

# **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-LM1**

## **Bi-directional Serial Data Interface IP Module**

Revision A  
Corresponding Hardware: Revision A

**IP-BiSerial-LM1**  
**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

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.

©1997,1998,1999,2001,2002 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 Mar 28th, 2002



---

---

# Table of Contents

---

---

<b>PRODUCT DESCRIPTION</b>	<b>6</b>
<b>THEORY OF OPERATION</b>	<b>8</b>
<b>ADDRESS MAP</b>	<b>11</b>
<b>PROGRAMMING</b>	<b>12</b>
<b>Register Definitions</b>	<b>14</b>
BIS_CNTL0	14
BIS_CNTL1	15
BIS_CNTL2	16
BIS_Vector	17
BIS_STAT0	17
BIS_STAT1	18
BIS_STAT2	19
BIS_CNTL7	19
BIS_CNTL11	20
BIS_FTX_W	21
BIS_FTX_R	21
BIS_FRX_W	21
BIS_FRX_R	22
PAE PAF Programming	22
<b>Interrupts</b>	<b>22</b>
<b>ID PROM</b>	<b>24</b>
<b>LOOP-BACK</b>	<b>25</b>
<b>IP MODULE IO INTERFACE PIN ASSIGNMENT</b>	<b>27</b>
<b>APPLICATIONS GUIDE</b>	<b>28</b>
<b>Interfacing</b>	<b>28</b>
<b>Construction and Reliability</b>	<b>29</b>
<b>Thermal Considerations</b>	<b>30</b>



<b>WARRANTY AND REPAIR</b>	<b>30</b>
<b>Service Policy</b>	<b>30</b>
Out of Warranty Repairs	31
<b>For Service Contact:</b>	<b>31</b>
<b>SPECIFICATIONS</b>	<b>32</b>
<b>ORDER INFORMATION</b>	<b>33</b>



---

---

# List of Figures

---

---

FIGURE 1	IP-BISERIAL-LM1 SERIAL PROTOCOL TIMING	8
FIGURE 2	IP-BISERIAL-LM1 INTERNAL ADDRESS MAP	11
FIGURE 3	IP-BISERIAL-LM1 CONTROL REGISTER 0 BIT MAP	14
FIGURE 4	IP-BISERIAL-LM1 CONTROL REGISTER 1 BIT MAP	15
FIGURE 5	IP-BISERIAL-LM1 CONTROL REGISTER 2 BIT MAP	16
FIGURE 6	IP-BISERIAL-LM1 STATUS REG 0 BIT MAP	17
FIGURE 7	IP-BISERIAL-LM1 STATUS REG 1 BIT MAP	18
FIGURE 8	IP-BISERIAL-LM1 STATUS REG 2 BIT MAP	19
FIGURE 9	IP-BISERIAL-LM1 CONTROL REGISTER 7 BIT MAP	19
FIGURE 10	IP-BISERIAL-LM1 CONTROL REGISTER 9 BIT MAP	20
FIGURE 11	IP-BISERIAL-LM1 CONTROL REGISTER 11 BIT MAP	20
FIGURE 12	IP-BISERIAL-LM1 ID PROM	24
FIGURE 13	IP-BISERIAL-LM1 LOGIC INTERFACE	26
FIGURE 14	IP-BISERIAL-LM1 IO INTERFACE	27

## 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 -LM1 version is a custom modification. The main modifications include switching to msb first format, active high strobe, Tx auto-start capability, and Rx message length verification.

In addition to the LM1 version other custom interfaces are available. Please see our web page for current protocols offered. If you do not find it there we can redesign 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-LM1 supports both 8 and 32 MHz IP Bus operation. The IP Clock, on-board oscillator or an external reference 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 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-LM1 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  $8 \cdot n$  bit transfers. The received bit stream is then padded with zeros to a 16-bit boundary and loaded into the input FIFO. The data length loaded is determined by the strobe signal. The host can then poll or wait for the message complete or FIFO almost-full interrupt. The message can then be read directly from the input FIFO.

The serial transmit channel reads data from the output FIFO and sends it out serially, msb first. The transmitter can be configured to auto-start when the last word of the message is written to the FIFO. The message can be any number of bytes long.

The IP-BISERIAL-LM1 utilizes several clock sources. The IP clock, reference oscillator, or user input clock can be 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-LM1. 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 IntReqO. The status is available for the FIFOs making it possible to operate in a polled mode.



## Theory of Operation

The IP-BISERIAL-LM1 is designed for the purpose of transferring data from one point to another with a serial protocol.

The IP-BISERIAL-LM1 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-LM1 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 Dynamic Engineering 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-LM1 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 LM1 timing for a 16-bit transfer is shown in figure 1. The clock is free running, the data is valid on the falling edge of the clock, and the active high strobe frames the data.

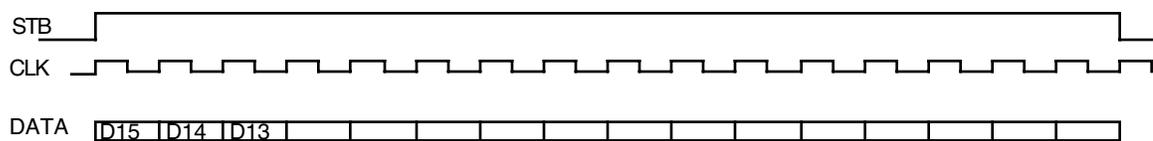


FIGURE 1

IP-BISERIAL-LM1 SERIAL PROTOCOL TIMING



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.

Data transmission can be initiated in manual or automatic start modes. In manual start mode, after the output FIFO is loaded and the start bit is detected high, the transfer begins. The data is read from the FIFO and loaded into the shift register. The MSB is then present at the output of the data buffer and the strobe is activated at the same time. One-half clock period later the Falling edge of the data clock is driven to the output clock buffer. One-half clock period later the data is transitioned to the next value. The MSB-1 is now on the data lines. The process repeats until the first word is transferred. Assuming that there is data to be sent in the FIFO a second word is read and loaded into the shift register and sent out onto the bus. The process is repeated until the FIFO is detected to be empty when it is time to load. The data stream is continuous.

In the automatic start mode the number of bytes to be sent is first entered in the register BIS\_CNTL7. The entire memory space is mapped to the FIFO ports such that a memory write performs a write to the Tx FIFO and a memory read performs a read from the Rx FIFO. When a memory write is performed, the address bits are compared to a value derived from the value in the length register. When the address matches the last address required to complete the required length the transmission is initiated. If an odd number of bytes are requested, only eight bits of the last word are shifted out.

The receive function is similar. When the Strobe is detected high, data is loaded into the receive shift register on the falling edge of the data clock. Once a word has been received the data is loaded into the receive FIFO. When the strobe goes inactive the transfer has been completed and an interrupt to the host is generated (if enabled). The receiver checks for overrun and framing errors and if an error is detected the appropriate bit in STAT1 is set. If only eight bits are received of the last word the RX\_BYTE signal is asserted and this is not seen as a framing error. The number of bytes expected to be received can be entered into register BIS\_CNTL9 and compared to the word count and RX\_BYTE signal to determine if the correct number of bytes were received. If the lengths don't match, the Rx length error bit is set in the STAT1 register.

The word count counter keeps track of the number of words received. At the end of the message, the counter loads the count into a latch and then



clears the counter to allow the next message to be received before the count is read through the STAT2 register. The count self clears when read and always reflects the length of the last message received. The counter is 15 bits wide and is mapped to bits 14-0 of the STAT2 register. Bit 15 of the register is an error bit that is set if the received count is not read before it is overwritten at the end of the next message. This does not interfere with the reception of messages, but is present solely for information.



## 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_STAT2	EQU \$0C	word	read
BIS_CNTL7	EQU \$0E	word	read/write
BIS_CNTL9	EQU \$12	word	read/write
BIS_CNTL11	EQU \$16	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-LM1 INTERNAL ADDRESS MAP

The address map provided is for the local decoding performed within the IP-BISERIAL-LM1. 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-LM1 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 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 and message length verification.

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 strobe 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 strobe is detected to be deasserted the interrupt request is asserted to let the host know that the data is available. The software can read the word count and set-up a loop counter to efficiently read the data from the FIFOs. If the next message starts and completes then the word count will be overwritten. In this case the FIFO empty flag can be used to control the amount of data to read. 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 the message length verification feature is enabled, the expected number of bytes should be written to BIS\_CNTL9. If the message length does not match this value the rx\_len\_err bit will be set in the STAT1 register.

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 LM1 design has both internal and external clock selection capability. 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. 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. If the auto-start feature is enabled, the number of bytes to be sent should be entered into BIS\_CNTL7. The FIFO is then loaded using MEM space accesses starting at zero offset and incrementing as if successive RAM locations were being accessed. When the last word that completes the desired length is written the transmission will automatically start. When the Tx interrupt is received the transmission has been completed and another message can be loaded.

Messages longer than 2K bytes can be accommodated by special ordering HW with larger FIFOs or by using the MT and Full flags on the FIFOs to poll during the transfer...fill the tx FIFO and when not full add more data until full. On the receive side poll and when not empty read the data until empty. The PAE flag is to provide a length programmable almost empty interrupt to allow the Tx side to operate in an interrupt driven mode with longer messages. Similarly the PAF can be used to provide an almost full interrupt to the receive side host 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.



## REGISTER DEFINITIONS

### BIS\_CNTLO

\$00 BISERIAL Control Register Port read/write

CONTROL REGISTER 0	
DATA BIT	DESCRIPTION
7	spare
6	spare
5	spare
4	TX AUTO-START ENABLE
3	INT FORCE
2	INT ENABLE FAE
1	INT ENABLE TX
0	START TX

FIGURE 3

IP-BISERIAL-LM1 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.

4. TX AUTO-START ENABLE when '1' enables the transmitter to start data transmission when the last word of a message is entered into the FIFO. The number of bytes to send is entered into BIS\_CNTL7 and the FIFO must be



written with MEM accesses with incrementing address.

## BIS\_CNTL1

\$02 BISERIAL Control Register Port read/write

CONTROL REGISTER 1	
DATA BIT	DESCRIPTION
7	ENABLE FIFO
6	spare
5	spare
4	RX AUTO-CLEAR ENABLE
3	RX BYTE COUNT COMPARE ENABLE
2	INT ENABLE FAF
1	INT ENABLE RX
0	STRT_RX

FIGURE 4

IP-BISERIAL-LM1 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.

3. RX BYTE COUNT COMPARE ENABLE is used to enable byte count checking of received messages. The expected byte count is written to BIS\_CNTL9 and at the end of a message the number of bytes received is compared to this value. If the lengths don't match the rx\_len\_err bit is set in the STAT1 register.

4. RX AUTO-CLEAR ENABLE when '1' enables the clearing of the start bit at the end of a message.



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
7	EXT/INT RX CLOCK
6	MASTER INT ENABLE
5	FRX_LD
4	FTX_LD
3	RX_RDY [spare output]
2	spare
1	spare
0	RX_TTL_422

FIGURE 5

IP-BISERIAL-LM1 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.

3. RX\_RDY bit is used to set the state of the Receiver Ready Bit. If the system needs Receiver Input Ready control then program the transmitter to output when the TX\_RDY bit in the STATO register is '1' and use the RX\_RDY to grant or block transmission. The default value is '0'. The signal can also be used as a general purpose output bit.

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.

7. EXT/INT RX CLOCK is used to select the reference clock to the Rx state machine and FIFOs. When '0' the IP clock is used to allow the interface to keep up with the IP Bus requirements. Default is '0', normal operation is 1.

## 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\_STATO

\$08 BISERIAL Status Port [read only]

Data Bit	Status	
11	TX_RDY	
10	BYTE_RX	
9	RX_STB	
8	TX_STB	
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-LM1 STATUS REG 0 BIT MAP

11. TX\_RDY when '1' indicates that the remote receiver is ready to receive data. This bit is used to hold off the transmitter.



10. BYTE\_RX when '1' indicates that the last message received contained an odd number of bytes. This bit is used in conjunction with the word count to determine the exact number of bytes that were received.

8,9. RX\_STB & TX\_STB are indicators that a data transfer is in progress.

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	
7	INT REQUEST	1 = interrupt pending
6	OVERRUN ERROR	1 = error detected
5	FRAME ERROR	1 = error detected
4	RX LENGTH ERROR	1 = error detected
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-LM1 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.

4-6. RX LENGTH, FRAME, and OVERRUN errors are tested for when a reception is in progress. If detected then the status bit is set and the reception continues. The word count shows the number of words.

7. 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

\$OC BISERIAL Status Port [read only]

Data Bit	Status
15	MC_ERR 1 = new count written before old count read
14 - 0	CNT14-0 word count

FIGURE 8

IP-BISERIAL-LM1 STATUS REG 2 BIT MAP

15 MC\_ERR is set when a new word count is loaded before status register 2 is read. The old count is over-written by the new count.

14-0. CNT14-0. The word count is incremented for each word loaded into the FIFOs. The latch storing the count is loaded when the reception is completed. The counter is cleared when the new reception starts, but the previous count remains latched until it is read or overwritten at the end of the current message. The software has until this time to read the count. This length of time is set by system constraints.

## BIS\_CNTL7

\$OE BISERIAL Control Register Port read/write

CONTROL REGISTER 7	
DATA BIT	DESCRIPTION
15-0	TX BYTE COUNT

FIGURE 9

IP-BISERIAL-LM1 CONTROL REGISTER 7 BIT MAP

15-0. TX BYTE COUNT: This value is used to determine when to start the transmit process when the auto-start feature is enabled. This occurs when enough data has been written to the FIFO to meet this byte requirement.



## BIS\_CNTL9

\$12 BISERIAL Control Register Port read/write

CONTROL REGISTER 9	
DATA BIT	DESCRIPTION
15-0	RX BYTE COUNT

FIGURE 10

IP-BISERIAL-LM1 CONTROL REGISTER 9 BIT MAP

15-0. RX BYTE COUNT: This value is used to compare to the actual number of bytes received. If the counts do not match the RX LENGTH ERROR bit is set in the STAT1 register.

## 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 11

IP-BISERIAL-LM1 CONTROL REGISTER 11 BIT MAP

### CLOCK PRE-SELECTOR

00	IP Clock
01	Oscillator
10	External
11	IP Clock

The clock pre-selector is used to select which reference clock to use with the divisor hardware (clock source). The external clock is brought in on the TX\_RDY\_422 pins.

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).

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 reference oscillator is 10 MHz. 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.

Please note that the 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.

### **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 'O' 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 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-LM1 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-LM1 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 strobe line is deactivated. The interrupt is mapped to interrupt request 0.



The CPU will respond by asserting INT. The hardware will automatically supply the appropriate interrupt vector and clear the request when accessed by the CPU. The source of the interrupt is obtained by reading BIS\_STAT1. The status remains valid until the status register bits are cleared.

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 before 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 input to the latch, 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. 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 master interrupt should be disabled and the individual interrupts of interest enabled. When BIS\_STAT1 shows an interrupt pending the appropriate FIFO action can take place and the enable toggled to remove the interrupt request 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 the INTACK cycle. 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 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-LM1 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. LM1 is set to 0x01 with a customer number of 0x04.

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	(\$04)
11	Driver ID, low byte	(\$01)
13	Driver ID, high byte	(\$00)
15	No of extra bytes used	(\$0C)
17	CRC	(\$39)

FIGURE 12

IP-BISERIAL-LM1 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
Strobe+	14 - 26
Strobe-	15 - 27
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-LM1. Pins marked n/c below are defined by the specification, but not used on the IP-BISERIAL-LM1. Also see the User Manual for your carrier board for more information.

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

IP-BISERIAL-LM1 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-LM1. Also see the User Manual for your carrier board for more information.

GND	REFCLK_422+	RXSTB+	1	26	
REFCLK_422-	GND	RXSTB-	2	27	
GND	GND	RX_RDY_TTL	3	28	
TXRDY+	GND	RX_DATA_TTL	4	29	
TXRDY-	GND	RX_CLK_TTL	5	30	
GND	GND	RX_STB_TTL	6	31	
TXDATA +	GND	TX_RDY_TTL	7	32	
TXDATA -	GND	GND	8	33	
GND	GND	R_C_TTL_IN	9	34	
TXCLK +	GND	GND	10	35	
TXCLK -	GND	GND	11	36	
GND	GND	GND	12	37	
TXSTB +	GND	GND	13	38	
TXSTB -	R_C_TTL_IN	GND	14	39	
GND	GND	GND	15	40	
RX_RDY_422+	GND	GND	16	41	
RX_RDY_422-	GND	TX_DTA_TTL	17	42	
GND	GND	GND	18	43	
RXDATA +	GND	GND	19	44	
RXDATA -	TX_CLK_TTL	GND	20	45	
GND	GND	GND	21	46	
RXCLK +	GND	GND	22	47	
RXCLK -	GND	TX_STB_TTL	23	48	
GND	GND	GND	24	49	
			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 14

IP-BISERIAL-LM1 IO INTERFACE

LM1 uses the 422 versions of data, clock, and strobe. Tx refers to the transmit and Rx refers to Receive relative to the BiSerial board. User out on RXRDY and user in on TXRDY pins.



# 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-LM1 when it is not powered can damage it, as well as the rest of the host system. This problem may be avoided by turning all power supplies on and off at the same time. Alternatively, the use of OPTO-22 isolation panels is recommended.

**Keep cables short.** Flat cables, even with alternate ground lines, are not suitable for long distances. IP-BISERIAL-LM1 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-LM1 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 dissipation. 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](mailto:support@dyneng.com)



# Specifications

Logic Interface:	IP Module Logic Interface
Serial Interface:	RS-485 RDY, Data, CLK, STB Rx and Tx
Tx CLK rates generated:	IP CLK, Oscillator, User CLK input 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) 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 transmission 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. <b>220</b> mA @ 5V



## Order Information

IP-BISERIAL-LM1	IP Module with 1 Tx and 1 Rx serial channel, Programmable data rates LM1 protocol support, RS-485 drivers and receivers 16 bit IP interface
Tools for IP-BISERIAL-LM1	IP-Debug-Bus - IP Bus interface extender <a href="http://www.dyneng.com/ipdbgbus.html">http://www.dyneng.com/ipdbgbus.html</a> IP-Debug-IO - IO connector breakout <a href="http://www.dyneng.com/ipdbgio.html">http://www.dyneng.com/ipdbgio.html</a>
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

