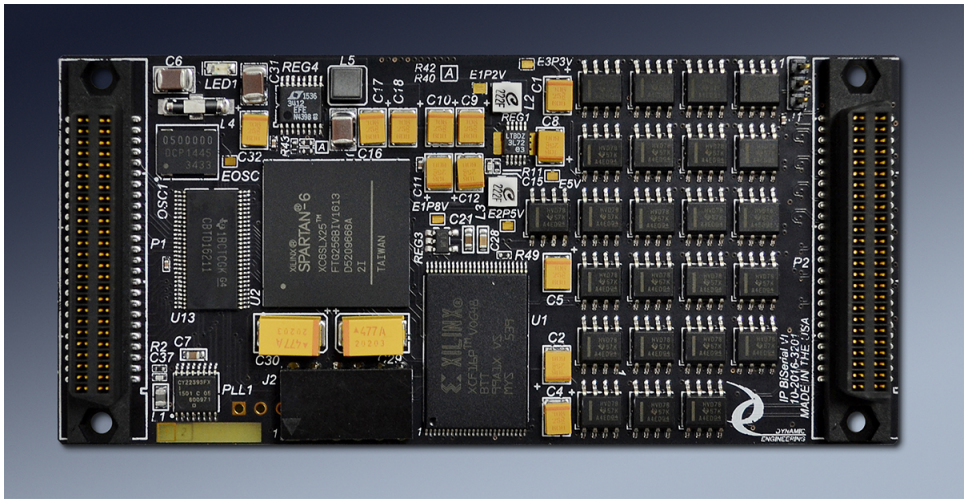
DYNAMIC ENGINEERING
150 DuBois St. Suite C Santa Cruz, CA 95060
831-457-8891 **Fax** 831-457-4793
sales@dyneng.com
www.dyneng.com
Est. 1988

User Manual

IP-BiSerial-VI-USER

User Interface

LVDS / RS485 / Mixed IndustryPack® Module



Manual Revision: 01p1

Corresponding Hardware: Revision 01
FLASH revision: NA Fab: 10-2016-3201

IP-BiSerial-VI-USER
User Interface
IndustryPack® Module

Revised 01/06/20

Table of Contents

List of Figures

Product Description

IP-BiSerial-VI-USER describes the use of the supplied VHDL reference to use with IP-BiSerial-VI. The design is intended for users who want to design their own VHDL. The reference design is based on the “BA27” model.

Xilinx Spartan VI SLX25 2I is the installed part type. Any mixture of inputs, outputs, half duplex operation is supported. When delivered this version will have a special test program loaded that will need to be replaced with the user FLASH implementation.

IP-BiSerial-VI is part of the IndustryPack® “IP” Module family of I/O components by Dynamic Engineering. IP-BiSerial-VI provides an IP Module type II compliant mechanical package with a Spartan 6 FPGA, FLASH, PLL(4 reference clocks to FPGA), 24 differential IO [LVDS and/or RS485] each with separately programmable terminations and direction controls. The FLASH is easily reprogrammable with the Xilinx Impact SW and USB adapter – a header is included for this purpose.

The Differential IO are routed to the connector with controlled impedance, matched length [FPGA to IO pin pairs], and pin definitions consistent with the Dynamic Engineering standard for differential IO on IP Modules [1,2 … 23,24 (25,50 grounds), 26,27…48,49]. This definition is implemented on Dynamic Engineering Carrier designs supporting differential pairs [PCIe3IP, PCIe5IP, PCI3IP, PCI5IP, VPX2IP] to name a few.

The reference design is implemented with a single level hierarchy – all logic at the Base

level. The IP Bus interface and decoding, master interrupt control, PLL programming [unused], clock generation, and Transmit and Receive logic plus FIFO memories.

The internal reference clock is derived from the 50 MHz oscillator.

IP-BiSerial-VI features 8 clockout/clock in pairs [IO connections between closely located pins where an IO is tied to a clock input pin]. This allows the muxed signal to be brought back into the FPGA and treated as a clock [Channel Reference Clock] With a lower number of channels the internal clock muxes could be used. With 8 channels resource limitations makes this approach problematic. This feature was borrowed from PMC-BiSerial-VI. *Two of the clocks are used in the reference design. These can be repurposed, reused or discarded as needed for your design.*

IP-BISERIAL-VI-USER conforms to the VITA standard. This guarantees compatibility with multiple IP Carrier boards. Since the IP maintains plug and software compatibility while mounted on different form factors, system prototyping may be done on one IP Carrier board, with final system implementation done on a different one.

The serial channels are supported by 4K by 16 bit FIFOs, which support word accesses. An on-board read/write path exists for loop-back testing.

The design specific details are comments on what is in the reference design. It is intended the user remove and replace or perhaps update logic blocks to incorporate the new features you desire.

The serial receive channel looks for data in 16 or 32 bit transfers. The received bit stream is loaded into the input FIFO. The data length loaded is determined by the programmed mode. The number of words loaded is determined by the once or continuous control. The host can poll or wait for the message complete or FIFO almost-full interrupt. The message can be read directly from the input FIFO. Several error conditions are checked including parity, and over-run.

The serial transmit channel reads data from the output FIFO and sends it out serially, lsb or msb first. Each word can have a parity bit, and it is automatically appended. The parity is programmable for odd and even. The message can be one or two words long. Data can be pre-stored in the FIFO and sent later with the start enable.

When the transmission is complete, an optional DonePulse can be generated on a separate IO line. The pulse is programmable for width and delay from the last falling edge of the transmitted serial data clock.

Interrupts are supported. The interrupt occurs at the end of the transmission whether data is received or sent or both. The programmable interrupts are available to provide an almost empty indicator for Tx and almost full indicator for Rx. The interrupts are individually maskable, and the interrupt vector is user programmable by a read/write register. The interrupt occurs on IntReq0. Status is available for the FIFO's making it possible to operate in a polled mode.

Theory of Operation

IP-BISERIAL-VI-USER is designed for the purpose of transferring data from one point to another with a serial protocol.

IP-BISERIAL-VI-USER features a Xilinx FPGA. The FPGA contains all of the registers and protocol controlling elements of the BISERIAL design. Only the IO, FLASH and Power Supplies are external to the Xilinx device.

A logic block within the Xilinx device contains the decoding and timing elements required for the host CPU to interface with the IP bus. The timing is referenced to the 8 or 32 MHz IP logic clock. The IP responds to the ID, INT, MEM, and IO selects. The DMA control lines are connected to the Xilinx for future revisions, but are not used at this time. The BISERIAL design requires wait states for read or write cycles to any address. Hold cycles are supported as required by the host processor. Data remains enabled during a read until the host removes the SEL line. Local timing terminates a write cycle prior to the SEL being deasserted. If no hold cycles are requested by the host, IP-BISERIAL-VI-USER is capable of supporting 16+ MB per second data transfer rate with a 32 MHz reference rate.

A 64 word decoder is included in the design. Easy to tie in registers or enable read back paths.

The serial I/O can support many protocols. The timing for a transfer is shown in figure 1. The clock is gated and normally low. The data switches on the falling edge, and is valid on the rising edge of the clock. The design is implemented with the data shifted to provide ½ period of set-up and ½ period of hold. The Transmit state machine uses the programmed reference clock running at 4X the transmit rate. Frequency options are user defined or 1 MHz transmit rate selections. The user selection is based in dividing the 50 MHz oscillator. The length can be 16,17,32,34 bits depending on parity. The data order can be reversed. The Done Pulse can be optionally asserted with a programmable delay after the last falling edge of the clock and a programmable width once asserted.

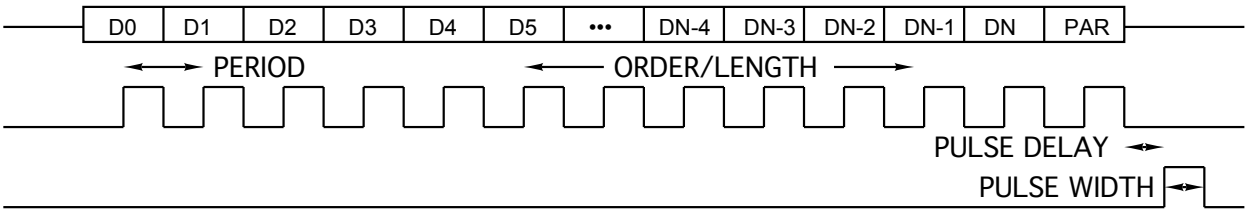


FIGURE 1 IP-BISERIAL-USER SERIAL PROTOCOL TIMING

When in 32 bit mode two 16 bit transfers are concatenated. If parity is enabled then 34 clock periods are used, and if disabled 32 are used. Parity is in the 17th and 34th bit positions when enabled.

32 bit mode requires two 16 bit data to be stored in the TX FIFO prior to transmission. A comparator is used to insure at least 2 data are available before the state machine can begin transmission in this mode. For 16 bit mode the FIFO empty flag is used to insure data is available.

The receive channel uses the IP clock to sample the incoming clock and look for a period where the clock is low for longer than the bit period, or inter-message gap. The time is defined in a programmable register. Once the clock has been in the off state for sufficient time to determine that a message is currently not being received the hardware begins to look for an active clock. The count is based on the IP clock – 8 or 32 MHz. The state-machine will add a small number of clock periods to transition between the start-up states and to account for metastability.

Once the programmed time has been achieved, the state-machine looks for rising edges in the received clock. Data is captured by the shift register. 16 or 17 active edges are counted based on the parity being checked or not. Once the data is captured, the data is moved to a secondary latch. This allows the second half of a 32 bit word to be captured in parallel with the parity test [or not] and the data loaded to the FIFO. If the mode is set to 32 bits a second word is “looked” for without waiting for a new gap between words.

When data is captured the state machine creates an interrupt request [maskable], and returns to the idle state. When the interrupt is requested the start bit is automatically cleared unless the continuous bit is set. If placed in continuous mode, the hardware will see the Enable is set when it returns to idle. Immediately the SM will look for a data gap, and then to capture more data.

If desired the interrupt for the RX can be disabled, and the FIFO almost full interrupt used instead. Over-run conditions are tested each time the state-machine loads new data into the FIFO. The parity and over-run errors are latched and held until explicitly cleared.

A pair of state machines within the FPGA control all transfers between the FIFO and IO. The Tx state machine reads from the transmit FIFO 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 FIFO.

Some IP Carriers support 32 or 64 bit data read and writes to IP slots performing an

auto-conversion to 16 bits saving data transfers. Dynamic Engineering carriers have auto-incrementing and static address conversion from 64/32 to 16 bits. FIFO's are on quad word addresses to allow optimization on PCIe based carriers with QW capability as well as PCI based carriers with LW capability.

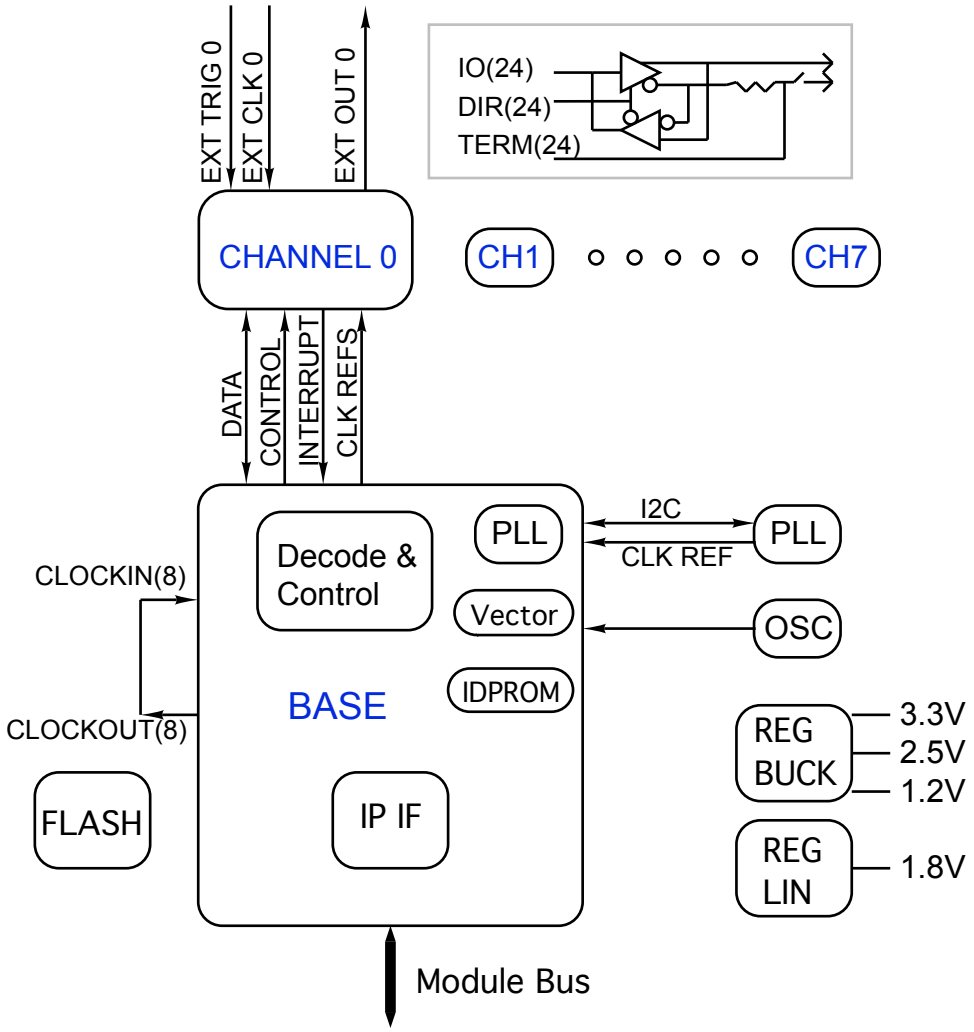


FIGURE 2 IP-BISERIAL-VI-USER BLOCK DIAGRAM

IP-BiSerial-VI incorporates 24 differential pairs, each of which can be LVDS or RS485. For the REF design, 4 IO are required for the primary interface. An additional 16 are used for a parallel port with 4 more spare IO.

The Spartan VI requires three voltages, derived via Buck regulators from the 5V IP standard voltage [3.3V, 2.5V, and 1.2V]. In addition, the FLASH requires a small amount of 1.8V and a linear regulator is provided for this purpose. The 3.3V rail is also used for the LVDS and RS-485 devices allowing mix and match assembly.

The PLL is not incorporated into the REF Design. It is in the VHDL and can be used if the PLLA-D are incorporated into the design.

The installed oscillator is a 50 MHz model with 10 ppm. Additional board level clocking options include the ClockOut to ClockIn network. Each of the 8 ClockOut signals is tied back to a clock input on the FPGA allowing flexibility. This clock bus is used within BA27 to allow the transmitter to use either the 4 MHz or a divided down reference to generate the output signal timing.

The FLASH is used to store the module firmware – in this case the User function.

The Base level of the design provides the IndustryPack interface, general decoding, card level status, IDPROM and Vector register plus board level features ⇔ PLL, interrupt masking, master channel enable.

The Reference design is “flat” with the Base and IO features at the same level. The decoded data bus is routed to each internal register to create control logic, status, data storage to support the serial IO and parallel ports.

Please refer the address/bit maps for details of the various registers and operation. *These will serve as a good reference to what the supplied reference design is doing, and what you will want to modify to make your own design.*

IP-BiSerial-VI-USER conforms to the IndustryPack® standard. This guarantees compatibility with compliant carrier boards. Because the IP-BiSerial-VI-USER 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 the PCIe3IP – PCI carrier for IP Modules can be used for development in a conventional PC. Later the hardware and software can be ported to the target. http://www.dyneng.com/PCIe3IP.html

With the Dynamic Engineering Windows driver collection for IP and carrier modules a Parent – Child architecture is employed. The IP portion of the driver is directly portable between the various Dynamic Engineering IP carriers [PCIe3IP, PCI3IP, PCIe5IP, PCI5IP, PC104pIP, PC104p4IP, cPCI2IP, cPCI4IP etc]. The parent portion of the driver contains the carrier specific design information. This means that software developed for the IP-BiSerial-VI-USER on one platform can be directly ported to another. PCI to

cPCI for example.

Designers can make use of the Dynamic Engineering carrier driver for non-Dynamic Engineering IP modules using the Generic IP capability built into the parent portion of the driver. IP modules that the carrier driver does not recognize are installed as “generic” and accessed with a address, data interface model. Software developed for the Generic mode can also be ported between modules. The generic model will be used for interfacing with the User design.

Please note: the IDPROM should have new data defined as part of the development effort to prevent the carrier driver from recognizing the BA27 design and not installing the Generic driver.

Address Maps

Address Map Base

For reference – implemented in the Reference Design and likely to be changed by the HW engineer.

IPBISVI_BA27_Base	0x00 //0 Base & Tx control register
IPBISVI_BA27_RxCntl	0x02 //1 Rx Control register
IPBISVI_BA27_Vector	0x04 //2 Interrupt vector register
IPBISVI_BA27_BaseStatus	0x06 //3 status read
IPBISVI_BA27_Rev	0x08 //4 Revision, Major&Minor, read only
IPBISVI_BA27_Info	0x0A //5 Spare port used by OS
IPBISVI_BA27_HalfDiv	0x0C //6 Set Divisor for Tx Rate
IPBISVI_BA27_DataDelay	0x0E //7 Set time between words
IPBISVI_BA27_FIFO	0x10 //8 Write to Tx, Read from RX FIFO
IPBISVI_BA27_AMT	0x12 //9 Set Tx Almost Empty level
IPBISVI_BA27_AFL	0x14 //10 Set Rx Almost Full Level
IPBISVI_BA27_FIFO_Status	0x16 //11 FIFO status read only
IPBISVI_BA27_Direction	0x18 //12 Parallel Port Direction Register
IPBISVI_BA27_Termination	0x1A //13 Parallel Port Termination Reg
IPBISVI_BA27_DataTx	0x1C //14 Parallel Port Transmit Data
IPBISVI_BA27_DataIO	0x1E //15 Parallel Port IO side read only
IPBISVI_BA27_TxFifoCnt	0x20 //16 Tx FIFO Data Count read only
IPBISVI_BA27_RxFifoCnt	0x22 //17 Rx FIFO Data Count read only
IPBISVI_BA27_DelayCnt	0x24 //18 DonePulse Delay lower word
IPBISVI_BA27_DelayCnt	0x26 //19 DonePulse Delay upper word
IPBISVI_BA27_PulseWidth	0x28 //20 DonePulse Pulse Width
IPBISVI_BA27_DonePulseCntl	0x2A //21 DonePulse Control

FIGURE 3 IP-BISERIAL-VI-USER ADDRESS MAP

Numbers following the // are the HW decode numbers based on the words – word 0, word 1 etc. There are a total of 64 available in the IP IO space.

Programming

Programming IP-BiSerial-VI-USER requires only the ability to read and write data from the host. The base address refers to the first user address for the slot in which the IP is installed.

The registers are organized as shown in the memory map and can be directly accessed with the offset to the IP Module position on the carrier. The base level has card level control and information. The channels repeat with the same relative offsets to the start of each address range shown.

Depending on the software environment it may be necessary to set-up the system software with IP-BiSerial-VI-USER “registration" data. Other OS may be more “plug and play”. The Dynamic Engineering Driver operates in a “plug and play” mode using parent ⇔ child architecture..

If the special functions are used, access to the IO space is utilized. The bit maps for the IO space registers follows in the next section.

Interrupts are available to alert the local host when an event has happened.

Register Definitions

IPBISVI_BA27_Base

IPBISVI_BA27_Base 0x0000 // 0 base control register offset

DATA BIT	DESCRIPTION
15	PLL_SDAT
14	PLL_SCLK
13	PLL_EN
12	ClockIoTxSel
11	CLK8_32
10	TxOrder
9	Spare
8	TxParityOff
7	TxOddEven
6	TxMode16_32
5	Tx Auto Clear En
4	IntForce
3	IntMasterEn
2	FaeIntEn
1	TxIntEn
0	TxStart

FIGURE 4 IP-BISERIAL-VI-BA27 BASE CONTROL REGISTER BIT MAP

CLK8_32: Can be used to tell the IP what the IP Clock rate is. Currently unused in this design.

IntMasterEn: when set causes allows programmed interrupt sources to assert Int0. Required for all interrupt types.

IntForce: when set causes an interrupt to be generated to the system. Useful for debugging and software test.

PLL IF programming.
The PLL reference frequency is 50 MHz. The PLL is a Cypress 22393. The PLL is programmed with the output file generated by the Cypress PLL programming tool. [CY3672 R3.01 Programming Kit or CyberClocks R3.20.00 or later.]

The .JED file is used by the Dynamic Driver to program the PLL. Programming the PLL is fairly involved and beyond the scope of this manual. For clients writing their own drivers it is suggested to get the Engineering Kit for this board including software, and to use the translation and programming files ported to your environment. This procedure will save you a lot of time. The output file from the Cypress tool can be passed directly to the Dynamic Driver [Linux or Windows] and used to program the PLL.

pll_en : Software output enable control for PLL
pll_sclk: Output to pll command clock
pll_s2 : grounded on the card.
pll_sdat : When PLL_EN is set the output follows the register bit otherwise held in tristate. When read the state of the IO pin is returned.

TxStart is set to send data. The bit is auto cleared at the end of a transmission.

TxIntEn is the Interrupt Enable bit for the Transmit channel. The default state is off. If enabled, and the master interrupt enable is also enabled an interrupt is requested when the transmission is complete. The interrupt is cleared by an Interrupt Acknowledge cycle or disabling the interrupt.

FaeIntEn is the Interrupt Enable bit for the Programmable Almost Empty condition. The default state is off. If enabled and the master interrupt enable is also enabled then an interrupt is requested when the FIFO level falls below the almost empty count. The interrupt is cleared by an Interrupt Acknowledge cycle or disabling the interrupt.

Tx Auto Clear En when ‘1’ enables the clearing of the start bit at the end of a transmitted message. When set to ‘0’ the transmitter will stay enabled at the end of a 16 or 32 bit transfer, and continue to send data-words until the transmit FIFO is empty.

TxMode16_32 when ‘1’ selects 32 bit transmission mode and when ‘0’ selects 16 bit mode. Data is sent lsb first and with programmable parity attached for 16/17 or 32/34 bits total.

TxOddEven when ‘1’ odd parity is selected for the transmitter. The parity bit is appended to the data sent and is related to the data by an XOR function. If odd parity is selected then the parity bit is set to force an odd number of bits to be set [D0-15,parity]. If the parity selection is even then the parity bit is set to cause an even number of bits to

be set. Example: 0x0102 = data. If odd parity is selected then the parity bit would be = ‘1” because there are 2 bits set within the data making an even total.

TxParOff when set suppresses the generation of parity. 16 or 32 bits are transmitted. When ‘0’ parity is generated and appended to the 16 bit data to make a 17 bit word. In 32 bit mode two 17 bit words are concatenated.

TxOrder when ‘1’ causes the Tx data bit order to be reversed to be Msb first. Parity is still last if sent. When ‘0’ Lsb first is the data order.

ClockIoTxSel when ‘1’ selects 4 MHz as the reference for the Tx state machine. When ‘0’ the divided oscillator frequency is used. See the HalfDiv register for setting this rate.

IPBISVI_BA27_RxCntl

IPBISVI_BA27_RxCntl 0x0002 // 2 Rx Control Register

DATA BIT	DESCRIPTION
15	Reset
14-11	spare
10	RxOrder
9	Spare
8	RxParityOff
7	RxOddEven
6	RxMode16_32
5	Rx Auto Clear En
4	spare
3	FifoBypassEn
2	FafIntEn
1	RxIntEn
0	RxStart

FIGURE 5 IP-BISERIAL-VI-BA27 RX CONTROL REGISTER BIT MAP

RxStart is set to receive data. The bit is can be auto cleared at the end of a reception.

RxIntEn is the Interrupt Enable bit for the Receive port. The default state is off. If enabled, and the master interrupt enable is also enabled an interrupt is requested when the reception is complete. The interrupt is cleared by an Interrupt Acknowledge cycle or disabling the interrupt.

FafIntEn is the Interrupt Enable bit for the Programmable Almost Full condition. The default state is off. If enabled and the master interrupt enable is also enabled then an interrupt is requested when the FIFO level rises above the almost full count. The interrupt is cleared by an Interrupt Acknowledge cycle or disabling the interrupt.

Rx Auto Clear En when ‘1’ enables the clearing of the start bit at the end of a received message. When set to ‘0’ the receiver will stay enabled at the end of a 16 or 32 bit transfer, and continue to receive data-words until the receive FIFO is full.

RxMode16_32 when ‘1’ selects 32 bit reception mode and when ‘0’ selects 16 bit mode. Data is expected lsb first and with programmable parity attached for 16/17 or 32/34 bits total.

RxOddEven when ‘1’ odd parity is selected for the receiver. The parity bit appended to the data sent is checked. If odd parity is selected then the parity bit is set to force an odd number of bits to be set [D0-15,parity]. If the parity selection is even then the parity bit is set to cause an even number of bits to be set. Example: 0x0102 = data. If odd parity is selected then the parity bit would be = ‘1” because there are 2 bits set within the data making an even total.

RxParOff when set suppresses the check of parity. 16 or 32 bits are expected if enabled. When ‘0’ parity is expected and appended to the 16 bit data to make a 17 bit word. In 32 bit mode two 17 bit words are concatenated.

RxOrder when ‘1’ causes the Rx data bit order to be reversed to be Msb first. Parity is still last if sent. When ‘0’ Lsb first is the data order stored into memory.

FifoBypassEn when ‘1’ causes the data written into the Tx FIFO to be transferred to the Rx FIFO. Data written to the Tx FIFO can then be read from the Rx FIFO. Data read / transferred is no longer available to transmit with the IO function. TxStart and RxStart should be disabled when using loop-back. When ‘0’ normal operation results with IO data received going into the Rx FIFO and IO data to transmit coming from the Tx FIFO.

Reset when set causes a reset to the FIFO’s and TX/RX state machines. Other registers are not affected by this reset. Return to ‘0’ for normal operation.

IPBISVI_BA27_VECTOR
IPBISVI_BA27_Vector 0x0004 // 2 IP vector port
Vector Port

DATA BIT	DESCRIPTION
15-8	Undefined
7-0	vector

FIGURE 6 IP-BISERIAL-VI-BA27 VECTOR BIT MAP

If the system uses a vectored interrupt approach the vector port should be initialized to the value assigned to this device. IP-BiSerial-VI-BA27 can be used as vectored or auto-vectored. When auto-vectored this port is unused. The Status port can be used to determine the source of any pending interrupts from IP-BiSerial-VI-BA27.

Default is 0xFF for data.

IPBISVI_BA27_BaseStatus

IPBISVI_BA27_BaseStatus 0x0006 // 3 base Status register

Status Register

DATA BIT	DESCRIPTION
15	IntReq
14	4 MHz Locked
13	ParityErr
12	OverRunErr
11	FafInt
10	FaeInt
9	RxInt
8	TxInt
7-1	'0'
0	IntReq0

FIGURE 7 IP-BISERIAL-VI-BA27 INTERRUPT STATUS BIT MAP

IntReq0: when set indicates an enabled Tx or Rx interrupt request is pending. Signal level before the Master Interrupt Enable.

IntReq When IntReq0 or ForceInt is set and the Master Interrupt Enable is set this bit will be set.

TxInt when Set indicates the Transmitter has completed sending 16 ⇔ 34 bits as defined in the control register. This bit will be set each time a transmission completes potentially creating an interrupt stream if the Auto Clear is not enabled and if there are more than 1 data sets loaded into the FIFO. This is a sticky bit and is cleared by writing back to the same port with this bit set.

RxInt when Set indicates the Receiver has completed receiving 16 ⇔ 34 bits as defined in the control register. This bit will be set each time a reception completes potentially creating an interrupt stream if the Auto Clear is not enabled and if there are more than 1 data sets received from the IO port. This is a sticky bit and is cleared by writing back to the same port with this bit set.

FaeInt when Set indicates the Tx FIFO has transitioned from not almost empty to a level below the programmed threshold. This is a sticky bit and is cleared by writing back to the same port with this bit set.

FafInt when Set indicates the Rx FIFO has transitioned from not almost full to a level

above the programmed threshold. This is a sticky bit and is cleared by writing back to the same port with this bit set.

ParityErr is a sticky bit. When set a Parity Error has been detected. This can happen when parity is not set to match the transmitter both for existence and type. Clear by writing back with the same bit position set.

OverRunErr is a sticky bit. When set an Over Run condition has been detected. This will happen when the Rx FIFO is full when the receiver tries to write another received data word. If this is happening, try lowering the Almost Full level and using that interrupt to cause a data read from the Rx FIFO. Clear by writing back with the same bit position set.

4 MHz Locked when set indicates the internal clock generator [PLL] is in lock and the 4 MHz is available to use. The 4 MHz is derived from the 50 MHz oscillator [multipled to 500 and divided down to 4]. The Oscillator is a 10 ppm unit.

IPBISVI_BA27_REV

IPBISVI_BA27_REV 0x0008 // 4
Node Address and Options Register

DATA BIT	DESCRIPTION
15-8	RevisionMajor
7-0	RevisionMinor

FIGURE 8 IP-BISERIAL-VI-BA27 REVISION BIT MAP

RevisionMajor:
This field is reported via the IDPROM as well [revision]. It is rolled when major changes occur.

RevisionMinor:
This field is only read from this location. It is rolled when minor changes occur – usually during development to allow SW tracking of HW revisions without using the Major Revision field.

IPBISVI_BA27_INFO

IPBISVI_BA27_INFO 0x000A // 5
Location Register

DATA BIT	DESCRIPTION
15-11	spare
10-3	Carrier Switch
2-0	Carrier Slot

FIGURE 9 IP-BISERIAL-VI-BA27 INFO BIT MAP

The location register can be updated by the carrier driver during initialization. IP-BiSerial-VI-BA27 Driver can access this information later to determine the carrier and location on the carrier that this node is installed into. Once the IP-BiSerial-VI-BA27 Driver is started the user software can use this as a general purpose register. The IP Driver stores a local copy in RAM to allow the user software to determine which node it is talking to when multiple nodes are present in a PCI/PCIe based system with dynamic addressing. Please note that this function is supported on all Dynamic Engineering carriers and may not be supported on other products.

In a Windows system the user software will query for the installed devices and the devices will be returned in order. The issue is that the order can change and there is no way to directly tell with software which card you are controlling at the moment. The user software can retrieve the devices present and then match them up to the physical hardware based on the carrier switch setting and slot on the carrier.

In some systems, knowing which board is controlling which machine can be important. Please see the software manual and userap reference software for examples of working with multiple cards. The userap software prints out the device number and associated slot and switch settings. Your software can use the information without printing out for proper access control.

This is only important if you have multiple cards visible to the same CPU.

IPBISVI_BA27_HalfDiv

IPBISVI_BA27_HalfDiv x0C // 6

DATA BIT	DESCRIPTION
15-0	HalfDiv

FIGURE 10 IP-BISERIAL-VI-BA27 HALFDIV BIT MAP

HalfDiv is programmed with the divisor used for the first level of division within the user specified counter. The reference for the counter is the oscillator = 50 MHz in this design. The output of the user specified counter is further divided by 2 to create a square wave no matter the programmed divisor set.

Example: Desire 1 MHz. reference for Transmission. Last stage is /2 => divide 50 MHz to get 2 MHz. = 25. Please note: the transmitter further reduces by 4 to create a 250 KHz transmission in the example. The first stage divider is preloaded with x1 and rolls over to x1 when the programmed count is reached.

IPBISVI_BA27_DataDelay

IPBISVI_BA27_DataDelay 0x0E // 7

DATA BIT	DESCRIPTION
15-0	Data Delay

FIGURE 11 IP-BISERIAL-VI-BA27 DATA DELAY BIT MAP

DataDelay: is set to determine when a reception is not underway. The count is based on the 50 MHz oscillator and with the 16 bit range can define required delays up to 1.3 mS.

The receiver state machine auto-bauds with the received data stream looking for rising edges. To prevent data being captured mid-transfer, the timeout is used to make sure the received data is between transmissions when starting. The Time should be specified based on the inter-word gap used in your system and the expected frequency of the data.

IPBISVI_BA27_FIFO

IPBISVI_BA27_FIFO 0x10 // 8

DATA BIT	DESCRIPTION
15-0	FIFO Data

FIGURE 12 IP-BISERIAL-VI-BA27 FIFO

FIFO: Tx and Rx functions are supported with separate 4095x16 FIFO's. Writing to this address puts data into the Tx FIFO. Reading removes data from the Rx FIFO.

If ByPassEn is set the data written to the Tx FIFO can be read from the Rx FIFO. If in standard operating mode, the data written to the Tx FIFO will be read by the Tx State Machine and transmitted as programmed. Data from the IO will be received and stored into the Rx FIFO.

Status for the FIFO's is available in the FIFO Status register

IPBISVI_BA27_AMT

IPBISVI_BA27_AMT 0x12 // 9

DATA BIT	DESCRIPTION
15-0	Tx FIFO Almost Empty

FIGURE 13 IP-BISERIAL-VI-BA27 AMT BIT MAP

The Reference count to determine if the Tx FIFO is “Almost Empty” is set in this register. When the count is below the programmed [less than] level the status bit will be set and an interrupt can be generated [from the transition to being Almost Empty]. Knowing the value programmed allows a SW loop to refill the TxFIFO without needing to read the FIFO count.

IPBISVI_BA27_AFL

IPBISVI_BA27_AFL 0x14 // 10

DATA BIT	DESCRIPTION
15-0	Rx FIFO Almost Full

FIGURE 14 IP-BISERIAL-VI-BA27 AFL BIT MAP

The Reference count to determine if the Rx FIFO is “Almost Full” is set in this register. When the count is above the programmed [greater than] level the status bit will be set and, an interrupt can be generated [from the transition to being Almost Full]. Knowing the value programmed allows a SW loop to read the RxFIFO without needing to read the FIFO count.

IPBISVI_BA27_FIFO_Status

IPBISVI_BA27_FIFO_Status 0x16 // 11

DATA BIT	DESCRIPTION
15-8	'0'
8	TxIdle
7	'0'
6	Rx FIFO AFL
5	Rx FIFO Full
4	Rx FIFO MT
3	'0'
2	Tx FIFO AMT
1	Tx FIFO Full
0	Tx FIFO MT

FIGURE 15 IP-BISERIAL-VI-BA27 FIFO STATUS BIT MAP

MT is set when no data is in the corresponding FIFO. x000 is the count

Full is set when the FIFO has no room left – xFFF is the count.

AMT is set when the count is below the programmed threshold. For example with the threshold set to x10 if the TxFifoCnt less than or equal to x0F the bit will be set.

AFL is set when the count is above the programmed threshold. For example with the threshold set to xFF0 if the RxFifoCnt greater than or equal to xFF1 the bit will be set.

TxIdle is set when the Transmit State Machine is in the Idle State. Use to monitor if the transmitter is busy.

IPBISVI_BA27_Direction

IPBISVI_BA27_Direction 0x18 // 12

DATA BIT	DESCRIPTION
15-0	Parallel Port Direction

FIGURE 16 IP-BISERIAL-VI-BA27 DIRECTION BIT MAP

Each Bit of the Direction Control Register corresponds to a bit in the Parallel Port. Bit 0 is for IO 8 … Bit 15 for IO 23. When set the transceiver is programmed to transmit, when cleared the transceiver is programmed to receive. The FPGA IO are programmed to support the transceiver definitions.

IPBISVI_BA27_Termination

IPBISVI_BA27_Termination 0x1A // 13

DATA BIT	DESCRIPTION
15-0	Parallel Port Termination

FIGURE 17 IP-BISERIAL-VI-BA27 TERMINATION BIT MAP

Each Bit of the Termination Control Register corresponds to a bit in the Parallel Port. Bit 0 is for IO 8 … Bit 15 for IO 23. When set the switch is programmed to terminate, when cleared the switch is programmed to not terminate the differential signal.

IPBISVI_BA27_DataTx

IPBISVI_BA27_DataTx 0x1C // 14

DATA BIT	DESCRIPTION
15-0	Parallel Port Transmit Data

FIGURE 18 IP-BISERIAL-VI-BA27 TX DATA BIT MAP

Each Bit of the Transmit Data Control Register corresponds to a bit in the Parallel Port. Bit 0 is for IO 8 … Bit 15 for IO 23. The IO on the FPGA are programmed to use the definitions in this register for each bit programmed to transmit. Any bits not programmed to transmit will remain in tri-state allowing input data on those bits.

IPBISVI_BA27_DataIO

IPBISVI_BA27_DataIO 0x1E // 15

DATA BIT	DESCRIPTION
15-0	Parallel Port IO Data

FIGURE 19 IP-BISERIAL-VI-BA27 IO DATA BIT MAP

Each Bit of the IO Data Control Register corresponds to a bit in the Parallel Port. Bit 0 is for IO 8 … Bit 15 for IO 23. The port reflects the state of the IO lines both defined as Tx and defined as Rx. The lines are registered to prevent metastable events while reading.

IPBISVI_BA27_TxFifoCnt

IPBISVI_BA27_TxFifoCnt 0x20 // 16

DATA BIT	DESCRIPTION
15-0	Tx FIFO Count

FIGURE 20 IP-BISERIAL-VI-BA27 TX FIFO COUNT

This read only port returns the current value of the Transmit FIFO. Count is zero extended to 16 bits.

IPBISVI_BA27_RxFifoCnt

IPBISVI_BA27_RxFifoCnt 0x22 // 17

DATA BIT	DESCRIPTION
15-0	Rx FIFO Count

FIGURE 21 IP-BISERIAL-VI-BA27 RX FIFO COUNT

This read only port returns the current value of the Receive FIFO. Count is zero extended to 16 bits.

IPBISVI_BA27_DelayCnt

IPBISVI_BA27_ DelayCnt 0x24,x26 // 18,19

DATA BIT	DESCRIPTION
23-0	Done Pulse Delay Count

FIGURE 22 IP-BISERIAL-VI-BA27 DP DELAY CNT

R/W port – 24 bits on adjacent 16 bit IP addresses to allow R/W as LW or as two 16 bit quantities. The count represents the delay from the last falling edge of the transmitted data clock to the rising edge of the Done Pulse. Valid for all values. Uses Tx Clock Reference as basis.

IPBISVI_BA27_PulseWidth

IPBISVI_BA27_PulseWidth 0x28 // 20

DATA BIT	DESCRIPTION
15-0	Done Pulse Width Definition

FIGURE 23 IP-BISERIAL-VI-BA27 DP PULSE WIDTH

R/W port – The count represents the Width of the Done Pulse when asserted. Uses Tx Clock Reference as basis. Hw will ignore if set to 0x00.

IPBISVI_BA27_DonePulseCntl

IPBISVI_BA27_DonePulseCntl 0x2A // 21

DATA BIT	DESCRIPTION
15-2	Spare
1	TxDoneEn - Set to Enable DonePulse
0	TxDoneAssert – SW force on

FIGURE 24 IP-BISERIAL-VI-BA27 DP CONTROL

TxDoneEn when set enables the Done Pulse to be asserted using the programmed

delay and width from ports DelayCount and PulseWidth. When not set, DonePulse is not asserted high.

TxDoneAssert when set requests the Tx StateMachine to transmit a DonePulse using the width parameter without transmitting any data or clock. The Tx StateMachine must not be enabled for transmission for this bit to work. This is a self-clearing bit – cleared after the transmission of the DonePulse.

Interrupts

IP-BiSerial-VI-USER interrupts are treated as auto-vectored. When the software enters into an exception handler to deal with an IP-BiSerial-VI-USER 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.

The interrupt is mapped to INT0 on the IP connector, which is mapped to a system interrupt via the host [carrier] device. The source of the interrupt is obtained by reading the Interrupt Status registers. The status remains valid until that bit in the status register is cleared.

When an interrupt occurs, the Master channel 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 status and enable the channel interrupt 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.

The base level has a master interrupt enable that affects all interrupt sources

Loop-back

The Engineering kit has reference software, which includes an external loop-back test. The test requires an external connections. We used IP-Debug-IO interconnected as shown below. *Table is for reference only – matches reference design – update to match your requirements*

From		**To**	
signal(port)	pins(P,N)	signal(port)	pins(P,N)
TxData(2)	3,4	RxData(0)	1,2
TxClk(3)	28,29	RxClk(1)	26,27
TxDP(5)	30,31	RxDP(4)	5,6
P0(8)	9,10	P8(16)	17,18
P1(9)	34,35	P9(17)	42,43
P2(10)	11,12	P10(18)	19,20
P3(11)	36,37	P11(19)	44,45
P4(12)	13,14	P12(20)	21,22
P5(13)	38,39	P13(21)	46,47
P6(14)	15,16	P14(22)	23,24
P7(15)	40,41	P15(23)	48,49

ID PROM

Every IP contains an ID PROM, whose size is at least 32 x 8 bits. The ID PROM aids in software auto configuration and configuration management. The user's software, or a supplied driver, may verify that the device it expects is actually installed at the location it expects, and is nominally functional. The ID PROM contains the manufacturing revision level of the IP. If a driver requires that a particular revision be present, it may check for it directly. The revision is also readable from the base revision register where both the major [reported in the PROM] and minor fields are available.

The location of the ID PROM in the host's address space is dependent on which carrier is used.

Standard data in the ID PROM on the IP-BiSerial-VI-BA27 is shown in the figure below. For more information on IP ID PROM's refer to the IP Module Logic Interface Specification.

Update the IDPROM to identify your design and invoke the generic driver support for your IP implementation. If you request we can assign a design number for your use.

Address	Data	
01	ASCII "I"	($49)
03	ASCII "P"	($50)
05	ASCII "A"	($41)
07	ASCII "H"	($48)
09	Manufacturer ID	($1E)
0B	Model Number	($0D) IP-BiSerial-VI
0D	Revision	($01)
0F	reserved	($01) Customer Number
11	Driver ID, low byte	($03) Design Number-BA27
13	Driver ID, high byte	($00)
15	No of extra bytes used	($0C)
17	CRC	($E5)

FIGURE 25 IP-BISERIAL-VI-USER ID PROM

IP-BiSerial-VI Logic Interface Pin Assignment

The figure below gives the pin assignments for the IP Module Logic Interface on the IP-BiSerial-VI. Pins marked n/c below are defined by the specification, but not used on the IP-BiSerial-VI. Also see the User Manual for your carrier board for more information.

GND	GND	1	26
CLK	+5V	2	27
Reset*	R/W*	3	28
D0	IDSEL*	4	29
D1	n/c	5	30
D2	MEMSEL*	6	31
D3	n/c	7	32
D4	INTSEL*	8	33
D5	n/c	9	34
D6	IOSEL*	10	35
D7	n/c	11	36
D8	A1	12	37
D9	n/c	13	38
D10	A2	14	39
D11	n/c	15	40
D12	A3	16	41
D13	INTREG0*	17	42
D14	A4	18	43
D15	n/c	19	44
BS0*	A5	20	45
BS1*	n/c	21	46
n/c	A6	22	47
n/c	Ack*	23	48
+5V	n/c	24	49
GND	GND	25	50

NOTE 1: The no-connect signals above are defined by the IP Module Logic Interface Specification, but not used by this IP. See the Specification for more information.

NOTE 2: The layout of the pin numbers in this table corresponds to the physical placement of pins on the IP connector. Thus this table may be used to easily locate the physical pin corresponding to a desired signal. Pin 1 is marked with a square pad on the IP Module.

FIGURE 26 IP-BISERIAL-VI LOGIC INTERFACE

IP-BiSerial-VI-USER IO Pin Assignment

The figure below gives the pin assignments for the IP Module IO Interface on the IP-BiSerial-VI-USER. Also see the User Manual for your carrier board for more information. IO#(signal name). Schematic shows IO#. Pairs are defined to be compatible with IP Standard Differential wiring – used on PCIe3IP, PCIe5IP etc.

IO0P(RxDataP)	IO1P(RxClkP)	1	26
IO0N(RxDataN)	IO1N(RxClkN)	2	27
IO2P(TxDataP)	IO3P(TxClkP)	3	28
IO2N(TxDataN)	IO3N(TxClkN)	4	29
IO4P(RxDpP)	IO5P(TxDpP)	5	30
IO4N(RxDpN)	IO5N(TxDpN)	6	31
IO6P(unusedP)	IO7P(unusedP)	7	32
IO6N(unusedN)	IO7N(unusedN)	8	33
IO8P(P0P)	IO9P(P1P)	9	34
IO8N(P0N)	IO9N(P1N)	10	35
IO10P(P2P)	IO11P(P3P)	11	36
IO10N(P2N)	IO11N(P3N)	12	37
IO12P(P4P)	IO13P(P5P)	13	38
IO12N(P4N)	IO13N(P5N)	14	39
IO14P(P6P)	IO15P(P7P)	15	40
IO14N(P6N)	IO15N(P7N)	16	41
IO16P(P8P)	IO17P(P9P)	17	42
IO16N(P8N)	IO17N(P9N)	18	43
IO18P(P10P)	IO19P(P11P)	19	44
IO18N(P10N)	IO19N(P11N)	20	45
IO20P(P12P)	IO21P(P13P)	21	46
IO20N(P12N)	IO21N(P13N)	22	47
IO22P(P14P)	IO23P(P15P)	23	48
IO22N(P14N)	IO23N(P15N)	24	49
IO_GND	IO_GND	25	50

NOTE: The layout of the pin numbers in this table corresponds to the physical placement of pins on the IP connector. Thus this table may be used to easily locate the physical pin corresponding to a desired signal. Pin 1 is marked on the IP Module.

FIGURE 27 IP-BISERIAL-VI-USER IO CONNECTOR PINOUT

IO_GND = AC / DC / Open based on J1 header shunt setting.

Applications Guide

Interfacing

The pin-out tables are displayed with the pins in the same relative order as the actual connectors. The pin definitions were chosen with noise immunity and cable compatibility in mind. The pairs should be connected with twisted pair wiring compatible with a 100 ohm system for best results.

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 485/LVDS devices rated voltages.

If induced noise is causing errors, please check the cabling and make sure the shields are properly tied to ground on one side. It may be necessary to go to higher grade cable. Please note that the shield ground on the card is connected to a header to allow user programming to a DC ground, AC [.1uF cap to ground] or open.

Construction and Reliability

IP Modules were conceived and engineered for rugged industrial environments. IP-BiSerial-VI is constructed out of 0.062 inch thick high temp ROHS compliant FR4 material.

Through hole and surface mounting of components are used.

The IP Module connectors are keyed and shrouded with Gold plated pins on both plugs and receptacles. They are rated at 1 Amp per pin, 200 insertion cycles minimum. These connectors make consistent, correct insertion easy and reliable.

The IP is secured against the carrier with four metric M2 stainless steel screws. The heads of the screws are countersunk into the IP. The four screws provide significant protection against shock, vibration, and incomplete insertion. For most applications they are not required.

The IP Module provides a low temperature coefficient of .89 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 IP. The coefficient means that if .89 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 IP-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 one degree differential temperature to the solder side of the board external cooling is easily accomplished.

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
831-457-4793 fax
support@dyneng.com

Specifications

Host Interface:	IP Module 8 and 32 MHz capable
IO Interface:	Serial Rx/Tx full duplex plus 16 bit parallel port
Tx Data rates generated:	PLLA and 50 MHz divided are available for Tx reference.
Software Interface:	Control Registers, Status Ports
Initialization:	Hardware Reset forces all registers [except vector] to 0.
Access Modes:	IO, Memory, ID, INT spaces (see memory map)
Wait States:	minimized based on programmed clock rate
Interrupt:	Programmable per channel/mode
Onboard Options:	Most Options are Software Programmable. Shunt for IO ground reference: open, DC, AC
Interface Options:	24 differential pairs plus reference on P2.
Dimensions:	Type II
Construction:	High temp ROHS compatible FR4 Multi-Layer Printed Circuit, Through Hole and SMT.
Temperature Coefficient:	.89 W/^{o}C for uniform heat across IP
Power:	Typical **180** mA @ 5V typical.
Temperature Range	–40C⇔85C or better rated components. Conformal Coating option for condensing environments

Order Information

IP-BiSerial-VI-USER	Reference design for IP-BiSerial-VI design. Requires at least one IP-BiSerial-VI board to be ordered. Standard IO = RS485.
-CC	Conformal Coating option
-ROHS	Change to ROHS processing. Without this option, standard leaded solder will be used.
-LVDS, MIXED	Additional IO Options for IP-BiSerial-VI.
Eng Kit–IP-BiSerial-VI	IP-Debug-IO - IO connector breakout IP-Debug-Bus - IP Bus interface extender

Note: The Engineering Kit is strongly recommended for first time purchases.