DYNAMIC ENGINEERING

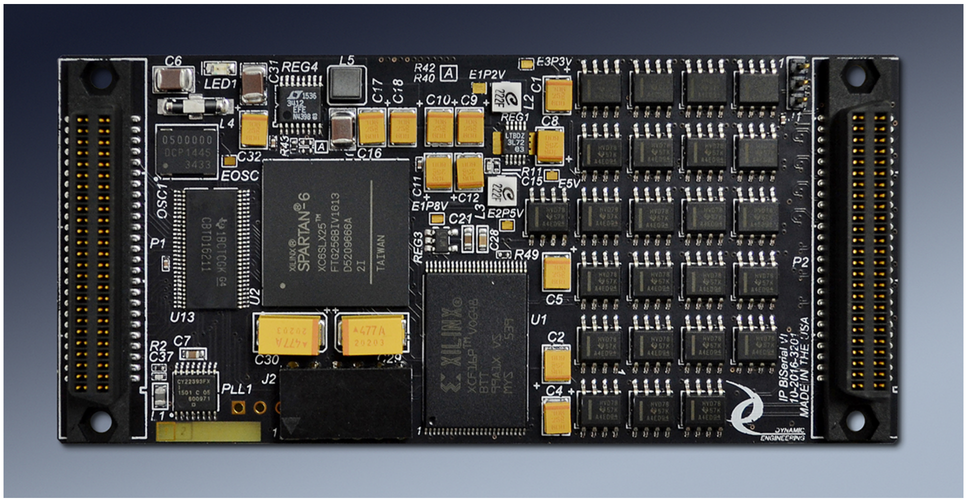
150 DuBois St. Suite C Santa Cruz, CA 95060
831-457-8891
sales@dyneng.com
www.dyneng.com
Est. 1988

User Manual

IP-BiSerial-VI-GPIO

Parallel interface with COS

IndustryPack® Module



Manual Revision: 1P1

Corresponding Hardware: Revision 01
FLASH revision: 0101 Fab: 10-2016-3201

IP-BiSerial-VI-GPIO
Serial Interface
IndustryPack® 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 IP Module carriers and compatible user-provided equipment. Connection of incompatible hardware is likely to cause serious damage.

Revised 6/2/20

Table of Contents

List of Figures

Product Description

IP-BiSerial-VI-GPIO provides 24 differential pairs organized as a parallel port. Each IO is independent with programming options for termination, direction, polarity, level or edge triggered with separate rising and falling enables. In addition, a choice of reference clocks with the local 50 MHz oscillator or programmable PLL. Filtered and direct IO access are also provided. Interrupts are programmable on a per line basis with 3 enables per line to cover the level, rising, and falling options. Registers are aligned with 32 bit accesses to allow 32 bit capable carriers to access directly.

Reference software for Win10 and Linux provide references for all of the modes of operation including setting up clocking, interrupts, using the parallel ports etc.

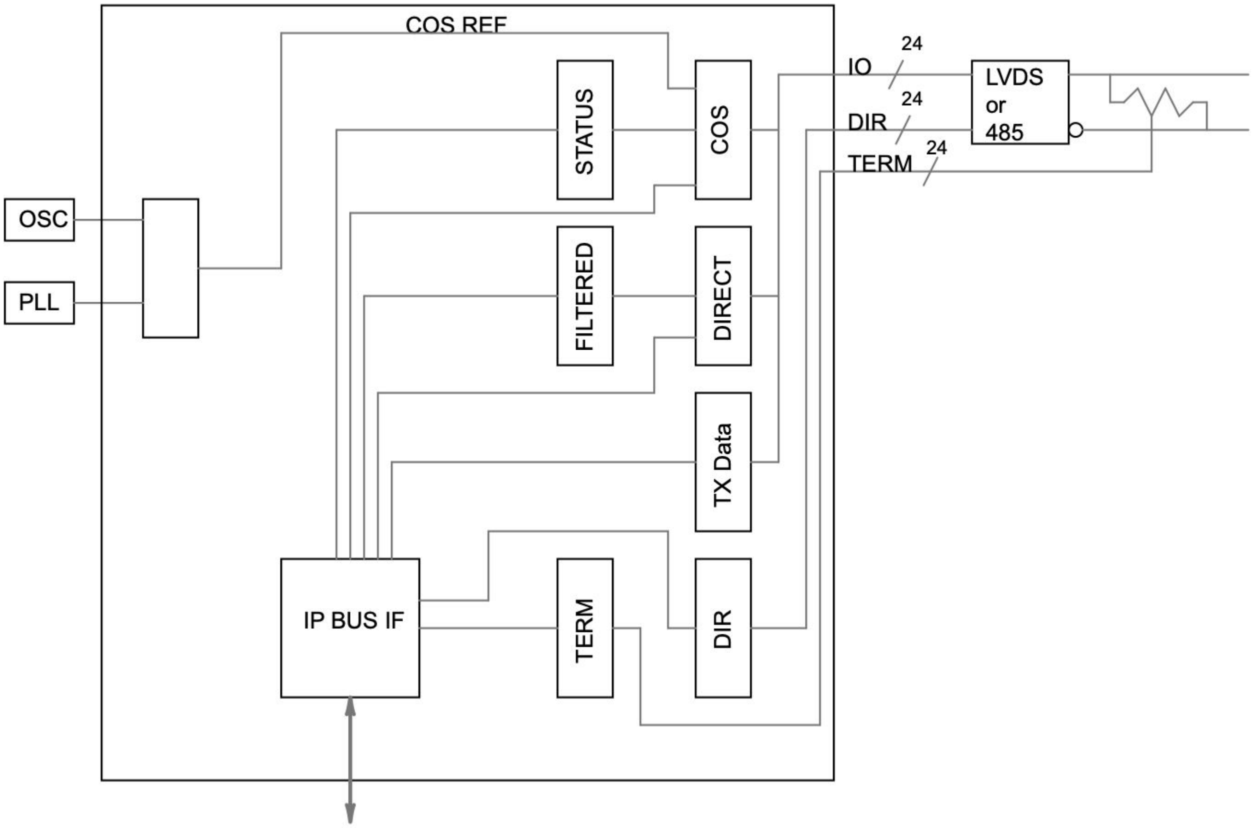


FIGURE 1 IP-BISERIAL-VI-GPIO FUNCTION

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.

GPIO is designed with a single level hierarchy – all logic at the Base level. The IP Bus interface and decoding, master interrupt control, PLL programming, clock generation, and Transmit and Receive logic.

The internal reference clock is selectable and can be derived from the 50 MHz oscillator or PLL.

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 is borrowed from PMC-BiSerial-VI.

IP-BISERIAL-VI-GPIO 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.

Each control is bitwise programmable for the IO. The IO controls are on LW boundaries to allow carriers with automatic LW conversion to have faster access times.

The 24 IO can be programmed to be transmitters or receivers with the direction selection. Termination is selectable per bit. Termination is accomplished with a 100 ohm resistor, and low impedance analog switch per bit.   Data from the Tx register is driven onto the lines where the direction is set to Tx.

To allow all 24 bits to be coordinated the base control register has an enable bit to allow the data to be written into the Tx local register and then moved to the output register at the same time.   This bit can be left in the enabled position if coordination between upper and lower is not required.

The received data can be read as Direct Data, Filtered Data, or COS status. Direct Data is the raw data after synchronization. Filtered Data has Polarity and EdgeLevel applied. Bits declared as edge are filtered out. Bits declared as inverted are reported after inversion. Active low signals on the IO can be read as active high in this manner.

The COS status is the captured change of state based on the settings in the enabled registers and the activity on the line. Separate Rising and Falling enables and status are supplied.

The sample rate for the COS is derived from the local oscillator or the PLL. The HalfDiv register is used to select the divisor to set the sample rate.

Interrupts are supported by GPIO. The interrupt occurs when the programmed event occurs. Status is provided in the Interrupt Status register – summary of Direct, COS Rising and COS Falling. 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 making it possible to operate in a polled mode. The COS interrupts are held until cleared by writing back to the respective status registers.

Product Details

IP-BiSerial-VI-GPIO is designed for the purpose of acting as a generic Parallel Port or monitoring data with Change Of State capabiity.

IP-BiSerial-VI-GPIO features a Xilinx FPGA. The FPGA contains all of the registers and protocol controlling elements of the BiSerial design.

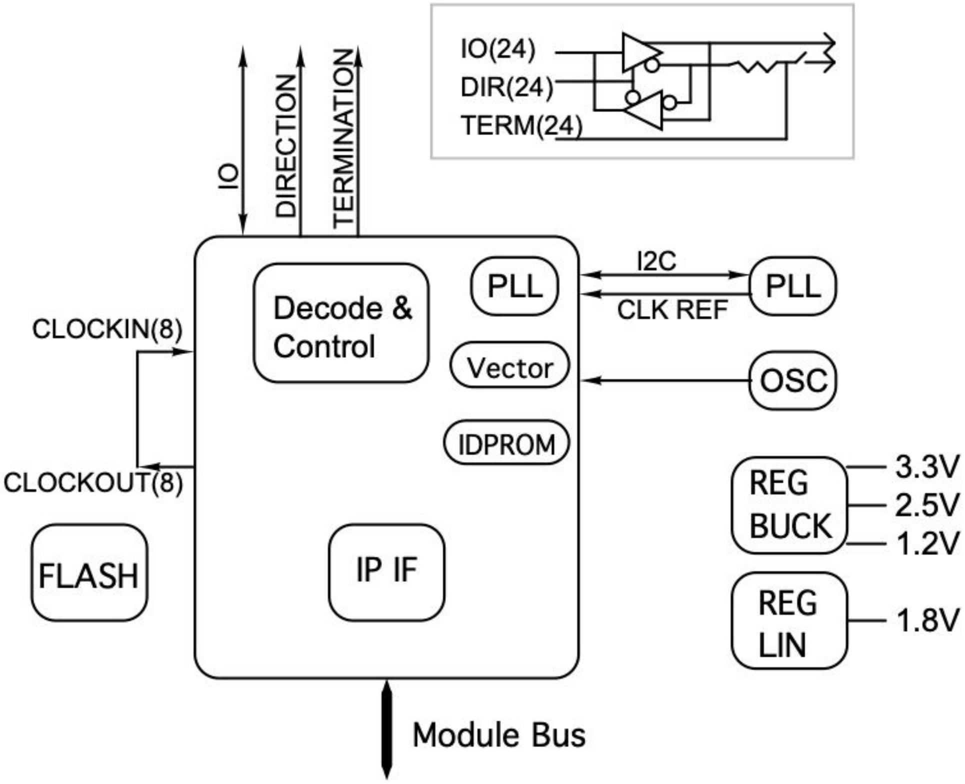


FIGURE 2 IP-BISERIAL-VI BLOCK DIAGRAM

IP-BiSerial-VI incorporates 24 differential pairs, each of which can be LVDS or RS485.

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 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.

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

GPIO is a flat design 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.

IP-BiSerial-VI-GPIO conforms to the IndustryPack® standard. This guarantees compatibility with compliant carrier boards. Because the IP-BiSerial-VI-GPIO 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-GPIO 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.

IP-BiSerial-VI-GPIO is tested with a combination of internal and external tests. The registers can be checked with R/W tests, the functions can be tested with the external loop-back fixture.

Interrupts are supported by IP-BiSerial-VI-GPIO. A force interrupt for software development and test is provided, plus an interrupt from multiple received data options. Status is provided separate from the interrupt to allow use in a polled environment.

Change of State or "COS" is used to detect the transition of a signal from high to low or vice-versa. Each IO has a separate detector for the rising edge and falling edge. If both are enabled COS is implemented. Otherwise technically, it is a rising or falling detector. The transitions are based on the external signal. With an active low signal and falling edge enabled the leading edge of an active transition would be selected independent of the polarity setting. (polarity is only applied to filtered data).

When the edge is detected, the corresponding status bit is set and remains set until explicitly cleared by writing to the status register with the corresponding bit set. Reset also clears stored status.

If the interrupt for active bit is enabled, the anded enable and status are ored together to create the rising or falling interrupt. If not enabled the status is available for polling. Please note the Master Interrupt Enable is also required for any of the programmed interrupts to create an interrupt to the host.

Level based interrupts are masked with the same interrupt enable bit and masked with the edgelevel selection. The Filtered data is the same data ahead of the interrupt enable being applied. Level data can also be polled in this manner. For level data the state of the IO line would need to change to remove the interrupt. In many cases the master or IO line level interrupt enable will need to be disabled until the line is in the ready state rather than the triggered state.

Address Maps

Address Map Base

```
IPBIS6GPIO_Base                 0x00  //0 Base & Tx control register
IPBIS6GPIO_MasterInt            0x02  //1 Master Interrupt Enable
IPBIS6GPIO_Vector               0x04  //2 Interrupt vector register
IPBIS6GPIO_IntStatus            0x06  //3 Interrupt status read
IPBIS6GPIO_Rev                  0x08  //4 Revision, Major&Minor, read only
IPBIS6GPIO_Info                 0x0A  //5 Spare port used by OS
IPBIS6GPIO_DataOut0             0x0C  //6 Tx Data 15-0
IPBIS6GPIO_DataOut1             0x0E  //7 Tx Data 23-16
IPBIS6GPIO_Direction0           0x10  //8 Direction 15-0
IPBIS6GPIO_Direction1           0x12  //9 Direction 23-16
IPBIS6GPIO_Termination0         0x14  //10 Termination 15-0
IPBIS6GPIO_Termination1         0x16  //11 Termination 23-16
IPBIS6GPIO_Polarity0            0x18  //12 Polarity 15-0
IPBIS6GPIO_Polarity1            0x1A  //13 Polarity 23-16
IPBIS6GPIO_EdgeLevel0           0x1C  //14 Edge or Level 15-0
IPBIS6GPIO_EdgeLevel1           0x1E  //15 Edge or Level 23-16
IPBIS6GPIO_IntEn0               0x20  //16 Interrupt Enable 15-0
IPBIS6GPIO_IntEn1               0x22  //17 Interrupt Enable 23-16
IPBIS6GPIO_Direct0              0x24  //18 Direct Read of IO 15-0
IPBIS6GPIO_Direct1              0x26  //19 Direct Read of IO 23-16
IPBIS6GPIO_CosRisingSt0         0x28  //20 COS Rising Status 15-0
IPBIS6GPIO_CosRisingSt1         0x2A  //21 COS Rising Status 23-16
IPBIS6GPIO_CosFallingSt0        0x2C  //22 COS Falling Status 15-0
IPBIS6GPIO_CosFallingSt1        0x2E  //23 COS Falling Status 23-16
IPBIS6GPIO_CosRisingEn0         0x30  //24 COS Rising Enable 15-0
IPBIS6GPIO_CosRisingEn1         0x32  //25 COS Rising Enable 23-16
IPBIS6GPIO_CosFallingEn0        0x34  //26 COS Falling Enable 15-0
IPBIS6GPIO_CosFallingEn1        0x36  //27 COS Falling Enable 23-16
IPBIS6GPIO_Filtered0            0x38  //28 Filtered Read of IO 15-0
IPBIS6GPIO_Filtered1            0x3A  //29 Filtered Read of IO 23-16
IPBIS6GPIO_HalfDiv              0x3C  //30 Set Divisor for Tx Rate
```

FIGURE 3 IP-BISERIAL-VI-GPIO ADDRESS MAP

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

Programming

Programming IP-BiSerial-VI-GPIO 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.   Channels when used, 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-GPIO "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.

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

The Dynamic Engineering drivers for Windows WFM (10) and Linux, manage the interaction with the carrier level driver. Please refer to the driver manuals for more information.   The manuals are downloadable from the webpage. Additional OS support can also be found on the website. Please contact Dynamic Engineering if you need additional SW options – VxWorks etc.

Register Definitions

IPBIS6GPIO_Base

0x0000 // 0 base control register offset

DATA BIT	DESCRIPTION
15	PLL_SDAT
14	PLL_SCLK
13	PLL_EN
12	Clock_COS_Sel
11	CLK8_32
10-5	Spare
4	ForceInt
3-2	Spare
1	Local Reset
0	LatchEn

FIGURE 4 IP-BISERIAL-VI-GPIO 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.

ForceInt: when set causes an interrupt to be generated to the system. Requires MasterIntEn to be enabled. Useful for debugging and software test.

Local Reset when ‘1’ causes an internal reset. ‘0’ for standard operation.

LatchEn when ‘1’ transfers data from the TX Data holding register to the output register. When ‘0’ data remains in the Tx Data register and is not propagated to the output register. Since it takes 2 - 16 bit writes to access all 24 bits of IO, LatchEn provides a mechanism to have all 24 update at the same time. In many cases LatchEn can be left in the enabled state. With 32 bit accesses the time between the 2 - 16 bit writes is << 1 uS [IP Clock rate dependent].

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. Please note: this means the PLL register bit is not returned when the port is read (since the PLL Data bit is returned in its place).

Clock_COS_Sel when ‘1’ selects the PLL as the reference for the local divider. When ‘0’ the local oscillator is used. See the HalfDiv register for setting the local divider. The PLL must be programmed. See the PLL section above. The oscillator reference [50 MHz] does not require added programming.

IPBIS6GPIO_MasterInt

0x0002 // 1 Master Interrupt Control Register

DATA BIT	DESCRIPTION
15-1	Spare
0	MasterIntEn

FIGURE 5 IP-BISERIAL-VI-GPIO MASTER INTERRUPT

MasterIntEn: when set allows programmed interrupt sources to assert Int0. Required for all interrupt types. When ‘0’ no interrupts will be requested, all status can be used for polling.

IPBIS6GPIO_VECTOR

0x0004 // 2 IP vector port

Vector Port

DATA BIT	**DESCRIPTION**
15-8	Undefined
7-0	vector

FIGURE 6 IP-BISERIAL-VI-GPIO 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-GPIO 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-GPIO.

Default is 0xFF for data.

IPBIS6GPIO_IntStatus

0x0006 // 3 Interrupt Status register

DATA BIT	DESCRIPTION
15	IntReqM
14-5	spare
4	ForceInt
3	LvlIntReq
2	CosFallingIntReq
1	CosRisingIntReq
0	IntReq0

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

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

IntReqM When IntReq0 is set and the Master Interrupt Enable is set this bit will be set.

CosRisingIntReq when Set indicates at least one Rising Status bit is set along with the corresponding interrupt enable. When cleared no interrupts are pending from this source.

CosFallingIntReq when Set indicates at least one Falling Status bit is set along with the corresponding interrupt enable. When cleared no interrupts are pending from this source.

LvlIntReq when Set indicates at least one bit declared as level, plus polarity, plus interrupt enable are true. When cleared no interrupts are pending from this source.

Note: COS interrupts are cleared by writing back to the Rising and or Falling status registers. Level interrupts are not cleared per se. Level interrupts are masked off until the IO line returns to the ready state, SW can then rearm the interrupt.

ForceInt is set when ForceInt is active in the base register. Slightly redundant having in both the base read and IntStatus read, and helpful for developing ISR routines to have the ForceInt behave as much like the IO interrupts as possible.

IPBIS6GPIO_REV

0x0008   //4 Revision Register

DATA BIT	DESCRIPTION
15-8	RevisionMajor
7-0	RevisionMinor

FIGURE 8   IP-BISERIAL-VI-GPIO 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.

Revisions (Major P Minor)
1p01 original – New Design 6/20

IPBIS6GPIO_INFO

0x000A // 5 Location Register

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

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

The location register can be updated by the carrier driver during initialization. IP-BiSerial-VI-GPIO 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-GPIO 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.

Data and Control registers are split into two 16 bit registers; shown as one 32 bit register in each of the following descriptions. The first offset points to the lower 15-0 data, the second points to the upper 31-16 data field. 23-16 are valid in the upper registers. Ports are accessible via 16 bit accesses or 32 bit transfers if the carrier supports this feature.

IPBIS6GPIO_DataOut0,1

x0C, x0E   // 6,7

DATA BIT	DESCRIPTION
31-0	Tx Data

FIGURE 10   IP-BISERIAL-VI-GPIO TX DATA

TX Data is transmitted on the IO lines when the corresponding direction bit is also set to transmit. Tx Data is a holding register. When written the data remains in this register until the LatchEn bit in the Base CNTL register is enabled. Data is transferred to the output register when LatchEn is set. LatchEn can be used to synchronize all 24 bits to be updated at once should that be necessary.

IPBIS6GPIO_Direction0,1

x10, x12 // 8,9 Direction Control Register

DATA BIT	DESCRIPTION
31-0	Direction

FIGURE 11 IP-BISERIAL-VI-GPIO DIRECTION BIT MAP

Each Bit of the Direction Control Register corresponds to a bit in the Parallel Port. 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.

IPBIS6GPIO_Termination0,1

x14, x16 // 10,11 Termination Control Register

DATA BIT	DESCRIPTION
31-0	Termination

FIGURE 12 IP-BISERIAL-VI-GPIO TERMINATION BIT MAP

Each Bit of the Termination Control Register corresponds to a bit in the Parallel Port. When set the analog switch is programmed to terminate, when cleared the switch is programmed to not terminate the differential signal.

IPBIS6GPIO_Polarity0,1

x18, x1A // 12,13 Polarity Control Register

DATA BIT	DESCRIPTION
31-0	Polarity

FIGURE 13 IP-BISERIAL-VI-GPIO POLARITY BIT MAP

Each Bit of the Polarity Control Register corresponds to a bit in the Parallel Port. When set the sense of the filtered data is inverted, when cleared the received data is left alone. For example, if the receive signal is active low the corresponding polarity bit is set to have it reflected as an active high signal in the filtered data. If the bit is to be used as an interrupt the enable is ANDed with the filtered data making the sense important to program. If not used for interrupt purposes it is up to the user to decide whether inversion is helpful or not. Please note: polarity does not affect the Direct Data port.

IPBIS6GPIO_EdgeLevel0,1

x1C, x1E // 14,15 Edge Level Select Control Register

DATA BIT	DESCRIPTION
31-0	EdgeLevel

FIGURE 14 IP-BISERIAL-VI-GPIO EDGELEVEL BIT MAP

Each Bit of the EdgeLevel Control Register corresponds to a bit in the Parallel Port. When set the bit is selected to be COS processed with Edge. When cleared the bit is selected for Level processing. The type of processing coupled with the interrupt enable selects how a particular IO is handled. The direct data is always available. The Rising and Falling status are always available. EdgeLevel is applied to filter which potential interrupters of each type are selected and the Interrupt Enable further selects which subset of those can actually generate an interrupt.

For example the direct data is filtered with EdgeLevel and only the level bits make it to the Filtered data [also treated with Polarity]. After the Filtered level the Interrupt Enable is applied to potentially allow interrupts from those sources.

IPBIS6GPIO_IntEn0,1

x20, x22 // 16,17 Interrupt Enable Control Register

DATA BIT	DESCRIPTION
31-0	Interrupt Enable

FIGURE 15 IP-BISERIAL-VI-GPIO INTEN BIT MAP

Each bit of the IntEn registers corresponds to an input bit. When set the IO can cause an interrupt. When cleared that IO is masked out. Please note: EdgeLevel selects which type of pre-processing is applied to the Input prior to the interrupt enable. Polarity, Rising, Falling are used to select the active level or edge to process.

IPBIS6GPIO_Direct0,1

x24, x26 // 18,19 Direct Data Read

DATA BIT	DESCRIPTION
31-0	Direct Data

FIGURE 16 IP-BISERIAL-VI-GPIO DIRECT BIT MAP

This is a read only register. The IO is synchronized and returned through this port. The state of the bit is the same as the encoded data on the differential pair. Polarity etc. is not applied at this port.

IPBIS6GPIO_CosRisingSt0,1

x28, x2A // 20,21 COS Rising Status

DATA BIT	DESCRIPTION
31-0	COS Rising Status

FIGURE 17 IP-BISERIAL-VI-GPIO RISING STATUS

The bits in the Rising Status register are set when the selected IO transitions from low to high. The programmed sample rate can affect how long it takes for the transition to be reflected in the status. See HalfDiv for more information. Please note: the corresponding CosRisingEn bit will need to be set in order to enable the capture of the corresponding rising edge. Status is cleared by writing to this port with the status bit to clear set.

IPBIS6GPIO_CosFallingSt0,1

x2C, x2E // 22,23 COS Falling Status

DATA BIT	DESCRIPTION
31-0	COS Falling Status

FIGURE 18 IP-BISERIAL-VI-GPIO FALLING STATUS

The bits in the Falling Status register are set when the selected IO transitions from high to low. The programmed sample rate can affect how long it takes for the transition to be reflected in the status. See HalfDiv for more information. Please note: the corresponding CosFallingEn bit will need to be set in order to enable the capture of the corresponding falling edge. Status is cleared by writing to this port with the status bit to clear set.

IPBIS6GPIO_CosRisingEn0,1

x30, x32 // 24,25 COS Rising Enable

DATA BIT	DESCRIPTION
31-0	COS Rising Enable

FIGURE 19 IP-BISERIAL-VI-GPIO RISING ENABLE

The bits in the Rising Enable register are set to enable rising edge detections of the corresponding IO bit. When cleared the rising edges of the IO bit are ignored.

IPBIS6GPIO_CosFallingEn0,1

x34, x36 // 22,23 COS Falling Enable

DATA BIT	DESCRIPTION
31-0	COS Falling Enable

FIGURE 20 IP-BISERIAL-VI-GPIO FALLING ENABLE

The bits in the Falling Enable register are set to enable falling edge detections of the corresponding IO bit. When cleared the falling edges of the IO bit are ignored.

IPBIS6GPIO_Filtered0,1

x38, x3A // 22,23 Filtered Data

DATA BIT	DESCRIPTION
31-0	Filtered Data

FIGURE 21 IP-BISERIAL-VI-GPIO FILTERED DATA

Read only port. The value of the port is the result of applying the EdgeLevel and Polarity to the IO. Please note: Direction is not a filter – transmitted bits can be filtered inputs if desired. The Polarity is applied on a bit by bit basis and then masked by the EdgeLevel definition to present the Level Selected data with the desired polarity. Please note: the output of the Filtered data is ANDed with the Interrupt Enable bits and then ORed together to create the level type interrupt request.

IPBIS6GPIO_HalfDiv

x3C // 30

DATA BIT	DESCRIPTION
15-0	HalfDiv

FIGURE 22 IP-BISERIAL-VI-GPIO 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 or PLLA in this design. The oscillator option is 50 MHz. PLLA is user programmable. See the Base Control register to select the reference for this module.

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. The first stage divider is preloaded with x1 and rolls over to x1 when the programmed count is reached.

Interrupts

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

An interrupt Vector can be read for systems requiring vectored interrupt processing.

The Windows and Linux SW packages provide both straight Interrupt Status read and ISR status read. The ISR when activated by an interrupt request, disables the master interrupt enable and reads the Rising, Falling, and Filtered data. The Rising and Falling status is then cleared within the handler. Reading the ISR structure returns all of the interrupt status saving multiple application level calls. The standard Interrupt Status read only returns the status without any other operations.

Loop-back

The Engineering kit has reference software, which includes an external loop-back test. The test requires external connections. We use IP-Debug-IO interconnected as shown below. There are 24 IO available.

A		**B**	
signal(port)	pins(P,N)	signal(port)	pins(P,N)
IO(0)	1,2	IO(12)	13,14
IO(1)	26,27	IO(13)	38,39
IO(2)	3,4	IO(14)	15,16
IO(3)	28,29	IO(15)	40,41
IO(4)	5,6	IO(16)	17,18
IO(5)	30,31	IO(17)	42,43
IO(6)	7,8	IO(18)	19,20
IO(7)	32,33	IO(19)	44,45
IO(8)	9,10	IO(20)	21,22
IO(9)	34,35	IO(21)	46,47
IO(10)	11,12	IO(22)	23,24
IO(11)	36,37	IO(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-GPIO is shown in the figure below. For more information on IP ID PROM’s refer to the IP Module Logic Interface Specification.

Address	Data	
01	ASCII "I"	($49)
03	ASCII "P"	($50)
05	ASCII "A"	($41)
07	ASCII "H"	($48)
09	Manufacturer ID	($1E) Dynamic Engineering
0B	Model Number	($0D) IP-BiSerial-VI
0D	Revision	($01) RevisionMajor
0F	reserved	($01) Customer Number
11	Driver ID, low byte	($04) Design Number-GPIO
13	Driver ID, high byte	($00)
15	No of extra bytes used	($0C)
17	CRC	($C8)

FIGURE 23 IP-BISERIAL-VI-GPIO ID PROM

IP-BISERIAL-VI-GPIO Logic Interface Pin Assignment

The figure below gives the pin assignments for the IP Module Logic Interface on the IP-BiSerial-VI-GPIO. Pins marked n/c below are defined by the specification, but not used on the IP-BiSerial-VI-GPIO. 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. 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 24 IP-BISERIAL-VI-GPIO LOGIC INTERFACE

IP-BISERIAL-VI-GPIO IO Pin Assignment

The figure below gives the pin assignments for the IP Module IO Interface on the IP-BISERIAL-VI-GPIO. 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	IO1P	1	26
IO0N	IO1N	2	27
IO2P	IO3P	3	28
IO2N	IO3N	4	29
IO4P	IO5P	5	30
IO4N	IO5N	6	31
IO6P	IO7P	7	32
IO6N	IO7N	8	33
IO8P	IO9P	9	34
IO8N	IO9N	10	35
IO10P	IO11P	11	36
IO10N	IO11N	12	37
IO12P	IO13P	13	38
IO12N	IO13N	14	39
IO14P	IO15P	15	40
IO14N	IO15N	16	41
IO16P	IO17P	17	42
IO16N	IO17N	18	43
IO18P	IO19P	19	44
IO18N	IO19N	20	45
IO20P	IO21P	21	46
IO20N	IO21N	22	47
IO22P	IO23P	23	48
IO22N	IO23N	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. 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 25 IP-BISERIAL-VI-GPIO 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 are conceived and engineered for rugged industrial environments. IP-BISERIAL-VI-GPIO is constructed out of 0.062 inch thick high temp ROHS compliant FR4 material.

Options are available for ROHS and standard processing

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-GPIO 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 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
support@dyneng.com

Specifications

Host Interface:	IP Module 8 and 32 MHz capable
IO Interface:	24 bit parallel port with COS.
COS sample rates generated:	50 MHz divided and PLL based are available for COS reference (SW Selectable).
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 bit/function
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/°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-GPIO	IP Module with 24 bit parallel port. Programmable COS or Level based interrupts and status, Direct and Filterered data. Bit level Direction, termination, Polarity, Rising and Falling control. RS-485 or LVDS.
-CC	Conformal Coating option
-ROHS	Change to ROHS processing. Without this option, standard leaded solder will be used.
-LVDS	RS-485 is standard for this design. To receive with LVDS IO add this option.
Eng Kit–IP-BISERIAL-VI-GPIO	IP-Debug-IO - IO connector breakout IP-Debug-Bus - IP Bus interface extender IP-BISERIAL-VI-GPIO Driver and reference software

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