

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-Miller

Bi-directional Serial Data Interface
IP Module

Revision A

Corresponding Hardware: Revision A

IP-BiSerial-Miller 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

©2003 by Dynamic Engineering.
IndustryPack is a registered trademark of GreenSpring
Computers Inc.
Other trademarks and registered trademarks are owned by their
respective manufactures.
Manual Revision A. Revised 10/31/03

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.



Table of Contents

PRODUCT DESCRIPTION	6
THEORY OF OPERATION	9
ADDRESS MAP	12
PROGRAMMING	13
REGISTER DEFINITIONS	15
BIS_CNTL0	15
BIS_CNTL1	17
BIS_CNTL2	18
BIS_VECTOR	19
BIS_STAT0	20
BIS_STAT1	21
BIS_STAT2	22
BIS_FTX_W	22
BIS_CNTL9	22
BIS_CNTL11	22
BIS_CNTL12, 13	23
BIS_FRX_W	24
BIS_FRX_R	24
BIS_FRX_R	24
PAE_PAF PROGRAMMING	25
INTERRUPTS	25
ID PROM	27
LOOP-BACK	28
IP MODULE LOGIC INTERFACE PIN ASSIGNMENT	29
IP MODULE IO INTERFACE PIN ASSIGNMENT	30



<u>APPLICATIONS GUIDE</u>	31
<u>Interfacing</u>	31
<u>Construction and Reliability</u>	32
<u>Thermal Considerations</u>	33
<u>WARRANTY AND REPAIR</u>	33
<u>Service Policy</u>	34
<u>Out of Warranty Repairs</u>	34
<u>For Service Contact:</u>	34
<u>SPECIFICATIONS</u>	35
<u>ORDER INFORMATION</u>	36



List of Figures

FIGURE 1	IP-BISERIAL-MLR SERIAL PROTOCOL TIMING	10
FIGURE 2	IP-BISERIAL-MLR INTERNAL ADDRESS MAP	12
FIGURE 3	IP-BISERIAL-MLR CONTROL REGISTER 0 BIT MAP	15
FIGURE 4	IP-BISERIAL-MLR CONTROL REGISTER 1 BIT MAP	17
FIGURE 5	IP-BISERIAL-MLR CONTROL REGISTER 2 BIT MAP	18
FIGURE 6	IP-BISERIAL-MLR STATUS REG 0 BIT MAP	20
FIGURE 7	IP-BISERIAL-MLR STATUS REG 1 BIT MAP	21
FIGURE 8	IP-BISERIAL-MLR CONTROL REGISTER 11 BIT MAP	22
FIGURE 9	IP-BISERIAL-MLR ID PROM	27
FIGURE 10	IP-BISERIAL-MLR LOGIC INTERFACE	29
FIGURE 11	IP-BISERIAL-MLR IO INTERFACE	30

Product Description

The 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. This version is a custom modification featuring Miller data encoding.

In addition to the -Mlr [Miller] version other custom interfaces are available. Please see our web page for current protocols offered. If you do not find what you want 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-Mlr supports both 8 and 32 MHz IP Bus operation. The on-board oscillator is 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 $2K\Omega$ 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 22Ω 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-Mlr 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 8K by 16-bit FIFOs, which support word accesses. An on-board read/write path exists for loop-back testing.

The serial receive channel first looks for a 2-clock interval, which represents a '0', '1' sequence. The '1' is entered in the receive shift register and subsequent bits are shifted in until a match of the 32-bit sync pattern is recognized. At this point the sync pattern and all following data are stored in the Rx FIFO.

Unless the frame check is disabled, the sync pattern is expected to be seen at the beginning of each following frame. If it is not, a framing error is latched and the receiver goes back to looking for the sync pattern. The frame length defaults to 64 32-bit words unless a different value is entered in the frame length register.

The host can poll or wait for the message complete or FIFO almost-full interrupt. The message is read directly from the input FIFO. Error conditions are checked including framing and FIFO over-run.

The serial transmit channel reads data from the output FIFO and sends it out serially, msb first. Data can be pre-stored in the FIFO and sent later with the Tx start bit.

The IP-BiSerial-Mlr utilizes the on-board oscillator 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-Mlr. An interrupt occurs at the end of the transmission when data is received or sent or both. The programmable FIFO level 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 data level status is available for the FIFOs making it possible to operate in a polled mode.





Theory of Operation

The IP-BiSerial-Mlr is designed for the purpose of transferring data from one point to another with a Miller encoded serial protocol.

The IP-BiSerial-Mlr 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-Mlr is 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-Mlr 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 -Mlr timing is shown in figure 1. Miller code data is determined solely by the interval between signal transitions. The only valid intervals are one, one and one-half, and two clock periods. An interval of two clocks signifies a data sequence of '0', '1'. This is the only interval that is not dependent on prior data and is therefore used to begin data decoding. An interval of one and one-half clocks indicates a '1' if the previous data was a '0' or a '0', '0' sequence if the previous data was a '1'. An interval of one clock indicates that the current data-bit is the same as the previous one.



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.

The IP-BiSerial-MlR receiver uses a 10x clock to detect signal transition intervals in the received data stream. An interval of up to 12 10x clocks is interpreted as a one-clock interval. An interval of 13-17 10x clocks is interpreted as a one and one-half clock interval. An interval of 18-31 clocks is interpreted as a two-clock interval. And an interval of 32 or more clocks is interpreted as a no-data condition and will cause the receive state machine to terminate the current reception.

The transmit state machine uses a programmed reference clock running at 2x the transmit data rate to create the required signal intervals. The user programs the clock control for the 10x clock rate and this clock is automatically divided-by 5 to generate the Tx clock. When the transmit state machine is started, data is read out of the FIFO and encoded into the Miller protocol and sent out on the TTL and RS-422 TxData lines. This process continues until there is no more data in the FIFO at which point the Tx interrupt occurs, if enabled, and the transmit state machine stops. If messages longer than the Tx FIFO size are desired, the Tx almost empty interrupt can be used to signal the loading of additional data before the FIFO becomes empty.

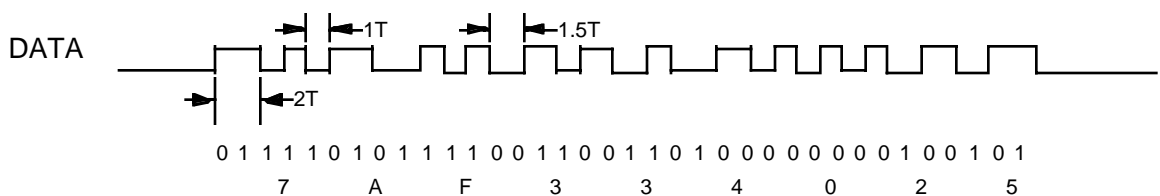


FIGURE 1 IP-BISERIAL-MLR SERIAL PROTOCOL TIMING

When the receive state machine detects a no-data condition, an Rx interrupt will occur, if enabled. This signals the user to read the received data from the Rx FIFO. Messages larger than The FIFO size can be handled



using the Rx almost full interrupt to read the data from the Rx FIFO before it becomes full, which causes a data overrun error condition to occur.

The FIFOs level status is also available from status register 0 if it is desired to run in a polled mode without interrupts

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_CNTLO	EQU \$00	word	read/write
BIS_CNTL1	EQU \$02	word	read/write
BIS_CNTL2	EQU \$04	word	read/write
BIS_VECTOR	EQU \$06	byte on word boundary	read/write
BIS_STAT0	EQU \$08	word	read
BIS_STAT1	EQU \$0A	word	read/write
BIS_STAT2	EQU \$0C	word	read
BIS_CNTL9	EQU \$12	word	read/write
BIS_CNTL11	EQU \$16	word	read/write
BIS_SYNC_LO	EQU \$18	word	read/write
BIS_SYNC_HI	EQU \$1A	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-MLR INTERNAL ADDRESS MAP

The address map provided is for the local decoding performed within the IP-BiSerial-Mlr. 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-Mlr 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. In this document, the base address refers to the starting address of the IO space of the slot in which the IP is installed.

In order to receive data the software is only required to enable the Rx state machine and FIFOs, configure the clock rate and optionally the frame size and sync pattern. 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 reception and enabling the state machine to only look for the sync pattern to initiate a reception.

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 interrupt mask set. When the start bit is set the hardware looks for a two-clock signal interval and then for a sync pattern match. In this manner the data received is protected from receiving partial messages. Once the sync pattern is detected the data loading process begins. When the Rx data stops, 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 IP-BiSerial-Mlr 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'. 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 transmission rate is 1/10 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, data can be loaded during transmission to save operational time. When the Tx interrupt is received the transmission has been completed.

The FIFOs are 8K x16 allowing 8K words to be stored. On the receive side the PAF can be used to provide an almost full interrupt to allow interrupt driven long message capability.

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 to be the IP clock, disable the FIFO, then enable the FIFO. Please note that the LD controls affect the mode that the FIFOs operate in after reset. Please see the register bit-maps for more information.



REGISTER DEFINITIONS

BIS_CNTLO

\$00 BISERIAL Control Register Port read/write

CONTROL REGISTER 0	
DATA BIT	DESCRIPTION
9	AUX 3 DIR
8	AUX 3 OUT
7	AUX 2 DIR
6	AUX 2 OUT
5	AUX 1 OUT
4	AUX 0 OUT
3	INT FORCE
2	INT ENABLE FAE
1	INT ENABLE TX
0	START TX

FIGURE 3

IP-BISERIAL-MLR 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 clearing the interrupt status latch 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 clearing the interrupt status latch 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 an effect.

5. AUX 0 OUT sets the level of the Tx clock output. Since no clock or strobe is transmitted in this design, the unused signals become auxiliary data signals that may be used for whatever the user desires.

6. AUX 1 OUT sets the level of the Tx strobe output. (see above)

7. AUX 2 OUT sets the level of the Tx ready output.

8. AUX 2 DIR sets the direction of the Tx ready RS-422 output. When this bit is a '1' the output is configured as a driver, when '0' it is a receiver.

9. AUX 3 OUT sets the level of the Rx ready output.

10. AUX 3 DIR sets the direction of the Rx ready RS-422 output. When this bit is a '1' the output is configured as a driver, when '0' it is a receiver. Note: Since the RxReady TTL and 422 outputs are driven from the same pin and the TTL signal is always an output, the TTL output is not valid when this bit is a '0'.



BIS_CNTL1

\$02 BISERIAL Control Register Port read/write

CONTROL REGISTER 1	
DATA BIT	DESCRIPTION
7	ENABLE FIFO
6	spare
5	RX INIT SYNC
4	RX AUTO-CLEAR ENABLE
3	spare
2	INT ENABLE FAF
1	INT ENABLE RX
0	START_RX

FIGURE 4

IP-BISERIAL-MLR 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 reception completes. The interrupt is cleared by clearing the interrupt status latch 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 clearing the interrupt status latch 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 reception and receive a new data transfer without software intervention.

5. RX INIT SYNC when '1' looks for the sync pattern only at the start of a message reception. If this bit is a '0' then a sync pattern is expected to begin each data frame.



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_422_TTL

FIGURE 5

IP-BISERIAL-MLR CONTROL REGISTER 2 BIT MAP

0. RX_422_TTL is the control bit to select whether the data should be received as TTL or 422 signals. Default state = 0 = TTL, a '1' selects 422 [485] 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 will still report 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.



BIS_STATO

\$08 BISERIAL Status Port [read only]

Data Bit	Status	
15	AUX 6 IN	Rx_Ready422
14	AUX 5 IN	Tx_Ready422
13	AUX 4 IN	Rx_Strobe422
12	AUX 3 IN	Rx_Clock422
10	AUX 2 IN	Rx_StrobeTTL
9	AUX 1 IN	Rx_ClockTTL
8	AUX 0 IN	Tx_ReadyTTL
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-MLR 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

15-12, 10-8 Auxiliary inputs: Since the IP-BiSerial-Mlr design uses only the data lines, the other serial connections have been assigned to Status bits in this register to read the input levels received. Note that the Tx and Rx Ready422 lines are valid only when the respective direction bit in the Tx control register is set to be an input.



BIS_STAT1

\$0A BISERIAL Interrupt/Error Status Port read/write

Data Bit	Status	
8	INT REQUEST	1 = interrupt pending
7	Frame Error	1 = error detected
6	Overrun Error	1 = error detected
5	Missed Count Error	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-MLR STATUS REG 1 BIT MAP

O-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 this status.

5. Missed Count Error is set to a '1' if a new Rx word count is latched before the old count has been read.

6-7. Overrun, and Frame 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. Frame is set if the sync pattern is not seen at the beginning of a data frame.

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_STAT2

\$0C BISERIAL Rx Word Count Status Port [read only]

Reading from this address returns the number of 32-bit words received in the last data message.

BIS_FTX_W

\$10 BISERIAL TX FIFO Write

By writing 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.

BIS_CNTL9

\$12 BISERIAL Frame Size Register Port read/write

This register specifies the number of 32-bit words in each data frame. The sync pattern is expected to begin each frame. If it is not seen and the init sync control bit is not set in the Rx configuration register, a framing error will be latched. If this register contains zeros (the reset condition), the default value of 64 (0x40) will be used.

BIS_CNTL11

\$16 BISERIAL Control Register Port read/write

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

FIGURE 8

IP-BISERIAL-MLR CONTROL REGISTER 11 BIT MAP

CLOCK PRE-SELECTOR

0	IP Clock
1	Oscillator



The clock pre-selector is used to select which reference clock to use with the divisor hardware (clock source). This bit is normally set to '1' for IP-BiSerial-Mlr use except during FIFO reset.

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 IP-BiSerial-Mlr use except during FIFO reset.

DIVISOR[11:0] are the clock divisor select bits. The clock source is divided by a 12-bit counter. The output frequency is $\{\text{reference} / [2^{(n+1)}]\}$, $n \geq 1$. 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 1.28 MHz (128 KHz. data rate) is achieved by selecting Oscillator reference, divided clock, and a factor of 16. $2^{(N+1)} = 16 \Rightarrow N = 7$. 0x3007 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_CNTL12, 13

\$18, 1A BISERIAL Sync Pattern Register Ports read/write

The 32-bit sync pattern is entered in these two registers. The lower 16 bits are contained in BIS_CNTL12 and the upper 16 bits are in BIS_CNTL13. If these registers contain zeros (the reset condition), the default value of 0x7af33402) will be used.



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 Rx reference clock needs to be set to the IP clock (BIS_CNTL11 bits 12, 13 = '0').

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_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 the FIFOs 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-MIr 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-MIr 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 clear the interrupt status latch or disable the interrupt. The STAT1 bits must be cleared or the particular interrupt disabled 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 the Tx interrupt and INT ENABLE RX enables 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 FIFO interrupts operate in much the same way. The programmable interrupts are triggered by FIFO level instead of Tx / Rx completion. The interrupts are cleared by 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 a particular device 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 IP's 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-Mlr 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. -Mlr is set to 0x01 with a customer number of 0x7.

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	(\$07)
11	Driver ID, low byte	(\$01)
13	Driver ID, high byte	(\$00)
15	No of extra bytes used	(\$0C)
17	CRC	(\$EB)

FIGURE 9

IP-BISERIAL-MLR ID PROM



Loop-back

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

422Data+	8 - 20
422Data-	9 - 21
TTLData	43 - 31

Auxiliary bits

422Clock+	11 - 23
422Clock-	12 - 24
TTL Clock	46 - 33
422Strobe+	14 - 26
422Strobe- -	15 - 27
TTLStrobe	49 - 35
422Ready+	17 - 5
422Ready- -	18 - 6
TTLReady	29 - 37



IP Module Logic Interface Pin Assignment

The figure below gives the pin assignments for the IP Module Logic Interface on the IP-BISERIAL-Mlr. Pins marked n/c below are defined by the specification, but not used on the IP-BISERIAL-Mlr. 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	
DO		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		INTREGO*		17	42	
D14		A4		18	43	
D15		n/c		19	44	
BSO*		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 10

IP-BISERIAL-MLR 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-Mlr. Also see the User Manual for your carrier board for more information.

GND	RXSTB +	1	26
REFCLK_422+	TXSTB -	2	27
REFCLK_422-	GND	3	28
GND	RXRDYTTL	4	29
TXRDY +	GND	5	30
TXRDY -	RXDATATTTL	6	31
GND	GND	7	32
TXDATA +	RXCLKTTL	8	33
TXDATA -	GND	9	34
GND	RXSTBTTL	10	35
TXCLK +	GND	11	36
TXCLK -	TXRDYTTL	12	37
GND	GND	13	38
TXSTB +	GND	14	39
TXSTB -	n/c	15	40
GND	GND	16	41
RXRDY +	GND	17	42
RXRDY -	TXDATATTTL	18	43
GND	GND	19	44
RXDATA +	GND	20	45
RXDATA -	TXCLKTTL	21	46
GND	GND	22	47
RXCLK +	GND	23	48
RXCLK -	TXSTBTTL	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 11

IP-BISERIAL-MLR IO INTERFACE

Mlr uses the TTL version of data. 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-BiSerial-Mlr 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 transceivers can handle relatively large offsets in voltage and are 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 through un-powered electronics to a minimum.

Keep cables short. Flat cables, even with alternate ground lines, are not suitable for long distances. The IP-BiSerial-Mlr 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-Mlr 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/°C for uniform heat disipation. This is based upon the temperature coefficient of the base FR4 material of 0.31 W/m-°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:	TTL TX Data, RX Data RS-485 equivalents. Four auxiliary RS-485, I/O, Three auxiliary TTL I/O.
Tx CLK rates generated:	IP CLK, Oscillator 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) Word in ID Space Word or LW in Memory space 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 Rx interrupt at end of reception Programmable Almost Empty Programmable Almost Full
DMA:	No Logic Interface DMA Support implemented at this time Memory space non-compelled supported to FIFOs.
Onboard Options:	All Options are Software Programmable
Interface Options:	50 pin flat cable 50 screw terminal block interface 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/°C for uniform heat across IP
Power:	Max. 220 mA @ 5V



Order Information

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

All information provided is Copyright Dynamic Engineering

