEPROM Control & User Bits register must be set to zero, and SWIZ to 1. The User Output bit defaults to 1.

Swizzle mode also requires that the 8 Bit Data Out and Command Out bytes be the high bytes of their 32 bit words.

The 9060 SD chip also has a big/little endian capability for direct slave reads/writes - but not DMA. The 9080 adds big/little endian control for DMA. See the Big/Little Endian Register in the PLX documentation.

SPD1,0 These bits control the timing of the handshake used to transfer data to the attached device. The Versatec specification requires minimum timing compatible with speed selection 1. Many plotters will be able to use speed 0 to advantage, but some will not tolerate timings this tight.

Four timing selections are available in each mode: Centronics, Versatec, and 4-edge.

Centronics:

Choice	Data Set-up	Strobe Width	Data Hold
#0	150ns	200ns	150ns
#1	250ns	300ns	200ns
#2	450ns	500ns	300ns
#3	550ns	800ns	450ns

Versatec:

Choice	Data Set-up	Strobe Width	Data Hold
#0	150ns	200ns	150ns*
#1	250ns	300ns	150ns*
#2	350ns	500ns	150ns*
#3	550ns	800ns	150ns*

* Minimum hold time - data held until plotter ready for next transfer.

4-edge:

Choice	Busy on to Strobe off	Busy off to Strobe on	Data to Strobe on	Busy off to Data change
#0	300ns	300ns	150ns	150ns
#1	400ns	400ns	250ns	150ns
#2	500ns	500ns	350ns	150ns
#3	700ns	700ns	550ns	150ns

In 4-edge mode, the strobe is held on until Busy is asserted, and held off until Busy is removed. Strobe length is dependent on Busy from the attached device.

Pulses to attached device - Versatec Clear, Remote Line Terminate, Remote Form Feed, Remote EOT, same width as strobe.

Versatec Reset, and Centronics Input Prime, are driven by a latch - pulse length is determined by software.

Device Control

Bit	Name	Function
07	FRDY	Force READY
06	RDYD	Disable READY
05	STBD	Disable Auto Strobe
04		reserved
03	ASTB	Assert Strobe
02	AAFD	Assert nAutoFd
01	DSEL	De-assert SEL IN
00	AINT	Assert nINIT

These latched bits control specific signals to the attached device. They can be used to implement ECP communications with a compatible printer/plotter. They may also be used to implement other device specific functions.

The user is referred to the IEEE P1284 specification and/or the Microsoft ECP specification for further, detailed information.

This register is cleared to all zeros by a bus initialize, MCLR in the Interface Control register, or via bit 30 (PCI software adapter reset) in the PLX EEPROM Control & User Bits register.

FRDY This bit forces the interface state to ready, regardless of the state of the device. It may be used when it is necessary to present data at the output lines, even though the device is not accepting data.

RDYD RDYD forces the interface to a not-ready state. It may be used for diagnostic purposes. It is included here primarily for completeness.

STBD This bit inhibits the 10115/10117 from generating a data strobe when data appears at the output of the FIFO. It may be used in special applications when it is necessary to present data at the output lines, but it is not desired to transmit the data to the device input buffers.

ASTB ASTB causes the unconditional assertion of the data strobe to the device.

AAFD This bit causes the nAutoFd signal to the device to be asserted. Centronics mode only.

DSEL Setting DSEL causes the SEL IN signal to the device to be de-asserted. Centronics mode only.

AINT This bit causes nINIT (Centronics INPUT PRIME, Versatec RESET) to be asserted. The device reset signal is also asserted by bus initialize, PCI Software Adapter reset in the PLX chip, or via the REST bit in the Interface Control register.

ASTB, AAFD, DSEL, AINT, and other control bits may be used when implementing IEEE P1284 communications, or for other special purposes. Refer to the '1284 specification for details.

Interface Control

Bit	Name	Function
07	MCLR	Master Clear Interface
06	SWAK	Software Acknowledge
05	RINT	Reset Interrupt Flag
04		reserved
03		reserved
02		reserved
01		reserved
00	REST	Reset Device

These latched bits are used to reset certain internal interface functions, as well as to send a reset to the attached device. Unlike the similar functions in "pulse" registers in earlier Tahoma Technology products, these bits are latched. A given bit must first be set to one, then to zero, to accomplish the required operation. Bits in this register should not be left set, or they will inhibit proper interface operation.

This register is cleared to all zeros by a bus initialize, or via bit 30 (PCI software adapter reset) in the PLX EEPROM Control & User Bits register. MCLR also clears the other bits in this register.

MCLR Resets all 10115/10117 functions except those contained in the PLX chip.

SWAK Setting this bit on and off simulates the receipt of an ACK pulse from a Centronics compatible device, or a false then true sequence on the Versatec READY line. It can be used to restore synchronization with the attached device after a handshake error, or for diagnostic purposes.

RINT Resets the Interrupt Flag (INTF) in the Interface Status register, removing the interrupt request from the PCI bus. RINT only succeeds if the condition causing the interrupt has been cleared, or the associated mask bit has been reset before RINT is issued.

REST This sends a rest signal to the attached device. The RESET is maintained until REST is cleared to zero.

Interface Status

Bit	Name	Function
07	DIRY	Device and Interface Ready
06	DRDY	Device Ready
05	INTF	Interrupt Flag
04	EMPT	FIFO Empty
03	NHLF	FIFO Not Half Full
02	NFLL	FIFO Not Full
01	PRNT	Print Mode
00	8BIT	8 Bit Data Path

DIRY — This is the master ready bit for the interface. It indicates that the FIFO is empty and the device is ready for another transfer or command. It may be enabled to cause an interrupt. It is not necessary for DIRY to be true before commands or data are written to the FIFO. It is only necessary that there be room in the FIFO.

DRDY — This bit follows the state of the internal device ready flip-flop. It indicates whether the device has become ready following the previous data or command transfer. It is primarily diagnostic in nature.

INTF — INTF sets when an interrupt mask and its associated condition are true. When it sets, an interrupt request is presented to the PLX chip, which will forward it to the PCI bus if the PLX PCI and local interrupt enable bits are set.

INTF stays set after the condition and/or mask are cleared. It will remain set until RINT in the Interface Control register is "pulsed".

EMPT — Indicates that the FIFO is empty, and that a data burst equal to the full depth of the FIFO may be issued. Standard FIFO depth is 16Kbytes, but this can be changed as a factory option. EMPT may be enabled to cause an interrupt.

NHLF — Indicates that the FIFO is less than half full + 1. When this bit is on, it indicates that a data burst equal to half the FIFO depth may be issued. NHLF may be enabled to cause an interrupt. This is probably the interrupt that will be used as a "master" interrupt by a device driver.

NFLL — NFLL is set whenever the FIFO is not full. It indicates that a single transfer may be made to the FIFO. It can be enabled to cause an interrupt.

PRNT — This bit indicates the state of the PRINT output when in Versatec mode. The board can be strapped to default to either print or plot mode. This bit can be used to determine the default strapping, or for diagnostic purposes. The default mode is invoked any time that the interface is reset, either by a bus initialize, a software reset from the PLX chip, master clear (MCLR) in the Interface Control register, or via the board-mounted momentary toggle switch.

8BIT — The 10115/10117 may be configured with an 8 bit internal path as a cost saving option. This will reduce performance in programmed I/O burst mode, but will have little effect on DMA efficiency. When in 8 bit mode, the registers that are actually only 8 bits wide must written as bytes, or multiple bytes will be unintentionally written. This bit may be monitored to determine the data path width.

Device Status

Bit	Name	Function
07	VTTL	0 = Versatec TTL Selected
06	VDIF	0 = Versatec Differential Selected
05	CENT	0 = Centronics Selected
04	VRDY	1 = Versatec READY asserted
03	CBSY	1 = Centronics BUSY asserted
02	PMTY	1 = Device Out of Paper
01	ONLN	1 = Device Selected (Online)
00	CFLT	1 = Centronics FAULT asserted

VTTL VTTL = 0 indicates that Versatec TTL interfacing has been selected. Interfacing mode is selected by an internal jumper cable on the 10115 board. The 10117 does not support the Versatec TTL interface.

VDIF VDIF = 0 indicates that Versatec differential (long lines) interfacing has been selected.

CENT CENT = 0 indicates that Centronics interfacing has been selected.

VRDY VRDY is a 1 whenever the Versatec READY- signal is asserted by the attached device. It is also 1 during the ACK- pulse from a Centronics device. Since the ACK- pulse is brief, this bit will normally be a 0 when a Centronics compatible device is attached.

CBSY CBSY is a 1 whenever an attached Centronics-compatible device is asserting the BUSY signal. BUSY is used in combination with ACK- to determine when the device is ready for data transfer. A Mode register BIT (IBSY) may be used to eliminate busy from the ready equation.

PMTY PMTY will be read as a 1 whenever the attached device is asserting its paper empty signal. This is true for both Versatec and Centronics-compatible devices. PMTY may be enabled to cause an interrupt.

ONLN ONLN = 1 indicates that a Versatec-compatible device is online, or a Centronics-compatible device is asserting the SEL signal. ONLN off may be enabled to cause an interrupt.

CFLT CFLT = 1 whenever a Centronics-compatible device is asserting the FAULT- signal. CFLT may be enabled to cause an interrupt.

Reverse Data

Data presented to the interface by the attached device may be read in this register. Reverse data transfers must be enabled by setting the REVD bit in the Mode register. There is no automatic handshake for reverse data transfers. The driver software must manipulate the output signals using a protocol appropriate for the specific attached device.

Auto LTR Count

The 10115/10117 is capable of automatically sending a Remote Line Terminate command to Versatec compatible devices. This may be useful when a plot smaller than the full paper width is being sent. The Auto LTR Count register is set to the number of bytes in the desired plot size ("count") minus 1, and Auto LTR is enabled using the appropriate command in the Command Out register. Until the Auto LTR mode is cleared, the board will issue a Remote Line Terminate for every "count" number of bytes transferred. This mode of operation can be particularly useful when transferring DMA buffers that are larger than a single plot line. It avoids having to zero-pad the ends of lines, or take an interrupt at the end of each line. This mode is less useful when doing programmed I/O bursts, as the burst size can be matched to the line size, and Remote Line Terminate commands issued after each burst. It is only necessary to enable interrupts when a burst has caused the FIFO to become more than half full.

The Auto LTR Count register is 16 bits wide, written and read as separate bytes at different addresses.

8 Bit Data Out

The 8 Bit Data Out register is used to transfer data to the 10115/10117's FIFO one byte at a time. It may be written as a 32, 16, or 8 bit value. Regardless of write width, four bytes will be written into the FIFO, with all but the low order byte discarded at the "far side". This must be considered when calculating how many transfers will fit into the FIFO. It takes four times as much FIFO space per valid data byte to write through this register than through the 32 Bit Data Out register or Burst Out Range. This register is provided as a way to transfer byte counts that are not multiples of four, and blocks that do not start on a quad byte boundary. It may be useful after a Programmed I/O burst, or before or after a DMA block transfer.

If a write to this register follows a DMA block, it is not necessary to check for space in the FIFO, or enable another interrupt, as the FIFO will be approximately half full, or less, following a DMA block. (DMA transfers are gated by half-full status, but do not necessarily end with the FIFO exactly half full +1. There may be less (or no) data in the FIFO at the end of a DMA transfer, or the FIFO may be half full +2.)

NOTE: If the board is configured for an 8 bit data path, this register must be written as a byte, and only a single byte will be written into the FIFO.

NOTE: If byte swizzling is enabled, the high byte of this register will be transferred to the FIFO.

Burst writes should not be used when writing to this register.

Command Out

The Command out register is used to write 8 bit commands to the FIFO. These are then interpreted at the "far side", and used as internal commands for the 10115/10117, or sent on as plotter commands. Internal commands, and some plotter commands, have no effect on the device ready state, while others set the local copy of device ready to not ready, and wait for the plotter to respond.

Command bytes may be written at any time there is sufficient room in the FIFO, and will flow along with data. Commands such as Versatec remote line terminate (VLTR) may be issued immediately following a DMA block, as the FIFO will be approximately half full (or less). A new DMA block can then be enabled without waiting for another interrupt.

Like the 8 bit Data Out register, the Command Out register also sends four bytes to the FIFO, with the unused bytes later discarded. This must be considered when operating the FIFO near full (not recommended!).

NOTE: If the board is configured for an 8 bit data path, this register must be written as a byte, and only a single byte will be written into the FIFO.

NOTE: If byte swizzling is enabled, the high byte of this register will be transferred to the FIFO.

Burst writes should not be used when writing to this register.

Command Byte	Function
7 6 5 4 3 2 1 0	
0 1 0 0 0 0 0 1	Versatec Clear (VCLR)*
0 1 0 0 0 0 1 0	Versatec Form Feed (VFED)*
0 1 0 0 0 1 0 0	Versatec E-O-T (VEOT)*
0 1 0 0 1 0 0 0	Versatec Line Terminate (VLTR)*
0 0 1 0 0 0 s p	Set Versatec Mode (VMOD)
0 0 0 a 0 0 0 d	Set I/O mode

s = 1 simultaneous print/plot p = 1 plot mode
s,p = 0,0 normal print mode

d = 1 data streaming d = 0 normal handshake

a = 1 Auto LTR a = 0 no Auto LTR

Data streaming mode causes the 10115/10117 to transmit data to the attached plotter as fast as possible, without waiting for a handshake. It is only suitable for devices compatible with this mode.

Auto LTR mode causes a Versatec Remote Line Terminate to be sent to the plotter for every "count" bytes transmitted, where "count" minus 1 has been previously written to the Auto LTR Count register.

* The Versatec commands marked with "*" clear the internal ready flip-flop. This will stay cleared until the plotter responds by toggling the READY line. These commands use the same plotter handshake as data transfers.

32 Bit Data Out

This register is used to transfer data to the FIFO 4 bytes at a time. Bytes will be sent to the attached device low byte first, unless byte swizzling is enabled.

This register must be written as a full 32 bit value, or data corruption will occur. If the board is configured for an 8 bit data path, writes may be 8, 16, or 32 bits wide, and data appropriate for the write width will be sent to the FIFO.

Burst writes should not be used when writing to this register.

PLX Run Time Registers

This range of addresses provides an alternate point of access to the PLX Run Time registers. They may also be accessed in their own address range - given by the base addresses for run time registers in the configuration register set. Accessing the run time registers, including the DMA registers, in the "IKON" space may simplify driver code somewhat, particularly if the 10115/10117 is mapped into low memory, or I/O space. On boards using the 9060 REV 3 chip (10115 REV 1 and 10117 REV 0), this is the only way to access the DMA registers.

These registers are not available in the "IKON" space in the REV 0 release of the 10115.

Local Chain Memory

This address range is provided on REV 1 of the 10115, which supports chaining DMA, but does not allow the DMA logic to access DMA chain "links" in PCI memory space. It is also provided on 10115 REV 2, which allows "links" in either PCI or local memory. In order to do DMA chaining on the 10115 REV 1, the chain "links" must be written into this memory. If chaining is not desired, this address range may be ignored.

The 10115 REV 0 and 3, and the 10117 do not support local memory. The 10115 REV 1 and 2 support local memory.

Burst Out Range

This range of addresses is used when doing burst 32 bit transfers to the 10115/10117 FIFO. Any size burst - up to the 4K maximum range - may be written to the FIFO at one time. Typically, an X86 style string move instruction would be used to burst data into the FIFO at 36Mbytes/second (maximum). The FIFO flags in the Interface Status register can be used to determine if there is room in the FIFO for the desired burst. Typically, transfers will be paced by the half full flag. If the FIFO is less than half full + 1, there will be at least 8Kbytes of available space, and two 4Kbyte bursts could be done without checking again, or taking additional interrupts. After the first (or second) burst, the software can check the half full flag again. If it is set, interrupts can be enabled, and the driver can sleep until FIFO space is available. There will be cases when the half full flag is set, but the driver "knows" approximately how much room is left (at a minimum). In these cases, the driver may choose to do additional bursts, or to send single bytes or commands to the FIFO. In general, it is not necessary, or desirable, to take an interrupt after each transfer.

In systems with caching memory management, it may be necessary to force the cache to drain to real memory (in this case the 10115/10117 FIFO), before doing an additional burst output, or enabling an interrupt. The not half full interrupt could occur immediately, if the cache has not begun to write to FIFO when the interrupt is enabled. If a second burst is made before the cache has emptied, it may overwrite the previous burst in cache.

DMA Operation

The 10115/10117 can operate as a 32 bit bus master. It is capable of reading data from PCI memory at the full specified 132Mbtes/second into the PLX chip's (small) FIFOs, and from there into the on board FIFOs at approximately 36Mbytes/second. The sustainable rate for bursts longer than a few quad words is 36Mbytes/second. The maximum FIFO to device rate is 2Mbytes/second, giving a maximum bus usage of approximately 5%.

NOTE: The 10115 REV 0 does not support DMA. The 10115 REV 1 supports chaining DMA with the chain "links" in local memory. The 10115 REV 2 allows "links" in local or PCI memory. The 10115 REV 3 requires that the "links" be in PCI memory.

The 10117 REV 0 supports DMA but not chaining. The 10117 REV 1 supports chaining with the "links" in PCI memory.

FIFO depth is 16Kbytes in a 32 bit configuration, and 4Kbytes in 8 bit mode. The 8 bit configuration is available as a cost-saving option. The following discussion applies to 32 bit operation.

All DMA transfers are 32 bits. Block lengths which are not exact multiples of 4 are handled by first transferring a DMA block which is an even multiple of 4, and then using the 8 bit Data Out register to complete the transfer. There may also be cases that require transferring single bytes before the DMA block. When a DMA block completes, the FIFO will be approximately half full - at a maximum - it may contain less data, and may even be empty, depending on block size and plotter speed. There will always be room for additional data or command transfers, without testing the FIFO for available space, or waiting for another interrupt. The "extra" data bytes necessary to complete the block, or any necessary command bytes, should simply be written to the 8 bit Data Out or Command Out register without waiting. Another DMA block may be enabled immediately. This approach requires only one interrupt per DMA block, including any "odd" bytes and commands.

NOTE: The 10115 was designed to use DMA channel 0 of the PLX 9060. Because of some early bugs in the PLX chip (which did not affect Tahoma Technology's designs) some PLX customers elected to use DMA channel 1 for single DMA channel designs. Because of this, PLX decided to use DMA channel 1 as the sole channel in the 9060 SD chip. At Tahoma Technology's urging, PLX has agreed to design the 9060 SD in such a way that it will respond to register accesses to DMA 0 and DMA 1. If PLX succeeds in this design effort, code written for the 9060 REV 3 chip should run unmodified on the 9060 SD, even though it accesses DMA channel 0.

Later 10115 artwork - board REV C and later - and all artwork revisions of the 10117 allow using either DMA channel when the 9060 REV 3 or 9080 REV 3 part is installed. Earlier 10115 board artwork versions do not connect DMA channel 1 to the Tahoma Technology logic.

DMA transfers are initiated by first setting up and enabling the DMA (and interrupt) control logic in the PLX chip. Non chaining DMA is done by first selecting non chaining mode in the appropriate DMA Mode register. The DMA Mode register must also be configured for a 32 bit local bus width (or 8 bit if configured for 8 bit operation), one internal wait state, disable Ready input, disable BTERM input, Burst enable, (non chaining), (Done Interrupt enable), constant local addressing mode, and Demand mode. The hex value to load into

the DMA Mode register for non chaining DMA with an interrupt when done is 0x1D07. Then load the PCI buffer address into the PCI Address register, the buffer size into the Transfer Size register, setting the Local Address register to point at the Burst Out Range (local address 0x1000), setting the direction in the Descriptor Pointer register, and then setting the Enable and Go bits in the DMA Command/Status register. It may also be appropriate to enable DMA and local interrupts in the Interrupt Control/Status register. DMA will run to completion, and generate an interrupt if enabled.

Chaining mode (with interrupt) is enabled by setting the appropriate DMA Mode register to 0x1F07. The descriptor pointer register should be loaded with the address of the first chaining "link", the PCI/local descriptor select bit, the transfer direction, and the intermediate interrupt enable, if desired (probably not). DMA is initiated by setting the Enable and Go bits in the appropriate DMA Command/Status register. The chaining "links" (descriptors) contain entries for PCI buffer address, buffer size, local address, and a pointer at the next "link". The pointer also contains an end of chain bit, an intermediate interrupt bit, and a transfer direction bit.

Once Chaining DMA has begun, it will transfer the buffers listed in the chaining "links" until it completes a "link" that has its end of chain bit set. At this time a Done interrupt will occur, if enabled.

Tahoma Burst Mode

Tahoma Technology has added an enhancement to the VPI handshake that allows data transfers to the plotter at up to 5Mbytes/second (4Mbytes/second in the current 10115/10117 implementation). This mode of operation also eliminates the effect of cable length on transfer rate.

Tahoma Burst Mode is only usable with VPI compatible plotters supporting this mode of operation. As of this writing, Tahoma Burst Mode is only used internally in some plot server/plotter combination systems. This mode is encouraged for new plotter development.

Tahoma Burst Mode refers to bursts of data from the board to the plotter. The term "burst" is used elsewhere in this manual to describe burst writes of data from the PCI bus to the board, and from the PLX bus interface chip to the board's FIFOs.

In a normal VPI handshake, the board waits for a ready indication from the plotter (-READY low), then stabilizes a data byte on the data lines, waits the required time, and pulses the data strobe (PICLK) for a minimum of 300ns. When the plotter has captured the data, it de-asserts READY. The board then waits for READY to assert again, indicating that another byte may be sent. The Pulsed Command handshake is similar, with the board issuing discrete pulses, rather than the data strobe. The maximum transfer rate using this handshake is approximately 2Mbytes/second, with 1Mbyte/second typical, depending on the plotter's capabilities.

Tahoma Burst Mode eliminates the wait for READY to cycle between bytes. In this mode, when the board sees plotter ready, it is allowed to burst up to 128 bytes (actually 128 plus one or two) without checking again for ready. The data strobe may be as short as 100ns, with 100ns between strobes. Data is stable 50ns prior to, and 50ns after, the data strobe. Following a burst, the board again monitors ready. The plotter uses a FIFO at its input. Ready is asserted as long as the FIFO has room for at least 256 bytes. When the plotter's input FIFO has less than 256 bytes of remaining room, it de-asserts READY. The 10115/10117 will pause data transmission within approximately 128 bytes (depending on cable length, there may actually be one or two bytes "in flight" in the cable - so more than 128 additional bytes may be sent following the de-assertion of READY). Pulsed Commands may also be issued using this handshake. Eliminating per-byte handshaking allows the transfer rate to approach 5Mbytes/second. Cable delays have only minimal effect on transfer rate, since ready is monitored once for 128 bytes rather than for each byte transferred. -READY is not required to transition (handshake) for each burst. The board is allowed to start a burst whenever -READY is asserted.

Enable burst mode by setting BRST in the Mode Register.

Tahoma Burst Mode offers considerable advantage over the traditional VPI handshake mode. It is possible to implement a plotter input interface that operates transparently with both bursting and non-bursting sources. Consult Tahoma Technology for details and design assistance.

Interrupts

The 10115/10117 is capable of generating an interrupt request on PCI interrupt line A. This interrupt will be translated to an ISA style interrupt selected by CMOS configuration. (On a typical X86 PCI system).

Interrupts are enabled by first setting up the system's interrupt logic -- see system documentation -- setting the PCI and local interrupt enable bits in the PLX Interrupt Control/Status register, and setting the appropriate mask bit in the Interrupt Mask register. Interrupts will occur whenever the mask and its associated condition register are true. The "and" of the mask and the condition will set the INTF flag bit, which will cause the local interrupt to be asserted. The interrupt code should clear the mask and/or remove the condition that caused the interrupt, and then "pulse" RINT to reset INTF.

NOTE: The 10115/10117 interrupts are level sensitive, not edge sensitive - like some earlier Tahoma Technology products. Setting a mask when the associated condition is true will cause an immediate interrupt. It is intended that an interrupt mask be set after issuing the command that will cause the interrupt. (DMA interrupts are enabled in the PLX chip before starting DMA). It may save machine overhead to check the condition before enabling interrupts, as in some cases, the command may complete before interrupts could be enabled. Enabling interrupts without checking will not cause a problem; there will be an (appropriate) immediate interrupt in the case where the command (or data transfer) completes quickly.

The DMA done interrupt is probably the most useful interrupt when high speed data transfer is desired. The DMA logic will attempt to keep the FIFO half full. The FIFO depth is 16Kbytes (in 32 bit mode). There will (probably) be data still in the FIFO when the DMA done interrupt occurs, giving time for another DMA block to be started before the FIFO empties. This will allow continuous transfer to the attached device.

When the DMA done interrupt occurs, there will still be room in the FIFO for data or commands. The "extra" bytes necessary to complete a block that was not an even multiple of four can be written immediately, without checking FIFO status. Similarly, commands may be issued without checking the FIFO, or incurring the overhead of additional interrupts.

It is allowable, and appropriate, to start another DMA block immediately after writing "extra" data bytes or commands that followed completion of an earlier DMA block. The DMA logic will not write data to the FIFO until it is less than half full. There is no danger of a data overrun. In this way, it is only necessary to incur one interrupt per DMA block, even if additional data bytes or commands are necessary between blocks.

Device Exerciser

The 10115 contains features which allow simple tests of the attached printer/plotter without software or CPU intervention. The 10117 does not have this capability. Board edge mounted toggle switches allow sending a repeating ASCII message to devices with character generators, or 0xFF bytes (all ones) in Versatec plot mode, which produces all black paper on a monochrome (or compatible) plotter.

The print/plot mode of the test is determined by a the toggle switch labeled "DEFAULT MODE". If the switch is in the "PLOT" position or in the center position, the default and test modes will be plot. This sets the interface to plot mode - if Versatec interfacing is selected - and attempts to send all ones to the device. If the switch is in the "PRINT" position, the mode will be print, and a repeating "ikonikonikon...." will be sent to the device.

Moving the momentary toggle switch labeled "RESET THEN TEST" to the "RESET" position causes the mode selected by the "DEFAULT MODE" switch to be entered into the boards logic. The "DEFAULT MODE" switch has no effect until the board is reset, either by the system, or software, or by the toggle switch. Moving the momentary switch to the "TEST" position causes the 10115 to begin issuing test data to the attached device. When in test mode, the slowest handshake speed is automatically selected. Test mode can be terminated by the toggle switch or by the host system.

It is important that the "DEFAULT MODE" switch be left in the appropriate position when the test is completed. The momentary switch should be toggled to "RESET" once this has been done.

Hardware Options

With the exception of default print/plot mode, and the selection of interface type (Versatec TTL or Differential, or Centronics) all hardware options on the 10115/10117 are selected by register bits. These bits are set by a combination of BIOS and the device driver.

Board Address Selection

The location of the 10115/10117 register sets in the I/O and memory address space of the host is determined by the BIOS. BIOS may locate the PLX portion of the register sets in I/O and/or memory space. The "IKON" portion of the register set may be located in I/O, low memory, or high memory space, as determined by the EEPROM installed at the factory. The driver software determines the locations of the registers by calling BIOS with requests to locate the board, and read its configuration registers.

Interrupt Selection

The 10115/10117 always drives PCI interrupt A. This is translated into an ISA style interrupt level as determined by CMOS configuration.

Device Interface Options

Interface Selection

The interface type used by the 10115 may be Versatec TTL, Versatec Differential, or Centronics (IEEE1284). The 10117 supports Versatec Differential and Centronics interfaces. Interface type is determined by the position of the jumper cable on the board. One end of the cable should be connected to the P1 connector ("I/O OUT"). The other end should be connected to P2 for Versatec TTL ("V-TTL"), P3 for Centronics ("CENT"), or P4 for Versatec Differential ("V-DIFF"). Most Versatec compatible devices use Differential interfacing, but there are a few that require a TTL interface. Consult printer/plotter documentation for the appropriate configuration.

Note: the board's interface selection ribbon cable MUST be configured to match the attached printer/plotter BEFORE installing the board.

Default Mode

The 10115/10117 may be configured to power up in either print or plot mode. The default mode of the 10115 is determined by the setting of the toggle switch labeled "DEFAULT MODE". When the board is reset, either by the host system or software, or by the "RESET" position of the other toggle switch, the default mode is entered into the board's logic.

There is also a 3-pin jumper provided for use on the 10117 or on low-cost 10115 boards that do not include the toggle switches. A jumper block may be positioned in either "PRINT" or "PLOT" on the 10115 to control the default mode of the board. On the 10117, installing the jumper block between pins 1 and 2 - the closest to the serial number, and the top of the board - causes the 10117 to default to plot mode. Installing the jumper between pins 2 and 3 - closest to the bottom of the board - selects print mode as the default.

NOTE: On the 10117, the jumper block MUST be installed (in either position). On the 10115, it will only be used if the toggle switches are not installed.

The print/plot mode only affects the 10115/10117 when Versatec type interfacing is selected. Typically, the device driver will override the jumper selected default mode once the software is started.

Timing

Timing of the handshake with the device is selected by setting the TSLx in the Mode register.

Four timing selections are available in each mode: Centronics, Versatec, and 4-edge.

Centronics:

Choice	Data Set-up	Strobe Width	Data Hold
#0	150ns	200ns	150ns
#1	250ns	300ns	200ns
#2	450ns	500ns	300ns
#3	550ns	800ns	450ns

Versatec:

Choice	Data Set-up	Strobe Width	Data Hold
#0	150ns	200ns	150ns*
#1	250ns	300ns	150ns*
#2	350ns	500ns	150ns*
#3	550ns	800ns	150ns*

* Minimum hold time - data held until plotter ready for next transfer.

4-edge:

Choice	Busy on to Strobe off	Busy off to Strobe on	Data to Strobe on	Busy off to Data change
#0	300ns	300ns	150ns	150ns
#1	400ns	400ns	250ns	150ns
#2	500ns	500ns	350ns	150ns
#3	700ns	700ns	550ns	550ns

In 4-edge mode, the strobe is held on until Busy is asserted, and held off until Busy is removed. Strobe length is dependent on Busy from the attached device.

Pulses to attached device - Versatec Clear, Remote Line Terminate, Remote Form Feed, Remote EOT- will be the same width as strobe.

Versatec Reset, and Centronics Input Prime, are driven by a latch - pulse length is determined by software.

BUSY

BUSY is normally used along with -ACK as part of the handshake response from a Centronics compatible device. If it is not used, it is normally pulled low (false) by the device. For the rare device that does not drive BUSY, and lets it float (true), a bit in the Mode register may be used to remove BUSY from the device ready equation. If IBSY is zero, BUSY will be used as part of the device ready determination. If IBSY is a one, BUSY will not be used. This bit has no effect when Versatec interfacing is selected.

Some Centronics compatible devices can take advantage of higher transfer rates if BUSY is used as the handshake signal, rather than -ACK. The 10115/10117 normally waits for BUSY to be de-asserted, and -ACK to complete before issuing the next data strobe. If a device always asserts BUSY after each data strobe, and ends BUSY before the end of its -ACK pulse, setting UBSY in the Mode register to one will provide a shorter handshake time. If the device does not always toggle BUSY after each strobe, setting UBSY will cause the handshake to fail. This bit has no effect when Versatec mode is selected.

Byte Swizzle

The 10115/10117 normally outputs the low byte of a quad byte value first. Setting the SWIZ bit in the Mode register will cause the high byte to be sent first. This also requires that the high byte be used in the 8 Bit Data Out and Command Out registers. Byte swizzle is not available when configured for an 8 bit data path.

Boards using the PLX 9060SD or 9080 chip may use the Big/Little Endian register in that chip to control byte order for slave access to the IKON registers and FIFO. The 9080 includes Big/Little Endian control of DMA transfers.

4 Edge Handshake

This mode allows for ECP style handshakes with the attached device. Four edge mode may be the fastest transfer mode for some devices, although normal handshaking is actually faster on the 10115/10117, if the device can take advantage of it. In 4 Edge mode, the device responds to the presence of the data strobe by asserting BUSY. The 10115/10117 responds by de-asserting the strobe. The device responds by de-asserting BUSY. The 10115/10117 may then assert strobe.

4 Edge mode is enabled by setting 4EDG in the Mode register. UBSY must also be set for ECP compatibility, otherwise, the 10115/10117 will perform 4 Edge handshakes using the data strobe and ACK.

Termination Selection

The termination resistor installed in the socket labeled TERM must agree with the interfacing mode selected, and with the device to which the interface is attached. The 10115/10117 is normally shipped with a 220/330 ohm network installed which is appropriate for most Centronics applications, and ALL Versatec applications, whether TTL or differential. For Centronics-type devices that use gates, rather than drivers to drive status, and handshake lines, it may be necessary to use a 470 ohm network.

A spare termination resistor network (470 ohms) is installed in the socket labeled "SPARE TERM" on the 10115. The 10117 does not have a spare termination socket. The extra resistor network is included in the 10117's shipping container.

IEEE1284 compliant devices will usually require the 470 ohm resistor network. If an inappropriate network is installed the usual symptom is no output, and device driver timeout. In some cases, using the inappropriate network causes intermittent operation.

Device Cabling

All interfacing modes are supported through a single 37-pin "D" subminiature connector on the rear panel of the interface. This connector may be used directly for Versatec applications (Versatec uses a 37-pin "D"). For Centronics applications, a special cable must be used which has a 37-pin "D" on one end and a 36-pin "delta ribbon" connector on the other. This cable may be used to go directly to the device (normally a "male" cable connector), or may be used with a "female" cable connector as an adapter to another manufacturer's "delta ribbon" to "delta ribbon" cable. It is also possible to fabricate other cable/connector combinations to match other existing cabling schemes.

Versatec Cable Run-List

Signal Name - TTL	Signal Name - Differential	37-pin "D"
IN01	INO1+D	1
GND	INO1-D	20
INO2	INO2+D	2
GND	INO2-D	21
IN03	INO3+D	3
GND	IN03-D	22
IN04	INO4+D	4
GND	IN04-D	23
IN05	IN05+D	5
GND	IN05-D	24
IN06	IN06+D	6
GND	IN06-D	25
IN07	IN07+D	7
GND	IN07-D	26
IN08	IN08+D	8
GND	IN08-D	27
CLEAR-	CLEAR-D	9
GND	CLEAR+D	28
PICLK	PICLK+D	10
GND	PICLK-D	29
READY-	READY-D	11
GND	READY+D	30
PRINT	PRINT+D	12
GND	PRINT-D	31
PARALLEL	N/C	13
ONLIN-	INOP-D	32
SPP-	SPP-D	14
GND	SPP+D	33
RESET-	RESET-D	15
GND	RESET+D	34
RFFED-	RFFED-D	16
GND	RFFED+D	35
REOTR-	REOTR-D	17
GND	REOTR+D	36
RLTER-	RLTER-D	18
GND	RLTER+D	37
NOPAP	INOP+D	19

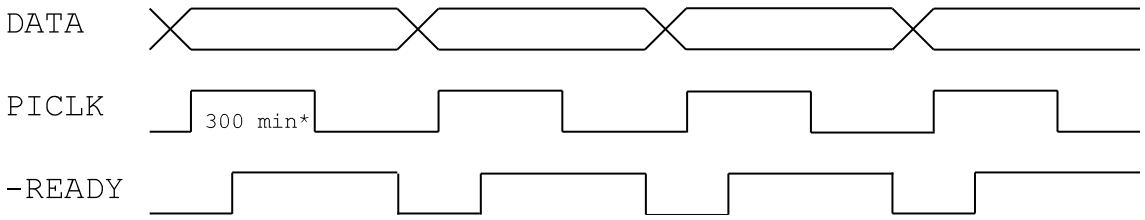
Centronics Cable Run-List

Signal Name	37-pin "D"	Delta Ribbon
STROBE-	1	1
GND	20	19
DATA1	2	2
GND	21	20
DATA2	3	3
GND	22	21
DATA3	4	4
GND	23	22
DATA4	5	5
GND	24	23
DATA5	6	6
GND	25	24
DATA6	7	7
GND	26	25
DATA7	8	8
GND	27	26
DATA8	9	9
GND	28	27
ACK-	10	10
GND	29	28
BUSY	11	11
GND	30	29
PE	12	12
GND	31	30
SELECT	13	13
INPUT PRIME-	32	31
AUTO FEED-	14	14
FAULT-	33	32
SELECT IN-	37	36

Handshake Timing Diagrams

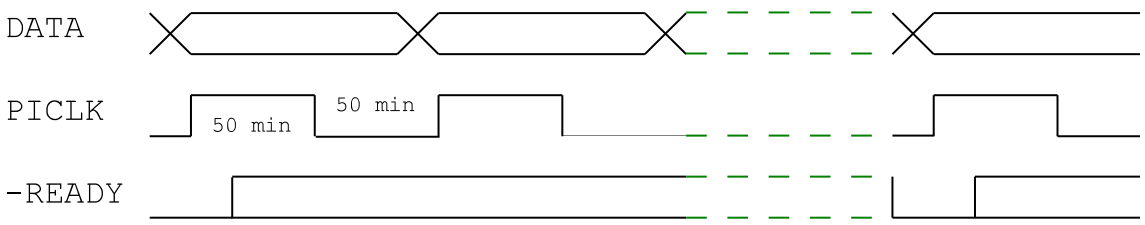
Timing diagrams shown below are typical for the data strobe (PICLK) and ready (-READY) handshake. The pulsed command (REMOTE EOT, etc) and ready handshake is similar - but the pulsed commands are negative going pulses. See manual text for handshake details.

Normal VPI Handshake



* 300ns is the VPI specification – 200ns timing selection available for plotters that support faster handshakes

Tahoma Burst Handshake



-READY examined before each burst. Burst length 1-128 bytes (possibly + 1 or 2 additional bytes)