**DYNAMIC ENGINEERING**
435 Park Dr., Ben Lomond, Calif. 95005
831-336-8891 **Fax** 831-336-3840
http://www.dyneng.com
sales@dyneng.com
Est. 1988

## User Manual

# IP-BiSerial-Q1

## Bi-directional Serial Data Interface IP Module

Revision A
Corresponding Hardware: Revision A

**IP-BiSerial-Q1**
Bi-directional Serial Data Interface
IP Module

Dynamic Engineering
435 Park Drive
Ben Lomond, CA 95005
831- 336-8891
831-336-3840 FAX
www.dyneng.com


Manual Revision A. Revised 03/12/03

# Table of Contents


**PRODUCT DESCRIPTION** 6

**THEORY OF OPERATION** 8

**ADDRESS MAP** 10

**PROGRAMMING** 11

**REGISTER DEFINITIONS** 13
BIS_CNTL0 13
BIS_CNTL1 14
BIS_CNTL2 15
BIS_VECTOR 16
BIS_STAT0 17
BIS_STAT1 17
BIS_CNTL11 18
BIS_CNTL15 19
BIS_FTX_W 19
BIS_FTX_R 20
BIS_FRX_W 20
BIS_FRX_R 20
PAE PAF PROGRAMMING 21

**INTERRUPTS** 21

**ID PROM** 23

**LOOP-BACK** 24

**IP MODULE IO INTERFACE PIN ASSIGNMENT** 26

**APPLICATIONS GUIDE** 27

**Interfacing** 27

**Construction and Reliability** 28

**Thermal Considerations** 29

**WARRANTY AND REPAIR** **29**

**Service Policy** **30**
Out of Warranty Repairs 30

**For Service Contact:** **30**

**SPECIFICATIONS** **31**

**ORDER INFORMATION** **32**

# List of Figures


| | | |
|---|---|---|
| FIGURE 1 | IP-BISERIAL-Q1 SERIAL PROTOCOL TIMING | 9 |
| FIGURE 2 | IP-BISERIAL-Q1 INTERNAL ADDRESS MAP | 10 |
| FIGURE 3 | IP-BISERIAL-Q1 CONTROL REGISTER 0 BIT MAP | 13 |
| FIGURE 4 | IP-BISERIAL-Q1 CONTROL REGISTER 1 BIT MAP | 14 |
| FIGURE 5 | IP-BISERIAL-Q1 CONTROL REGISTER 2 BIT MAP | 15 |
| FIGURE 6 | IP-BISERIAL-Q1 STATUS REG 0 BIT MAP | 17 |
| FIGURE 7 | IP-BISERIAL-Q1 STATUS REG 1 BIT MAP | 17 |
| FIGURE 8 | IP-BISERIAL-Q1 CONTROL REGISTER 11 BIT MAP | 18 |
| FIGURE 9 | IP-BISERIAL-Q1 CONTROL REGISTER DATA DELAY BIT MAP | 19 |
| FIGURE 10 | IP-BISERIAL-Q1 ID PROM | 23 |
| FIGURE 11 | IP-BISERIAL-Q1 LOGIC INTERFACE | 25 |
| FIGURE 12 | IP-BISERIAL-Q1 IO INTERFACE | 26 |

# Product Description

IP-BISERIAL is part of the IP Module family of modular I/O components. The IP-BISERIAL is capable of providing multiple serial protocols. The standard protocol implemented provides a Data, Clock and Strobe interface with Ready control. The –Q1 version is a custom modification featuring the use of a pulsed gated clock, and 16 or 32 bit data with parity.

In addition to the Q1 version other custom interfaces are available. Please see our web page for current protocols offered. If you do not find it there we can re-design the state machines and create a custom interface protocol. That protocol will then be offered as a “standard” special order product. Please contact Dynamic Engineering with your custom application.

The IP-BISERIAL-Q1 supports both 8 and 32 MHz IP Bus operation. The IP Clockis used to derive the reference clocks for the serial operation. Please be sure to select the proper clock divisors and source selector after reset to insure proper operation. Please refer to the programming section for details.

Both single ended and differential I/O are available on the serial signals. The differential drivers and receivers conform to the RS-485 specification (exceeds RS-422). The RS-485 input signals are terminated with 180Ω. The single ended driver signal is characterized as an open drain driver with 24 mA of sink. For convenience A 2KΩ pull-up is supplied on board, for faster termination a second pull-up can be added at the receiving end of the circuit. Single ended signals are received through 33Ω resistors. Care should be taken with the single ended signals, as transients can damage the board.

All configuration registers support read and write operations for maximum software convenience. Word and byte operations are supported (please refer to the memory map).

The IP-BISERIAL-Q1 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 1K by 16 bit FIFOs, which support word accesses. An on-board read/write path exists for loop-back testing.

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 first. Each word has a parity bit is 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 bit.

The IP-BISERIAL-Q1 utilizes the IP clock as a source for the clock generator. A programmable divider creates variable clock rates from the base rates. Please refer to the clock selection part of the programming section for details.

Interrupts are supported by the IP-BISERIAL-Q1. The interrupt occurs at the end of the transmission when 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. The status is available for the FIFOs making it possible to operate in a polled mode.

# Theory of Operation

The IP-BISERIAL-Q1 is designed for the purpose of transferring data from one point to another with a serial protocol.

The IP-BISERIAL-Q1 features a Xilinx FPGA. The FPGA contains all of the registers and protocol controlling elements of the BISERIAL design. Only the drivers, receivers, boot PROM and FIFOs are external to the Xilinx device.

The IP-BISERIAL-Q1 is a part of the IP Module family of modular I/O products. It meets the IP Module Vita Standard. In standard configuration it is a Type 1 mechanical with no components on the back of the board and one slot wide. Contact VITA for a copy of this specification. It is assumed that the reader is at least casually familiar with this document and logic design.

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, the IP-BISERIAL-Q1 is capable of supporting 16+ MB per second data transfer rate with a 32 MHz reference rate.

The serial I/O can support many protocols. The Q1 timing for a 16-bit transfer is shown in figure 1. The clock is gated and normally high. The data is valid on the rising edge of the clock. The base specification requires a 16 uS period with 4 uS low on the clock and 1 uS hold on the data after the rising edge of the clock. The design is implemented with the data shifted to provide 8 uS of set-up and 8 uS of hold when running with the 16 uS clock period. The Transmit state machine uses the programmed reference clock running at 4X the transmit rate to create the 3 periods of high and one of low time. If a frequency of 250 KHz is specified at the clock generator then the output rate will be the 62.5 KHz. Specified. Alternate rates can be achieved with the change of the base frequency. For example with a 2 Mhz reference, a 500 KHz transmit rate will result and still have the 1 uS of set-up and hold requirement.

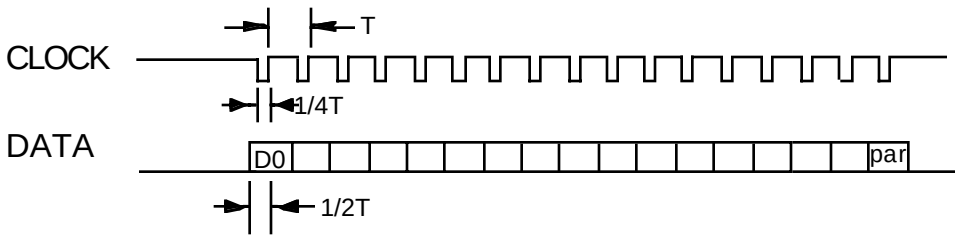


FIGURE 1 IP-BISERIAL-Q1 SERIAL PROTOCOL TIMING

The receive channel uses the IP clock to sample the incoming clock and look for a period where the clock is high for longer than the bit period. 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 not currently being received the hardware begins to look for an active clock. When the clock is determined to have gone low and then high the received data is captured by the shift register. After17 active edges have been found the data is stored into a secondary latch, the parity is tested, and the data moved to the storage FIFO. If the mode is set to 32 bits then a second word is “looked” for without waiting for a new gap between words. When the 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 then the hardware will see the start when it returns to idle and immediately look for a data gap and then to capture more data. In the continuous mode the proper size data can be captured and an interrupt stream sent to the host.

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 FPGA, and the FPGA and the data buffers. 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 bit data writes to IP slots performing an auto-conversion to 16 bits saving 1 / 2 of the data transfers. The PCI3IP and PCI5IP support auto-incrementing and static address conversion from 32 to 16 bits.

# Address Map

| Function | | Offset | Width | Type |
|---|---|---|---|---|
| BIS_CNTL0 | EQU | $00 | byte on word boundary | read / write |
| BIS_CNTL1 | EQU | $02 | byte on word boundary | read / write |
| BIS_CNTL2 | EQU | $04 | byte on word boundary | read / write |
| BIS_VECTOR | EQU | $06 | byte on word boundary | read / write |
| BIS_STAT0 | EQU | $08 | word | read |
| BIS_STAT1 | EQU | $0A | byte on word boundary | read / write |
| BIS_cntl11 | EQU | $16 | word | read / write |
| BIS_datadel | EQU | $1E | word | read / write |
| BIS_FTX_W | EQU | $10 | word | write |
| BIS_FTX_R | EQU | $22 | word | read |
| BIS_FRX_W | EQU | $20 | word | write |
| BIS_FRX_R | EQU | $40 | word | read |

FIGURE 2 IP-BISERIAL-Q1 INTERNAL ADDRESS MAP

The address map provided is for the local decoding performed within the IP-BISERIAL-Q1. The addresses are all offsets from the IO space base address. The BIS_FTX_W and BIS_FRX_R ports are also mapped to the entire MEM address space and the BISERIAL_IDPROM is mapped to the ID space. The carrier board that the IP is installed into provides these addresses.

# Programming

Programming the IP-BISERIAL-Q1 requires only the ability to read and write data in the host's I/O space. The base address is determined by the IP Carrier board. This documentation refers to the address where the IO space for the slot that the IP is installed in begins as the base address.

In order to receive data the software is only required to enable the Rx state machine, program the mode and FIFOs. If desired, the interrupt can be enabled and the interrupt vector written to the vector register. Data will be loaded into the FIFOs as it is received. Other options include automatic clearing of the start bit at the end of a message.

A typical sequence would be to first write to the vector register with the desired interrupt vector. For example $40 is a valid user vector for the Motorola 680x0 family. Please note that some carrier boards do not use the interrupt vector. The interrupt service routine should be loaded and the mask should be set. When the start bit is set the hardware looks to make sure that the sync is not active then begins looking for it to be active. In this manner the data received is protected from receiving partial messages. Once a new strobe assertion is detected the data loading process begins. When the word count is reached the interrupt request is asserted to let the host know that the data is available. The software can set-up a loop counter to efficiently read the data from the FIFOs. If the auto-clear feature is enabled, the Rx start bit will be cleared when the message completes and will have to be re-written before another message can be received. If continuous data is expected the PAF interrupt may be more interesting than the end of reception interrupt. Using the PAF interrupt would allow larger data blocks to be moved from the –Q1 to host memory.

The end of transmission interrupt will indicate to the software that an outgoing message has been started and that the message has terminated. If both the Tx and Rx interrupts are enabled then the SW needs to read BIS_STAT1 to see which source caused the interrupt. In order to avoid missing an interrupt, the bits in this register must be explicitly cleared by writing the appropriate bit as a '1'. The INTACK cycle will clear the actual interrupt. The interrupt status should be cleared after the INTACK cycle in order to avoid resetting the status bit. It is a good idea to clear the status bits before Start is enabled to insure that the RX_INT or TX_INT value read by the interrupt service routine came from the current reception.

Before transmitting data the FIFOs are enabled and the data loaded. The baud rate selections are used to select the output rate. The divided version

or the base rate can be selected for the reference rate on the Tx state machine. The transmission rate is 1 / 4 of the reference rate. Once the complete message is loaded and the controls set properly the start bit can be set to cause the transfer to begin. If a slow clock rate is selected and a long message is to be sent then data can be loaded during transmission to save operational time. When the Tx interrupt is received the transmission has been completed. The transmit state machine sends either one or two words based on the mode. The FIFO can be pre-loaded with multiple messages which are then sent when enabled.

The FIFOs are 1K x16 allowing 1K words to be stored. The transmission length is determined by the mode. On the receive side the PAF can be used to provide an almost full interrupt to allow interrupt driven long message capability. The continuous mode would need to be selected for this feature to be useful.

Refer to the Theory of Operation section above and the Interrupts section below for more information regarding the exact sequencing and interrupt definitions.

Initialization. The software is required to properly reset the FIFOs prior to use. The FIFOs are pointer based and if they are not properly reset the data transmitted may not match the data that was loaded. The basic procedure is : select the reference clock for the FIFO to be the IP clock, disable the FIFO, then enable the FIFO. Please note that the LD controls affect the mode that the FIFO enters after reset. Please see the register bit-maps for more information.

# REGISTER DEFINITIONS

## BIS_CNTL0

$00 BISERIAL Control Register Port read / write

| CONTROL REGISTER 0 | |
|---|---|
| **DATA BIT** | **DESCRIPTION** |
| 7 | spare |
| 6 | TX Odd Even Parity |
| 5 | TXmode16_32 |
| 4 | spare |
| 3 | INT FORCE |
| 2 | INT ENABLE FAE |
| 1 | INT ENABLE TX |
| 0 | START TX |

FIGURE 3 IP-BISERIAL-Q1 CONTROL REGISTER 0 BIT MAP

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

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

2. INT ENABLE FAE 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 to the almost empty count. The interrupt is cleared by an Interrupt Acknowledge cycle or disabling the interrupt.

3. INT FORCE is used to create an interrupt for test and software development purposes. Set the bit to cause the interrupt and clear the bit to remove the interrupt. Requires the master interrupt enable to be '1' to have effect.

5. TX mode16_32 when '1' selects 32 bit transmission mode and when '0' selects 16 bit mode. Data is sent lsb first and with parity attached for 17 or 34 bits total.

6. When the TX_ODD_EVEN bit = '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.

## BIS_CNTL1

$02 BISERIAL Control Register Port read / write

**CONTROL REGISTER 1**

| DATA BIT | DESCRIPTION |
|---|---|
| 7 | ENABLE FIFO |
| 6 | RX Odd Even Parity |
| 5 | RX mode16_32 |
| 4 | RX AUTO-CLEAR ENABLE |
| 3 | spare |
| 2 | INT ENABLE FAF |
| 1 | INT ENABLE RX |
| 0 | STRT_RX |

FIGURE 4 IP-BISERIAL-Q1 CONTROL REGISTER 1 BIT MAP

0. START_RX is used to enable the receive state machine to receive messages. If the auto-clear function is enabled, the start bit is auto-cleared at the end of a received message.

1. INT ENABLE RX is used to enable the receive interrupt. The default is disabled. If enabled and the master interrupt enable is also enabled then an interrupt is requested when the Strobe returns to the off state. The interrupt is cleared by an Interrupt Acknowledge cycle or disabling the interrupt.

2. INT ENABLE FAF is used to enable the Programmable Almost Full interrupt. The default is disabled. If enabled and the master interrupt enable is also enabled then an interrupt is requested when FIFO fills to the almost full condition. The interrupt is cleared by an Interrupt Acknowledge cycle or disabling the interrupt.

4. RX AUTO-CLEAR ENABLE when '1' enables the clearing of the start bit at the end of a message. When set to '0' the receiver will stay enabled at

the end of a 16 or 32 bit transfer and receive a new data transfer without software intervention.

5. RX mode16_32 when '1' selects 32 bit reception mode and when '0' selects 16 bit mode. Data is received lsb first and with parity attached for 17 or 34 bits total.

6. When the RX_ODD_EVEN bit = '1' odd parity is selected for the receiver. The parity bit is appended to the data received and is related to the data by an XOR function. The parity is calculated for the incoming word based on the odd-even selection and tested against the parity received with the data. If the parity bit does not match the parity calculated an error is flagged. The error is set in the Status Register (1). This setting should match the transmitter that the receiver is connected to for proper operation.

7. ENABLE FIFO is used to reset the FIFOs. The default state is reset and the FIFOs must be enabled to be used to store data. This signal must be set high then low then high while the FIFO clock is running to cause a proper reset with guaranteed flag operation. We suggest using the IP clock as the reference for this process. Please refer to FTX_LD and FRX_LD (control reg 2).

## BIS_CNTL2

$04 BISERIAL Control Register Port read/write

**CONTROL REGISTER 2**

| DATA BIT | DESCRIPTION |
|---|---|
| 6 | MASTER INT ENABLE |
| 5 | FRX_LD |
| 4 | FTX_LD |
| 3 | spare |
| 2 | spare |
| 1 | spare |
| 0 | RX_TTL_422 |

FIGURE 5 IP-BISERIAL-Q1 CONTROL REGISTER 2 BIT MAP

0. RX_TTL_422 is the control bit to select whether the data, clock, and strobe should be received as TTL or 422 signals. Default state = 0 = 422 [485] a '1' selects TTL inputs.

4,5. FRX_LD is tied to the RX FIFO WE2/_LD pin. FTX_LD is tied to the Tx FIFO WE2/_LD pin. When the FIFOs are taken out of reset it is possible to

set-up the FIFO to accept commands to program the levels at which the programmable almost empty and programmable almost full signals operate. ***In the standard transfer mode these pins are set hi before CLR_FIFO is released to use as a second WE control pin***. If the PAE and PAF flags are used at a different than default depth then the flags will require programming. The default is 7 from full or 7 before empty. Please refer to the PAE PAF Programming section or the Cypress data sheet for the proper sequence.

6. MASTER INT ENABLE is the master interrupt enable. Default is 0. If set to 1 then the RX, TX, PAE, or PAF interrupts can occur based on individual interrupt enables. If the master interrupt enable is off [0] then no interrupts will be generated. The status register can still see the interrupt requests to allow polled operation.

**BIS_VECTOR**

$06 BISERIAL Interrupt Vector Port read / write
The interrupt vector for the BISERIAL is stored in this byte wide register. This read / write register is initialized to 0XFF upon power-on reset or software reset. The vector is stored in the odd byte location [D7..0]. The vector should be initialized before the interrupt is enabled or the mask is lowered. The interrupt is automatically cleared when the CPU acknowledges the interrupt.

## BIS_STAT0

$08 BISERIAL Status Port [read only]

| Data Bit | Status | |
|---|---|---|
| 7 | FTX_FF | 1 = transmit FIFO full |
| 6 | FTX_PAE | 1 = transmit FIFO almost empty |
| 5 | FTX_MT_1 | 1 = transmit FIFO 1 empty |
| 4 | FTX_MT_0 | 1 = transmit FIFO 0 empty |
| 3 | FRX_FF_0 | 1 = receive FIFO 0 full |
| 2 | FRX_FF_1 | 1 = receive FIFO 1 full |
| 1 | FRX_PAF | 1 = receive FIFO almost full |
| 0 | FRX_MT | 1 = receive FIFO empty |

FIGURE 6 IP-BISERIAL-Q1 STATUS REG 0 BIT MAP

7-0. The FIFO flags are active high. When the empty bit is '1' then the FIFO is empty. When the empty flag is '0' then the FIFO has at least one piece of data stored. When the Full Flag is set the FIFO is full. When not set then the FIFO still has room

## BIS_STAT1

$0A BISERIAL Status Port [read only]

| Data Bit | Status | |
|---|---|---|
| 8 | INT REQUEST | 1 = interrupt pending |
| 7 | Parity Error | 1 = error detected |
| 6 | Overrun Error | 1 = error detected |
| 5 | gnd | set to '0' |
| 4 | gnd | set to '0' |
| 3 | FAF_INT | 1 = interrupt pending |
| 2 | FAE_INT | 1 = interrupt pending |
| 1 | RX_INT | 1 = Interrupt pending |
| 0 | TX_INT | 1 = Interrupt pending |

FIGURE 7 IP-BISERIAL-Q1 STATUS REG 1 BIT MAP

0-3. RX_INT, TX_INT, FAF_INT, FAE_INT are set when the respective interrupt conditions exist and the interrupts are enabled. The master interrupt can be disabled and still have the benefit of the status.

6-7. Overrun, and Parity errors are tested for when a reception is in

progress. If detected then the status bit is set and the reception continues. Overrun is set when the FIFO is full at the time data needs to be written to the FIFO. Parity is set if the parity calculated on the received data does not match the parity received.

8. INT REQUEST is high when an interrupt condition exists. This bit is not masked by the MASTER INT ENABLE.

Note: All the status bits in this register are individually clearable by writing a '1' to the corresponding bit.

## BIS_CNTL11

$16 BISERIAL Control Register Port read / write

| CONTROL REGISTER 11 | |
|---|---|
| **DATA BIT** | **DESCRIPTION** |
| 14-13 | CLOCK PRE-SELECTOR |
| 12 | CLOCK POST-SELECTOR |
| 11-0 | DIVISOR |

FIGURE 8 IP-BISERIAL-Q1 CONTROL REGISTER 11 BIT MAP

CLOCK PRE-SELECTOR

| | |
|---|---|
| 00 | IP Clock |
| 01 | Oscillator |
| 10 | Oscillator |
| 11 | IP Clock |

The clock pre-selector is used to select which reference clock to use with the divisor hardware (clock source). Please note that the Q1 design is hardwired to use the IP clock as a reference. The Preselector has no effect. The table is included for customer reference in case an oscillator or other clock is desired in the future.

POST-SELECTOR when '1' sets the output clock to the divided clock, when '0' sets the output clock to the pre-selector reference value (clock source). This bit is normally set to '1' for -Q1 use.

DIVISOR[11-0] are the clock divisor select bits. The clock source is divided by a 12-bit counter. The output frequency is {reference / [2(n+1)]}, $n \geq 1$. The reference oscillator is not installed. The counter divides by N+1 due to counting from 0 to n before rolling over. The output is then divided by 2 to produce a square wave output.

The desired frequency of 250 KHz. Is achieved by selecting IP reference, divided clock and a factor of 32. 2(N+1) = 32 => N = 15. 0x100F would be the correct value to write to the Clock Control Register [Bis_cntl11].

Please note that the standard 485 buffers are rated for 10 MHz. With most systems the larger divisors will be used. The smaller divisors are provided for use with external oscillators and the external clock line. Faster 485 buffers [40 MHz] are available by special order.

## BIS_CNTL15

$1E BISERIAL Data Delay Control Register Port read / write

| CONTROL REGISTER 11 | |
|---|---|
| **DATA BIT** | **DESCRIPTION** |
| 15-0 | datadel |

FIGURE 9 IP-BISERIAL-Q1 CONTROL REGISTER DATA DELAY BIT MAP

The value in BIS_CNTL15 will control the number of clocks [IP] that are counted with the received clock in the high state before the received clock is determined to be in a gap between words. We used 1.5 periods for this value in our testing. Please note the value will change based on 8 or 32 MHz IP reference rate and the actual clock rate in use. With 16 uS system period and 32 MHz IP clock the value would be 768.
(16*1.5) / [.0313]

## BIS_FTX_W

$10 BISERIAL TX FIFO Write

By writing a to this address data is loaded into the Tx FIFO. In addition a write to the Memory Space (any address) will also write to the Tx FIFO. The memory space write is a useful feature for software that would naturally auto-increment the address, or systems where the addresses are auto-incremented. This is the mechanism used in the auto-start mode. The write address is used to determine when sufficient data has been written to complete a message.

### BIS_FTX_R

$22 BISERIAL TX FIFO Read

A loop-back path is provided for the Tx FIFO to allow the host to read the data stored in the Tx FIFO. Reading from this address fetches data from the Tx FIFO. The clock must be set to IP for source and '0' for the post-selector before reading from this register. *Once the data is read from the FIFO the data is no longer available for transmission*.

### BIS_FRX_W

$20 BISERIAL RX FIFO Write

A loop-back path is provided for the Rx FIFO to allow the host to load data into the Rx FIFO. Writing to this address loads data into the Rx FIFO. This operation competes with and should not be performed during normal operation. The clock selector (BIS_CNTL2 bit7) needs to be set to internal for the Rx reference clock.

### BIS_FRX_R

$40 BISERIAL RX FIFO Read

The data stored in the Receive FIFO can be accessed through this port. A read from the Memory space (any address) will also access this port. Use the memory space for 32 bit auto-incrementing accesses and faster transfers if your carrier supports that. Non-compelled DMA is also an option with the MEM space access.

## PAE PAF PROGRAMMING

To use the PAF and PAE flags with a setting other than the default of seven, the FIFO must be programmed. The FIFOs are configured as two 8-bit data paths in parallel The upper byte is where the flags are generated [D15-8], the lower byte flags are not used. This data path carries the programming information when the FIFO is taken out of reset with the LD controls set low. The Rx FIFO is written with the loop-back path and requires the clock settings for loop-back. The Rx and Tx FIFO can be programmed using the following procedure.

¥ reset theFIFOs and place into the programmable flag mode
¥ program the PAE/PAF flag to trigger at new value
¥ put into operational mode

¥ set LD control low with enable hi — set low for flag levels to be programmed
¥ set LD control low with enable low
¥ set LD control low with enable hi to clear the FIFO with the WE control in the load mode

¥ program the PAE [and PAF] to be at the new value
¥ first write with LD low = LSB of PAE, second = upper bits of PAE
third = LSB of PAF and 4th = upper bits of PAF
¥ raise LD Control hi to resume normal FIFO operation

# INTERRUPTS

All IP Module interrupts are vectored. The vector from the IP-BISERIAL-Q1 comes from a vector register loaded as part of the initialization process. The vector register can be programmed to any 8 bit value. The default value is $FF which is sometimes not a valid user vector. The software is responsible for choosing a valid user vector.

The IP-BISERIAL-Q1 state-machines generate an interrupt request when a transmission or reception is complete and the enable bit in the control registers are set. The transmission is considered complete when the last bit is clocked out. The interrupt is mapped to interrupt request 0. The CPU will respond by asserting INT. The hardware will automatically supply the appropriate interrupt vector. The source of the interrupt is obtained by reading BIS_STAT1. The status remains valid until the status register bits are cleared. The interrupt remains asserted until the status is cleared, or the interrupt enables are disabled.

The interrupt level seen by the CPU is determined by the IP Carrier board being used. The master interrupt can be disabled or enabled through the BIS_CNTL2 register. The individual enables for Tx and Rx are controllable through BIS_CNTL0 and BIS_CNTL1. The enables operate after the

interrupt holding latch, which stores the request for the CPU. Once the interrupt request is set, the way to clear the request is to reset the board, service the request, or disable the interrupt. Toggling the interrupt enable low will clear the interrupt, but the STAT1 bits must be cleared before the Master interrupt enable is re-asserted or another interrupt will be generated. The interrupt enables can be set back to enabled immediately after clearing the status bits. INT ENABLE TX enables and clears the Tx interrupt and INT ENABLE RX enables and clears the Rx interrupt.

If operating in a polled mode and making use of the interrupts for status then the interrupts should be disabled and the status register polled. When BIS_STAT1 shows an interrupt pending the appropriate FIFO action can take place then BIS_STAT1 is written to clear that “interrupt request” before starting the next transfer.

Power on initialization will provide a cleared interrupt request, interrupts disabled, and interrupt vector of $FF.

The programmable interrupts operate in much the same way. The programmable interrupts are triggered by FIFO level instead of Tx / Rx completion. The interrupts are cleared with the individual enables or writing to the status register. The levels are programmable to allow the software to respond before the FIFO is empty or full so that longer transfers can be handled without using larger FIFOs.

## 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 location of the ID PROM in the host's address space is dependent on which carrier is used. Normally the ID PROM space is directly above the IPs I/O space, or at IP-base + $80. Macintosh drivers use the ID PROM automatically.

Standard data in the ID PROM on the IP-BISERIAL-Q1 is shown in the figure below. For more information on IP ID PROMs refer to the IP Module Logic Interface Specification, available from Dynamic Engineering.

Each of the modifications to the IP-BiSerial-IO board will be recorded with a new code in the DRIVER ID location. Q1 is set to 0x01 with a customer number of 0x6.

| 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 | ($01) |
| 0D | Revision | ($A0) |
| 0F | reserved | ($06) |
| 11 | Driver ID, low byte | ($01) |
| 13 | Driver ID, high byte | ($00) |
| 15 | No of extra bytes used | ($0C) |
| 17 | CRC | ($BA) |

FIGURE 10 IP-BISERIAL-Q1 ID PROM

## Loop-back

The Engineering kit has reference software, which includes an external loop-back test. The test requires an external cable with the following pins connected.

| | |
|---|---|
| Data+ | 8 - 20 |
| Data- | 9 - 21 |
| Clock+ | 11 - 23 |
| Clock- | 12 – 24 |

## IP Module Logic Interface Pin Assignment

The figure below gives the pin assignments for the IP Module Logic Interface on the IP-BISERIAL-Q1. Pins marked n/c below are defined by the specification, but not used on the IP-BISERIAL-Q1. 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 | | n/c | | 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 11 IP-BISERIAL-Q1 LOGIC INTERFACE

# IP Module IO Interface Pin Assignment

The figure below gives the pin assignments for the IP Module IO Interface on the IP-BISERIAL-Q1. Also see the User Manual for your carrier board for more information.

| | | | | | |
|---|---|---|---|---|---|
| GND | n/c | | 1 | 26 | |
| REFCLK_422+ | | n/c | 2 | | 27 |
| REFCLK_422- | GND | | 3 | 28 | |
| GND | | n/c | 4 | | 29 |
| n/c | GND | | 5 | 30 | |
| n/c | | rx data_TTL | 6 | | 31 |
| GND | GND | | 7 | 32 | |
| TXDATA + | | rx clk_TTL | 8 | | 33 |
| TXDATA - | GND | | 9 | 34 | |
| GND | | n/c | 10 | | 35 |
| TXCLK + | GND | | 11 | 36 | |
| TXCLK - | | n/c | 12 | | 37 |
| GND | GND | | 13 | 38 | |
| n/c | | GND | 14 | | 39 |
| n/c | n/c | | 15 | 40 | |
| GND | | GND | 16 | | 41 |
| n/c | GND | | 17 | 42 | |
| n/c | | tx dataTTL | 18 | | 43 |
| GND | GND | | 19 | 44 | |
| RXDATA + | | GND | 20 | | 45 |
| RXDATA - | tx clk TTL | | 21 | 46 | |
| GND | | GND | 22 | | 47 |
| RXCLK + | GND | | 23 | 48 | |
| RXCLK - | | n/c | 24 | | 49 |
| GND | GND | | 25 | 50 | |

NOTE 1: 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 12 IP-BISERIAL-Q1 IO INTERFACE

Q1 uses the 422 versions of data, clock, and strobe. Tx refers to the transmit and Rx refers to Receive relative to the BiSerial board. N/C refers to pins connected to active devices not used in this implementation. Most carriers connect the IP IO 1:1 to the connectors on the carrier. Please check with your carrier for connector mapping.

# Applications Guide

## Interfacing

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 IP-OctalSerial-FCS 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. The RS-485 transcievers can handle relatively large offsets in voltage and is designed to operate when parts of the system are powered and parts are not. It is better design practice to keep the voltage offsets minimized and the potential for current flowing though un-powered electronics to a minimum.

**Keep cables short**. Flat cables, even with alternate ground lines, are not suitable for long distances. IP-BISERIAL-Q1 does not contain special input protection.

**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, by applying voltage less than ground or more than +5 volts with the IP powered. With the IP unpowered, driven input voltages should be kept within .7 volts of ground potential.

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

Many flat cable interface products are available from third party vendors to assist you in your system integration and debugging. These include connectors, cables, test points, 'Y's, 50 pin in-line switches, breakout boxes, etc.

## Construction and Reliability

IP Modules were conceived and engineered for rugged industrial environments. The IP-BISERIAL-Q1 is constructed out of 0.062 inch thick FR4 material.

Through hole and surface mounting of components are used. IC sockets use gold plated screw machine pins. High insertion and removal forces are required, which assists in the retention of components. If the application requires unusually high reliability or is in an environment subject to high vibration, the user may solder the corner pins of each socketed IC into the socket, using a grounded soldering iron.

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 0.89 W/$^{o}$C for uniform heat disipation. 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 0.89 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 BISERIAL 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 then 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

Dynamic Engineering warrants this product to be free from defects in workmanship and materials under normal use and service and in its original, unmodified condition, for a period of one year from the time of purchase. If the product is found to be defective within the terms of this warranty, Dynamic Engineering's sole responsibility shall be to repair, or at Dynamic Engineering's sole option to replace, the defective product. The product must be returned by the original customer, insured, and shipped prepaid to Dynamic Engineering. All replaced products become the sole property of Dynamic Engineering.

Dynamic Engineering's warranty of and liability for defective products is limited to that set forth herein. Dynamic Engineering disclaims and excludes all other product warranties and product liability, expressed or implied, including but not limited to any implied warranties of merchandisability or fitness for a particular purpose or use, liability for negligence in manufacture or shipment of product, liability for injury to persons or property, or for any incidental or consequential damages.

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.

## 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. The current minimum repair charge is $100. 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
435 Park Dr.
Ben Lomond, CA 95005
831-336-8891
831-336-3840 fax
Internet Address support@dyneng.com

# Specifications

| | |
|---|---|
| Logic Interface: | IP Module Logic Interface |
| Serial Interface: | RS-485 TX Data, TX Clk, RX Data, RX ClK and TTL equivalents |
| Tx CLK rates generated: | IP CLK, plus 12 bit divider to create output frequencies |
| Software Interface: | Control Registers, ID PROM, Vector Register, Status Ports, FIFOs |
| Initialization: | Hardware Reset forces all registers to 0 except the Vector Register which resets to 0XFF. |
| Access Modes: | Word in IO Space (see memory map)<br>Word in ID Space<br>Word or LW in Memory space<br>Vectored interrupt |
| Access Time: | back-to-back cycles in 500ns (8MHz.) or 125 nS (32 MHz.) to / from FIFO |
| Wait States: | 1 to ID space, 2 to IO, MEM, or INT space except for loop-back FIFO access |
| Interrupt: | Tx interrupt at end of transmission<br>Rx interrupt at end of reception<br>Programmable Almost Empty<br>Programmable Almost Full |
| DMA: | No Logic Interface DMA Support implemented at this time<br>Memory space non-compelled supported to FIFOs. |
| Onboard Options: | All Options are Software Programmable |
| Interface Options: | 50 pin flat cable<br>50 screw terminal block interface<br>User cable |
| Dimensions: | Standard Single IP Module. 1.8 x 3.9 x 0.344 (max.) inches |
| Construction: | FR4 Multi-Layer Printed Circuit, Through Hole and Surface Mount Components. Programmable parts are socketed. |
| Temperature Coefficient: | 0.89 W / $^{O}$C for uniform heat across IP |
| Power: | Max. **220** mA @ 5V |

# Order Information

| | |
|---|---|
| IP-BISERIAL-Q1 | IP Module with 1 Tx and 1 Rx serial channel,<br>Programmable data rates<br>Q1 protocol support,<br>RS-485 drivers and receivers<br>16 bit IP interface |
| Tools for IP-BISERIAL-Q1 | IP-Debug-Bus - IP Bus interface extender<br>http://www.dyneng.com/ipdbgbus.html<br>IP--Debug-IO - IO connector breakout<br>http://www.dyneng.com/ipdbgio.html<br>HDRterm50 50 position terminal block breakout from ribbon cable<br>http://www.dyneng.com/HDRterm50.html |
| Eng Kit–IP-BISERIAL | IP-Debug-IO - IO connector breakout<br>IP-Debug-Bus - IP Bus interface extender<br>Technical Documentation,<br>1. IP-BISERIAL Schematic<br>2. IP-BISERIAL Reference test software<br>Data sheet reprints are available from the manufacturer's web site<br>reference software. |