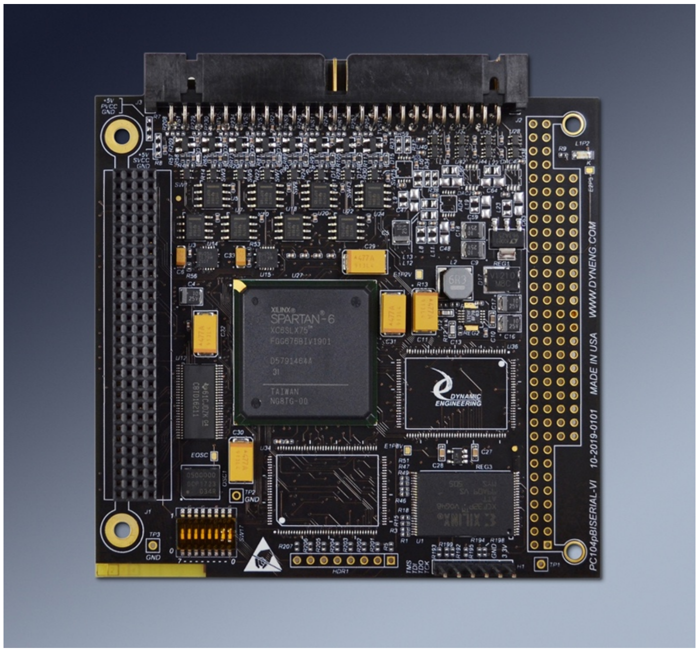
DYNAMIC ENGINEERING
150 DuBois St. Suite C, Calif. 95060
831-457-8891
sales@dyneng.com
www.dyneng.com
Est. 1988

User Manual

PC104p-BiSerial-VI-BA14

WRA / RCB Transmitter / Receiver
8 TX redundant
8 Inputs – software selectable
PCI-104 Module



Revision 01P1
Corresponding Hardware: Revision 01
FLASH revision 5
10-2019-01

PC104p-BiSerial-VI-BA14
WRA / RCB Interface
PCI-104 Module

This document contains information of proprietary interest to Dynamic Engineering. It has been supplied in confidence and the recipient, by accepting this material, agrees that the subject matter will not be copied or reproduced, in whole or in part, nor its contents revealed in any manner or to any person except to meet the purpose for which it was delivered.

Dynamic Engineering has made every effort to ensure that this manual is accurate and complete. Still, the company reserves the right to make improvements or changes in the product described in this document at any time and without notice. Furthermore, Dynamic Engineering assumes no liability arising out of the application or use of the device described herein.

The electronic equipment described herein generates, uses, and can radiate radio frequency energy. Operation of this equipment in a residential area is likely to cause radio interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Dynamic Engineering's products are not authorized for use as critical components in life support devices or systems without the express written approval of the president of Dynamic Engineering.

This product has been designed to operate with PC/104p Module stacks and compatible user-provided equipment. Connection of incompatible hardware is likely to cause serious damage.

Revised 4/11/19.

Table of Contents

List of Figures

Product Description

PC104p-BiSerial-VI-BA14 is part of the PC/104p Module family of modular I/O components by Dynamic Engineering. PC104p-BiSerial-VI is capable of providing multiple protocols. The BA14 protocol implemented provides one serial Input and Output each with 8 IO to select. The serial format is used with WRA / RCB equipment. Start bit, CMD/Data bit, MSB first, 16 bit words, Stop bit. In addition, there are 4 marking bits between words [minimum] and 21 bits between messages.

PC104p-BiSerial-VI is an update to the PC104p-BiSerial-VI base design featuring Spartan-VI 75 Industrial temperature FPGA and improved DAC and ADC devices. Increased memory within the FPGA will allow for more options in terms of buffering, programmed update rates etc. TVS devices are attached to the IO to provide 400W of over voltage protection. Series resistors are available – usually installed with 0 ohms – to provide edge rate control should that be needed. Pull-up and Pull-down devices [not normally installed] are available for half duplex situations where the terminations are not in the cable.

PC104p-BiSerial-VI-BA14 is ported from PC104p-BiSerial-VI-BA14 with some updates. 8Kx32 internal FIFO's are an update from the 2K deep versions used in the "VI".

The memory is supported with DMA to allow for streaming data from the system into host memory and from the host memory to the system. The ports are full duplex.

The BA14 version is minimized and does not include the ADC, DAC, PLL, TTL or external FIFO's.

Other custom interfaces are available. We will redesign the state machines and create a custom interface protocol. That protocol will then be offered as a "standard" special order product. Please see our web page for current protocols offered. Please contact Dynamic Engineering with your custom application.

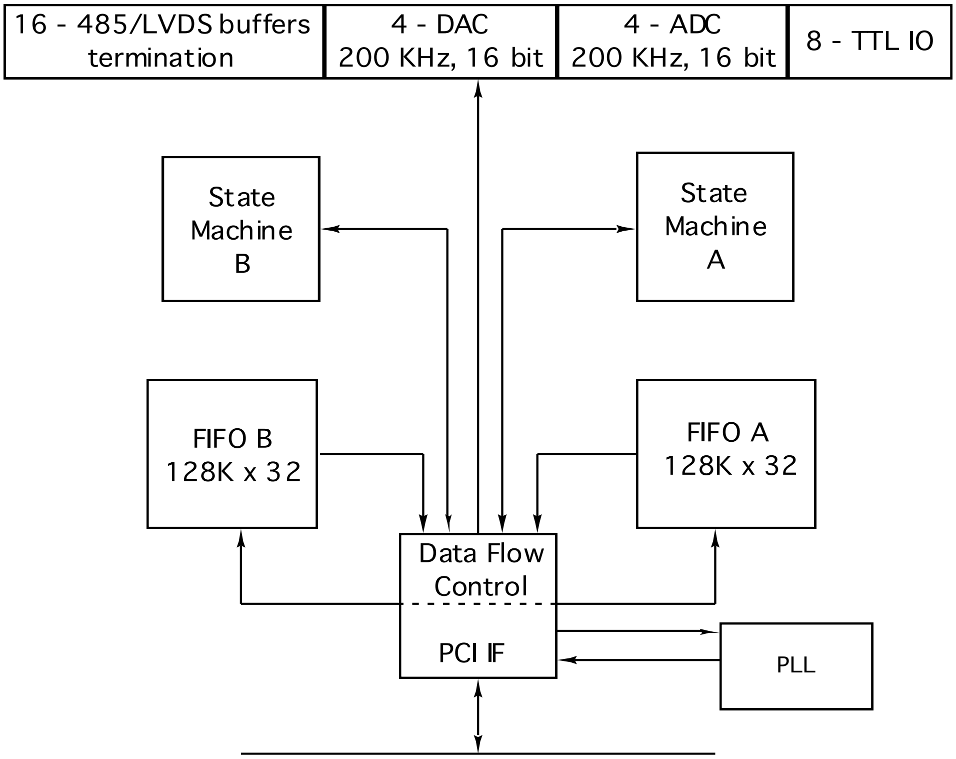


FIGURE 1 PC104P-BISERIAL–VI BLOCK DIAGRAM

The configuration shown in Figure one makes use of two external [to the FPGA] FIFO's. The external FIFO's are 128K deep x 32 bits wide. Some designs do not require as much memory, and are more efficiently implemented using the internal FIFO's. Internal FIFO's can be configured using the block RAM within the Xilinx.

Sixteen differential I/O are provided for the serial/parallel signals. The drivers and receivers conform to the LVDS or RS-485 specification (exceeds RS-422 specification). The LVDS or RS-485 input signals are selectively terminated with 100Ω. The termination resistors are in discrete packages to allow flexible termination options for custom formats and protocols. Optional pullup/pulldown resistors can also be installed to provide a logic '1' on undriven lines. The routes are matched length and impedance controlled.

The terminations and transceivers are programmable through the Spartan VI device to provide the proper mix of outputs, inputs and terminations needed for a specific protocol implementation. BA14 Serial interface uses 16 differential IO. 0-7 are allocated to the 8 input signals and 8-15 are used for the transmit port. All 16 can be programmed to be Parallel data or the WRA /RCB function via the source control register. The terminations are programmable for all IO.

All configuration registers support read and write operations for maximum software convenience, and all addresses are long word aligned.

PC104p-BiSerial-VI conforms to the PC/104p standard. This guarantees compatibility with multiple PC/104p boards. Because the PC/104p may be mounted on different form factors, while maintaining plug and software compatibility, system prototyping may be done on one Carrier board, with final system implementation on a different one. For example, PCI2PC104p – PCI carrier for PCI-104 can be used for development in a conventional PC. Later the hardware and software can be ported to the system.
http://www.dyneng.com/pci2pc104p.html

The serial format for transmit and receive is specific to the system used with the BA14 design. The electrical format is RS422. The data is similar in format to UART. The start bit is active low, the stop bit is active high, the marking state is high. The data is sent MSB first.

The hardware for the receiver tests for 21+ 1's then waits for the start bit. The data is sampled with a 50 MHz. clock. Data has a base rate of 6.25 MHz [8 samples per period]. The hardware counts from the approximate center of the start bit and saves the next 17 bits to the FIFO storage. At the end of the word, the marking state is tested to determine if an inter-word or inter-message gap has been detected. If inter-word the hardware scans for the next start bit and appends the data to the message. If an inter-message gap is detected the interrupt request is set and the hardware scans for the start of the next message.

The transmitter uses a reference oscillator to operate. The oscillator value can be changed or the divisor used to provide alternate frequencies. The onboard PLL is not used for the BA14. The PLL could be implemented to provide more programmable transmit frequencies and or a new reference rate for the receiver if higher frequencies are desired.

The Transmitter sends the data stored within the FIFO. The data is transmitted until the FIFO is empty. An interrupt can be generated when the transmission is completed. The Transmitter enable bit is auto-cleared when the transmission is completed with a mask to allow this feature to be disabled. If disabled then the transmitter will send the next data as soon as data becomes available within the FIFO. In all cases a new message will have at least 21 1's pre-pended to the data. The transmitter starts tracking the marking state as soon as the hardware is powered up to allow for a quick start with a new message. When the FIFO goes empty the marking state is also tracked to allow for a shorter inter-message gap when data is loaded to send again.

In a loop-back test, the receiver will need to be enabled with the transmitter connected long enough to provide the 21 1's.

Interrupts are supported by the PC104p-BiSerial-VI-BA14. An interrupt can be configured to occur at the end of a transmitted message. An interrupt can be set at the end of a reception. All interrupts are individually maskable and a master interrupt enable is also provided to disable all interrupts simultaneously.

The BA14 design is implemented with the idea of offloading the CPU as much as possible. One channel of receive and one channel of transmit DMA [scatter gather capable] are implemented with large buffer capabilities. Link lists can be used to reduce the amount of CPU interaction required to capture large data streams.

The base specification allows for messages to be up to 32 words long. The messages are only limited by the system's ability to adsorb or transmit the DMA data. Smaller messages can be transferred within the specification limits or longer ones as desired.

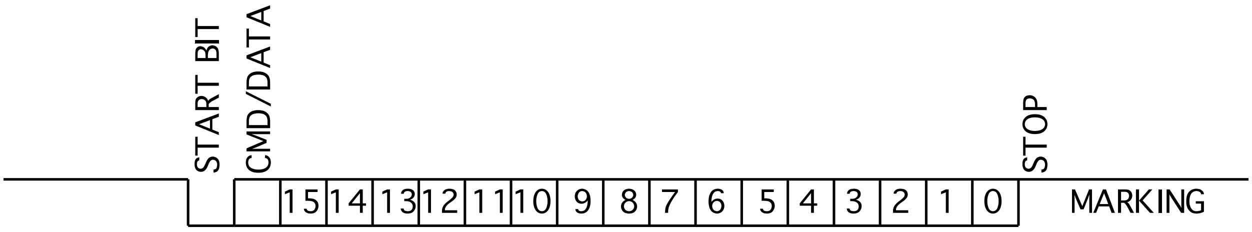


FIGURE 2 PC104P-BISERIAL–VI-BA14 TIMING

Theory of Operation

The PC104p-BiSerial-VI-BA14 is designed to transfer data from one point to another with a simple serial protocol.

PC104p-BiSerial-VI-BA14 features a Xilinx Spartan VI FPGA. The FPGA contains all of the registers and protocol controlling elements of the BA14 implementation. Only the transceivers, and switches are external to the Xilinx device in this application.

The PCI interface to the host CPU is controlled by a logic block within the Xilinx. PC104p-BiSerial-VI is capable of supporting 40 MBytes per second into and out of the FIFO's with single word reads and writes.

In reality in most systems the transfer rate will be 2-4 LW/uS for approximately 12Mbytes per second. With DMA the data is transferred at 33 MHz. with 32 bit words for a transfer rate of 132 Mbytes/sec when active. With most PC systems there is enough system overhead to reduce the effective transfer rate to approximately 50% of the maximum or 66 Mbytes/sec.

With 2 channels, DMA, and 66 Mbytes/sec available there are 33 Mbytes/sec [8.25 MLW/Sec] available per channel on the BA14. With 6.25 MHz. data received or transmitted and converted to 17 [32 with overhead] bit words the effective receive or transmit rate is 272 K LW/sec. Plenty of margin for the data transfer ⇔ system memory.

The DMA channels are independent allowing the data streams to be independently written/Read to/from separate buffers in system memory. With BA14 each port and direction includes a separate DMA engine allowing multiple transfers to happen “in parallel” without CPU intervention.

Only the transmitter and receiver on port 0 are implemented due to IO mux requirements 1:8 and 8:1. If your system can use the BA14 function and would prefer a PMC, we can port this design to PMC-BiSerial-VI and provide 2 transmitters and 2 receivers with the same 8 IO expansion/reduction on each channel. The PMC version has 32+ transceivers available.
http://www.dyneng.com/pmc_biserial_VI.html

PC104p-BiSerial-VI can support many protocols. PC104p-BiSerial-VI–BA14 uses a serial “UART-like” protocol.

State machines within the FPGA control all transfers between the internal registers and FPGA logic, and the FPGA and the data buffers. The TX state machine reads from the transmit data register and loads the shift register before sending the data. The RX state machine receives data from the data buffers, and takes care of moving data from the shift register into the RX storage.

Data is read from the TX channel FIFO and loaded into the shift register. When the data is loaded the Start, Stop and mandatory marking bits are also loaded. The Start bit is immediately visible on the IO with the CMD/DATA following next. One bit period later the data is transitioned to the next value. This process repeats until the first word is transferred. If more data is available, the process repeats for the second word. Etc. It is a good idea to preload the FIFO with the complete message prior to enabling to insure proper operation. Please refer to the register bit definitions for more details.

The data rate is set by an external oscillator for TX and by the system for RX. The 485 transceivers are rated for 50 MBPS. The interface is synchronous. The external system may or may not be operating when the BA14 is enabled to capture data. The external signals are sampled and operated on by a state-machine operating at the oscillator clock rate [50 MHz].

The receive shift register coupled with the receive state-machine to capture the data. The data signal is synchronized and then sampled. When the channel receive state-machine is enabled it searches for the first valid transition from high to low. To be valid the data must have been in the marking state for at least 21 bit periods to insure that a new message is being captured. After the transition the shift register captures data for the next 17 bits. The data is asynchronous; a counter is used to track the center of the bit periods as determined by the start bit. After each group of 17 bits, the data is captured into a storage register, and the FIFO interface signaled that there is data available to store into the FIFO. The data is extended [upper bits] with “0” to be 32 bits wide.

The receiver stays enabled until software disables the function.

The transmitter is automatically disabled at the end of a message to allow for re-loading of a contiguous message before re-starting. The transmitter marks when “off” and keeps track of the off time to allow a faster re-start in the case where some, or perhaps all of the 21 bit times have happened before enabling a new message.

Address Map

```
PC104P_BISVI_BASE            0x0000   // 0 base control register offset
PC104P_BISVI_ID_SW           0x0004   // 1 ID & Switch Register offset
PC104P_BISVI_STATUS          0x0008   // 2 base Status register
PC104P_BISVI_DIR_TERM        0x000C   // 3 direction and termination register

PC104P_BISVI_PARDAT_485      0x0010   // 4  parallel 485 data IO DATA
PC104P_BISVI_PARDAT_485_RDBK 0x0014 // 5  parallel 485 data reg read-back
PC104P_BISVI_PARCNTL         0x0018   // 6  parallel data control register
PC104P_BISVI_TTL_IO          0x001C   // 7  TTL Data out and TTL Data in

PC104P_BISVI_TTL_RDBK        0x0020   // 8  TTL Data out register read-back
PC104P_BISVI_FIFO_A          0x0024   // 9  External FIFO A write,read port
PC104P_BISVI_FIFO_B          0x0028   // 10  External FIFO B write,read port
PC104P_BISVI_DAC0            0x002C   // 11  DAC 0 data port

PC104P_BISVI_DAC1            0x0030   // 12  DAC 1 data port
PC104P_BISVI_DAC2            0x0034   // 13  DAC 2 data port
PC104P_BISVI_DAC3            0x0038   // 14  DAC 3 data port
PC104P_BISVI_ADC0            0x003C   // 15  ADC 0 CMD/data port

PC104P_BISVI_ADC1            0x0040   // 16  ADC 1 CMD/data port
PC104P_BISVI_ADC2            0x0044   // 17  ADC 2 CMD/data port
PC104P_BISVI_ADC3            0x0048   // 18  ADC 3 CMD/data port
PC104P_BISVI_ASTAT           0x004C   // 19  DAC/ADC Status port

PC104P_BISVI_base0           0x0078   // 30  base control for channel 0
PC104P_BISVI_int0            0x007C   // 31  INT latch and status 0
PC104P_BISVI_burstin0        0x0080   // 32  burst in 0
PC104P_BISVI_burstout0       0x0084   // 33  burst out 0
PC104P_BISVI_fiforw0         0x0088   // 34  single read / write 0
PC104P_BISVI_txamtc0         0x008C   // 35  TX Almost Empty Count 0
PC104P_BISVI_rxafc0          0x0090   // 36  RX Almost Full Count 0
PC104P_BISVI_txfifowc0       0x0094   // 37  TX FIFO Word Count 0
PC104P_BISVI_rxfifowc0       0x0098   // 38  RX FIFO Word Count 0
PC104P_BISVI_spare0          0x009C   // 39  Spare 0

PC104P_BISVI_base1           0x00A0   // 40  base control for channel 1
PC104P_BISVI_int1            0x00A4   // 41  INT latch and status 1
PC104P_BISVI_burstin1        0x00A8   // 42  burst in 1
PC104P_BISVI_burstout1       0x00AC   // 43  burst out 1
PC104P_BISVI_fiforw1         0x00B0   // 44  single read / write 1
PC104P_BISVI_txamtc1         0x00B4   // 45  TX Almost Empty Count 1
PC104P_BISVI_rxafc1          0x00B8   // 46  RX Almost Full Count 1
PC104P_BISVI_txfifowc1       0x00BC   // 47  TX FIFO Word Count 1
PC104P_BISVI_rxfifowc1       0x00C0   // 48  RX FIFO Word Count 1
PC104P_BISVI_spare1          0x00C4   // 49  Spare 1
```

PC104P_BISVI_base2	0x00C8	// 50 base control for channel 2
PC104P_BISVI_int2	0x00CC	// 51 INT latch and status 2
PC104P_BISVI_burstin2	0x00D0	// 52 burst in 2
PC104P_BISVI_burstout2	0x00D4	// 53 burst out 2
PC104P_BISVI_fiforw2	0x00D8	// 54 single read / write 2
PC104P_BISVI_txamtc2	0x00DC	// 55 TX Almost Empty Count 2
PC104P_BISVI_rxafc2	0x00E0	// 56 RX Almost Full Count 2
PC104P_BISVI_txfifowc2	0x00E4	// 57 TX FIFO Word Count 2
PC104P_BISVI_rxfifowc2	0x00E8	// 58 RX FIFO Word Count 2
PC104P_BISVI_spare2	0x00EC	// 59 Spare 2
PC104P_BISVI_base3	0x00F0	// 60 base control for channel 3
PC104P_BISVI_int3	0x00F4	// 61 INT latch and status 3
PC104P_BISVI_burstin3	0x00F8	// 62 burst in 3
PC104P_BISVI_burstout3	0x00FC	// 63 burst out 3
PC104P_BISVI_fiforw3	0x00100	// 64 single read / write 3
PC104P_BISVI_txamtc3	0x00104	// 65 TX Almost Empty Count 3
PC104P_BISVI_rxafc3	0x00108	// 66 RX Almost Full Count 3
PC104P_BISVI_txfifowc3	0x0010C	// 67 TX FIFO Word Count 3
PC104P_BISVI_rxfifowc3	0x00110	// 68 RX FIFO Word Count 3
PC104P_BISVI_spare3	0x00114	// 69 Spare 3

//FIFO_TX0_SIZE 1 // n x 32 internal
FIFO_BA14_TX0_SIZE 8192 // 8192 x 32 internal FIFO

//FIFO_RX0_SIZE 1 // n x 32 internal FIFO
FIFO_BA14_RX0_SIZE 8192 // 8192 x 32 internal FIFO

FIGURE 3 PC104P-BISERIAL-VI-BA14 ADDRESS MAP

The address map provided is for the local decoding performed within the PC104p-BiSerial-VI-BA14. The addresses are all offsets from a base address, which is assigned by the system when the PCI bus is configured.

The VendorId = 0x10EE. The CardId = 0x0026. The ID's and Address definitions retained from the BIS3 version.

Current FLASH revision - see switch port.

Note: ADC, DAC, upper ports are shown for reference. Not used in BA14.

Programming

Programming the PC104p-BiSerial-VI-BA14 requires only the ability to read and write data from the host. The base address is determined during system configuration of the PCI bus. The base address refers to the first user address for the slot in which the PMC is installed.

Depending on the software environment it may be necessary to set-up the system software with the PC104p-BiSerial-VI-BA14 "registration" data. For example, in WindowsNT there is a system registry, which is used to identify the resident hardware. Other OS may be more “plug and play”.

In order to receive data the software is only required to enable the receiver for the channel(s) of interest. The data will be captured once the receiver is synchronized with the “LR” signal and the data stored into the associated FIFO. The data can be read based on the FIFO status or DMA programmed and enabled to provide automatic transfer of data to the system memory.

To transmit, data should be written into the channel 0 TX FIFO and the transmitter enabled. The initial write before enable needs to be enough data to guarantee that the FIFO will not go empty before more data can be written. You can overlap writing the message with transmitting the message if your system can guarantee timing on the writes. With a less real time system, preloading the entire message is a good idea.

Interrupts are used to help manage the DMA process. When a programmed transfer is completed the interrupt can be generated to alert the host to program a new transfer to a new location. The transfers can be programmed to be quite large and are independent for each channel allowing the CPU interaction to be minimized.

The Dynamic Engineering BA14 driver for Windows 7 manages the interaction and can set-up the DMA for you. Please refer to the driver manual for more information.

Register Definitions

PC104P_BISVI_BASE

PC104P_BISVI_BASE 0x0000 // 0 base control register offset

BASE Control Register

DATA BIT	DESCRIPTION
31	Select_Disable
30-2	Spare and reserved bits
1	intforce
0	m_inten

FIGURE 4 PC104P-BISERIAL-VI-BA14 BASE CONTROL REGISTER BIT MAP

M_inten when set allows the PC104p-Biserial-VI-BA14 to generate interrupts at the card level. There are additional local enables for each interrupt type at the channel level.

Intforce when set causes an interrupt to be generated to the system. Useful for debugging and software test. Please note that M_inten must be enabled [set] for intforce to have an effect.

Select_Disable when set [‘1’] disables the update of the Mux control lines from the user switch. If changing switch settings and not wanting to change clk settings etc. then set this bit high. ‘0’ allows the mux to be updated from the switch. The mux selects which clock, IDSEL etc are used by the board based on Switch positions 0,1 [of the 8].

Mostly used to allow test of the switch, changing the settings without breaking the PCI interface.

PC104P_BISVI_ID_SW

```
PC104P_BISVI_ID_SW            0x0004   // 1 ID & Switch Register offset
```

Switch and Revision Port

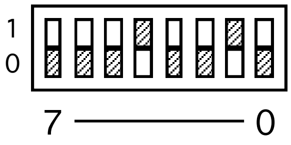
DATA BIT	**DESCRIPTION**
31-16	Spare
15-8	revision
7-0	switch in

FIGURE 5 PC104P-BISERIAL-VI-BA14 STATUS BIT MAP

The dip switch can be read with the lower eight bits of this port. Direct mapping.

The Switch Read Port has the user bits. The user bits are connected to the eight dip-switch positions. The switches allow custom configurations to be defined by the user and for the software to identify a particular board by its switch settings and to configure it accordingly.

The Dip-switch is marked on the silk-screen with the positions of the digits and the '1' and '0' definitions. The numbers are hex coded. The example shown would produce 0x12 when read.



The bits 1,0 are used to select the IDSEL, PCI Clock, Request / Grant and interrupt level. An external mux is used for the clock and IDSEL lines. The Request, Grant, and Interrupt logic is internal to the Xilinx. Bits 1,0 should be set to correspond to the level of the Biserial within the PC104p stack. The CPU would be the base level. Position “00” would be immediately above the CPU, position “01” would be the next slot up etc.

The revision is for the FLASH is currently 0x5. It is a good idea for your software to check the FLASH revision to make sure the HW matches the SW definitions.

PC104P_BISVI_STATUS

PC104P_BISVI_STATUS 0x0008 // 2 base Status register

CONTROL RX

DATA BIT	DESCRIPTION
31	int_stat
30-1	Spare
0	loc_int

FIGURE 6 PC104P-BISERIAL-VI-BA14 INTERRUPT STATUS BIT MAP

LOC_INT = Force_int for this implementation. INT_STAT is the masked version of FORCE_INT. For designs implementing other non-channel resources more interrupt status will be available. Please also see the channel interrupts for more information.

PC104P_BISVI_DIR_TERM

PC104P_BISVI_DIR_TERM 0x000C // 3 direction and termination register

CONTROL DIR_TERM REGISTER

DATA BIT	**DESCRIPTION**		
15-0	DIRection	15-0	0 = read 1 = drive
31-16	TERMination	15-0	1 = terminated

FIGURE 7 PC104P-BISERIAL-VI-BA14 DIRECTION TERMINATION CONTROL BIT MAP

The direction for each of the 16 differential pairs is controlled through this port. The port defaults to zero, which corresponds to tri-stating the drivers and no terminations enabled.

Pull-up and Pull-down resistors built into some '485 interface devices may make the signal appear to be driven (if open) when in the tri-stated mode. Enabling the termination on a tri-stated line will yield approximately 2.5V on each side of the tri-stated driver.

CONTROL	CORRESPONDING IO BITS
DIR_15-0	IO_15-0
TERM_15.0	IO_15..0

Parallel termination resistors are supplied on each differential pair along with a switch to allow the user to select which lines are terminated and where. In some systems it will make sense to terminate the lines in the cable and in others it will make sense to use the onboard terminations.

The terminations for the receive groups should be set to terminate with the user software in most cases. If the Parallel Port is set to be an input with the direction bits the corresponding termination bits should also be set.

PC104P_BISVI_PARDAT_485

PC104P_BISVI_PARDAT_485 0x0010 // 4 parallel 485 data IO DATA

Parallel Data (485) IO Port

DATA BIT	DESCRIPTION
31-16	Spare
15-0	parallel output data

FIGURE 8 PC104P-BISERIAL-VI 485 DATA IO BIT MAP

There are 16 potential output bits in the parallel port. The Direction and Termination register sets the direction of the bits. When the direction is set to output and the source control is set to parallel port the bit definitions from this register are driven onto the corresponding parallel port lines.

Writing to this register puts data onto the enabled data lines [direction set].

Reading from this port returns all of the IO lines. It is possible that the output data does not match the IO data in the case of the Direction bits being set to input.

PC104P_BISVI_PARDAT_485_RDBK

PC104P_BISVI_PARDAT_485_RDBK 0x0014 // 5 parallel 485 data reg read-back

Parallel Data Register Read-back Port

DATA BIT	DESCRIPTION
15-0	15-0

FIGURE 9 PC104P-BISERIAL-VI-BA14 485 DATA RDBK BIT MAP

To read the contents of the PARDAT_485 port access this port. Read only. This is the direct read of the register rather than the IO signals.

PC104P_BISVI_PARCNTL

```
PC104P_BISVI_PARCNTL          0x0018   // 6  parallel data control register
```

Parallel Port Control

DATA BIT	**DESCRIPTION**
31-16	Spare
15-0	Parallel Port Source Definitions

FIGURE 10 PC104P-BISERIAL-VI-BA14 PARALLEL CONTROL BIT MAP

Each of the Parallel Port bits has a corresponding source control bit. When the bit is set '1' the parallel data is used [PC104P_BISVI_PARDAT_485]. When '0' the defined IO is used. The BA14 design uses bits 7-0 for the receive port, and has bits 15-8 available as a transmit port. The control bits for 15-0 should be set to 0 to use the IO definitions instead of the parallel port.

Please note that the direction & termination control bits need to be set to make the port bits act as inputs or outputs. For the BA14 the bits 7-0 should be set to xXX00 to leave the lower bits as inputs for the IO function. Most likely bits 27-16 should be set to x00FF to terminate the inputs. The Upper Direction bits should be set for the channels it is intended to transmit on. Any number of bits can be selected. Unselected bits will be left in the tri-stated state.

PC104P_BISVI_TTL_IO

```
PC104P_BISVI_TTL_IO           0x001C   // 7  TTL Data out and TTL Data in
```

TTL Parallel Port Data

DATA BIT	**DESCRIPTION**
31-16	Spare
15-8	TTL BitWise enables. ‘1’ = enabled [output]
7-0	TTL Parallel Port Source Definitions

FIGURE 11 PC104P-BISERIAL-VI-BA14 TTL PARALLEL DATA BIT MAP

The 8 Bit TTL parallel port is accessed through this register. Writing to the port controls the output data. Reading from the port provides the state of the TTL IO lines. The Enables are tied to the active low buffer enables. Setting a ‘1’ on the corresponding control bit will put a ‘0’ on the Tx IO Buffer Enable. Default is to off. Must be set to drive the line.

To read-back the TTL output data register use the PC104P_BISVI_TTL_RDBK port. The external line definitions can be different than the register definitions because the lines are tri-stated for a ‘1’ definition and driven low for a ‘0’ bit definition.

PC104P_BISVI_TTL_RDBK

```
PC104P_BISVI_TTL_RDBK        0x0020   // 8  TTL Data out register read-back
```

TTL Register Read-Back

DATA BIT	**DESCRIPTION**
31-16	Spare
15-8	TTL BitWise enables. ‘1’ = enabled [output]
7-0	TTL Parallel Port Register Read-Back

FIGURE 12 PC104P-BISERIAL-VI-BA14 TTL PARALLEL DATA BIT MAP

To read the register contents of the TTL output definition port read from this register.

Addresses x24 – x4C, xA0 - x114 are not used in the BA14 design.

```
PC104P_BISVI_FIFO_A            0x0024   // 9  External FIFO A write,read port
PC104P_BISVI_FIFO_B            0x0028   // 10  External FIFO B write,read port
PC104P_BISVI_DAC0              0x002C   // 11  DAC 0 data port

PC104P_BISVI_DAC1              0x0030   // 12  DAC 1 data port
PC104P_BISVI_DAC2              0x0034   // 13  DAC 2 data port
PC104P_BISVI_DAC3              0x0038   // 14  DAC 3 data port
PC104P_BISVI_ADC0              0x003C   // 15  ADC 0 CMD/data port

PC104P_BISVI_ADC1              0x0040   // 16  ADC 1 CMD/data port
PC104P_BISVI_ADC2              0x0044   // 17  ADC 2 CMD/data port
PC104P_BISVI_ADC3              0x0048   // 18  ADC 3 CMD/data port
PC104P_BISVI_ASTAT           0x004C   // 19  DAC/ADC Status port
```

PC104P_BISVI_BASE 0

PC104P_BISVI_base0 //30 base control for channel 0

Parallel Port Control

DATA BIT	DESCRIPTION
31-13	Spare
12	TX_ENABLE_CLRDIS
11	Spare
10-8	CH_SEL(2-0)
7	RX ENABLE
6	TX ENABLE
5	FORCE INT
4	MINTEN
3	DMA RDEN
2	DMA WREN
1	BYPASS
0	RST

FIGURE 13 PC104P-BISERIAL-VI-BA14 DATA SOURCE BIT MAP

RST when ‘1’ causes the FIFO for the channel to reset. TX and RX.

Bypass when ‘1’ causes the TX FIFO to automatically be read and loaded into the RX FIFO to implement Loop-back testing. Set to ‘0’ for normal operation.

DMA WREN and DMA RDEN are interrupt enables for the DMA transfers. Set to enable.

MINTEN is the channel master interrupt enable. Enabled when ‘1’. Use to enable the RX and TX channel interrupts. DMA can be used with or instead of these interrupts.

Force Int when set and MINTEN is also set causes an interrupt to be generated from this channel.

TX Enable when set ‘1’ starts the Transmit IO state-machine. The State-machine will idle until the Valid signal from the TX FIFO is true. Data will then be sent at 2.5 Mbits/sec until the FIFO data is not valid when needed or the enable is set to ‘0’. Please note that channel 0 is the only channel with the transmitter instantiated.

RX Enable when set ‘1’ starts the Receive IO state-machine. The state-machine will wait for an edge on the data strobe [LR] and then start to capture data. Each time 32 bits is captured data is written to the RX FIFO. The process continues until disabled '0'.

CH_SEL bits are used to select the IO channel to use for receiving.

Ch_SEL	CHANNEL
000	IO 0
001	IO 1
010	IO 2
011	IO 3
100	IO 4
101	IO 5
110	IO 6
111	IO 7

TX_ENABLE_CLRDIS when set causes the auto-clear at the end of a transmission to be disabled. The effect is that the hardware stays enabled so any new data written to the TX FIFO will [potentially] immediately begin to be transmitted.

PC104P_BISVI_INT 0

PC104P_BISVI_int0 // 31 INT latch and status 0

Channel Status Port

DATA BIT	DESCRIPTION
15	DMA RD INT
14	DMA WR INT
13	DMA RD ERR
12	DMA WR ERR
11	LOC_INT
10	RCVR OVFL
9	INTTX_CH
8	INTRX_CH
7	RX FF VALID
6	RX FF FULL
5	RX FF AFL
4	RX FF MT
3	gnd
2	TX FF FULL
1	TX FF AMT
0	TX FF MT

FIGURE 14 PC104P-BISERIAL-VI-BA14 DATA SOURCE BIT MAP

Reading from the register provides status. Writing to the register with bits 15-12,10,9,8 set will clear the stored condition.

TX FF MT is the Transmit FIFO Empty status bit. When the internal TX FIFO is empty this bit will be set.

TX FF AMT is the Transmit FIFO Almost Empty status bit. When the FIFO data level is at or below the programmed “Almost Empty” point this bit will be set.

TX FF FULL is the Transmit FIFO Full status bit. When the FIFO is full this bit is set.

RX FF MT is the Receive FIFO Empty status bit. When the internal receive FIFO is empty this bit is set.

RX FF AFL is the Receive FIFO Almost Full status bit. When the FIFO is full to the programmed almost full level or above this bit is set.

RX FF FULL is the Receive FIFO Full status bit. When the FIFO is full this bit is set.

RX FF VALID is a status bit which is set to indicate that there is valid data ready to be read from the FIFO. The FIFO has an external pipeline to support the DMA and back-up requirements. The data from the FIFO is pre-read into the pipeline to be ready to transfer. When valid data is in the last stage of the pipeline the valid bit is set. Please note that the FIFO may be empty and this bit still set as the pipeline is emptied.

LOC_INT is set when a channel IO interrupt or Force Interrupt has occurred. In this design the IO does not have interrupts leaving only the Force Interrupt condition.

DMA write is associated with writing data into the card and the TX direction. DMA read is associated with reading data from the card and the RX direction. The DMA hardware within the Biserial 3 takes care of the actual data transfer.

DMA WR ERR is set when an error occurs during a DMA write transfer. This can be a target or master abort or an incorrect direction bit in a descriptor pointer.

DMA RD ERR is set when an error occurs during a DMA read transfer.

DMA RD INT is set when a DMA read transfer completes.

DMA WR INT is set when a DMA write transfer completes.

INTRX_CH, INTTX_CH The channel interrupts are set when the transmitter completes sending data or the receiver has received a message. If the hardware is being used for larger message based transfers the DMA interrupts can be used instead to manage the flow based on FIFO condition rather than individual messages.

Receiver Overflow is set if the receiver FIFO is full when it is time to write the next data. The condition is stored until cleared.

PC104P_BISVI_BURSTIN 0

PC104P_BISVI_burstin0 // 32 burst in 0

Burst IN Port

DATA BIT	**DESCRIPTION**
31-0	Burst in starting address

FIGURE 15 PC104P-BISERIAL-VI-BA14 BURST IN BIT MAP

Burst In is a VHDL module designed by Dynamic Engineering to manage DMA transfers of data from host memory to the local memory. In the case of the PC104p-BiSerial-VI the local memory is FIFO based.

The software will access the Burst In address – located in host memory, and supply the hardware with the location of the initial DMA descriptor. The hardware will then fetch the pointers to the data from the location supplied. The data will be read from the host memory and the transfers controlled by the FIFO level. The burst in module will attempt to keep the FIFO close to full and the IO module will read the data and transmit.

The process is compatible with scatter gather DMA techniques. The Dynamic Driver for Windows will manage this process for you.

If you are writing your own driver, each list entry consists of three 32-bit words. The linked list descriptor order is:

DmaPciAddress PCI address of data block
DmaLength Length of data block
NextPointer PCI address of next descriptor (bit 0 is set to indicate that this is the last list entry - bit1 is set if the transfer is from the device to PCI memory)

PC104P_BISVI_BURSTOUT 0

PC104P_BISVI_burstout0 // 33 burst out 0

Burst IN Port

DATA BIT	DESCRIPTION
31-0	Burst in starting address

FIGURE 16 PC104P-BISERIAL-VI-BA14 BURST OUT BIT MAP

Burst Out is a VHDL module designed by Dynamic Engineering to manage DMA transfers of data from local memory to the host memory. In the case of the PC104p-BiSerial-VI the local memory is FIFO based.

The software will access the Burst Out address and supply the hardware with the location of the initial DMA descriptor – located in host memory. The hardware will then fetch the pointers to the data from the location supplied. The data will be read from the local memory and the transfers controlled by the FIFO level. The burst out module will attempt to keep the FIFO close to empty and the IO module will load data into the FIFO as data is received from the external system.

The process is compatible with scatter gather DMA techniques. The Dynamic Driver for Windows will manage this process for you.

The link list order is the same as for the Burst In process.

PC104P_BISVI_FIFORW 0

PC104P_BISVI_fiforw0 // 34 single read / write 0,1,2,3

Single access Read / Write Port

DATA BIT	DESCRIPTION
31-0	FIFO Data

FIGURE 17 PC104P-BISERIAL-VI-BA14 FIFO R/W BIT MAP

To do non- DMA read and write accesses to the FIFO's use this port. The TX FIFO can be written to and the RX FIFO read from for each channel. If the Bypass mode is enabled the data will automatically be moved from TX to RX FIFO's allowing for DMA or single word access loop-back testing. For small data transfers this port can be easier to use than setting up the DMA. For larger transfers or time critical operation the DMA method is preferred.

PC104P_BISVI_TXAMTC 0

PC104P_BISVI_txamtc0 // 35 TX Almost Empty Count 0,1,2,3

TX Almost Empty Count Port

DATA BIT	DESCRIPTION
15-0	Count Data

FIGURE 18 PC104P-BISERIAL-VI-BA14 TX AMT COUNT BIT MAP

The TX Almost Empty Count is programmed with this register. The Register is read-write. The count is used to determine the channels Almost Empty state. When this level is true the DMA will move more data into the FIFO.

PC104P_BISVI_RXAFC 0,1,2,3

PC104P_BISVI_rxafc0,1,2,3 // 36,46,56,66 RX Almost Full Count 0,1,2,3

RX Almost Full Count Port

DATA BIT	DESCRIPTION
15-0	Count Data

FIGURE 19 PC104P-BISERIAL-VI-BA14 RX AF COUNT BIT MAP

The RX Almost Full Count is programmed with this register. The Register is read-write. The count is used to determine the channels Almost Full state. When this level is true the DMA will move more data out of the FIFO.

PC104P_BISVI_TX FIFO WORD COUNT 0,1,2,3

PC104P_BISVI_txfifowc0,1,2,3 // 37,47,57,67 TX FIFO Word Count 0,1,2,3

TX FIFO Word Count

DATA BIT	DESCRIPTION
15-0	Count Data

FIGURE 20 PC104P-BISERIAL-VI-BA14 TX FIFO COUNT BIT MAP

The TX FIFO level can be read with this port. The level is compared with the programmed level to create the TX Almost Empty flag. This port is read only.

PC104P_BISVI_RX FIFO WORD COUNT 0,1,2,3

PC104P_BISVI_rxfifowc0,1,2,3   // 38,48,58,68 RX FIFO Word Count 0,1,2,3

TX FIFO Word Count

DATA BIT	DESCRIPTION
15-0	Count Data

FIGURE 21   PC104P-BISERIAL-VI-BA14 RX FIFO COUNT BIT MAP

The RX FIFO level can be read with this port. The level is compared with the programmed level to create the RX Almost Full flag. This port is read only.

Interrupts

PC104p-BiSerial-VI-BA14 interrupts are treated as auto-vectored. When the software enters into an exception handler to deal with a PC104p-BiSerial-VI-BA14 interrupt the software must read the status register(s) to determine the cause(s) of the interrupt, clear the interrupt request(s) and process accordingly. Power on initialization will provide a cleared interrupt request and interrupts disabled.

For example, the PC104p-BiSerial-VI-BA14 DMA and state machines generate an interrupt request when a programmed transfer is complete and the interrupt enable and Master interrupt enable bits are set.

The interrupt is mapped to INTA [or B or C or D based on the switch setting] on the PC/104p connector, which is mapped to a system interrupt when the PCI bus configures. The source of the interrupt is obtained by reading the Interrupt Status register. The status remains valid until that bit in the status register is explicitly cleared.

When an interrupt occurs, the Master interrupt enable should be cleared and the status register read to determine the cause of the interrupt. Next perform any processing needed to remove the interrupting condition, clear the latched bit and set the Master interrupt enable bit high again.

The individual enables operate after the interrupt holding latches, which store the interrupt conditions for the CPU. This allows for operating in polled mode simply by monitoring the Interrupt Status register.

Loop-back

The Engineering kit has reference software, which includes an external loop-back test. The BA14 version of the PC104p-BiSerial-VI utilizes a 50 pin right angle header connector. The test requires an external cable [ribbon] with the following pins connected. Upper 8 to lower 8.

SIGNALs	S	D0
IO8+=>0	17	1
IO8-	18	2
IO9+=>1	19	3
IO9-	20	4
IO10+=>2	21	5
IO10-	22	6
IO11+=>3	23	7
IO11-	24	8
IO12+=>4	25	9
IO12-	26	10
IO13+=>5	27	11
IO13-	28	12
IO14+=>6	29	13
IO14-	30	14
IO15+=>7	31	15
IO15-	32	16

PC104p-BiSerial-VI-BA14 Header Pin Assignment

The figure below gives the pin assignments for the header connector on the PC104p-BiSerial-VI design. Please note that the Analog and TTL IO are not installed on this version. GND* is a plane which is tied to GND through a 0805 0Ω resistor. DC, AC or open are options. For customized version, or other options, contact Dynamic Engineering.

IO_0P IN0+	IO_0m IN0-	1	2
IO_1P IN1+	IO_1m IN1-	3	4
IO_2P IN2+	IO_2m IN2-	5	6
IO_3P IN3+	IO_3m IN3-	7	8
IO_4P IN4+	IO_4m IN4-	9	10
IO_5P IN5+	IO_5m IN5-	11	12
IO_6p IN6+	IO_6m IN6-	13	14
IO_7p IN7+	IO_7m IN7-	15	16
IO_8p OUT0+	IO_8m OUT0-	17	18
IO_9p OUT1+	IO_9m OUT1-	19	20
IO_10p OUT2+	IO_10m OUT2-	21	22
IO_11p OUT3+	IO_11m OUT3-	23	24
IO_12p OUT4+	IO_12m OUT4-	25	26
IO_13p OUT5+	IO_13m OUT5-	27	28
IO_14p OUT6+	IO_14m OUT6-	29	30
IO_15p OUT7+	IO_15m OUT7-	31	32
GND*	GND*	33	34
ADC0	TTL0	35	36
ADC1	TTL1	37	38
ADC2	TTL2	39	40
ADC3	TTL3	41	42
DAC0	TTL4	43	44
DAC1	TTL5	45	46
DAC2	TTL6	47	48
DAC3	TTL7	49	50

FIGURE 22 PC104P-BISERIAL-VI CONNECTOR PINOUT

Applications Guide

Interfacing

The pin-out tables are displayed with the pins in the same relative order as the actual connectors. The pin definitions are defined with noise immunity in mind. The pairs are chosen to match standard ribbon cable pairing to allow a low cost commercial cable to be used for the interface.

Some general interfacing guidelines are presented below. Do not hesitate to contact the factory if you need more assistance.

Watch the system grounds. All electrically connected equipment should have a fail-safe common ground that is large enough to handle all current loads without affecting noise immunity. Power supplies and power-consuming loads should all have their own ground wires back to a common point.

Power all system power supplies from one switch. Connecting external voltage to the PC104p-BiSerial-VI when it is not powered can damage it, as well as the rest of the host system. This problem may be avoided by turning all power supplies on and off at the same time. Alternatively, the use of OPTO isolation panels is recommended.

Keep cables short. Flat cables, even with alternate ground lines, are not suitable for long distances. The PC104p-BiSerial-VI does contain input protection. The connector is pinned out for a standard Header cable to be used. The twisted pairs are defined to match up with the PC104p-BiSerial-VI pin definitions. It is suggested that this standard cable be used for most of the cable run.

Custom cables can be manufactured with discrete wire header and direct connection to your mating equipment.

Terminal Block. We offer a high quality 50-screw terminal block that directly connects to the ribbon cable. The terminal block can mount on standard DIN rails. HDRterm50 [http://www.dyneng.com/HDRterm50.html]

We provide the components. You provide the system. Safety and reliability can be achieved only by careful planning and practice. Inputs can be damaged by static discharge, or by applying voltage outside of the RS-485 devices rated voltages.

Construction and Reliability

PC/104p Modules were conceived and engineered for rugged industrial environments. PC104p-BiSerial-VI is constructed out of high temperature 0.062 inch thick FR4 material.

Through hole and surface mounting of components are used.

The PC/104p device is secured into the stack with high insertion force pins and four screws attached to the 4 stand-offs. The four screws provide significant protection against shock, vibration, and incomplete insertion.

The PC/104p Module provides a low temperature coefficient of 1.7 W/^{o}C for uniform heat. This is based upon the temperature coefficient of the base FR4 material of 0.31 W/m-^{o}C, and taking into account the thickness and area of the PC/104p. The coefficient means that if 1.7 Watts are applied uniformly on the component side, then the temperature difference between the component side and solder side is one degree Celsius.

Thermal Considerations

The PC104p-BiSerial-VI design consists of CMOS circuits. The power dissipation due to internal circuitry is very low. It is possible to create a higher power dissipation with the externally connected logic. If more than one Watt is required to be dissipated due to external loading; forced air cooling is recommended. With the two degree differential temperature to the solder side of the board external cooling is easily accomplished.

Dynamic Engineering PC104p devices ground the stand-off support pads – tied into the ground plane [complete and at least 1 oz]. Regulators and higher dissipation devices are well bonded to the ground plane to allow for thermal transfer as well as electrical operation.

Warranty and Repair

Please refer to the warranty page on our website for the current warranty offered and options.

http://www.dyneng.com/warranty.html

Service Policy

Before returning a product for repair, verify as well as possible that the suspected unit is at fault. Then call the Customer Service Department for a RETURN MATERIAL AUTHORIZATION (RMA) number. Carefully package the unit, in the original shipping carton if this is available, and ship prepaid and insured with the RMA number clearly written on the outside of the package. Include a return address and the telephone number of a technical contact. For out-of-warranty repairs, a purchase order for repair charges must accompany the return. Dynamic Engineering will not be responsible for damages due to improper packaging of returned items. For service on Dynamic Engineering Products not purchased directly from Dynamic Engineering, contact your reseller. Products returned to Dynamic Engineering for repair by other than the original customer will be treated as out-of-warranty.

Out of Warranty Repairs

Out of warranty repairs will be billed on a material and labor basis. Customer approval will be obtained before repairing any item if the repair charges will exceed one half of the quantity one list price for that unit. Return transportation and insurance will be billed as part of the repair and is in addition to the minimum charge.

For Service Contact:

Customer Service Department
Dynamic Engineering
150 DuBois St., Suite C
Santa Cruz, CA 95060
831-457-8891
support@dyneng.com

Specifications

Host Interface:	PCI-104 - 32 bit PCI bus
Serial Interface:	1 RX and 1 TX serial RCB / WRA protocol ports. 8 IO ports selected in SW.
Tx Data rates generated:	6.25 Mhz
Rx Data rates accepted:	6.25 MHz
Software Interface:	Control Registers, Status Ports
Initialization:	Hardware Reset forces all registers to 0.
Access Modes:	LW boundary Space (see memory map)
Wait States:	1 for all addresses
Interrupt:	DMA Read and write interrupts for each channel Software interrupt
DMA:	2 channels of DMA, TX and RX channels.
Onboard Options:	All Options are Software Programmable
Interface Options:	50 pin ribbon cable or discrete wire 50 screw terminal block interface
Dimensions:	Standard Single PCI-104 Module.
Construction:	FR4 Multi-Layer Printed Circuit, Through Hole and Surface Mount Components.
Temperature Coefficient:	1.7 W/^{O}C for uniform heat across PC/104p
Power:	Max. **TBD** mA @ 5V

Order Information

PC104p-BiSerial-VI-BA14	PCI-104p Module with RCB / WRA protocol
HDRterm50	50 position screw terminal adapter http://www.dyneng.com/HDRterm50.html
HDRcabl50	50 position ribbon cable http://www.dyneng.com/HDRribn50.html