
# IP-MP-SERIAL

## Dual-Channel
## High-Speed Serial
## IndustryPack®

User Manual


Hardware Revision: D
Part Number: 89004100 Rev.4.1
20050315

**IP-MP-SERIAL**
**Dual-Channel**
**High-Speed Serial**
**IndustryPack®**

**SBS Technologies, Inc.**
**1284 Corporate Center Drive**
**St. Paul, MN 55121**

**Tel: (651) 905-4700**

**FAX: (651) 905-4701**
**Email: support.commercial@sbs.com**

**http://www.sbs.com**



SBS Technologies, Inc. reserves the right to make any changes in the devices or device specifications contained herein at any time and without notice. Customers are advised to verify all information contained in this document.

The electronic equipment described herein generates, uses and may radiate radio frequency energy, which can cause radio interference. SBS Technologies, Inc. assumes no liability for any damages caused by such interference.

SBS Technologies' products are **not** authorized for use as critical components in medical applications such as life support equipment, without the express consent of the General Manager of SBS Technologies, Inc. Commercial Group.

This product has been designed to operate with IndustryPack, PC•MIP or PMC modules or carriers and compatible user-provided equipment. Connection of incompatible hardware is likely to cause serious damage. SBS Technologies, Inc. assumes no liability for any damages caused by such incompatibility.

# Table of Contents

# List of Figures

# Product Description

***Note:*** *This manual includes information about Revision C and Revision D IP-MP-SERIAL cards. Both versions are software compatible. Differences between the revisions are described throughout this manual and are summarized in Appendix A.*

The IP-MP-SERIAL is part of the family of IndustryPack® (IP) I/O modules. This IP is built to run on an 8 MHz or a 32 MHz IndustryPack bus. The IP-MP-SERIAL fully supports synchronous and asynchronous operation with all five modem handshake signals. IP-MP-SERIAL provides internal data buffering, serial data rates up to 10 megabits per second (Mb/s), and support for nearly all serial protocols. The IP-MP-SERIAL incorporates the Zilog Z16C30 Universal Multi-Protocol Serial Controller (USC) which includes four 32-byte FIFO buffers. A FIFO buffer is provided for both transmit and receive data on each of the two independent channels. Protocols supported by the Z16C30 include HDLC, SDLC, Bisync, NRZ, NRZI, and Biphase. Additional capabilities include 16- and 32-bit CRC generation and checking, one to eight bits per character, one address/data bit, MIL STD 1553B protocol support, sync character stripping, preamble generation, and digital phase-locked loop circuits for clock recovery. Refer to the Zilog documentation for details on Z16C30 programming.

The IP-MP-SERIAL includes an onboard 3.6864 MHz oscillator that can be divided by the Z16C30 to create the standard bit rates from 50 b/s to 115.2 Kb/s. A table of time constant values for these bit rates is provided near the end of the Programming section. A separate location is provided on the IP for a user oscillator. The I/O interface provides four external clock lines to permit full duplex synchronous operation on both channels. The interface for each channel includes an external transmit and receive clock. The transmit clock can be programmed to be either an input to the IP or an output from the IP. The receive clock is only an input.

The two channels within the Z16C30 USC device are almost completely independent. Each channel has a baud rate generator, a digital phase-locked loop, a vector register, an interrupt controller, and a protocol controller. Refer to the Zilog documentation for details.

The Zilog Z16C30 does have one limitation that needs to be considered when implementing synchronous serial interfaces. The reference clock required for the baud rate generator or the digital phase lock loop can only be input to the Z16C30 on either the Receive Clock (RxC) pin or the Transmit Clock (TxC) pin of the channel. If the synchronous serial interface requires the IP-MP-SERIAL to supply the transmit clock and receive the receive clock, then only the receive clock can be used to generate the transmit clock. If a half size oscillator is available that generates the exact frequency of the transmit clock, then there is another option. The oscillator can be installed in the User Oscillator position (OSC1 in card Revision C and COSC2 in Revision D) on the IP-MP-SERIAL and its output can be sent to both the Z16C30 TxC pin and the remote receiver. This is achieved by programming the Z16C30 TxC pin as an input and programming IPCR_B to output the Transmit Clock and enable the driver. Refer to the Zilog documentation for details on Z16C30 programming.

Independent interrupt vectors and controls are provided for each IP-MP-SERIAL channel. The IP can be configured so both channels share Interrupt Request Zero or each channel interrupts on a separate Interrupt Request line. When a single Interrupt Request line is used, Channel A is given priority over Channel B in the event of simultaneous interrupt requests. Interrupt priority and sequencing is controlled by a programmable device. Contact SBS Technologies if your application has special requirements.

The EIA-232, EIA-422, and EIA-485 compatible interface circuit (formerly RS-232, RS-422, and RS-485) facilitates interfacing with a wide variety of equipment. In Revision C cards, 5 SIP resistor packages are used to define the electrical characteristics of the serial I/O interface. For Revision D cards, two jumper setting and four SIP resistor packages are used instead.

The IP-MP-SERIAL supports DMA transfers when combined with a carrier board that also supports IndustryPack DMA transfers. IndustryPack Control Register A can be programmed to select one of four DMA configurations for the IP-MP-SERIAL.

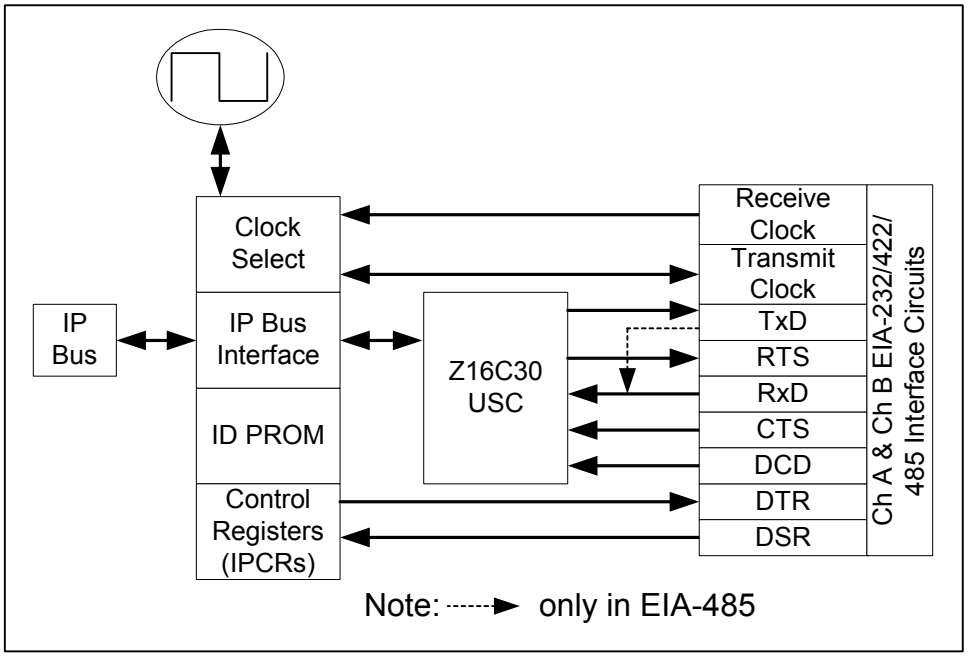


**Figure 1 IP-MP-SERIAL Block Diagram**

# Communication Capabilities

The IP-MP-SERIAL uses a Zilog Z16C30 Universal Serial Controller (USC) that provides internal data buffering, serial data rates up to 10 Mb/s, and support for nearly all serial protocols. The Z16C30 includes a 32-byte FIFO buffer for each transmit and receive data channel. The Channel Mode Register (CMR) of each channel of the Z16C30 is programmed to configure the IP for operate in any of the following modes.

**Asynchronous Mode:** The receiver and transmitter can handle data at a rate of 1/16, 1/32, or 1/64 the clock rate. The receiver rejects start bits less than one-half a bit time and will not erroneously assemble characters following a framing error. The transmitter is capable of sending stop bits anywhere in the range of 1/16 to two stop bits per character in 1/16 bit increments.

**Isochronous Mode:** Both the transmitter and receiver may operate on asynchronous data using a 1x clock. The transmitter can send one or two stop bits.

**Asynchronous with Code Violations Mode:** This is similar to Isochronous mode except that the start bit is replaced by a three bit-time code violation pattern as in MIL-STD-1553B. The transmitter can send zero, one or two stop bits.

**Monosync Mode:** In this mode, a single character is used for synchronization. The sync character can be either eight bits long with an arbitrary data character length, or programmed to match the data character length. The receiver is capable of automatically stripping sync characters from the received data stream. The transmitter may be programmed to automatically send a CRC on either an under-run or at the end of a programmed message length.

**Bisync Mode:** This mode is identical to monosync mode except that character synchronization requires two successive characters for synchronization. The two characters need not be identical.

**HDLC/SDLC Mode:** In this mode, the receiver recognizes flags, performs optional address matching, accommodates extended address fields, 8-bit or 16-bit control fields and logical control fields, performs zero deletion and CRC checking. The receiver is capable of receiving shared-zero flags, recognizing the abort sequence and receiving arbitrary length messages. The transmitter automatically sends opening and closing flags, performs zero insertion and can be programmed to send an abort, an extended abort, a flag or CRC and a flag on transmit under-run. The transmitter can also automatically send the closing flag with optional CRC at the end of a programmed message length. Shared-zero flags are selected in the transmitter and a separate character length may be programmed for the last character in the frame.

**Bisync Transparent Mode:** In this mode, the synchronization pattern is DLE-SYN, programmed to be selected from either ASCII or EBCDIC encoding. The receiver recognizes control character sequences and automatically handles CRC calculations without CPU intervention. The transmitter can be programmed to send SYN, DLE-SYN, CRC-SYN, or CRC-DLE-SYN upon underrun and can automatically send the closing DLE-SYN with optional CRC at the end of a programmed message length.

**NBIP Mode:** This mode is identical to asynchronous mode except that the receiver checks for the status of an additional address/data bit between the parity bit and the stop bit. The value of this bit is FIFOed along with the data. This bit is automatically inserted in the transmitter with the value that is FIFOed with the transmit data.

**802.3 Mode:** This mode implements the data format of IEEE 802.3 with 16-bit address compare. In this mode, the DCD and CTS modem handshake signals are used to implement the carrier sense and collision detect interactions with the receiver and transmitter.

**Slaved Monosync Mode:** This mode is available only in the transmitter and allows the transmitter operating as though it were in monosync mode to send data that is byte-synchronous to the data being received by the receiver.

**HDLC Loop Mode:** This mode is also available only in the transmitter and allows operation in an HDLC loop configuration. In this mode, the receiver is programmed to operate in HDLC mode so that the transmitter echoes received messages. Upon receipt of a particular bit pattern, a sequence of seven consecutive ones, the transmitter breaks the loop and inserts its own frame(s).

# Data Encoding

The Z16C30 used on the IP-MP-SERIAL may be programmed to encode and decode the serial data in any of eight different ways as shown in Figure 2 and described below. The high and low levels shown in Figure 2 are observed at the TxD and receive data (RxD) pins of the Z16C30. These levels are inverted by the EIA-232 drivers and receivers between the Z16C30 and the I/O interface. The EIA-422/485 drivers and receivers do not invert the data signals. The transmitter encoding method is selected independent of the receiver decoding method. The encoding method is configured by the programming of the Encoding Field of the Transmit Mode Register (TMR) of each IP channel. The decoding method is configured by the programming of the Decoding Field of the Receive Mode Register (RMR) of each IP channel.

**NRZ:** This is the default encoding method. In this encoding method a one is represented by a High level for the duration of the bit cell and a zero is represented by a Low level for the duration of the bit cell.

**NRZB:** This encoding method inverts the NRZ data.

**NRZI-Mark:** This encoding method represents a one is by a transition at the beginning of the bit cell. That is, the level present in the preceding bit cell is reversed. A zero is represented by the absence of a transition at the beginning of the bit cell.

**NRZI-Space:** For NRZI-Space encoding, a one is represented by the absence of a transition at the beginning of the bit cell. The level of the previous bit cell is maintained. A zero is represented by a transition at the beginning of the bit cell.

**Biphase-Mark:** This encoding scheme is also known as FM1. A one in Biphase-Mark encoding is represented by a transition at the beginning of the bit cell and another transition at the center of the bit cell. A zero is represented by a transition at the beginning of the bit cell only.

**Biphase-Space:** This encoding scheme is also known as FM0. A Biphase-Space encoded one is represented by a transition at the beginning of the bit cell only. A zero is represented by a transition at the beginning of the bit cell and second transition at the center of the bit cell.

**Biphase-Level:** This encoding scheme is also known as Manchester encoding. This encoding method represents a one by a High level during the first half of the bit cell and a Low level during the second half of the bit cell. A zero is represented by a Low during the first half of the bit cell and a High during the second half of the bit cell.

**Differential Biphase-Level:** This encoding scheme is also known as Differential Manchester encoding. In Differential Biphase-Level encoding, a one is represented by a transition at the center of the bit cell that has the opposite polarity from the transition at the center of the preceding bit cell. A zero is represented by a transition at the center of the bit cell that has the same polarity as the transition at the center of the preceding bit cell. In both cases there may be transitions at the beginning of the bit cell to set up the level required to make the correct center transition.

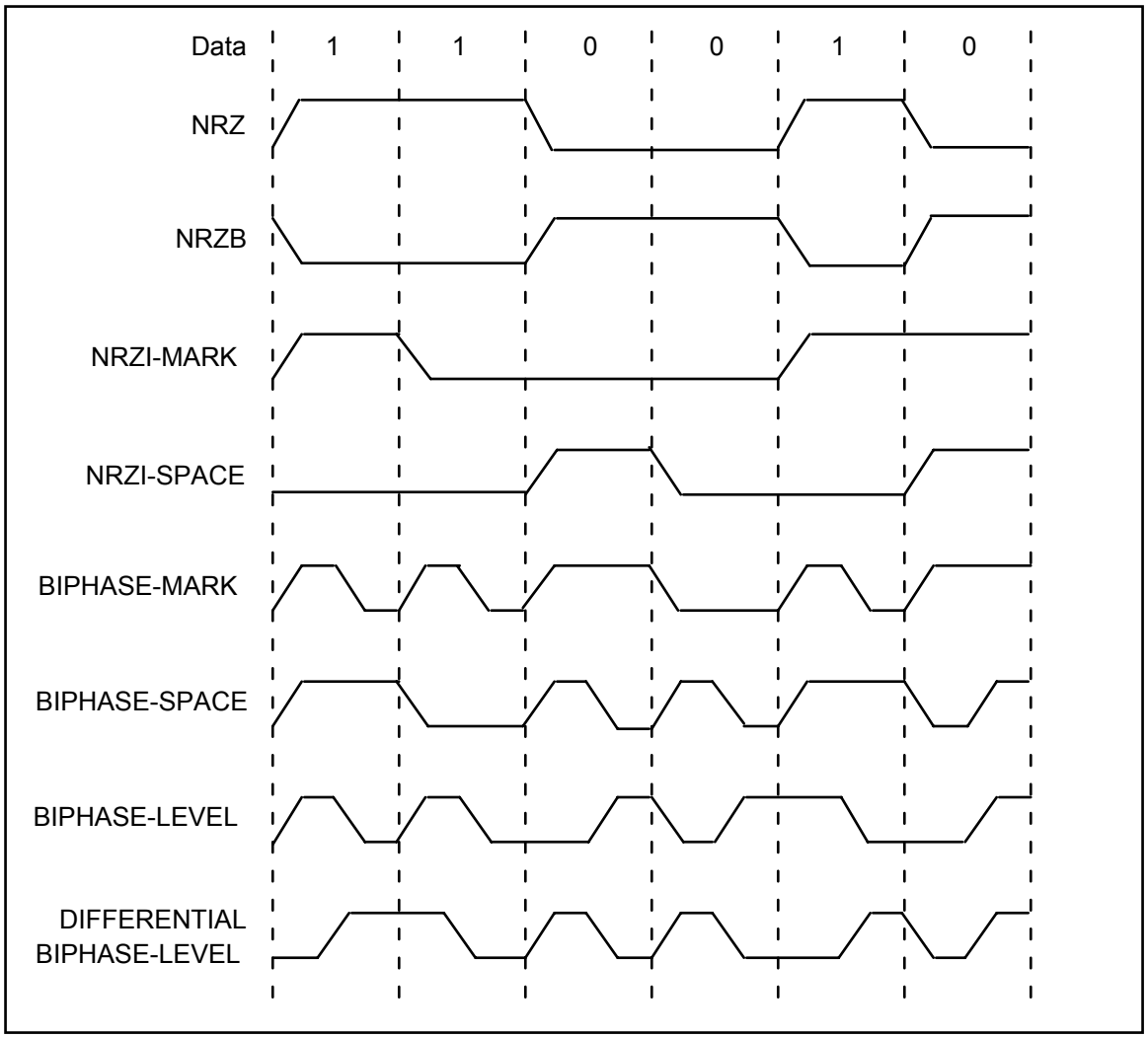


**Figure 2 Data Encoding**

# DMA Operation

The IP-MP-SERIAL is one of the first IndustryPacks designed to support Standard IP Bus DMA transfers. These DMA cycles are modified I/O cycles that use address line A1 to indicate which DMA Request is being acknowledged. The IP-MP-SERIAL maps all DMA transfers to the data register of the selected channel of the Z16C30. There are relative few carrier boards that support DMA transfers over the IP Bus. This section describes how the Motorola MVME162FX can be used to perform DMA accesses to the IP-MP-SERIAL.

The Z16C30 can be programmed to generate up to four DMA request signals, two for each channel. The IP-MP-SERIAL supports four DMA modes that map two of the four DMA request signals to the two IndustryPack Bus DMA Request lines. The IP Carrier board DMA channel that responds to the Transmitter DMA Requests generated by IP-MP-SERIAL must be configured to respond with a Standard DMA Acknowledge (I/O) write cycle. These cycles can only transfer data to the Transmit Data Register of the Z16C30. The Z16C30 can be programmed so that DMA transfers to the Transmit Data Register can configure other Z16C30 Registers. The DMA channel that responds to the Receiver DMA Requests from the IP-MP-SERIAL must be configured to respond with a Standard DMA Acknowledge (I/O) read cycle. These cycles can only transfer data from the Receive Data Register of the Z16C30. The Z16C30 can be configured to transfer receive status register data on the tail of received messages by DMA. Figure 3 lists the four DMA modes and shows the DMA Requests that must be acknowledged with DMA I/O Reads and the ones that must be acknowledged with DMA I/O Writes.

| DMA Mode | DMAReq0 | DMAReq1 |
|---|---|---|
| 00 | Channel B Receiver (DMA I/O Read) | Channel A Receiver (DMA I/O Read) |
| 01 | Channel B Transmitter (DMA I/O Write) | Channel A Transmitter (DMA I/O Write) |
| 10 | Channel B Transmitter (DMA I/O Write) | Channel B Receiver (DMA I/O Read) |
| 11 (Default) | Channel A Transmitter (DMA I/O Write) | Channel A Receiver (DMA I/O Read) |

**Figure 3 DMA Request Modes**

Both word and byte DMA transfers to the Transmit Data Register and from the Receive Data Register are supported. The data path width is defined by the state of the BS0* and BS1* signals on the IP Bus during the DMA transfer. The IP-MP-SERIAL performs a 16-bit data transfer when BS0* and BS1* are zero and an 8-bit data transfer when BS0* is a zero and BS1* is a one. Software should never try to make a byte access to the upper byte of the Z16C30 data registers.

No DMA requests or transfers can occur until the corresponding DMA Enable bit of IPCR_A is set to ‘0’ to enable DMA interface logic. IPCR_A bit 3 enables DMA Request 0 and bit 4 enables DMA Request 1. The IP-MP-SERIAL board will only respond to the DMA acknowledge cycles made to DMA Request channels that have the enable bit set to ‘0’. The IP-MP-SERIAL board will ignore all DMA acknowledge cycles made to DMA Request channels that have the enable bit is set to ‘1’. If both bits 3 and 4 of IPCR_A are set to ‘1’, the IP-MP-SERIAL board will ignore all DMA acknowledge cycles. On most systems this will result in a Bus Time Out Error since the IP does not acknowledge the access.

Bit 3 of IPCR_A is automatically reset to disable DMA transfers on DMA Request 0 if the IP Carrier board asserts DMA End during a DMA Acknowledge cycle to DMA Request 0. Bit 4 of IPCR_A is automatically reset to disable DMA transfers on DMA Request 1 if the IP Carrier board asserts DMA End during a DMA Acknowledge cycle to DMA Request 1. The DMA End signal is never asserted by the IP-MP-SERIAL.

The DMA request thresholds for the transmitter and receiver of each channel of the Z16C30 are set by writing 7 hex to bits 15..12 of the corresponding Command/Status Registers (RCSR_A, RCSR_B, TCSR_A, or TCSR_B) to select the Request Level register. Next the 8 bit threshold level value is written to bits 15..8 of the corresponding Interrupt Control Registers (RICR_A, RICR_B, TICR_A, or TICR_B). After writing the Request Level, it is recommended that the hex value of 5 be written to bits 15..12 of the corresponding Command/Status Register to protect the Request Level threshold from being accidentally modified. The Z16C30 does not request a DMA transfer until there is one more character or empty location in the FIFO buffers than is specified in the Request Level threshold field. Because of this requirement it is possible that the last character of a received message may make the received character count equal to but not greater than the request threshold so no DMA request is generated until the first character of the next message is received. This is most likely to happen when DMA transfers are made by byte accesses rather than word accesses. By using DMA word accesses and setting the receive threshold to one byte or one byte less than the message length will cause all data to be transferred. Near the end of the Programming section of this manual is a table of register values that can be used to initialize the Motorola MVME162FX and IP-MP-SERIAL for DMA operation.

# Resistor and Jumper Settings (Rev. C)

This section applies Revision C cards only. Five SIP resistor packages are used to define the electrical characteristics of the serial I/O interface. RN1 and RN2 are pull-up resistors used to control the current available to the “reference” pin of the differential receivers during single ended operation. For EIA-232 operation the ‘+’ side of the differential receiver is biased at approximately 1.2 volts. The ‘–’ side of the differential receiver is used as the EIA-232 input. RN3, RN4 and RN5 are discrete resistor packages used to terminate the EIA-422 differential inputs. On the standard IP, RN3, RN4 and RN5 are 100Ω resistors. The user may select and install alternate SIP resistors to optimize the termination for a specific application. RN3, RN4 and RN5 should not be installed for single ended configurations to prevent the input voltage from being applied to the reference pin. RN1 and RN2 should not be installed during differential operation. However; in most cases, the small current drain added by RN1 and RN2 will not result in improper operation. Figure 4 below summarizes the recommended Resistor Pack configurations for the various serial interface configurations. The factory default resistor pack configuration has all of the resistor pack locations filled with the values listed in Figure 4 below, and is suitable in most cases for EIA-422 or EIA-485 operation.

| Serial Interface | Resistor Packs Installed | Resistor Packs Removed |
|---|---|---|
| EIA-232 | RN1 and RN2 (2KΩ) | RN3, RN4 and RN5 (100Ω) |
| EIA-422 | RN3, RN4 and RN5 (100Ω) | RN1 and RN2 (2KΩ) |
| EIA-485 | RN3, RN4 and RN5 (100Ω) | RN1 and RN2 (2KΩ) |

**Figure 4 IP-MP-SERIAL Rev C Resistor Pack Configuration**

To configure the IP-MP-SERIAL for different serial interfaces on each channel it is necessary to cut off some of the pins on the resistor packages so that RN3, RN4 and RN5 can be used to terminate the differential inputs on one channel without corrupting the operation of the EIA-232 inputs on the other channel. For example to operate Channel A in EIA-232 mode and Channel B in either EIA-422 or EIA-485 mode; remove pins 4 and 5 from RN1; pins 2, 4 and 5 from RN2; pins 1, 2, 3 and 4 from RN3; pins 1, 2, 5 and 6 from RN4; and pins 1, 2, 3 and 4 from RN5.

To support EIA-485 operation, the EIA-422 transmit data (TxD) line of both channels is equipped with an EIA-422 transceiver. Under software control, this pair of lines can be either transmit data or receive data. The transmit clock (TxC) line of each channel is also equipped with an EIA-422 transceiver. Instead of 100Ω termination resistors, the transmit data lines have 2KΩ bias networks with jumper options. The transmit clock lines have both 100Ω termination resistors and 2KΩ bias networks with jumper options. Removing the jumpers disables the bias networks. The bias resistors on the plus terminal of the EIA-422 transceiver can be connected to either ground or +12 volts. The minus terminal network can be connected to either ground or –12 volts. If either the transmit data or clock is to be used as an EIA-232 receiver, the plus terminal of the transceiver should be biased to ground and the minus terminal of the transceiver used as an input if the signal is to be inverted. If no inversion is desired, then set the minus terminal bias network jumpers to ground and use the plus terminal of the EIA-422 transceiver as the input. The standard configuration of these jumpers sets the plus terminal bias network to +12 volts and the minus terminal bias network to ground. Experimenting with the bias network jumper settings can reduce the noise introduced by a missing or unterminated serial interface cable. **See Appendix A for a summary of differences between IP-MP-SERIAL Rev C and Rev D boards.**

There are four sets of jumpers on the IP-MP-SERIAL. These jumpers determine how the transmit data and transmit clock lines of each channel are biased. Figure 5 lists the

factory default setting of these jumpers. Figure 6 lists the functions of all possible jumper settings. Figure 7 is an assembly diagram of the IP-MP-SERIAL showing the locations and factory default setting of the jumpers.

| Jumper | Setting | Function |
|---|---|---|
| E1 | Pin 1 to Pin 2 | Bias the Channel A plus transmit data (TxD_A+) and plus transmit clock (TxC_A+) lines to +12 volts |
| E2 | Pin 2 to Pin 3 | Bias the Channel A minus transmit data (TxD_A–) and minus transmit clock (TxC_A–) lines to ground |
| E3 | Pin 1 to Pin 2 | Bias the Channel B plus transmit data (TxD_B+) and plus transmit clock (TxC_B+) lines to +12 volts |
| E4 | Pin 2 to Pin 3 | Bias the Channel B minus transmit data (TxD_B–) and minus transmit clock (TxC_B–) lines to ground |

**Figure 5 Factory Default Jumper Settings for Rev C Board**

| Jumper | Setting | Function |
|---|---|---|
| E1 | Removed | Disable the Channel A plus transmit data (TxD_A+) and plus transmit clock (TxC_A+) the bias networks |
| E1 | Pin 1 to Pin 2 | Bias the Channel A plus transmit data (TxD_A+) and plus transmit clock (TxC_A+) lines to +12 volts |
| E1 | Pin 2 to Pin 3 | Bias the Channel A plus transmit data (TxD_A+) and plus transmit clock (TxC_A+) lines to ground |
| E2 | Removed | Disable the Channel A minus transmit data (TxD_A–) and minus transmit clock (TxC_A–) the bias networks |
| E2 | Pin 1 to Pin 2 | Bias the Channel A minus transmit data (TxD_A–) and minus transmit clock (TxC_A–) lines to –12 volts |
| E2 | Pin 2 to Pin 3 | Bias the Channel A minus transmit data (TxD_A–) and minus transmit clock (TxC_A–) lines to ground |
| E3 | Removed | Disable the Channel B plus transmit data (TxD_B+) and plus transmit clock (TxC_B+) the bias networks |
| E3 | Pin 1 to Pin 2 | Bias the Channel B plus transmit data (TxD_B+) and plus transmit clock (TxC_B+) lines to +12 volts |
| E3 | Pin 2 to Pin 3 | Bias the Channel B plus transmit data (TxD_B+) and plus transmit clock (TxC_B+) lines to ground |
| E4 | Removed | Disable the Channel B minus transmit data (TxD_B–) and minus transmit clock (TxC_B–) the bias networks |
| E4 | Pin 1 to Pin 2 | Bias the Channel B minus transmit data (TxD_B–) and minus transmit clock (TxC_B–) lines to –12 volts |
| E4 | Pin 2 to Pin 3 | Bias the Channel B minus transmit data (TxD_B–) and minus transmit clock (TxC_B–) lines to ground |

**Figure 6 Jumper Settings for Rev C Board**

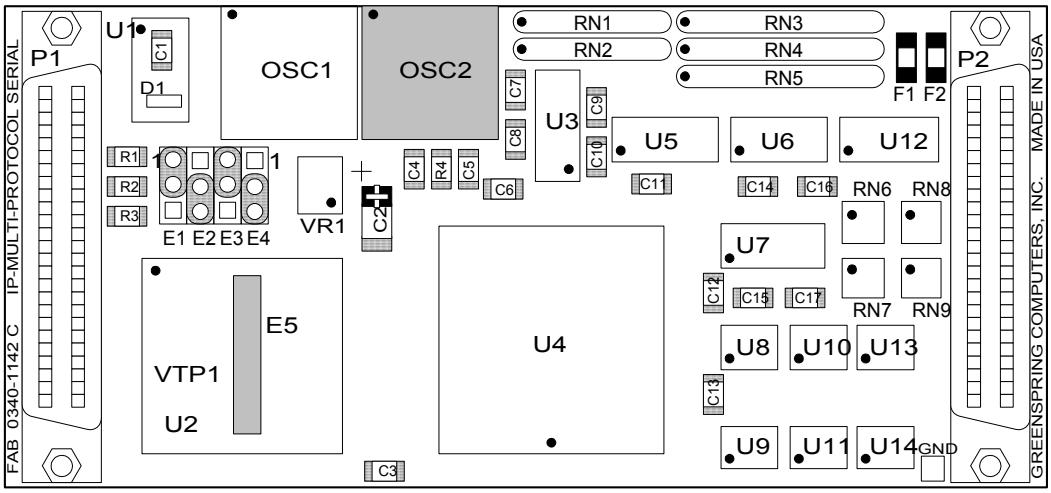


**Figure 7 Factory Jumper and Resistor Network Settings for Rev C Board**

## Option 95121

The IP-MP-SERIAL Revision C can include Option 95121 which uses the IP Clock to sample the received External Receive Clock and the External Transmit Clock. Noise on the I/O cable that is not filtered out by the EIA-422 receivers is filtered out by this sampling. Clock sampling limits the maximum frequency of the received external clocks to 3.5 MHz on the 8 MHz IP Bus.

# Resistor and Jumper Settings (Rev. D)

Jumper E1 and E4 are used to select single ended or differential I/O interface for Channel A and Channel B. RN1, RN2, RN3, and RN4 are discrete resistor packages used to terminate the EIA-422 differential inputs and EIA-485 transceivers. On the standard IP, RN1, RN2, RN3, and RN4 are 100Ω resistors. The user may select and install alternate SIP resistors to optimize the termination for a specific application. RN1, RN2, RN3, and RN4 should be removed for single ended configurations. Figure 8 summarizes the recommended resistor packs configurations and jumper settings for the various serial interface configurations. Notice that RN3 and RN4 are removed by the factory. These resistor packs are required for EIA-422/485 operations. However there are two options of how to install these resistor packs, which is described later in this section.

| Serial Interface | | Jumper Settings | Resistor Packs |
|---|---|---|---|
| Channel A | EIA-232 | E1: Pin 2 to Pin 3 | RN2 and RN3 removed |
| | EIA-422 | E1: Pin 1 to Pin 2 | RN2 and RN3 (100Ω) installed |
| | EIA-485 | E1: Pin 1 to Pin 2 | RN2 and RN3 (100Ω) installed |
| Channel B | EIA-232 | E4: Pin 2 to Pin 3 | RN1 and RN4 removed |
| | EIA-422 | E4: Pin 1 to Pin 2 | RN1 and RN4 (100Ω) installed |
| | EIA-485 | E4: Pin 1 to Pin 2 | RN1 and RN4 (100Ω) installed |

**Figure 8 IP-MP-SERIAL I/O Interface Configuration**

It is possible to configure the IP-MP-SERIAL for different serial interfaces on each channel. For example, to operate Channel A in EIA-232 mode and Channel B in either EIA-422 or EIA-485 mode: connect jumper E1's pin 2 to pin 3, remove RN2 and RN3, connect jumper E4's pin 1 to pin 2 and install RN1 and RN4

To support EIA-485 operation, the EIA-422 transmit data (TxD) line of both channels is equipped with an EIA-422/485 transceiver. Under software control, this pair of lines can be either transmit data or receive data. The transmit clock (TxC) line of each channel is also equipped with an EIA-422/485 transceiver. The transmit data lines have 2KΩ bias networks with jumper options.

The user has the option to terminate TxD line of each channel. By default, SIP resistors are installed with all locations of RN3 and RN4 filled as shown in Figure 11, therefore terminating the TxD lines. If termination of the TxD lines is not desired, it is necessary to cut off pin 1 and pin 2 of the provided SIP resistors and install them as shown in Figure 12 with location 5 and 6 for RN3 and RN4 empty. RN3 corresponds to Channel A and RN4 to Channel B. To terminate Channel A's TxD line only, SIP resistors need to fill all locations of RN3 but only locations 1 to 4 of RN4. IP-MP-SERIAL Rev C boards have their TxD lines terminated. **See Appendix A for a summary of differences between IP-MP-SERIAL Rev C and Rev D boards.**

The transmit clock lines have both 100Ω termination resistors and 2KΩ bias networks with jumper options. Removing the jumpers disables the bias networks. The bias resistors on the plus terminal of the EIA-422/485 transceiver can be connected to either ground or +12 volts. The minus terminal network can be connected to either ground or –12 volts. If the EIA-422/485 transceiver that connects to TxD lines is to be used as an EIA-232 receiver, the plus terminal of the transceiver should be biased to ground and the minus terminal of the transceiver used as an input if the signal is to be inverted. If no inversion is desired, then set the minus terminal bias network jumpers to ground and use the plus terminal of the EIA-422/485 transceiver as the input. The standard configuration of these jumpers sets the plus terminal bias network to +12 volts and the minus terminal bias network to ground. Experimenting with the bias network jumper settings can reduce

the noise introduced by a missing or unterminated serial interface cable. In IP-MP-SERIAL Rev C, EIA-422/485 that carry TxC lines can also be used as EIA-232 receivers. In the same fashion, EIA-422/485 that carry TxD lines can be used as EIA-232 receivers. However, in IP-MP-SERIAL Rev D, a dedicated EIA-232 receiver that connects to TXC lines is available and it is recommended to use these receivers to receive TXC during EIA-232 operation. Contact SBS Technologies if your application requires the use of EIA422/485 transceivers that connects to TXC as EIA-232 receivers**. See Appendix A for a summary of differences between IP-MP-SERIAL Rev C and Rev D boards.**

There are 6 sets of jumpers on the IP-MP-SERIAL. Four of the jumpers (E2, E3, E5, and E6) determine how the transmit data and transmit clock lines of each channel are biased. Figure 9 lists the factory default setting of these jumpers. Figure 10 lists the functions of all possible bias jumper settings. Figures 11 is assembly diagram of the IP-MP-SERIAL showing the locations for setting of the jumpers.

| Jumper | Setting | Function |
|---|---|---|
| E2 | Pin 1 to Pin 2 | Bias the Channel A plus transmit data (TxD_$A_A$) and plus transmit clock (TxC_$A_A$) lines to +12 volts |
| E3 | Pin 2 to Pin 3 | Bias the Channel A minus transmit data (TxD_$A_B$) and minus transmit clock (TxC_$A_B$) lines to ground |
| E5 | Pin 1 to Pin 2 | Bias the Channel B plus transmit data (TxD_$B_A$) and plus transmit clock (TxC_$B_A$) lines to +12 volts |
| E6 | Pin 2 to Pin 3 | Bias the Channel B minus transmit data (TxD_$B_B$) and minus transmit clock (TxC_$B_B$) lines to ground |

**Figure 9 Factory Default Jumper Settings for Bias Circuit: Rev D**

| Jumper | Setting | Function |
|---|---|---|
| E2 | Removed | Disable the Channel A plus transmit data (TxD_$A_A$) and plus transmit clock (TxC_$A_A$) the bias networks |
| E2 | Pin 1 to Pin 2 | Bias the Channel A plus transmit data (TxD_$A_A$) and plus transmit clock (TxC_$A_A$) lines to +12 volts |
| E2 | Pin 2 to Pin 3 | Bias the Channel A plus transmit data (TxD_$A_A$) and plus transmit clock (TxC_$A_A$) lines to ground |
| E3 | Removed | Disable the Channel A minus transmit data (TxD_$A_B$) and minus transmit clock (TxC_$A_B$) the bias networks |
| E3 | Pin 1 to Pin 2 | Bias the Channel A minus transmit data (TxD_$A_B$) and minus transmit clock (TxC_$A_B$) lines to –12 volts |
| E3 | Pin 2 to Pin 3 | Bias the Channel A minus transmit data (TxD_$A_B$) and minus transmit clock (TxC_$A_B$) lines to ground |
| E5 | Removed | Disable the Channel B plus transmit data (TxD_$B_A$) and plus transmit clock (TxC_$B_A$) the bias networks |
| E5 | Pin 1 to Pin 2 | Bias the Channel B plus transmit data (TxD_$B_A$) and plus transmit clock (TxC_$B_A$) lines to +12 volts |
| E5 | Pin 2 to Pin 3 | Bias the Channel B plus transmit data (TxD_$B_A$) and plus transmit clock (TxC_$B_A$) lines to ground |
| E6 | Removed | Disable the Channel B minus transmit data (TxD_$B_B$) and minus transmit clock (TxC_$B_B$) the bias networks |
| E6 | Pin 1 to Pin 2 | Bias the Channel B minus transmit data (TxD_$B_B$) and minus transmit clock (TxC_$B_B$) lines to –12 volts |
| E6 | Pin 2 to Pin 3 | Bias the Channel B minus transmit data (TxD_$B_B$) and minus transmit clock (TxC_$B_B$) lines to ground |

**Figure 10 Jumper Settings for Bias Circuit: Rev D**

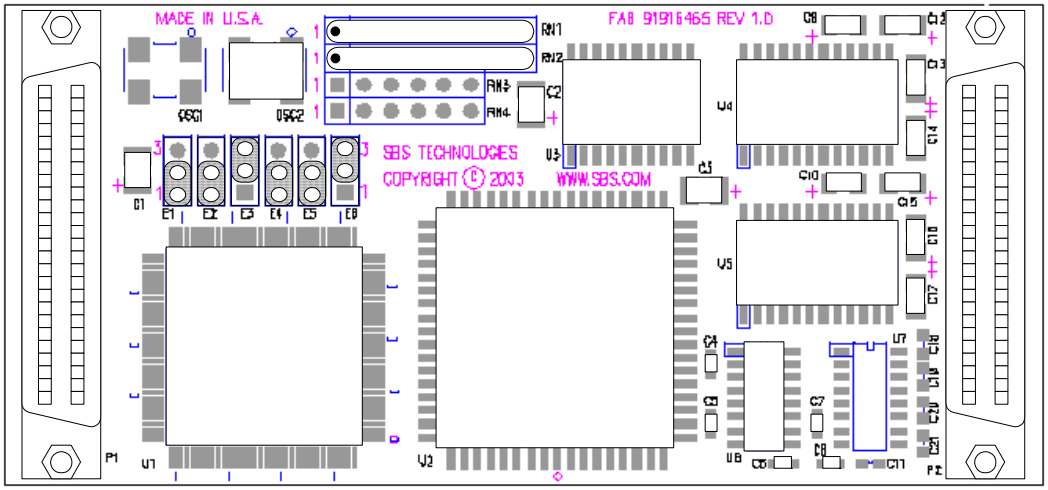


**Figure 11 Factory Jumper and Resistor Network Settings with TxD Lines Terminated: Rev D**

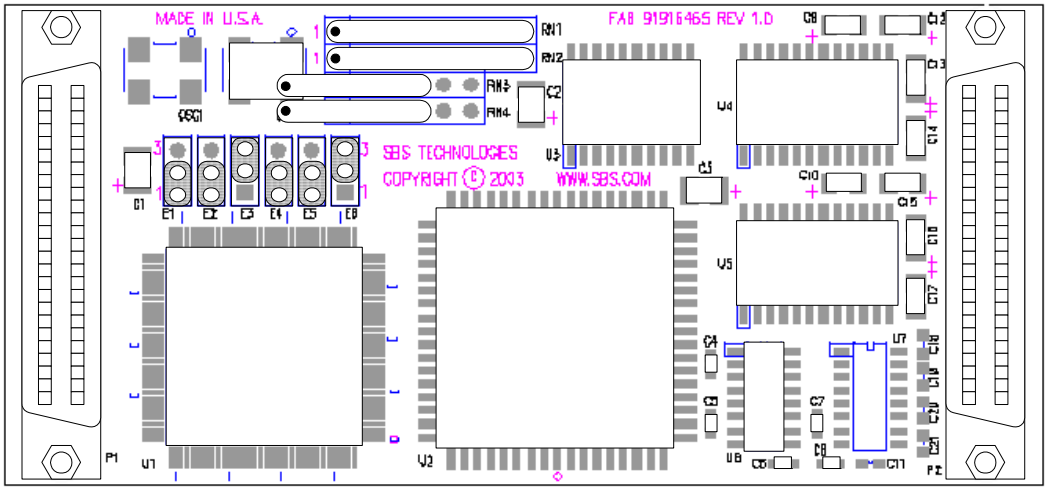


**Figure 12 Resistor Network Location with TxD Lines Unterminated : Rev D (Compatible to Rev C Board Factory Default)**

# VMEbus Addressing

The address map of the IP-MP-SERIAL on the VMEbus is given in Figures 13 - 18. All registers are word registers accessed in IP I/O space. The Receive/Transmit Data Register can be accessed as either a word or a byte. Byte accesses to this register are made on data lines D7..D0. Word accesses are made on data lines D15..D0. Software should never try to make a byte access to the upper byte of the Receive/Transmit Data Register. The IP-MP-SERIAL Control Registers (IPCR_A and IPCR_B) must always be accessed as words. The first access to the Z16C30 must be a word write to the Bus Configuration Register (BCR). All of the other registers in the Z16C30 can be accessed as either words or bytes. Word accesses are made on data lines D15..D0. Low byte addresses are accessed on D7..D0. High byte addresses are accessed on D15..D8.

Interrupt mapping is a function of the selected carrier board. See your IP Carrier board User Manual for more information.

| Register Name | 68K Address | Word Access |
|---|---|---|
| Channel Command/Address Register (CCAR_B) | $00 | Word, R/W |
| Channel Mode Register (CMR_B) | $02 | Word, R/W |
| Channel Command/Status Register (CCSR_B) | $04 | Word, R/W |
| Channel Control Register (CCR_B) | $06 | Word, R/W |
| Bus Configuration Register [first write only] (BCR) | $08 | Word, W |
| IndustryPack Control Register B (IPCR_B) | $0A | Word, R/W |
| Test Mode Data Register (TMDR_B) | $0C | Word, R/W |
| Test Mode Control Register (TMCR_B) | $0E | Word, R/W |
| Clock Mode Control Register (CMCR_B) | $10 | Word, R/W |
| Hardware Configuration Register (HCR_B) | $12 | Word, R/W |
| Interrupt Vector Register (IVR_B) | $14 | Word, R/W |
| I/O Control Register (IOCR_B) | $16 | Word, R/W |
| Interrupt Control Register (ICR_B) | $18 | Word, R/W |
| Daisy-Chain Control Register (DCCR_B) | $1A | Word, R/W |
| Misc. Interrupt Status Register (MISR_B) | $1C | Word, R/W |
| Status Interrupt Control (SICR_B) | $1E | Word, R/W |
| Receive/Transmit Data Register (RDR/TDR_B) | $20 | Word, R/W |
| Receive Mode Register (RMR_B) | $22 | Word, R/W |
| Receive Command/Status Register (RCSR_B) | $24 | Word, R/W |
| Receive Interrupt Control Register (RICR_B) | $26 | Word, R/W |
| Receive Sync Register (RSR_B) | $28 | Word, R/W |
| Receive Count Limit Register (RCLR_B) | $2A | Word, R/W |
| Receive Character Count Register (RCCR_B) | $2C | Word, R/W |
| Time Constant 0 Register (TC0R_B) | $2E | Word, R/W |
| Alias of address $20 | $30 | Word, R/W |
| Transmit Mode Register (TMR_B) | $32 | Word, R/W |
| Transmit Command/Status Register (TCSR_B) | $34 | Word, R/W |
| Transmit Interrupt Control Register (TICR_B) | $36 | Word, R/W |
| Transmit Sync Register (TSR_B) | $38 | Word, R/W |
| Transmit Count Limit Register (TCLR_B) | $3A | Word, R/W |
| Transmit Character Count Register (TCCR_B) | $3C | Word, R/W |
| Time Constant 1 Register (TC1R_B) | $3E | Word, R/W |

**Figure 13 Word VMEbus Address Map (Channel B)**

| Register Name | 68K Address | High Byte Access |
|---|---|---|
| Channel Command/Address Register (CCAR_B) | $00 | H Byte, R/W |
| Channel Mode Register (CMR_B) | $02 | H Byte, R/W |
| Channel Command/Status Register (CCSR_B) | $04 | H Byte, R/W |
| Channel Control Register (CCR_B) | $06 | H Byte, R/W |
| Test Mode Data Register (TMDR_B) | $0C | H Byte, R/W |
| Test Mode Control Register (TMCR_B) | $0E | H Byte, R/W |
| Clock Mode Control Register (CMCR_B) | $10 | H Byte, R/W |
| Hardware Configuration Register (HCR_B) | $12 | H Byte, R/W |
| Interrupt Vector Register (IVR_B) | $14 | H Byte, R/W |
| I/O Control Register (IOCR_B) | $16 | H Byte, R/W |
| Interrupt Control Register (ICR_B) | $18 | H Byte, R/W |
| Daisy-Chain Control Register (DCCR_B) | $1A | H Byte, R/W |
| Misc. Interrupt Status Register (MISR_B) | $1C | H Byte, R/W |
| Status Interrupt Control (SICR_B) | $1E | H Byte, R/W |
| Receive Mode Register (RMR_B) | $22 | H Byte, R/W |
| Receive Command/Status Register (RCSR_B) | $24 | H Byte, R/W |
| Receive Interrupt Control Register (RICR_B) | $26 | H Byte, R/W |
| Receive Sync Register (RSR_B) | $28 | H Byte, R/W |
| Receive Count Limit Register (RCLR_B) | $2A | H Byte, R/W |
| Receive Character Count Register (RCCR_B) | $2C | H Byte, R/W |
| Time Constant 0 Register (TC0R_B) | $2E | H Byte, R/W |
| Transmit Mode Register (TMR_B) | $32 | H Byte, R/W |
| Transmit Command/Status Register (TCSR_B) | $34 | H Byte, R/W |
| Transmit Interrupt Control Register (TICR_B) | $36 | H Byte, R/W |
| Transmit Sync Register (TSR_B) | $38 | H Byte, R/W |
| Transmit Count Limit Register (TCLR_B) | $3A | H Byte, R/W |
| Transmit Character Count Register (TCCR_B) | $3C | H Byte, R/W |
| Time Constant 1 Register (TC1R_B) | $3E | H Byte, R/W |

**Figure 14 High Byte VMEbus Address Map (Channel B)**

| Register Name | 68K Address | Low Byte Access |
|---|---|---|
| Channel Command/Address Register (CCAR_B) | $01 | L Byte, R/W |
| Channel Mode Register (CMR_B) | $03 | L Byte, R/W |
| Channel Command/Status Register (CCSR_B) | $05 | L Byte, R/W |
| Channel Control Register (CCR_B) | $07 | L Byte, R/W |
| Test Mode Data Register (TMDR_B) | $0D | L Byte, R/W |
| Test Mode Control Register (TMCR_B) | $0F | L Byte, R/W |
| Clock Mode Control Register (CMCR_B) | $11 | L Byte, R/W |
| Hardware Configuration Register (HCR_B) | $13 | L Byte, R/W |
| Interrupt Vector Register (IVR_B) | $15 | L Byte, R/W |
| I/O Control Register (IOCR_B) | $17 | L Byte, R/W |
| Interrupt Control Register (ICR_B) | $19 | L Byte, R/W |
| Daisy-Chain Control Register (DCCR_B) | $1B | L Byte, R/W |
| Misc. Interrupt Status Register (MISR_B) | $1D | L Byte, R/W |
| Status Interrupt Control (SICR_B) | $1F | L Byte, R/W |
| Receive/Transmit Data Register (RDR/TDR_B) | $21 | L Byte, R/W |
| Receive Mode Register (RMR_B) | $23 | L Byte, R/W |
| Receive Command/Status Register (RCSR_B) | $25 | L Byte, R/W |
| Receive Interrupt Control Register (RICR_B) | $27 | L Byte, R/W |
| Receive Sync Register (RSR_B) | $29 | L Byte, R/W |
| Receive Count Limit Register (RCLR_B) | $2B | L Byte, R/W |
| Receive Character Count Register (RCCR_B) | $2D | L Byte, R/W |
| Time Constant 0 Register (TC0R_B) | $2F | L Byte, R/W |
| Alias of address $21 | $31 | L Byte, R/W |
| Transmit Mode Register (TMR_B) | $33 | L Byte, R/W |
| Transmit Command/Status Register (TCSR_B) | $35 | L Byte, R/W |
| Transmit Interrupt Control Register (TICR_B) | $37 | L Byte, R/W |
| Transmit Sync Register (TSR_B) | $39 | L Byte, R/W |
| Transmit Count Limit Register (TCLR_B) | $3B | L Byte, R/W |
| Transmit Character Count Register (TCCR_B) | $3D | L Byte, R/W |
| Time Constant 1 Register (TC1R_B) | $3F | L Byte, R/W |

**Figure 15 Low Byte VMEbus Address Map (Channel B)**

See Programming section below and Zilog documentation for register definition details.

All registers are word registers accessed in IP I/O space. The Receive/Transmit Data Register can be accessed as either a word or a byte. Byte accesses to this register are made on data lines D7..D0. Word accesses are made on data lines D15..D0. Software should never try to make a byte access to the upper byte of the Receive/Transmit Data Register. The IP-MP-SERIAL Control Registers (IPCR_A and IPCR_B) must always be accessed as words. The first access to the Z16C30 must be a word write to the Bus Configuration Register (BCR). All of the other registers in the Z16C30 can be accessed as either words or bytes. Word accesses are made on data lines D15..D0. Low byte addresses are accessed on D7..D0. High byte addresses are accessed on D15..D8.

Interrupt mapping is a function of the selected carrier board. See your IP Carrier board User Manual for more information.

| Register Name | 68K Address | Word Access |
|---|---|---|
| Channel Command/Address Register (CCAR_A) | $40 | Word, R/W |
| Channel Mode Register (CMR_A) | $42 | Word, R/W |
| Channel Command/Status Register (CCSR_A) | $44 | Word, R/W |
| Channel Control Register (CCR_A) | $46 | Word, R/W |
| Spare Address, No Register | $48 | Word, R/W |
| IndustryPack Control Register A (IPCR_A) | $4A | Word, R/W |
| Test Mode Data Register (TMDR_A) | $4C | Word, R/W |
| Test Mode Control Register (TMCR_A) | $4E | Word, R/W |
| Clock Mode Control Register (CMCR_A) | $50 | Word, R/W |
| Hardware Configuration Register (HCR_A) | $52 | Word, R/W |
| Interrupt Vector Register (IVR_A) | $54 | Word, R/W |
| I/O Control Register (IOCR_A) | $56 | Word, R/W |
| Interrupt Control Register (ICR_A) | $58 | Word, R/W |
| Daisy-Chain Control Register (DCCR_A) | $5A | Word, R/W |
| Misc. Interrupt Status Register (MISR_A) | $5C | Word, R/W |
| Status Interrupt Control (SICR_A) | $5E | Word, R/W |
| Receive/Transmit Data Register (RDR/TDR_A) | $60 | Word, R/W |
| Receive Mode Register (RMR_A) | $62 | Word, R/W |
| Receive Command/Status Register (RCSR_A) | $64 | Word, R/W |
| Receive Interrupt Control Register (RICR_A) | $66 | Word, R/W |
| Receive Sync Register (RSR_A) | $68 | Word, R/W |
| Receive Count Limit Register (RCLR_A) | $6A | Word, R/W |
| Receive Character Count Register (RCCR_A) | $6C | Word, R/W |
| Time Constant 0 Register (TC0R_A) | $6E | Word, R/W |
| Alias of address $60 | $70 | Word, R/W |
| Transmit Mode Register (TMR_A) | $72 | Word, R/W |
| Transmit Command/Status Register (TCSR_A) | $74 | Word, R/W |
| Transmit Interrupt Control Register (TICR_A) | $76 | Word, R/W |
| Transmit Sync Register (TSR_A) | $78 | Word, R/W |
| Transmit Count Limit Register (TCLR_A) | $7A | Word, R/W |
| Transmit Character Count Register (TCCR_A) | $7C | Word, R/W |
| Time Constant 1 Register (TC1R_A) | $7E | Word, R/W |

**Figure 16 Word VMEbus Address Map (Channel A)**

| Register Name | 68K Address | High Byte Access |
|---|---|---|
| Channel Command/Address Register (CCAR_A) | $40 | H Byte, R/W |
| Channel Mode Register (CMR_A) | $42 | H Byte, R/W |
| Channel Command/Status Register (CCSR_A) | $44 | H Byte, R/W |
| Channel Control Register (CCR_A) | $46 | H Byte, R/W |
| Spare Address, No Register | $48 | H Byte, R/W |
| IndustryPack Control Register A (IPCR_A) | $4A | H Byte, R/W |
| Test Mode Data Register (TMDR_A) | $4C | H Byte, R/W |
| Test Mode Control Register (TMCR_A) | $4E | H Byte, R/W |
| Clock Mode Control Register (CMCR_A) | $50 | H Byte, R/W |
| Hardware Configuration Register (HCR_A) | $52 | H Byte, R/W |
| Interrupt Vector Register (IVR_A) | $54 | H Byte, R/W |
| I/O Control Register (IOCR_A) | $56 | H Byte, R/W |
| Interrupt Control Register (ICR_A) | $58 | H Byte, R/W |
| Daisy-Chain Control Register (DCCR_A) | $5A | H Byte, R/W |
| Misc. Interrupt Status Register (MISR_A) | $5C | H Byte, R/W |
| Status Interrupt Control (SICR_A) | $5E | H Byte, R/W |
| Receive Mode Register (RMR_A) | $62 | H Byte, R/W |
| Receive Command/Status Register (RCSR_A) | $64 | H Byte, R/W |
| Receive Interrupt Control Register (RICR_A) | $66 | H Byte, R/W |
| Receive Sync Register (RSR_A) | $68 | H Byte, R/W |
| Receive Count Limit Register (RCLR_A) | $6A | H Byte, R/W |
| Receive Character Count Register (RCCR_A) | $6C | H Byte, R/W |
| Time Constant 0 Register (TC0R_A) | $6E | H Byte, R/W |
| Transmit Mode Register (TMR_A) | $72 | H Byte, R/W |
| Transmit Command/Status Register (TCSR_A) | $74 | H Byte, R/W |
| Transmit Interrupt Control Register (TICR_A) | $76 | H Byte, R/W |
| Transmit Sync Register (TSR_A) | $78 | H Byte, R/W |
| Transmit Count Limit Register (TCLR_A) | $7A | H Byte, R/W |
| Transmit Character Count Register (TCCR_A) | $7C | H Byte, R/W |
| Time Constant 1 Register (TC1R_A) | $7E | H Byte, R/W |

**Figure 17 High Byte VMEbus Address Map (Channel A)**

| Register Name | 68K Address | Low Byte Access |
|---|---|---|
| Channel Command/Address Register (CCAR_A) | $41 | L Byte, R/W |
| Channel Mode Register (CMR_A) | $43 | L Byte, R/W |
| Channel Command/Status Register (CCSR_A) | $45 | L Byte, R/W |
| Channel Control Register (CCR_A) | $47 | L Byte, R/W |
| Spare Address, No Register | $49 | L Byte, R/W |
| IndustryPack Control Register A (IPCR_A) | $4B | L Byte, R/W |
| Test Mode Data Register (TMDR_A) | $4D | L Byte, R/W |
| Test Mode Control Register (TMCR_A) | $4F | L Byte, R/W |
| Clock Mode Control Register (CMCR_A) | $51 | L Byte, R/W |
| Hardware Configuration Register (HCR_A) | $53 | L Byte, R/W |
| Interrupt Vector Register (IVR_A) | $55 | L Byte, R/W |
| I/O Control Register (IOCR_A) | $57 | L Byte, R/W |
| Interrupt Control Register (ICR_A) | $59 | L Byte, R/W |
| Daisy-Chain Control Register (DCCR_A) | $5B | L Byte, R/W |
| Misc. Interrupt Status Register (MISR_A) | $5D | L Byte, R/W |
| Status Interrupt Control (SICR_A) | $5F | L Byte, R/W |
| Receive/Transmit Data Register (RDR/TDR_A) | $61 | L Byte, R/W |
| Receive Mode Register (RMR_A) | $63 | L Byte, R/W |
| Receive Command/Status Register (RCSR_A) | $65 | L Byte, R/W |
| Receive Interrupt Control Register (RICR_A) | $67 | L Byte, R/W |
| Receive Sync Register (RSR_A) | $69 | L Byte, R/W |
| Receive Count Limit Register (RCLR_A) | $6B | L Byte, R/W |
| Receive Character Count Register (RCCR_A) | $6D | L Byte, R/W |
| Time Constant 0 Register (TC0R_A) | $6F | L Byte, R/W |
| Alias of address $61 | $71 | L Byte, R/W |
| Transmit Mode Register (TMR_A) | $73 | L Byte, R/W |
| Transmit Command/Status Register (TCSR_A) | $75 | L Byte, R/W |
| Transmit Interrupt Control Register (TICR_A) | $77 | L Byte, R/W |
| Transmit Sync Register (TSR_A) | $79 | L Byte, R/W |
| Transmit Count Limit Register (TCLR_A) | $7B | L Byte, R/W |
| Transmit Character Count Register (TCCR_A) | $7D | L Byte, R/W |
| Time Constant 1 Register (TC1R_A) | $7F | L Byte, R/W |

**Figure 18 Low Byte VMEbus Address Map (Channel A)**

See Programming section below and Zilog documentation for register definition details.

# NuBus Addressing

NuBus addressing requires computing the address from the byte addresses given above under VMEbus Addressing. The formula is:

$$NuBus\_byte\_address = (VMEbus\_byte\_address \times 2) - 1$$

All low byte data is still transferred on data lines D7..D0 and all high byte data is still transferred on data lines D15..D8.

Word addresses on the NuBus are the same as for VME. Word data is transferred on data lines D15..D0.

Interrupt mapping is a function of the selected carrier board. See your IP Carrier board User Manual for more information.

# ISA (PC-AT) Bus Addressing

The address map of the IP-MP-SERIAL on the ISA bus is given in Figures 19 through 24 below. All registers are word registers accessed in IP I/O space. The Receive/Transmit Data Register can be accessed as either a word or a byte. Byte accesses to this register are made on data lines D7..D0. Word accesses are made on data lines D15..D0. Software should never try to make a byte access to the upper byte of the Receive/Transmit Data Register. The IP-MP-SERIAL Control Registers (IPCR_A and IPCR_B) must always be accessed as words. The first access to the Z16C30 must be a word write to the Bus Configuration Register (BCR). All of the other registers in the Z16C30 can be accessed as either words or bytes. Word accesses are made on data lines D15..D0. Low byte accesses are made on data lines D7..D0. This byte is the even byte in Intel family host architectures, and the odd byte in Motorola 68K host architectures. High byte accesses are made on data lines D15..D8. This byte is the odd byte in Intel family host architectures, and the even byte in Motorola 68K host architectures.

| Register Name | Intel Address | Word Access |
|---|---|---|
| Channel Command/Address Register (CCAR_B) | $00 | Word, R/W |
| Channel Mode Register (CMR_B) | $02 | Word, R/W |
| Channel Command/Status Register (CCSR_B) | $04 | Word, R/W |
| Channel Control Register (CCR_B) | $06 | Word, R/W |
| Bus Configuration Register [first write only] (BCR) | $08 | Word, W |
| IndustryPack Control Register B (IPCR_B) | $0A | Word, R/W |
| Test Mode Data Register (TMDR_B) | $0C | Word, R/W |
| Test Mode Control Register (TMCR_B) | $0E | Word, R/W |
| Clock Mode Control Register (CMCR_B) | $10 | Word, R/W |
| Hardware Configuration Register (HCR_B) | $12 | Word, R/W |
| Interrupt Vector Register (IVR_B) | $14 | Word, R/W |
| I/O Control Register (IOCR_B) | $16 | Word, R/W |
| Interrupt Control Register (ICR_B) | $18 | Word, R/W |
| Daisy-Chain Control Register (DCCR_B) | $1A | Word, R/W |
| Misc. Interrupt Status Register (MISR_B) | $1C | Word, R/W |
| Status Interrupt Control (SICR_B) | $1E | Word, R/W |
| Receive/Transmit Data Register (RDR/TDR_B) | $20 | Word, R/W |
| Receive Mode Register (RMR_B) | $22 | Word, R/W |
| Receive Command/Status Register (RCSR_B) | $24 | Word, R/W |
| Receive Interrupt Control Register (RICR_B) | $26 | Word, R/W |
| Receive Sync Register (RSR_B) | $28 | Word, R/W |
| Receive Count Limit Register (RCLR_B) | $2A | Word, R/W |
| Receive Character Count Register (RCCR_B) | $2C | Word, R/W |
| Time Constant 0 Register (TC0R_B) | $2E | Word, R/W |
| Alias of address $20 | $30 | Word, R/W |
| Transmit Mode Register (TMR_B) | $32 | Word, R/W |
| Transmit Command/Status Register (TCSR_B) | $34 | Word, R/W |
| Transmit Interrupt Control Register (TICR_B) | $36 | Word, R/W |
| Transmit Sync Register (TSR_B) | $38 | Word, R/W |
| Transmit Count Limit Register (TCLR_B) | $3A | Word, R/W |
| Transmit Character Count Register (TCCR_B) | $3C | Word, R/W |
| Time Constant 1 Register (TC1R_B) | $3E | Word, R/W |

**Figure 19 Word ISA Bus Address Map (Channel B)**

| Register Name | Intel Address | High Byte Access |
|---|---|---|
| Channel Command/Address Register (CCAR_B) | $01 | H Byte, R/W |
| Channel Mode Register (CMR_B) | $03 | H Byte, R/W |
| Channel Command/Status Register (CCSR_B) | $05 | H Byte, R/W |
| Channel Control Register (CCR_B) | $07 | H Byte, R/W |
| IndustryPack Control Register B (IPCR_B) | $0B | H Byte, R/W |
| Test Mode Data Register (TMDR_B) | $0D | H Byte, R/W |
| Test Mode Control Register (TMCR_B) | $0F | H Byte, R/W |
| Clock Mode Control Register (CMCR_B) | $11 | H Byte, R/W |
| Hardware Configuration Register (HCR_B) | $13 | H Byte, R/W |
| Interrupt Vector Register (IVR_B) | $15 | H Byte, R/W |
| I/O Control Register (IOCR_B) | $17 | H Byte, R/W |
| Interrupt Control Register (ICR_B) | $19 | H Byte, R/W |
| Daisy-Chain Control Register (DCCR_B) | $1B | H Byte, R/W |
| Misc. Interrupt Status Register (MISR_B) | $1D | H Byte, R/W |
| Status Interrupt Control (SICR_B) | $1F | H Byte, R/W |
| Receive Mode Register (RMR_B) | $23 | H Byte, R/W |
| Receive Command/Status Register (RCSR_B) | $25 | H Byte, R/W |
| Receive Interrupt Control Register (RICR_B) | $27 | H Byte, R/W |
| Receive Sync Register (RSR_B) | $29 | H Byte, R/W |
| Receive Count Limit Register (RCLR_B) | $2B | H Byte, R/W |
| Receive Character Count Register (RCCR_B) | $2D | H Byte, R/W |
| Time Constant 0 Register (TC0R_B) | $2F | H Byte, R/W |
| Transmit Mode Register (TMR_B) | $33 | H Byte, R/W |
| Transmit Command/Status Register (TCSR_B) | $35 | H Byte, R/W |
| Transmit Interrupt Control Register (TICR_B) | $37 | H Byte, R/W |
| Transmit Sync Register (TSR_B) | $39 | H Byte, R/W |
| Transmit Count Limit Register (TCLR_B) | $3B | H Byte, R/W |
| Transmit Character Count Register (TCCR_B) | $3D | H Byte, R/W |
| Time Constant 1 Register (TC1R_B) | $3F | H Byte, R/W |

**Figure 20 High Byte ISA Bus Address Map (Channel B)**

| Register Name | Intel Address | Low Byte Access |
|---|---|---|
| Channel Command/Address Register (CCAR_B) | $00 | L Byte, R/W |
| Channel Mode Register (CMR_B) | $02 | L Byte, R/W |
| Channel Command/Status Register (CCSR_B) | $04 | L Byte, R/W |
| Channel Control Register (CCR_B) | $06 | L Byte, R/W |
| IndustryPack Control Register B (IPCR_B) | $0A | L Byte, R/W |
| Test Mode Data Register (TMDR_B) | $0C | L Byte, R/W |
| Test Mode Control Register (TMCR_B) | $0E | L Byte, R/W |
| Clock Mode Control Register (CMCR_B) | $10 | L Byte, R/W |
| Hardware Configuration Register (HCR_B) | $12 | L Byte, R/W |
| Interrupt Vector Register (IVR_B) | $14 | L Byte, R/W |
| I/O Control Register (IOCR_B) | $16 | L Byte, R/W |
| Interrupt Control Register (ICR_B) | $18 | L Byte, R/W |
| Daisy-Chain Control Register (DCCR_B) | $1A | L Byte, R/W |
| Misc. Interrupt Status Register (MISR_B) | $1C | L Byte, R/W |
| Status Interrupt Control (SICR_B) | $1E | L Byte, R/W |
| Receive/Transmit Data Register (RDR/TDR_B) | $20 | L Byte, R/W |
| Receive Mode Register (RMR_B) | $22 | L Byte, R/W |
| Receive Command/Status Register (RCSR_B) | $24 | L Byte, R/W |
| Receive Interrupt Control Register (RICR_B) | $26 | L Byte, R/W |
| Receive Sync Register (RSR_B) | $28 | L Byte, R/W |
| Receive Count Limit Register (RCLR_B) | $2A | L Byte, R/W |
| Receive Character Count Register (RCCR_B) | $2C | L Byte, R/W |
| Time Constant 0 Register (TC0R_B) | $2E | L Byte, R/W |
| Alias of address $20 | $30 | L Byte, R/W |
| Transmit Mode Register (TMR_B) | $32 | L Byte, R/W |
| Transmit Command/Status Register (TCSR_B) | $34 | L Byte, R/W |
| Transmit Interrupt Control Register (TICR_B) | $36 | L Byte, R/W |
| Transmit Sync Register (TSR_B) | $38 | L Byte, R/W |
| Transmit Count Limit Register (TCLR_B) | $3A | L Byte, R/W |
| Transmit Character Count Register (TCCR_B) | $3C | L Byte, R/W |
| Time Constant 1 Register (TC1R_B) | $3E | L Byte, R/W |

**Figure 21 Low Byte ISA Bus Address Map (Channel B)**

Interrupt mapping is a function of the selected carrier board. See your IP Carrier board User Manual for more information.

See Programming section below and Zilog documentation for register definition details.

All registers are word registers accessed in IP I/O space. The Receive/Transmit Data Register can be accessed as either a word or a byte. Byte accesses to this register are made on data lines D7..D0. Word accesses are made on data lines D15..D0. Software should never try to make a byte access to the upper byte of the Receive/Transmit Data Register. The IP-MP-SERIAL Control Registers (IPCR_A and IPCR_B) must always be accessed as words. The first access to the Z16C30 must be a word write to the Bus Configuration Register (BCR). All of the other registers in the Z16C30 can be accessed as either words or bytes. Word accesses are made on data lines D15..D0. Low byte accesses are made on data lines D7..D0. This byte is the even byte in Intel family host architectures, and the odd byte in Motorola 68K host architectures. High byte accesses are made on data lines D15..D8. This byte is the odd byte in Intel family host architectures, and the even byte in Motorola 68K host architectures.

Interrupt mapping is a function of the selected carrier board. See your IP Carrier board User Manual for more information.

| Register Name | Intel Address | Word Access |
|---|---|---|
| Channel Command/Address Register (CCAR_A) | $40 | Word, R/W |
| Channel Mode Register (CMR_A) | $42 | Word, R/W |
| Channel Command/Status Register (CCSR_A) | $44 | Word, R/W |
| Channel Control Register (CCR_A) | $46 | Word, R/W |
| Spare Address, No Register | $48 | Word, R/W |
| IndustryPack Control Register A (IPCR_A) | $4A | Word, R/W |
| Test Mode Data Register (TMDR_A) | $4C | Word, R/W |
| Test Mode Control Register (TMCR_A) | $4E | Word, R/W |
| Clock Mode Control Register (CMCR_A) | $50 | Word, R/W |
| Hardware Configuration Register (HCR_A) | $52 | Word, R/W |
| Interrupt Vector Register (IVR_A) | $54 | Word, R/W |
| I/O Control Register (IOCR_A) | $56 | Word, R/W |
| Interrupt Control Register (ICR_A) | $58 | Word, R/W |
| Daisy-Chain Control Register (DCCR_A) | $5A | Word, R/W |
| Misc. Interrupt Status Register (MISR_A) | $5C | Word, R/W |
| Status Interrupt Control (SICR_A) | $5E | Word, R/W |
| Receive/Transmit Data Register (RDR/TDR_A) | $60 | Word, R/W |
| Receive Mode Register (RMR_A) | $62 | Word, R/W |
| Receive Command/Status Register (RCSR_A) | $64 | Word, R/W |
| Receive Interrupt Control Register (RICR_A) | $66 | Word, R/W |
| Receive Sync Register (RSR_A) | $68 | Word, R/W |
| Receive Count Limit Register (RCLR_A) | $6A | Word, R/W |
| Receive Character Count Register (RCCR_A) | $6C | Word, R/W |
| Time Constant 0 Register (TC0R_A) | $6E | Word, R/W |
| Alias of address $60 | $70 | Word, R/W |
| Transmit Mode Register (TMR_A) | $72 | Word, R/W |
| Transmit Command/Status Register (TCSR_A) | $74 | Word, R/W |
| Transmit Interrupt Control Register (TICR_A) | $76 | Word, R/W |
| Transmit Sync Register (TSR_A) | $78 | Word, R/W |
| Transmit Count Limit Register (TCLR_A) | $7A | Word, R/W |
| Transmit Character Count Register (TCCR_A) | $7C | Word, R/W |
| Time Constant 1 Register (TC1R_A) | $7E | Word, R/W |

**Figure 22 Word ISA Bus Address Map (Channel A)**

| Register Name | Intel Address | High Byte Access |
|---|---|---|
| Channel Command/Address Register (CCAR_A) | $41 | H Byte, R/W |
| Channel Mode Register (CMR_A) | $43 | H Byte, R/W |
| Channel Command/Status Register (CCSR_A) | $45 | H Byte, R/W |
| Channel Control Register (CCR_A) | $47 | H Byte, R/W |
| Spare Address, No Register | $49 | H Byte, R/W |
| IndustryPack Control Register A (IPCR_A) | $4B | H Byte, R/W |
| Test Mode Data Register (TMDR_A) | $4D | H Byte, R/W |
| Test Mode Control Register (TMCR_A) | $4F | H Byte, R/W |
| Clock Mode Control Register (CMCR_A) | $51 | H Byte, R/W |
| Hardware Configuration Register (HCR_A) | $53 | H Byte, R/W |
| Interrupt Vector Register (IVR_A) | $55 | H Byte, R/W |
| I/O Control Register (IOCR_A) | $57 | H Byte, R/W |
| Interrupt Control Register (ICR_A) | $59 | H Byte, R/W |
| Daisy-Chain Control Register (DCCR_A) | $5B | H Byte, R/W |
| Misc. Interrupt Status Register (MISR_A) | $5D | H Byte, R/W |
| Status Interrupt Control (SICR_A) | $5F | H Byte, R/W |
| Receive Mode Register (RMR_A) | $63 | H Byte, R/W |
| Receive Command/Status Register (RCSR_A) | $65 | H Byte, R/W |
| Receive Interrupt Control Register (RICR_A) | $67 | H Byte, R/W |
| Receive Sync Register (RSR_A) | $69 | H Byte, R/W |
| Receive Count Limit Register (RCLR_A) | $6B | H Byte, R/W |
| Receive Character Count Register (RCCR_A) | $6D | H Byte, R/W |
| Time Constant 0 Register (TC0R_A) | $6F | H Byte, R/W |
| Transmit Mode Register (TMR_A) | $73 | H Byte, R/W |
| Transmit Command/Status Register (TCSR_A) | $75 | H Byte, R/W |
| Transmit Interrupt Control Register (TICR_A) | $77 | H Byte, R/W |
| Transmit Sync Register (TSR_A) | $79 | H Byte, R/W |
| Transmit Count Limit Register (TCLR_A) | $7B | H Byte, R/W |
| Transmit Character Count Register (TCCR_A) | $7D | H Byte, R/W |
| Time Constant 1 Register (TC1R_A) | $7F | H Byte, R/W |

**Figure 23 High Byte ISA Bus Address Map (Channel A)**

| Register Name | Intel Address | Low Byte Access |
|---|---|---|
| Channel Command/Address Register (CCAR_A) | $40 | L Byte, R/W |
| Channel Mode Register (CMR_A) | $42 | L Byte, R/W |
| Channel Command/Status Register (CCSR_A) | $44 | L Byte, R/W |
| Channel Control Register (CCR_A) | $46 | L Byte, R/W |
| Spare Address, No Register | $48 | L Byte, R/W |
| IndustryPack Control Register A (IPCR_A) | $4A | L Byte, R/W |
| Test Mode Data Register (TMDR_A) | $4C | L Byte, R/W |
| Test Mode Control Register (TMCR_A) | $4E | L Byte, R/W |
| Clock Mode Control Register (CMCR_A) | $50 | L Byte, R/W |
| Hardware Configuration Register (HCR_A) | $52 | L Byte, R/W |
| Interrupt Vector Register (IVR_A) | $54 | L Byte, R/W |
| I/O Control Register (IOCR_A) | $56 | L Byte, R/W |
| Interrupt Control Register (ICR_A) | $58 | L Byte, R/W |
| Daisy-Chain Control Register (DCCR_A) | $5A | L Byte, R/W |
| Misc. Interrupt Status Register (MISR_A) | $5C | L Byte, R/W |
| Status Interrupt Control (SICR_A) | $5E | L Byte, R/W |
| Receive/Transmit Data Register (RDR/TDR_A) | $60 | L Byte, R/W |
| Receive Mode Register (RMR_A) | $62 | L Byte, R/W |
| Receive Command/Status Register (RCSR_A) | $64 | L Byte, R/W |
| Receive Interrupt Control Register (RICR_A) | $66 | L Byte, R/W |
| Receive Sync Register (RSR_A) | $68 | L Byte, R/W |
| Receive Count Limit Register (RCLR_A) | $6A | L Byte, R/W |
| Receive Character Count Register (RCCR_A) | $6C | L Byte, R/W |
| Time Constant 0 Register (TC0R_A) | $6E | L Byte, R/W |
| Alias of address $60 | $70 | L Byte, R/W |
| Transmit Mode Register (TMR_A) | $72 | L Byte, R/W |
| Transmit Command/Status Register (TCSR_A) | $74 | L Byte, R/W |
| Transmit Interrupt Control Register (TICR_A) | $76 | L Byte, R/W |
| Transmit Sync Register (TSR_A) | $78 | L Byte, R/W |
| Transmit Count Limit Register (TCLR_A) | $7A | L Byte, R/W |
| Transmit Character Count Register (TCCR_A) | $7C | L Byte, R/W |
| Time Constant 1 Register (TC1R_A) | $7E | L Byte, R/W |

**Figure 24 Low Byte ISA Bus Address Map (Channel A)**

See Programming section below and Zilog documentation for register definition details.

# Programming

## General

All of the control logic on the IP-MP-SERIAL is contained in an Altera FPGA. After a power on reset or master reset the FPGA undergoes a programming process that requires 330 milliseconds. During this programming time the IP will not respond to Carrier board accesses, the result is a Bus Error. The errors can be eliminated by designing the system software so that the IP is not accessed during the first 330 milliseconds after a master reset.

The phrase “Logic Interface” refers to the Electrical/Logical interface between an IndustryPack board like the IP-MP-SERIAL and the carrier board on which it is installed. The phrase “I/O Interface” refers to the I/O connector on the IndustryPack that carries the serial I/O signals between the IndustryPack and the carrier board.

Independent interrupt vectors and controls are provided for each IP-MP-SERIAL channel. By default both serial channels of the IP-MP-SERIAL share Interrupt Request Zero (IRQ0). For simultaneous interrupt requests in this mode Channel A is given priority over Channel B. If Channel B asserts an interrupt request before Channel A, then the Channel A interrupt request is held off until the Channel B interrupt is acknowledged. Interrupts can also be configured so that the Channel A interrupt request is presented to the carrier board on Interrupt Request Zero (IRQ0) and the Channel B interrupt request is presented on Interrupt Request One (IRQ1). Simultaneous interrupt requests in this mode are presented to the carrier board as simultaneous interrupts on IRQ0 and IRQ1.

The IP-MP-SERIAL provides a clock multiplexer so that software can select either the on board 3.6864 MHz oscillator, the user supplied oscillator, the external transmit clock or the external receive clock as the clock source for either the Receive Clock (RxC) pin or the Transmit Clock (TxC) pin of each Z16C30 channel. The clock multiplexer introduces a delay that results in a skew between the external transmit and receive clocks and the data. **See IP-MP-SERIAL Specifications at the end of this manual for the external clock to data skew value.**

When using the IP-MP-SERIAL to implement synchronous serial interfaces a limitation of the Zilog Z16C30 must be considered. The reference clock required for the digital phase lock loop or baud rate generator of each channel can be input to the Z16C30 on only the Receive Clock (RxC) pin or the Transmit Clock (TxC) pin of that channel. If the synchronous serial interface requires the IP to send a transmit clock and use a receive clock then the receive clock must be used to generate the transmit clock. If a half size oscillator that generates the exact frequency of the transmit clock is available, then it can be installed in the User Oscillator position (OSC2) on the IP and its output can be sent to both the Z16C30 TxC pin and the remote receiver. This is achieved by programming the Z16C30 TxC pin as an input and programming IPCR_B to enable the Transmit Cock driver. Writing zeros to bits 3 through 5 of the I/O Control Register in the Z16C30 configures the TxC pin as an input. The I/O Control Register for Channel B (IOCR_B) is at offset 16 hex from the base address of the IP. The I/O Control Register for Channel A (IOCR_A) is at offset 56 hex. Writing ones to bit six and bit seven of IP Control Register B (IPCR_B) at offset 0A hex configures the Channel A transmit clock transceiver as a driver. Writing a zero to bit one of IP Control Register B (IPCR_B) configures the Channel A transmit clock multiplexer to output the selected clock. Writing ones to bit eight and bit nine of IP Control Register B (IPCR_B) at offset 0A hex configures the Channel B transmit clock transceiver as a driver. Writing a zero to bit fourteen of IP Control Register B (IPCR_B) configures the Channel B transmit clock multiplexer to output the selected clock.

## Register Definitions

For detailed information about all of the registers in the IP-MP-SERIAL register map except for the two IndustryPack Control Registers (IPCR_A & IPCR_B), refer to the Zilog Z16C30 documentation. The IPCRs are special registers that are outside the Z16C30. These registers occupy unused addresses in the Z16C30 register map. The control lines to the USC are not activated for accesses to the IPCRs. All registers are word registers accessed in IP I/O space. All of the registers in the Z16C30 can be accessed as either words or bytes. Word accesses are made on data lines D15..D0. The low byte of a Z16C30 register is accessed on D7..D0. The high byte of a Z16C30 register is accessed on D15..D8. The Receive/Transmit Data Register can be accessed as either a word or a low byte. Software should never try to make a byte access to the upper byte of the Receive/Transmit Data Register. The IP-MP-SERIAL Control Registers (IPCR_A and IPCR_B) must always be accessed as words. The first access to the Z16C30 must be a word write to the Bus Configuration Register (BCR).

## Bus Configuration Register (BCR)

The Bus Configuration Register specifies the basic hardware interface protocol used to access the Zilog Z16C30. The Z16C30 interprets the first write to the device after reset or power on as a write to the BCR. This must be a word write. The control logic on the IP-MP-SERIAL requires that the Z16C30 use a 16 bit multiplexed address/data bus, a right shifted address and the WAIT*/RDY* pin as an access acknowledge handshake signal. This configuration is programmed by writing the hex word ‘0005’ to the BCR register at address offset 8.

## IndustryPack Control Register B (IPCR_B)

| **Data Bit** | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| **Rd / Wrt** | Singl_Int | TxCBO– | RxCAS1 | RxCAS0 | TxCAS1 | TxCAS0 | TxCBIE– | TxCBOE |

| **Data Bit** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **Rd / Wrt** | TxCAIE– | TxCAOE | B485 RE– | B422 RE– | A485 RE– | A422 RE– | TxCAO– | RST– |

**Figure 25 IndustryPack Control Register B**

The IPCR_B performs seven control functions: software reset of the Z16C30 USC, enable the EIA-232/422/485 transmit clock receivers, data drivers, and data receivers, Channel A transmit clock selection multiplexer, Channel A receive clock selection multiplexer, and the interrupt interface configuration. To maximize software compatibility with IP-Universal Serial, the IP-MP-SERIAL defines bits 0 through 7 of IPCR_B the same as those bits used in the IP-Universal Serial IPCR. To maximize software compatibility with other versions of the IP-MP-SERIAL board, all of the used bits in IPCR_B are defined the same for all IP-MP-SERIAL boards. The initial value of IPCR_B after a master reset is $FEBF.

IPCR_B  Bit [0] — S/W Reset [R/W] (active low)
The S/W Reset bit is combined with the hardware master reset and driven to the Z16C30 USC. Power-up and master reset leave this bit set to ‘1’ so that the Z16C30 is not held reset. Bit 0 should be set low and then returned high to create a software controlled reset pulse. The reset signal must be low (Bit 0 set to ‘0’) for 170 nanoseconds and high (Bit 0 set to ‘1’) for 70 nanoseconds before the next access involving the USC is made. With the timing constraints of the 8 MHz IP Interface, back-to-back instructions can be used to create the reset pulse. On a 32 MHz IP Interface back-to-back instructions are likely to produce a 185 nanosecond reset pulse. Additional delay is required to meet the 70 nanosecond delay before the next Z16C30 access. Note that the USC will not respond to CPU accesses while the S/W Reset bit is ‘0’. In most systems this causes a bus time out.

IPCR_B  Bit [1] — Channel A Transmit Clock Output Mode [R/W] (active low)
The IP-MP-SERIAL supports a Transmit Clock for Channel A that can be either an input or output. Power on reset or master reset sets Bit 1 to ‘1’. Setting IPCR_B Bit 1 to ‘0’ configures the Channel A Transmit Clock logic to output the selected clock. The clock selected by the Channel A Transmit Clock Multiplexer is applied to the TxCA pin of the Z16C30 and to the EIA-232 and EIA-422 Transmit Clock drivers when Bit 1 is set to ‘0’ and Bits 11 and 10 of the IPCR_B are not set to ‘0’ and ‘1’ respectively. Setting Bits 11 and 10 of the IPCR_B to ‘0’ and ‘1’ connects the EIA-422 Transmit Clock transceiver, the EIA-232 Transmit Clock driver and the TxCA pin of the Z16C30 so that the transmit clock generated by the Z16C30 can be sent out over the User I/O interface. Programming details for these bits are given in the IPCR_B Bit [10] and Bit [11] section below. The CCR, HCR, IOCR, CMR, and CMCR registers of the Z16C30 are programmed to create the desired bit rate. Refer to the Zilog documentation for bit map definitions of the various registers. The clock transmitted by the Channel A EIA-422 transceiver is also available on the Channel A EIA-232 clock driver.

IPCR_B  Bit [2] — Channel A EIA-422 (EIA-232) Receive Data Enable [R/W] (active low)
When IPCR_B Bit 2 is ‘0’, the Channel A EIA-422 Receive Data receiver is enabled. This receiver is also used to enable the EIA-232 Receive Data. The configuration of the jumpers and the resistor packs to support either EIA-422 or EIA-232 mode is described in the Resistor and Jumper Settings section (Figure 4).  IPCR_B bits 2 and 3 must not both be ‘0’ at the same time to receive valid data. If both of these bits are ‘0’ then a hardware conflict is created because the Channel A Receive Data input of the Z16C30 USC (RxDA) has two sources. Bits 2 and 3 are both reset to ‘1’ by a master reset and must be configured correctly to receive valid data.

IPCR_B  Bit [3] — Channel A EIA-485 Receive Data Enable [R/W] (active low)
When IPCR_B Bit 3 is ‘0’, the Channel A EIA-485 Receive Data receiver is enabled. EIA-485 defines a single differential pair for both transmit and receive data. The IP-MP-SERIAL uses the TxD pair on Channel A for EIA-485 data. Writing a ‘0’ to this bit enables the IP-MP-SERIAL to receive EIA-485 data on the Channel A transmit/receive data pair. It is possible to create hardware conflicts with improper enable selection. For example, IPCR_B bits 2 and 3 must not both be ‘0’ at the same time to receive valid data. If both of these bits are ‘0’ then the Channel A Receive Data input of the Z16C30 USC (RxDA) will have two sources. Bits 2 and 3 are both reset to ‘1’ by a master reset. These bits must be configured correctly before valid data can be received.

| Configuration | Bits 3 and 2 | |
|---|---|---|
| **INVALID** | **0** | **0** |
| EIA-232 or EIA-422 | 1 | 0 |
| EIA-485 | 0 | 1 |
| Receivers Disabled (**Reset**) | 1 | 1 |

**Figure 26  Channel A Data Receiver Controls**

IPCR_B Bit [4] — Channel B EIA-422 (EIA-232) Receive Data Enable [R/W] (active low)
When IPCR_B Bit 4 is '0', the Channel B EIA-422 Receive Data receiver is enabled. This receiver is also used to enable the EIA-232 Receive Data. The configuration of the jumpers and the resistor packs to support either EIA-422 or EIA-232 mode is described in the Resistor and Jumper Settings section (Figure 4). IPCR_B bits 4 and 5 must not both be '0' at the same time to receive valid data. If both of these bits are '0' then a hardware conflict is created because the Channel B Receive Data input of the Z16C30 USC (RxDB) has two sources. Bits 4 and 5 are both reset to '1' by a master reset and must be configured correctly to receive valid data.

IPCR_B Bit [5] — Channel B EIA-485 Receive Data Enable [R/W] (active low)
When IPCR_B Bit 5 is '0', the Channel B EIA-485 Receive Data receiver is enabled. EIA-485 defines a single differential pair for both transmit and receive data. The IP-MP-SERIAL uses the TxD pair on Channel B for EIA-485 data. Writing a '0' to this bit enables the IP-MP-SERIAL to receive EIA-485 data on the Channel B transmit/receive pair. It is possible to create a hardware conflict with improper enable selection. For example, IPCR_B bits 4 and 5 must not both be '0' at the same time to receive valid data. If both of these bits are '0' then the Channel B receive data input of the Z16C30 USC (RxDB) will have two sources. Bits 4 and 5 are both reset to '1' by a master reset. These bits must be configured correctly before valid data can be received.

| Configuration | Bits 5 and 4 | |
|---|---|---|
| **INVALID** | **0** | **0** |
| EIA-232 or EIA-422 | 1 | 0 |
| EIA-485 | 0 | 1 |
| Receivers Disabled (**Reset**) | 1 | 1 |

**Figure 27 Channel B Data Receiver Controls**

IPCR_B Bit [6] — Channel A EIA-422 Transmit Clock Driver Enable [R/W] (active high)
The IP-MP-SERIAL supports a Transmit Clock for Channel A that can be either an input or output. Power on reset or master reset disables both the EIA-422 Transmit Clock driver and receiver. Setting Bit 6 to '1' enables the Channel A EIA-422 Transmit Clock transceiver as a differential driver. IPCR_B Bit 1 must be set to '0' to output the clock selected by the Channel A Transmit Clock Multiplexer to both the Channel A EIA-422 clock transceiver and the Channel A EIA-232 clock driver. Bits 11 and 10 of the IPCR_B must be set to '0' and '1' respectively to connect the Channel A EIA-422 Transmit Clock transceiver and the EIA-232 Transmit Clock driver to the Z16C30 TxCA pin. Programming details for these bits are given in the IPCR_B Bit [10] and Bit [11] section below. The CCR, HCR, IOCR, CMR, and CMCR registers of the Z16C30 are programmed to create the desired bit rate. Refer to the Zilog documentation for bit map definitions of the various registers.

IPCR_B Bit [7] — Channel A Transmit Clock Receiver Enable [R/W] (active low)
The Transmit Clock for Channel A can be either a single ended or differential input or an output. Power on reset disables the Transmit Clock transmitter and receiver. In EIA-422 mode, writing a '0' to Bit 7 enables the Channel A EIA-422 Transmit Clock transceiver as a receiver. In EIA-232 mode, writing a '0' to Bit 7 enables the Channel A EIA-232 Transmit Clock receiver. Refer to the Resistor and Jumper Settings section (Figure 4) for information on configuring the jumpers and the resistor packages for either EIA-422 or EIA-232 mode. Bits 11 and 10 of the IPCR_B must be set to '0' and '1' respectively to connect the Channel A Transmit Clock transceiver and the Z16C30 TxCA pin. Programming details for these bits are given in the IPCR_B Bit [10] and Bit [11] section below. The CCR, HCR, IOCR, CMR, and CMCR registers of the Z16C30 are programmed to create the desired bit rate. Refer to the Zilog documentation for details on these registers. The clock received by the EIA-422 transceiver is applied to the EIA-232 clock driver.

| Configuration | Bits 7 and 6 | |
|---|---|---|
| Enable External Transmit Clock Receiver | 0 | 0 |
| **INVALID** | **0** | **1** |
| External Transmit Clock Disabled (**Reset**) | 1 | 0 |
| Enable External Transmit Clock Driver | 1 | 1 |

**Figure 28 Channel A Transmit Clock Transceiver Controls**

IPCR_B Bit [8] — Channel B EIA-422 Transmit Clock Driver Enable [R/W] (active high)
The IP-MP-SERIAL supports a Transmit Clock for Channel B that can be either an input or output. Power on reset or master reset disables both the EIA-422 Transmit Clock driver and receiver. Setting Bit 8 to '1' enables the Channel B EIA-422 Transmit Clock transceiver as a differential driver. IPCR_B Bit 14 must be set to '0' to output the clock selected by the Channel B Transmit Clock Multiplexer to both the Channel B EIA-422 clock transceiver and the Channel B EIA-232 clock driver. Bits 13 and 12 of the IPCR_A must be set to '0' and '1' respectively to connect the Channel B EIA-422 Transmit Clock transceiver and the EIA-232 Transmit Clock driver to the Z16C30 TxCB pin. Programming details for these bits are given in the IPCR_A Bit [12] and Bit [13] section below. The CCR, HCR, IOCR, CMR, and CMCR registers of the Z16C30 are programmed to create the desired bit rate. Refer to the Zilog documentation for bit map definitions of the various registers.

IPCR_B Bit [9] — Channel B Transmit Clock Receiver Enable [R/W] (active low)
The Transmit Clock for Channel B can be either a single ended or differential input or an output. Power on reset disables the Transmit Clock transmitter and receiver. In EIA-422 mode, writing a '0' to Bit 9 enables the Channel B EIA-422 Transmit Clock transceiver as a receiver. In EIA-232 mode, writing a '0' to Bit 9 enables the Channel B EIA-232 Transmit Clock receiver. Refer to the Resistor and Jumper Settings section (Figure 4) for information on configuring the jumpers and the resistor packages for either EIA-422 or EIA-232 mode. Bits 13 and 12 of the IPCR_A must be set to '0' and '1' respectively to connect the Channel B Transmit Clock transceiver and the Z16C30 TxCB pin. Programming details for these bits are given in the IPCR_A Bit [12] and Bit [13] section below. The CCR, HCR, IOCR, CMR, and CMCR registers of the Z16C30 are programmed to create the desired bit rate. Refer to the Zilog documentation for details on these registers. The clock received by the EIA-422 transceiver is applied to the EIA-232 clock driver.

| Configuration | Bits 9 and 8 | |
|---|---|---|
| Enable External Transmit Clock Receiver | 0 | 0 |
| **INVALID** | **0** | **1** |
| External Transmit Clock Disabled (**Reset**) | 1 | 0 |
| Enable External Transmit Clock Driver | 1 | 1 |

**Figure 29 Channel B Transmit Clock Transceiver Controls**

IPCR_B Bits [10] and [11] — Channel A Transmit Clock Selection [R/W]
The Channel A Transmit Clock source may be selected from one of four sources. Bits 10 and 11 of IPCR_B control the Transmit Clock Multiplexer. Power on reset selects the on board 3.6864 MHz oscillator. The user supplied oscillator, external transmit clock and external receive clock are other options. The selected clock is applied to the TxCA pin of the Z16C30, which should be programmed as an input. Setting Bit 1 of the IPCR_B to '0' configures the Channel A Transmit Clock Multiplexer to output the Transmit Clock. In this mode, the selected clock is also applied to the EIA-422 Transmit Clock transceiver and the EIA-232 Transmit Clock driver. Programming the multiplexer to select the external transmit clock and setting IPCR_B Bit 1 to '0' allows the Z16C30 TxCA pin to send a

clock to the external clock transmitter. In this mode the TxCA pin must be programmed as an output. The EIA-232 Transmit Clock driver is always enabled. Bit 6 of IPCR_B must be programmed to '1' to enable the EIA-422 transceiver as a driver.

| Configuration | Bits 11 and 10 | |
|---|---|---|
| Select the External Receive Clock | 0 | 0 |
| Select the External Transmit Clock | 0 | 1 |
| Select the User Supplied Oscillator | 1 | 0 |
| Select the 3.6864 MHz Oscillator (**Reset**) | 1 | 1 |

**Figure 30 Channel A Transmit Clock Multiplexer Controls**

IPCR_B Bits [12] and [13] —- Channel A Receive Clock Selection [R/W]
The Channel A Receive Clock for the IP-MP-SERIAL may be selected from one of four sources. Bits 12 and 13 of IPCR_B control the Receive Clock Multiplexer. Power on reset selects the on board 3.6864 MHz oscillator. Other options include the user supplied oscillator, the external transmit clock and the external receive clock. The selected clock is applied to the RxCA pin of the Z16C30. The RxCA pin of the Z16C30 should always be configured as an input.

| Configuration | Bits 13 and 12 | |
|---|---|---|
| Select the External Receive Clock | 0 | 0 |
| Select the External Transmit Clock | 0 | 1 |
| Select the User Supplied Oscillator | 1 | 0 |
| Select the 3.6864 MHz Oscillator (**Reset**) | 1 | 1 |

**Figure 31 Channel A Receive Clock Multiplexer Controls**

IPCR_B Bit [14] —- Channel B Transmit Clock Output Mode [R/W] (active low)
The IP-MP-SERIAL supports a Transmit Clock for Channel B that can be either an input or output. Power on reset or master reset sets Bit 14 to '1'. Setting IPCR_B Bit 14 to '0' configures the Channel B Transmit Clock logic to output the selected clock. The clock selected by the Channel B Transmit Clock Multiplexer is applied to the TxCB pin of the Z16C30 and to the EIA-232 and EIA-422 Transmit Clock drivers when Bit 14 is set to '0' and Bits 13 and 12 of the IPCR_B are not set to '0' and '1' respectively. Setting Bits 13 and 12 of the IPCR_A to '0' and '1' connects the EIA-422 Transmit Clock transceiver, the EIA-232 Transmit Clock driver and the TxCB pin of the Z16C30 so that the transmit clock generated by the Z16C30 can be sent out over the User I/O interface. Programming details for these bits are given in the IPCR_B Bit [13] and Bit [12] section above. The CCR, HCR, IOCR, CMR, and CMCR registers of the Z16C30 are programmed to create the desired bit rate. Refer to the Zilog documentation for bit map definitions of the various registers. The clock transmitted by the Channel A EIA-422 transceiver is also available on the Channel A EIA-232 clock driver.

IPCR_B Bit [15] —- Single/Dual Interrupt Requests [R/W]
Bit 15 of IPCR_B controls the interrupt request logic. When bit 15 is '1', both Channel A and Channel B use Interrupt Request Zero (IRQ0) to present interrupts to the carrier board. In the event of simultaneous interrupt requests from the Z16C30, Channel A is granted priority over Channel B. Setting bit 15 to '0' configures the interrupt interface so that Channel A interrupt requests are presented to the carrier board on Interrupt Request Zero (IRQ0) and Channel B interrupt requests are presented on Interrupt Request One (IRQ1). When bit 15 is '0', simultaneous interrupt requests from the Z16C30 are presented to the carrier board as simultaneous interrupts on IRQ0 and IRQ1. Interrupt priority in this mode is determined by the carrier board and system configuration.

The IPCR_B performs seven control functions: software reset of the Z16C30 USC, enable the EIA-232/422/EIA-485 transmit clock receivers, data drivers and receivers, Channel A transmit clock selection multiplexer, Channel A receive clock selection multiplexer, and the interrupt interface configuration. As an aid for programming IPCR_B, Figure 32 lists some IPCR_B configuration values.

| IP-MP-SERIAL IPCR_B Configuration | | IPCR_B Data (hex) |
|---|---|---|
| Remove Software Reset of Z16C30 (initial state after power on reset) | | $FEBF |
| Software Reset of Z16C30 | | $FEBE |
| Channels A and B share IRQ0 for interrupts<br>RxCA = 3.6864 MHz<br>Ext. EIA-422 TxCB & TxCA disabled<br>Channel B–EIA-232/EIA-422 | TxCA = 3.6864 MHz<br>EIA-232 TxCB & TxCA Inactive<br>Channel A–EIA-232/EIA-422 | $FEAB |
| Channels A and B share IRQ0 for interrupts<br>RxCA = 3.6864 MHz<br>Ext. TxCB & TxCA disabled<br>Channel B–EIA-232/EIA-422 | TxCA = 3.6864 MHz<br>EIA-232 TxCB & TxCA Inactive<br>Channel A–EIA-485 | $FEA7 |
| Channels A and B share IRQ0 for interrupts<br>RxCA = 3.6864 MHz<br>Ext. TxCB & TxCA disabled<br>Channel B–EIA-485 | TxCA = 3.6864 MHz<br>EIA-232 TxCB & TxCA Inactive<br>Channel A–EIA-232/EIA-422 | $FE9B |
| Channels A and B share IRQ0 for interrupts<br>RxCA = 3.6864 MHz<br>Ext. TxCB & TxCA disabled<br>Channel B–EIA-485 | TxCA = 3.6864 MHz<br>EIA-232 TxCB & TxCA Inactive<br>Channel A–EIA-485 | $FE97 |
| Channels A and B share IRQ0 for interrupts<br>RxCA = User MHz<br>Ext. TxCB & TxCA disabled<br>Channel B–EIA-232/EIA-422 | TxCA = 3.6864 MHz<br>EIA-232 TxCB & TxCA Inactive<br>Channel A–EIA-232/EIA-422 | $EEAB |
| Channels A and B share IRQ0 for interrupts<br>RxCA = 3.6864 MHz<br>Ext. TxCB & TxCA disabled<br>Channel B–EIA-232/EIA-422 | TxCA = User MHz<br>EIA-232 TxCB & TxCA Inactive<br>Channel A–EIA-485 | $FAA7 |
| Channels A and B share IRQ0 for interrupts<br>RxCA = User MHz<br>Ext. TxCB disabled, EIA-232 Inactive<br>Channel B–EIA-485 | TxCA = User MHz<br>Ext. TxCA Out, EIA-232 Output<br>Channel A–EIA-232/EIA-422 | $EAD9 |
| Channels A and B share IRQ0 for interrupts<br>RxCA = Ext. RxCA<br>Ext. TxCB disabled<br>Channel B–EIA-485 | TxCA = Ext. TxCA Input<br>Ext. TxCA Input<br>Channel A–EIA-485 | $C617 |
| Channels A and B share IRQ0 for interrupts<br>RxCA = Ext. TxCA Input<br>Ext. TxCB disabled<br>Channel B–EIA-232/EIA-422 | TxCA = Ext. RxCA<br>Ext. TxCA Input<br>Channel A–EIA-232/EIA-422 | $D22B |
| Channels A and B share IRQ0 for interrupts<br>RxCA = Ext. RxCA<br>Ext. TxCB Out, EIA-232 Output<br>Channel B–EIA-485 | TxCA = Ext. TxCA Output<br>Ext. TxCA Out, EIA-232 Output<br>Channel A–EIA-485 | $85D5 |
| Channels A and B share IRQ0 for interrupts<br>RxCA = 3.6864 MHz<br>Ext. TxCB & TxCA disabled<br>Channel B–EIA-485 | TxCA = 3.6864 MHz<br>EIA-232 TxCB & TxCA Inactive<br>Channel A–EIA-232/EIA-422 | $FE9B |
| Channel A interrupts on IRQ0<br>RxCA = 3.6864 MHz<br>Ext. TxCB & TxCA disabled<br>Channel B–EIA-485 | Channel B interrupts on IRQ1<br>TxCA = 3.6864 MHz<br>EIA-232 TxCB & TxCA Inactive<br>Channel A–EIA-232/EIA-422 | $7E9B |

**Figure 32 Some Valid IPCR_B Configurations**

Refer to IPCR_A for DMA Enable.

IndustryPack Control Register A (IPCR_A)

| Data Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8<br>**RO** |
|---|---|---|---|---|---|---|---|---|
| **Rd / Wrt** | RxCBS 1 | RxCBS 0 | TxCBS1 | TxCBS0 | DMA– | B422D | DTRB | DSRB |

| Data Bit | 7 | 6 | 5<br>**RO** | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **Rd / Wrt** | A422D | DTRA | DSRA | DMAEN1– | DMAEN0– | 8/32 MHz | DMAMS1 | DMAMS0 |

**Figure 33 IndustryPack Control Register A**

The IPCR_A performs six control functions: DMA Request signal mapping, DMA Channel enabling, EIA-422 driver enabling, DSR activation during DMA Mode, Channel B transmit and receive clock selection. The functions of some of the bits in IPCR_A are enabled only when DMA Mode is enabled. To maximize software compatibility with IP-Universal Serial, the IP-MP-SERIAL supports the use of the DMA Request lines from the Z16C30 as the EIA-422 driver enables and Data Terminal Ready (DTR) controls. When DMA Mode is enabled by writing a '0' to IPCR_A Bit 11, these lines are controlled by bits in IPCR_A. The IPCR_A bits that monitor the state of the Data Set Ready (DSR) receivers are read only. All other bits in IPCR_A are read/write. Do not change any of the DMA control bits during DMA transfers. The initial value of the read/write bits of IPCR_A after a master reset is $FFFF. The read only bits (bits 5 and 8) may be either '0' or '1' depending on the state of the DSR lines being monitored.

IPCR_A Bit [0] and [1] — DMA Mode Selection [R/W]
The IP-MP-SERIAL supports four DMA Modes. IPCR_A Bits 0 and 1 control the selection of the DMA Mode. The selected mode determines the mapping of the four DMA Request lines from the Z16C30 to the two IP DMA Request lines. Power on reset selects full duplex DMA on Channel A. Other options include full duplex DMA on Channel B, transmit only DMA on both Channels A and B, and receive only DMA on both Channels A and B. No DMA requests are presented to the carrier board until IPCR_A bits 4 and/or 3 are set to '0' to enable the DMA channel.

| Configuration | DMA Request Mapping | Bits 1 and 0 | |
|---|---|---|---|
| Receive DMA | RxREQB to DMAReq0 (DMA I/O Read)<br>RxREQA to DMAReq1 (DMA I/O Read) | 0 | 0 |
| Transmit DMA | TxREQB to DMAReq0 (DMA I/O Write)<br>TxREQA to DMAReq1 (DMA I/O Write) | 0 | 1 |
| Channel B DMA | TxREQB to DMAReq0 (DMA I/O Write)<br>RxREQB to DMAReq1 (DMA I/O Read) | 1 | 0 |
| Channel A DMA (**Reset**) | TxREQA to DMAReq0 (DMA I/O Write)<br>RxREQA to DMAReq1 (DMA I/O Read) | 1 | 1 |

**Figure 34 DMA Mode Controls**

IPCR_A  Bit [2] — IP Bus Clock Rate [R/W]
Bit 2 of IPCR_A is used to inform the IP-MP-SERIAL of the IP Bus Clock rate. This is necessary to provide optimum performance when accessing the Z16C30. Power on reset selects 8 MHz. Writing a '0' to bit 2 selects 32 MHz. If the IP Bus Clock is 8 MHz and bit 2 is set to '0' then additional wait states will be inserted in each Z16C30 access and performance will be slowed. If the IP Bus Clock is 32 MHz and bit 2 is set to '1' then timing violations will occur on accesses to the Z16C30 and performance will be unreliable.

IPCR_A  Bit [3] — DMA Enable for DMAReq0 [R/W] (active low)
When a '0' is written to IPCR_A Bit 3, the use of the DMAReq0 interface logic is enabled. While Bit 3 is a '0', the DMA request from the Z16C30 that is mapped to the IP DMAReq0 signal is presented to the IP Carrier board. No DMA requests will be generated on DMAReq0 until IPCR_A Bit 3 is set to '0'. The IP-MP-SERIAL board ignores all DMAAck* Zero accesses made by the IP Carrier board if bit 3 of IPCR_A is set to '1'. On most systems this will result in a Bus Time Out Error since the IP does not acknowledge the access. If the IP Carrier board asserts DMAEnd during a DMAAck Zero I/O cycle to the IP-MP-SERIAL, Bit 3 of IPCR_A is set to '1'. The IP-MP-SERIAL never asserts DMAEnd. Also see note in Bit [4].

IPCR_A  Bit [4] — DMA Enable for DMAReq1 [R/W] (active low)
When a '0' is written to IPCR_A Bit 4, the use of the DMAReq1 interface logic is enabled. While Bit 4 is a '0', the DMA request from the Z16C30 that is mapped to the IP DMAReq1 signal is presented to the IP Carrier board. No DMA requests will be generated on DMAReq1 until IPCR_A Bit 4 is set to '0'. The IP-MP-SERIAL board ignores all DMAAck* One accesses made by the IP Carrier board if bit 4 of IPCR_A is set to '1'. On most systems this will result in a Bus Time Out Error since the IP does not acknowledge the access. If the IP Carrier board asserts DMAEnd during a DMAAck zero I/O cycle to the IP-MP-SERIAL, Bit 3 of IPCR_A is set to '1'. The IP-MP-SERIAL never asserts DMAEnd. See also note in Bit [4].

IPCR_A Bit [4] – DMA Enable for DMAReq1 [R/W] (active low)
When a '0' is written to IPCR_A Bit 4, use of the DMAReq1 interface logic is enabled. While Bit 4 is a '0', the DMA request from the Z16C30 that is mapped to the IP DMAReq1 signal is presented to the IP carrier board. No DMA requests will be generated on DMAReq1 until IPCR_A Bit 4 is set to '0'. The IP-MP-SERIAL board ignores all DMAAck* One accesses made by the IP carrier board if Bit 4 of IPCR_A is set to '1'. On most systems this will result in a Bus Time Out Error since the IP does not acknowledge the access. If the IP carrier board asserts DMAEnd during a DMAAck One I/O cycle to the IP-MP-SERIAL, Bit 4 of IPCR_A is set to '1'. The IP-MP-SERIAL never asserts DMAEnd.

**Note:** This note documents a recent change on how bits 3 and 4 can be modified. This is necessary to avoid inadvertent change to either bit while one is changed. A write to bits 3 and 4 achieves the following:

Bit [4,3] = 00  DMA enable both Channel 0 and 1
Bit [4,3] = 01  DMA enable Channel 1, no change to Channel 0
Bit [4,3] = 10  DMA enable Channel 0, no change to Channel 1
Bit [4,3] = 11  DMA disable both Channel 0 and 1

This change should be compatible with your existing installed software, otherwise contact the factory.

IPCR_A  Bit [5] — Channel A Data Set Ready Input [R] (active high)
Bit 5 of IPCR_A monitors the state of the Channel A Data Set Ready (DSR_A) receiver. An active EIA-232 DSR input is read as a '1' in Bit 5 of IPCR_A.

IPCR_A  Bit [6] — Channel A DMA Mode Data Terminal Ready [R/W] (active high)
The output of IPCR_A Bit 6 is used only when DMA Mode is established by setting IPCR_A Bit 11 to ‘0’. In DMA Mode, IPCR_A Bit 6 is used to control the Channel A Data Terminal Ready (DTR_A) driver instead of the TxREQA pin of the Z16C30. Writing a ‘1’ to IPCR_A Bit 6 turns on the EIA-232 DTR_A driver. If IPCR_A Bit 7 is set to ‘1’ to enable the EIA-422 drivers, then the EIA-422 DTR_A pair is also turned on by writing a ‘1’ to IPCR_A Bit 6. The IP-MP-SERIAL supports the use of TxREQA as the control for DTR_A to maintain software compatibility with IP-Universal Serial. If it is preferrable to use IPCR_A Bit 6 to control the DTR_A driver, the IP-MP-SERIAL may be programmed to operate in DMA Mode without ever enabling the DMA channels.

IPCR_A  Bit [7] — Channel A DMA Mode EIA-422 Driver Enable [R/W] (active high)
The output of IPCR_A Bit 7 is used only when DMA Mode is established by setting IPCR_A Bit 11 to ‘0’. In DMA Mode, IPCR_A Bit 7 is used to enable the EIA-422 drivers on the Channel A Request To Send (RTS_A) and Data Terminal Ready (DTR_A) lines instead of the RxREQA pin of the Z16C30. Writing a ‘1’ to IPCR_A Bit 7 enables the EIA-422 drivers. Power on reset sets bits 7 and 11 to ‘1’ so the EIA-422 drivers are not enabled. Writing a ‘0’ to bit 7 disables the EIA-422 drivers in DMA Mode and can reduce power consumption. The IP-MP-SERIAL supports the use of RxREQA as the control for the EIA-422 drivers to maintain software compatibility with IP-Universal Serial. If it is preferred to use IPCR_A Bit 7 to control the EIA-422 driver, the IP-MP-SERIAL may be programmed to operate in DMA Mode without ever enabling the DMA channels.

IPCR_A  Bit [8] — Channel B Data Set Ready Input [R] (active high)
Bit 8 of IPCR_A monitors the state of the Channel B Data Set Ready (DSR_B) receiver. An active EIA-232 DSR input is read as a ‘1’ in Bit 8 of IPCR_A.

IPCR_A  Bit [9] — Channel B DMA Mode Data Terminal Ready [R/W] (active high)
The output of IPCR_A Bit 9 is used only when DMA Mode is established by setting IPCR_A Bit 11 to a ‘0’. In DMA Mode, IPCR_A Bit 9 is used to control the Channel B Data Terminal Ready (DTR_B) driver instead of the TxREQB pin of the Z16C30. Writing a ‘1’ to IPCR_A Bit 9 turns on the EIA-232 DTR_B driver. If IPCR_A Bit 10 is set to ‘1’ to enabled the EIA-422 drivers, then the EIA-422 DTR_B pair is also turned on by writing a ‘1’ to IPCR_A Bit 9. The IP-MP-SERIAL supports the use of TxREQB as the control for DTR_B to maintain software compatibility with IP-Universal Serial. If it is preferred to use IPCR_A Bit 9 to control the DTR_B driver, the IP-MP-SERIAL may be programmed to operate in DMA Mode without ever enabling the DMA channels.

IPCR_A  Bit [10] — Channel B DMA Mode EIA-422 Driver Enable [R/W] (active high)
The output of IPCR_A Bit 10 is used only when DMA Mode is established by setting IPCR_A Bit 11 to ‘0’. In DMA Mode, IPCR_A Bit 10 is used to enable the EIA-422 drivers on the Channel B Request To Send (RTS_A) and Data Terminal Ready (DTR_A) lines instead of the RxREQB pin of the Z16C30. Writing a ‘1’ to IPCR_A Bit 10 enables the EIA-422 drivers. Power on reset sets bits 10 and 11 to ‘1’ so the EIA-422 drivers are not enabled. Writing a ‘0’ to bit 10 disables the EIA-422 drivers in DMA Mode and can reduce power consumption. The IP-MP-SERIAL supports the use of RxREQB as the control for the EIA-422 drivers to maintain software compatibility with IP-Universal Serial. If it is preferred to use IPCR_A Bit 10 to control the EIA-422 driver, the IP-MP-SERIAL may be programmed to operate in DMA Mode without ever enabling the DMA channels.

IPCR_A  Bit [11] — DMA Transfer Mode Select [R/W] (active low)
When a ‘0’ is written to IPCR_A Bit 11, DMA Mode is selected. In DMA Mode, IPCR_A Bits 6, 7, 9 and 10 are used instead of the TxREQA, RxREQA, TxREQB, and RxREQB pins of the Z16C30 to control DTR_A, the Channel A EIA-422 DSR and DTR drivers enable, DTR_B and the Channel B EIA-422 DSR and DTR drivers enable. IPCR_A Bit 11 must be set to ‘0’ for DMA transfers. IPCR_A Bit 11 may be set to ‘0’ if it is preferred to use IPCR_A bits instead of the Z16C30 pins to control the serial interface drivers.

| DMA Mode IPCR_A [11] = 0 | Non-DMA Mode IPCR_A [11] = 1 | Function |
|---|---|---|
| IPCR_A [6] | TxREQA Pin | DTR_A |
| IPCR_A [7] | RxREQA Pin | EIA-422_A  Dr. En. |
| IPCR_A [9] | TxREQB Pin | DTR_B |
| IPCR_A [10] | RxREQB Pin | EIA-422_B  Dr. En. |

**Figure 35  Control Sources for DTR and the EIA-422 Driver for DSR and DTR**

IPCR_A  Bits [12] and [13] — Channel B Transmit Clock Selection [R/W]
The Channel B Transmit Clock for the IP-MP-SERIAL may be selected from one of four sources. Bits 12 and 13 of IPCR_A control the Transmit Clock Multiplexer. Power on reset selects the on board 3.6864 MHz oscillator. Other options include the user supplied oscillator, the external transmit clock and the external receive clock. The selected clock is applied to the TxCB pin of the Z16C30, which should be programmed as an input. Setting Bit 14 of the IPCR_B to ‘0’ configures the Channel B Transmit Clock Multiplexer to output the Transmit Clock. In this mode, the clock selected by the Transmit Clock Multiplexer is also applied to the Channel B Transmit Clock EIA-422 clock transceiver and EIA-232 clock driver. When IPCR_B Bit 14 is set to ‘0’ and the Transmit Clock multiplexer is set to select the external transmit clock, the clock output on the TxCB pin of the Z16C30 is applied to the external clock EIA-422 transceiver and EIA-232 clock driver. In this configuration the TxCB must be programmed as an output. The EIA-232 Transmit Clock driver is always enabled. Bit 8 of IPCR_B must be programmed to ‘1’ to enable the EIA-422 transceiver as a driver.

| Configuration | Bits 13 and 12 | |
|---|---|---|
| Select the External Receive Clock | 0 | 0 |
| Select the External Transmit Clock | 0 | 1 |
| Select the User Supplied Oscillator | 1 | 0 |
| Select the 3.6864 MHz Oscillator (**Reset**) | 1 | 1 |

**Figure 36  Channel B Transmit Clock Multiplexer Controls**

IPCR_A  Bits [14] and [15] — Channel B Receive Clock Selection [R/W]
The Channel B Receive Clock for the IP-MP-SERIAL may be selected from one of four sources. Bits 14 and 15 of IPCR_A control the Receive Clock Multiplexer. Power on reset selects the on board 3.6864 MHz oscillator. Other options include the user supplied oscillator, the external transmit clock and the external receive clock. The selected clock is applied to the RxCB pin of the Z16C30. The RxCB pin of the Z16C30 should always be configured as an input.

| Configuration | Bits 15 and 14 | |
|---|---|---|
| Select the External Receive Clock | 0 | 0 |
| Select the External Transmit Clock | 0 | 1 |
| Select the User Supplied Oscillator | 1 | 0 |
| Select the 3.6864 MHz Oscillator (**Reset**) | 1 | 1 |

**Figure 37  Channel B Receive Clock Multiplexer Controls**

The IPCR_A performs six control functions. As an aid to programming IPCR_A, Figure 38 lists common configuration values for IPCR_A.

| IP-MP-SERIAL IPCR_A Configuration | | IPCR_A Data (hex) |
|---|---|---|
| RxCB = 3.6864 MHz<br>Non DMA Mode<br>DSR_B (RO) Inactive<br>DSR_A (RO) Inactive<br>IP Bus Clock = 8 MHz | TxCB = 3.6864 MHz<br>Bits 10 & 9 ignored<br>Bits 7 & 6 ignored<br>Both DMA Channels Disabled<br>Channel A DMA Selected | $F81F |
| RxCB = User MHz<br>Non DMA Mode<br>DSR_B (RO) Inactive<br>DSR_A (RO) Active<br>IP Bus Clock = 8 MHz | TxCB = 3.6864 MHz<br>Bits 10 & 9, ignored<br>Bits 7 & 6 ignored<br>Both DMA Channels Disabled<br>Channel A DMA Selected | $B83F |
| RxCB = 3.6864 MHz<br>DMA Mode<br>DTR_B, DSR_B Active<br>DTR_A, DSR_A Inactive<br>IP Bus Clock = 8 MHz | TxCB = User MHz<br>EIA-422_B Drivers Enabled<br>EIA-422_A Drivers Enabled<br>Both DMA Channels Disabled<br>Channel A DMA Selected | $E79F |
| RxCB = User MHz<br>DMA Mode<br>DTR_B, DSR_B Inactive<br>DTR_A, DSR_A Active<br>IP Bus Clock = 8 MHz | TxCB = User MHz<br>EIA-422_B Drivers Enabled<br>EIA-422_A Drivers Enabled<br>DMA Channel 0 Enabled<br>Channel A DMA Selected | $A4F7 |
| RxCB = Ext. RxCB<br>DMA Mode<br>DTR_B, DSR_B Active<br>DTR_A, DSR_A Active<br>IP Bus Clock = 8 MHz | TxCB = Ext. TxCB<br>EIA-422_B Drivers Disabled<br>EIA-422_A Drivers Disabled<br>DMA Channel 1 Enabled<br>Channel A DMA Selected | $136F |
| RxCB = Ext. TxCB<br>DMA Mode<br>DTR_B, DSR_B Active<br>DTR_A, DSR_A Active<br>IP Bus Clock = 8 MHz | TxCB = Ext. RxCB<br>EIA-422_B Drivers Enabled<br>EIA-422_A Drivers Enabled<br>Both DMA Channels Enabled<br>Channel A DMA Selected | $47EF |
| RxCB = Ext. RxCB<br>DMA Mode<br>DTR_B, DSR_B Active<br>DTR_A, DSR_A Active<br>IP Bus Clock = 8 MHz | TxCB = Ext. TxCB<br>EIA-422_B Drivers Enabled<br>EIA-422_A Drivers Enabled<br>Both DMA Channels Enabled<br>Channel B DMA Selected | $17E6 |
| RxCB = Ext. RxCB<br>DMA Mode<br>DTR_B, DSR_B Active<br>DTR_A, DSR_A Active<br>IP Bus Clock = 32 MHz | TxCB = Ext. TxCB<br>EIA-422_B Drivers Enabled<br>EIA-422_A Drivers Enabled<br>Both DMA Channels Enabled<br>Transmit DMA Selected | $17E1 |
| RxCB = Ext. RxCB<br>DMA Mode<br>DTR_B, DSR_B Active<br>DTR_A, DSR_A Active<br>IP Bus Clock = 32 MHz | TxCB = Ext. TxCB<br>EIA-422_B Drivers Enabled<br>EIA-422_A Drivers Enabled<br>Both DMA Channels Enabled<br>Receive DMA Selected | $17E0 |

**Figure 38 Some Valid IPCR_A Configurations**

When DMA Mode is not selected, IPCR_A Bit 11 is ‘1’, data written to IPCR_A Bits 6, 7, 9 and 10 is not used but is stored and can be read back. These bits replace the DMA request pins of the Z16C30 when DMA Mode is selected. Data written to IPCR_A Bits 5 and 8 is always ignored. These read only bits are used to monitor the DSR inputs from the IP I/O Interface.

## Transmission Driver Enables

The EIA-232 drivers on the single ended Transmit Data, Transmit Clock, Request To Send and Data Terminal Ready lines are always enabled. The single ended and differential transmission signals are routed to separate I/O Interface pins.

Several pins on the Z16C30 can be used for general I/O or DMA handshake signals. On the IP-MP-SERIAL the DMA Receive Request (RxREQ) pins are used as general I/O when IPCR_A Bit 11 is '1', the default state. When IPCR_A Bit 11 is set to '0', IPCR_A bits 7 and 10 are substituted for the RxREQ pins so that DMA transfers can be performed. The RxREQ pin of each channel or the corresponding IPCR_A bit is used to control the EIA-485/EIA-422 drivers on the differential Request To Send and Data Terminal Ready lines of that channel. The RxREQ pin or corresponding IPCR_A bit must be programmed to output a '1' to enable the drivers. Programming the RxREQ pin or IPCR_A bit to output a '0' disables the drivers. **Refer to the Zilog documentation for details on programming the I/O Control Register bits in the USC that controls the RxREQ pin.** TheRxREQ pin control is contained in Z16C30's I/O Control Register.

The DMA Receive Acknowledge (RxACK) signals are not used to perform DMA transfers. The RxACK pin of each channel is used only to control the EIA-485/EIA-422 driver on the differential Transmit Data line of that channel. The RxACK pin must be programmed to output a '1' to enable the drivers. Programming the RxACK pin to output a '0' disables the drivers. **Refer to the Zilog documentation for details on programming the Hardware Configuration Register bits in the USC that controls the RxACK pin.** The RxACK pin control is contained in the Z16C30's Hardware Configuration Register.

| DMA Mode IPCR_A [11] = 0 IPCR_A Bit | Non DMA Mode IPCR_A [11] = 1 Z16C30 Pin | State | Z16C30 Register | EIA-422 Drivers Enabled |
|---|---|---|---|---|
| 7 | RxREQA | Output 1 | I/O Control | RTS_A and DTR_A |
| | RxACKA | Output 1 | Hardware Config. | TxD_A |
| 10 | RxREQB | Output 1 | I/O Control | RTS_B and DTR_B |
| | RxACKB | Output 1 | Hardware Config. | TxD_B |

**Figure 39 Transmission Line Driver Controls**

## Modem Control Signals

The Clear To Send (CTS) and Data Carrier Detect (DCD) pins of the Z16C30 can be used as a transmitter enable (CTS) and as a receiver enable (DCD) or as general inputs. The current state of the Data Set Ready (DSR) lines can be read through bits 5 and 8 of IPCR_A. On the IP-MP-SERIAL the DMA Transmit Request (TxREQ) handshake signals are used to support Data Terminal Ready (DTR) when IPCR_A Bit 11 is '1', the default state. When IPCR_A Bit 11 is set to '0', IPCR_A bits 6 and 9 are used to support DTR instead of the TxREQ pins so that DMA transfers can be performed. The DMA Transmit Acknowledge (TxACK) signals are not use for DMA transfers and are wired to the Request To Send (RTS) drivers. **Refer to the Zilog documentation for details on programming the I/O Control Register bits in the USC that controls the CTS, DCD, TxREQ and TxACK pins.** The CTS, DCD, and TxREQ pin control is contained in Z16C30's I/O Control Register. The TxACK pin control is contained in the Z16C30's Hardware Configuration Register. The Z16C30 can be programmed to generate an interrupt on either the rising or falling edge of the CTS and DCD inputs. The IP-MP-SERIAL does not generate an interrupt for any transitions on DSR.

| DMA Mode IPCR_A [11] = 0 IPCR_A Bit | Non DMA Mode IPCR_A [11] = 1 Z16C30 Pin | Z16C30 Register | Modem Control Signals |
|---|---|---|---|
| 6 | TxREQA | I/O Control | DTR_A |
| | TxACKA | Hardware Config. | RTS_A |
| | CTSA | I/O Control | CTS_A |
| | DCDA | I/O Control | DCD_A |
| 5 | | | DSR_A |
| 9 | TxREQB | I/O Control | DTR_B |
| | TxACKB | Hardware Config. | RTS_B |
| | CTSB | I/O Control | CTS_B |
| | DCDB | I/O Control | DCD_B |
| 8 | | | DSR_B |

**Figure 40 Modem Controls**

## IP Control Registers Function Diagrams

Figures 41 through 43 summarize the functions of the IndustryPack Control register (IPCR_A and IPCR_B) on the IP-MP-SERIAL in a graphical format. The bold lines are the control lines driven by the IP Control Registers. Figure 41 shows the control register bits that enable the Channel A data receivers and control the Transmit and Receive Clock Multiplexers. Figure 42 shows the control register bits that perform the same functions for Channel B. Figure 43 shows the Control Register bits that configure the Interrupt Request logic. This logic either maps both Channel A and Channel B interrupt requests to Interrupt Request 0 or maps the Channel A interrupt request to Interrupt Request 0 and the Channel B interrupt request to Interrupt Request 1. Figure 43 also shows the control register bits that configure the DMA Mode logic. This is the logic that maps the four DMA requests form the Z16C30 to the two IP interface DMA Request lines. When the IP-MP-SERIAL is not configured for DMA Mode, it supports the use of the DMA Request pins of the Z16C30 as modem handshake signals to maximize software compatibility with the IP-Universal Serial.

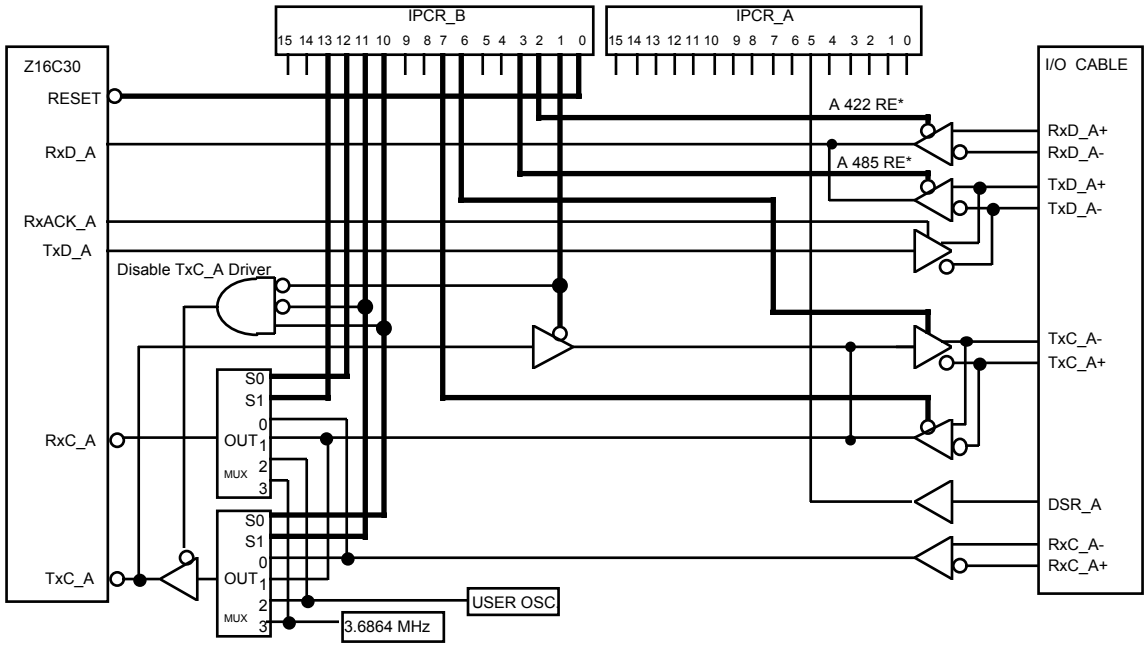


**Figure 41 Channel A Clock and Data Control**

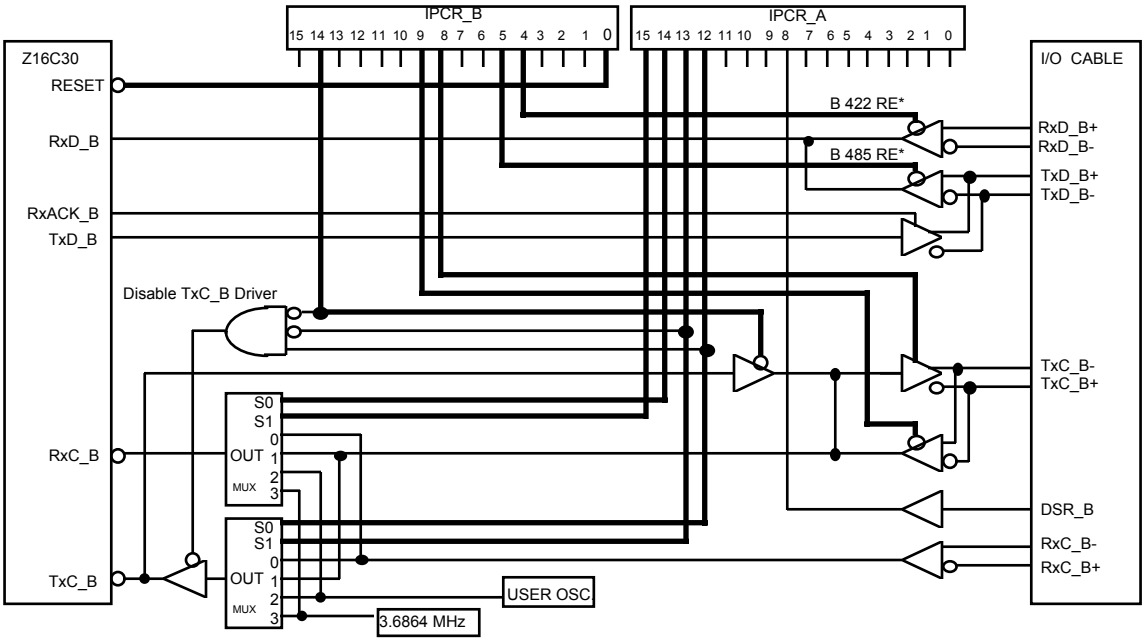


**Figure 42 Channel B Clock and Data Control**

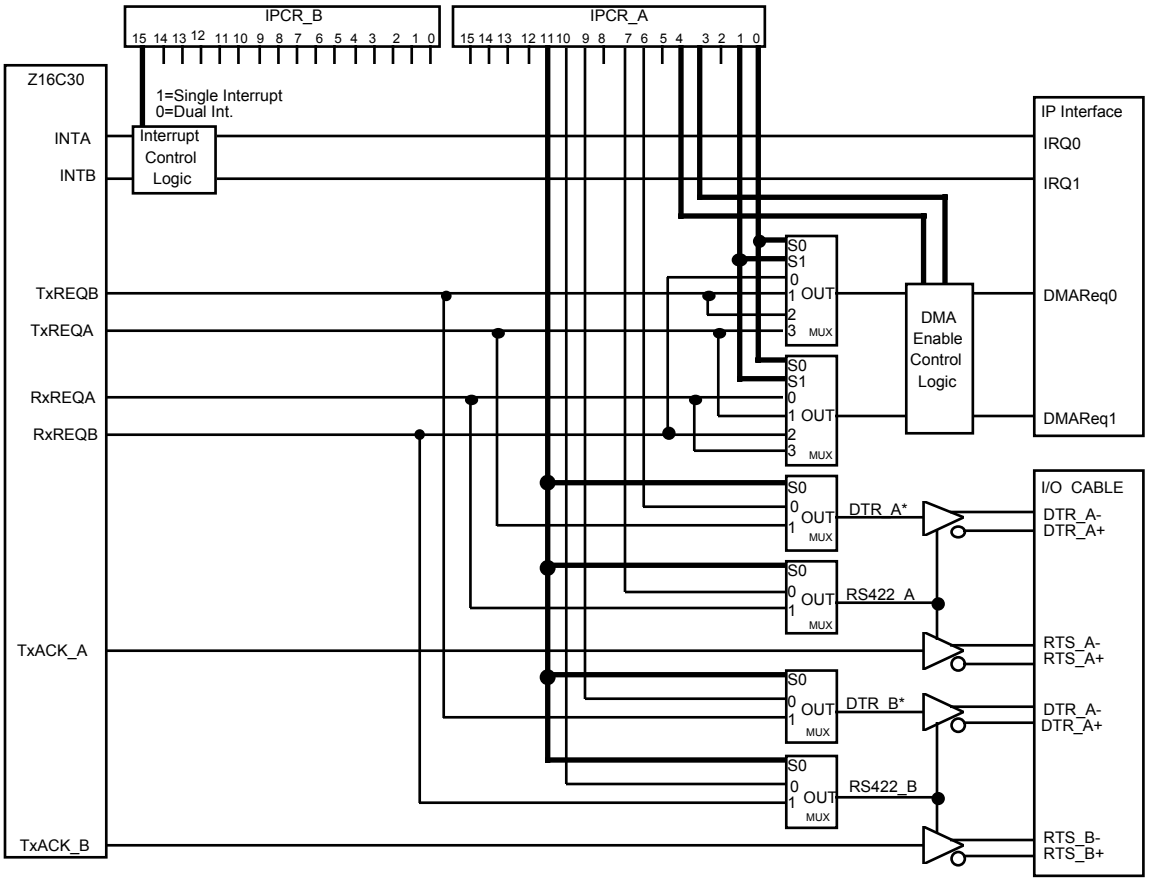


**Figure 43 Interrupt and DMA Control**

## Multi-Protocol Serial Sample Test Configuration

The following series of register accesses to an IP-MP-SERIAL will allow the user to output ASCII characters at 9600 b/s from pin 3 of the IP I/O interface. This can be observed with an oscilloscope, logic probe, lighted break-out box, or an EIA-232 terminal.

The base address of the IndustryPack I/O Space should be added to the register offset given in Figure 44 below when using a debugger to write the values. This is commonly $FF6000 for Pack A of the VIPC310 Carrier Board, and is $FFF58000 for Pack A of the Motorola MVME162.

All values in the Figure 44 below are in hexadecimal, and should be written using WORD (16-bit) accesses.

Note that for proper EIA-232 operation, resistor packs RN3, RN4 and RN5 should be removed from the IP-MP-SERIAL prior to its installation on the carrier board.

| Register | Offset | Value |
|---|---|---|
| IPCR_B | $0A | $FEAB |
| IPCR_A | $4A | $FFFF |
| BCR | $08 | $0005 |
| CCAR_A | $40 | $0400 |
| CCAR_A | $40 | $0000 |
| CMCR_A | $50 | $0324 |
| HCR_A | $52 | $0001 |
| TC0R_A | $6E | $0017 |
| RMR_A | $62 | $0002 |
| TMR_A | $72 | $0002 |

**Figure 44 IndustryPack Control Register Values for Testing Channel A, EIA–232**

Characters may now be written to the Channel A Transmit Data Register for conversion to serial data. Word writes to offset $60 will result in the transmission of two characters, while byte writes to offset $61 will result in the transmission of a single character.

Changing the value written to the HCR (offset $52) to $0081 will drive the TxACKA pin of the Z16C30 low and activate RTS_A. This will allow most EIA-232 terminals to send data to the IP-MP-SERIAL on pin 5 of the IP I/O interface. The data received can be read from the Receive Data Register using word reads from offset $60, or byte reads from offset $61.

Channel B can be tested in s similar manner. The series of IP-MP-SERIAL register accesses listed in Figure 45 will allow the user to output ASCII characters at 9600 b/s from pin 28 of the IP I/O interface. This can be observed with an oscilloscope, logic probe, lighted break-out box, or an EIA-232 terminal.

| Register | Offset | Value |
|---|---|---|
| IPCR_B | $0A | $FEAB |
| IPCR_A | $4A | $FFFF |
| BCR | $08 | $0005 |
| CCAR_B | $00 | $0400 |
| CCAR_B | $00 | $0000 |
| CMCR_B | $10 | $0324 |
| HCR_B | $12 | $0001 |
| TC0R_B | $2E | $0017 |
| RMR_B | $22 | $0002 |
| TMR_B | $32 | $0002 |

**Figure 45 IndustryPack Control Register Values for Testing Channel B, EIA–232**

Characters may now be written to the Channel B Transmit Data Register for conversion to serial data. Word writes to offset $20 will result in the transmission of two characters, while byte writes to offset $21 will result in the transmission of a single character.

Changing the value written to the HCR (offset $12) to $0081 will drive the TxACKB pin of the Z16C30 low and activate RTS_B. This will allow most EIA-232 terminals to send data to the IP-MP-SERIAL on pin 30 of the IP I/O interface. The data received can be read from the Receive Data Register using word reads of offset $20, or byte reads of offset $21.

## Asynchronous Bit Rate Time Constants

The Channel Mode Register of each channel (CMR_A & CMR_B) of the Z16C30 is programmed to specify the communications protocol to be used. This register defaults to Asynchronous mode with 16 clocks per bit. Figure 43 below lists time constants that should be loaded into the Time Constant Register (TC0R_A, TC1R_A, TC0R_B or TC1R_B) to generate standard asynchronous bit rates from 50 b/s to 115.2 Kb/s from the 3.6864 MHz on board oscillator. The Baud Rate Generators on the Z16C30 divide the input clock by the time constant plus one. The following formula can be used to calculate the time constant for 32 clocks per bit asynchronous mode, for 64 clocks per bit asynchronous mode and for one clock per bit synchronous mode. The term “Hz” is either 3686400 for the on board oscillator or the frequency of the User Supplied Oscillator.

$$Time\_Constant = \left( \frac{Hz}{Bit\_Rate \times Clocks\_per\_Bit} \right) - 1$$

| Bit Rate | 16x Time Constant Decimal | 16x Time Constant Hexadecimal | Error |
|---|---|---|---|
| 115,200 | 1 | $0001 | 0% |
| 76,800 | 2 | $0002 | 0% |
| 57,600 | 3 | $0003 | 0% |
| 38,400 | 5 | $0005 | 0% |
| 19,200 | 11 | $000B | 0% |
| 9600 | 23 | $0017 | 0% |
| 7200 | 31 | $001F | 0% |
| 4800 | 47 | $002F | 0% |
| 3600 | 63 | $003F | 0% |
| 2400 | 95 | $005F | 0% |
| 2000 | 114.2 | $0072 | 0.1739% |
| 1800 | 127 | $007F | 0% |
| 1200 | 191 | $00BF | 0% |
| 600 | 383 | $017F | 0% |
| 300 | 767 | $02FF | 0% |
| 150 | 1535 | $05FF | 0% |
| 134.5 | 1712 | $06B0 | 0.0007% |
| 110 | 2093.5 | $082D | 0.0260% |
| 75 | 3071 | $0BFF | 0% |
| 50 | 4607 | $11FF | 0% |

**Figure 46 Time Constants for Standard Asynchronous Bit Rates**

## Multi-Protocol Serial Sample DMA Configuration

Figure 47 lists register values that initialize the Motorola MVME162FX and IP-MP-SERIAL for DMA operation. These register values assume that the IP is in slot A on the MVME162FX and that the Channel A transmit data output is connected to the Channel A receive data input. DMA Channel A on the MVME162FX processes DMAReq0, which is the IP transmit DMA request. DMA Channel B on the MVME162FX processes DMAReq1, which is the IP receive DMA request. The receive DMA threshold is set to one byte in the receive FIFO. The transmit DMA threshold is set to 24 empty bytes in the transmit FIFO. After setting the threshold, the Command/Status register is programmed to read the number of characters in the FIFO buffers so that the threshold is not accidentally modified. The MVME162FX DMA byte counts are set to transfer 82 bytes. This sample DMA configuration expects 82 bytes of data to be loaded at $FFE0E200 before the registers are written. As soon as the DMA Channel A of the MVME162FX is enabled, the data at $FFE0E200 will be transferred to the Channel A transmitter of the IP. As soon as the DMA Channel B of the MVME162FX is enabled, the data in the Channel A receiver FIFO of the IP will be DMA transferred to memory starting at $FFE0E260.

| Register | Address | Value | Comment |
|---|---|---|---|
| 162 IP Clock (8 MHz) | $FFFBC01D | $00 | 8 MHz IP Clock |
| 162 IP Clock (32 MHz) | $FFFBC01D | $01 | 32 MHz IP Clock |
| 162 DMA Arbitration | $FFFBC01E | $00 | Round Robin |
| 162 DMA A Cntl. Reg. 1 | $FFFBC024 | $08 | 16 bit accesses |
| 162 DMA A Cntl. Reg. 2 | $FFFBC025 | $1D | Write, DMAEnd & Time Out on |
| 162 DMA A Mem. Addr. | $FFFBC028 | $FFE0E200 | Pointer to Tx Mem. Buff. |
| 162 DMA A IP Addr. | $FFFBC02C | $00 | Must be 0 for Std. DMA |
| 162 DMA A Byte Count | $FFFBC030 | $52 | 82 byte transfer |
| 162 DMA B Cntl. Reg. 1 | $FFFBC03C | $0A | 16 bit accesses, DMA B to slot A |
| 162 DMA B Cntl. Reg. 2 | $FFFBC03D | $19 | Read, DMAEnd & Time Out on |
| 162 DMA B Mem. Addr. | $FFFBC040 | $FFE0E260 | Pointer to Rx Mem. Buff. |
| 162 DMA B IP Addr. | $FFFBC044 | $00 | Must be 0 for Std. DMA |
| 162 DMA B Byte Count | $FFFBC048 | $52 | 82 byte transfer |
| IPCR_B | $FFF5800A | $FEAB | 3.6864 MHz Clk, 422 Rec. Ena. |
| IPCR_A (8 MHz) | $FFF5804A | $F7E7 | DMA Mode Ena., 8 MHz IP Clk. |
| IPCR_A (32 MHz) | $FFF5804A | $F7E3 | DMA Mode Ena., 32 MHz IP Clk. |
| IP BCR | $FFF58008 | $0005 | Z16C30 Bus Cntl. Reg. Std. Value |
| IP CCAR_A | $FFF58040 | $0400 | Reset Z16C30_A |
| IP CCAR_A | $FFF58040 | $0000 | Clear Reset Z16C30_A |
| IP IOCR_A | $FFF58056 | $0500 | TxRq & RxRq DMA, CTS & DCD in |
| IP CMCR_A | $FFF58050 | $0324 | TxC Pin>BRG0>TxC, BRG0>RxC |
| IP TC0R_A | $FFF5806E | $0017 | Time Const. for 9600 bps |
| IP HCR_A | $FFF58052 | $0081 | TxACK Pin Low, BRG0 on |
| IP RCSR_A | $FFF58064 | $7000 | Setup to write RxREQ threshold |
| IP RICR_A | $FFF58066 | $0100 | DMA REQ = RxFIFO > 1 byte |
| IP RCSR_A | $FFF58064 | $5000 | Protect threshold |
| IP TCSR_A | $FFF58074 | $7000 | Setup to write TxREQ threshold |
| IP TICR_A | $FFF58076 | $1800 | DMA REQ = TxFIFO < 8 bytes |
| IP TCSR_A | $FFF58074 | $5000 | Protect threshold |
| IP RMR_A | $FFF58062 | $0002 | Enable IP, Unconditional Recv. |
| IP TMR_A | $FFF58072 | $0002 | Enable IP, Unconditional Txmt. |
| 162 DMA A Enable Reg. | $FFFBC022 | $01 | Enable DMA Writes to TxD FIFO |
| 162 DMA B Enable Reg. | $FFFBC03A | $01 | Enable DMA Reads from RxD FIFO |

**Figure 47 MVME162FX and IP Register Values for DMA Testing**

# ID PROM

Every IP contains an ID PROM with a size of 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 a particular revision IP, it may check for it directly.

Standard data in the ID PROM on the IP-MP-SERIAL is shown in Figure 48. For more information on IP ID PROMs, refer to the IndustryPack Logic Interface Specification, available from SBS Technologies. The ID PROM on the IP-MP-SERIAL is implemented in the Altera FPGA device.

The location of the ID PROM in the host's address space is dependent on the carrier board used. For most VMEbus carriers 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. RM1260 address may be derived from Figures 45 below by multiplying the addresses given by two, then subtracting one. RM1270 addresses may be derived by multiplying the addresses given by two, then adding one.

| Address | Description | Rev C Contents | Rev D Contents |
|---|---|---|---|
| 3F<br><br>19 | (available for user) | | |
| 17 | CRC for bytes used | 6C | 08 |
| 15 | Number of bytes used | 0C | 0C |
| 13 | Driver ID, high byte | 00 | 00 |
| 11 | Driver ID, low byte | 00 | 00 |
| 0F | Reserved | 00 | 00 |
| 0D | Revision | C2 | D1 |
| 0B | Model number IP-MP-SERIAL | 51 | 51 |
| 09 | Manufacturer ID SBS Technologies | F0 | F0 |
| 07 | ASCII "H" | 48 | 48 |
| 05 | ASCII "A" | 41 | 41 |
| 03 | ASCII "P" | 50 | 50 |
| 01 | ASCII "I" | 49 | 40 |

**Figure 48 Standard ID PROM Data (hex)**

# I/O Pin Wiring

This section lists the I/O Interface pin assignments for IP-MP-SERIAL.

The pin numbers given in Figures 49 and 50 correspond to numbers on the 50-pin IndustryPack I/O connector (P2), to the wires on a 50-pin flat cable plugged into a standard IP Carrier board and to the screw terminal numbers on the IP-Terminal block.

| Flat Cable Channel A | D-Shell | I/O | EIA-232 Signal | EIA-422/EIA-485 Signal |
|---|---|---|---|---|
| 1 | 1 | | GND | GND |
| 3 | 2 | O | TxD | |
| 5 | 3 | I | RxD | $RxD_B$ |
| 7 | 4 | O | RTS | |
| 9 | 5 | I | CTS | $CTS_A$ |
| 11 | 6 | I | DSR | $DSR_A$ |
| 13 | 7 | | GND | GND |
| 15 | 8 | I | DCD | $DCD_A$ |
| 17 | 9 | I | | $RxC_B$ |
| 19 | 10 | I | | $DCD_B$ |
| 21 | 11 | O | | $TxD_A$ |
| 23 | 12 | I/O | | $TxC_A$ |
| 25 | 13 | I | | $CTS_B$ |
| 2 | 14 | O | | $TxD_B$ |
| 4 | 15 | I/O | TxC (in) | $TxC_B$ |
| 6 | 16 | I | | $RxD_A$ |
| 8 | 17 | I | RxC | $RxC_A$ |
| 10 | 18 | O | | $RTS_A$ |
| 12 | 19 | O | | $RTS_B$ |
| 14 | 20 | O | DTR | |
| 16 | 21 | O | | $DTR_A$ |
| 18 | 22 | I | | $DSR_B$ |
| 20 | 23 | O | | $DTR_B$ |
| 22 | 24 | O | TxC | |
| 24 | 25 | | | Fused +5V |

**Figure 49 I/O Pin Definitions (Channel A)**

The Channel A differential Transmit Clock pair ($TxC_A$ & $TxC_B$) can be programmed as either an input or an output. For EIA-232 operation, the $TxC_B$ pins can be used as an input to an EIA-232 receiver while EIA-422/485 that connects to TxC pairs are disabled on the receive side.

| Flat Cable Channel B | D-Shell | I/O | EIA-232 Signal | EIA-422/EIA-485 Signal |
|---|---|---|---|---|
| 26 | 1 | | GND | GND |
| 28 | 2 | O | TxD | |
| 30 | 3 | I | RxD | $RxD_B$ |
| 32 | 4 | O | RTS | |
| 34 | 5 | I | CTS | $CTS_A$ |
| 36 | 6 | I | DSR | $DSR_A$ |
| 38 | 7 | | GND | GND |
| 40 | 8 | I | DCD | $DCD_A$ |
| 42 | 9 | I | | $RxC_B$ |
| 44 | 10 | I | | $DCD_B$ |
| 46 | 11 | O | | $TxD_A$ |
| 48 | 12 | I/O | | $TxC_A$ |
| 50 | 13 | I | | $CTS_B$ |
| 27 | 14 | O | | $TxD_B$ |
| 29 | 15 | I/O | TxC (in) | $TxC_B$ |
| 31 | 16 | I | | $RxD_A$ |
| 33 | 17 | I | RxC | $RxC_A$ |
| 35 | 18 | O | | $RTS_A$ |
| 37 | 19 | O | | $RTS_B$ |
| 39 | 20 | O | DTR | |
| 41 | 21 | O | | $DTR_A$ |
| 43 | 22 | I | | $DSR_B$ |
| 45 | 23 | O | | $DTR_B$ |
| 47 | 24 | O | TxC | |
| 49 | 25 | | | Fused +5V |

**Figure 50 I/O Pin Definitions (Channel B)**

The Channel B differential Transmit Clock pair ($TxC_A$ & $TxC_B$) can be programmed as either an input or an output. For EIA-232 operation, the $TxC_B$ pins can be used as an input to an EIA-232 receiver while EIA-422/485 that connects to TxC pairs are disabled on the receive side.

The IP-MP-SERIAL supports EIA-485 bi-directional, multipoint, operation on the $TxD_A$ and $TxD_B$ pins. Refer to the Programming section earlier in this manual for details on controlling the transceiver.

# IP Logic Interface Pin Assignment

Figure 51 shows the pin assignments for the IndustryPack Logic Interface on the IP-MP-SERIAL. Pins marked n/c below are defined by the specification, but are not used on IP-MP-SERIAL. Also see the User Manual for your IP Carrier board for more information.

| | | | |
|---|---|---|---|
| GND | GND | 1 | 26 |
| CLK | +5V | 2 | 27 |
| Reset* | R/W* | 3 | 28 |
| D0 | IDSel* | 4 | 29 |
| D1 | DMAReq0* | 5 | 30 |
| D2 | MEMSel* | 6 | 31 |
| D3 | DMAReq1* | 7 | 32 |
| D4 | INTSel* | 8 | 33 |
| D5 | DMAAck* | 9 | 34 |
| D6 | IOSel* | 10 | 35 |
| D7 | n/c | 11 | 36 |
| D8 | A1 | 12 | 37 |
| D9 | DMAEnd* | 13 | 38 |
| D10 | A2 | 14 | 39 |
| D11 | n/c | 15 | 40 |
| D12 | A3 | 16 | 41 |
| D13 | IntReq0* | 17 | 42 |
| D14 | A4 | 18 | 43 |
| D15 | IntReq1* | 19 | 44 |
| BS0* | A5 | 20 | 45 |
| BS1* | n/c | 21 | 46 |
| –12V | A6 | 22 | 47 |
| +12V | Ack* | 23 | 48 |
| +5V | n/c | 24 | 49 |
| GND | GND | 25 | 50 |

Note 1: The no-connect (n/c) signals above are defined by the IndustryPack 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 IndustryPack.

**Figure 51 Logic Interface Pin Assignment**

# Construction and Reliability

IndustryPacks were conceived and engineered for rugged industrial environments. The IP-MP-SERIAL is constructed out of 0.062 inch thick FR4 V0 material. The six copper layers consist of two signal layers on the top and bottom, and four internal layers. Two internal layers are dedicated to power and ground planes. Two additional layers are used for signal wiring.

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

The IndustryPack connectors are keyed, shrouded and have 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 to the carrier with four M2 metric 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 IndustryPack provides a low temperature coefficient of 0.89 W/°C for uniform heat. This is based on the temperature coefficient of the base FR4 material of 0.31 W/m-°C, taking into account the thickness and area of the IP. This coefficient means that if 0.89 Watts is applied uniformly on the component side, then the temperature difference between the component and the solder side is one degree Celsius.

# Repair

## Service Policy

Before returning a product for repair, verify as soon 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 it is available, and ship prepaid and insured with the RMA number clearly written on the outside of the package. Include the return address and telephone number of a technical contact. For out-of-warranty repairs, a purchase order for repair charges must accompany the return. SBS Technologies will not be responsible for damages due to improper packaging of returned items. For service of SBS Technologies modular I/O products not purchased directly from SBS Technologies, contact your reseller. Products returned to SBS Technologies for repair by other than the original customer will be treated as out-of-warranty.

SBS Technologies, Inc.
1284 Corporate Center Drive
St. Paul, MN  55121-1245

Tel: (651) 905-4700

FAX: (651) 905-4701

Email: support.commercial@sbs.com

# IP-MP-SERIAL Specifications

This section gives the technical specification for the standard IP-MP-SERIAL.

| | |
|---|---|
| Reset Recovery Time: | 330 milliseconds |
| Number of Channels: | Two serial channels<br>Up to 10 Mb/s data rate<br>Address/Data bit support<br>16- or 32-bit CRC generation/checking<br>Low power CMOS |
| I/O Interface: | Serial synchronous and asynchronous<br>EIA-232, EIA-422 and EIA-485<br>HDLC, SDLC, Bisync, NRZ, NRZI, Biphase. |
| Software Interface: | Two Control Registers<br>Z16C30 Control Registers<br>ID PROM |
| Interrupt: | Programmable from Z16C30 |
| Initialization: | Master Reset initializes Z16C30<br>Software reset through IP Control Register B also resets Z16C30 |
| Access Modes: | Word I/O Space to IP Control Registers<br>Byte/Word I/O Space to all Z16C30 Reg.<br>Byte/Word in ID Space<br>Vectored interrupt |
| Wait States 8 MHz IP Clock (Rev. C): | Depends on access type. ID PROM requires one.<br>IPCR_A and IPCR_B require one.<br>Z16C30 USC I/O requires 5 typically<br>Z16C30 USC Interrupt Ack. requires 6 typically<br>[125 nS per wait state] |
| Wait States 8 MHz IP Clock (Rev. D): | Depends on access type. ID PROM requires one.<br>IPCR_A and IPCR_B require one.<br>Z16C30 USC I/O requires 5 typically<br>Z16C30 USC Interrupt Ack. requires 5 typically<br>[125 nS per wait state] |
| Wait States 32 MHz IP Clock (Rev. D): | Depends on access type. ID PROM requires one.<br>IPCR_A and IPCR_B require one.<br>Z16C30 USC I/O requires 8 typically<br>Z16C30 USC Interrupt Ack. requires 13 typically<br>[31.25 nS per wait state] |
| Wait States 32 MHz IP Clock (Rev. D): | Depends on access type. ID PROM requires one.<br>IPCR_A and IPCR_B require one.<br>Z16C30 USC I/O requires 8 typically<br>Z16C30 USC Interrupt Ack. requires 10 typically<br>[31.25 nS per wait state] |

Synchronous Clock/Data Skew (Rev. C):

| | | |
|---|---|---|
| From Ext.TxCA in | to TxCA pin | = 16.8 nS |
| From Ext. TxCA in | to RxCA pin | = 17.2 nS |
| From TxCA pin | to Ext. TxCA out | = 8.4 nS |
| From Ext. RxCA in | to TxCA pin | = 16.1 nS |
| From Ext. RxCA in | to RxCA pin | = 15.6 nS |
| From Ext. TxCB in | to TxCB pin | = 11.0 nS |
| From Ext. TxCB in | to RxCB pin | = 11.0 nS |
| From TxCB pin | to Ext. TxCB out | = 7.2 nS |
| From Ext. RxCB in | to TxCB pin | = 10.4 nS |
| From Ext. RxCB in | to RxCB pin | = 10.6 nS |

Synchronous Clock/Data Skew -- Option 95121 on 8 MHz IP Bus: (Rev. C):

| | | |
|---|---|---|
| From Ext.TxCA in | to TxCA pin | = 265.0 nS |
| From Ext. TxCA in | to RxCA pin | = 265.0 nS |
| From Ext. RxCA in | to TxCA pin | = 265.0 nS |
| From Ext. RxCA in | to RxCA pin | = 265.0 nS |
| From Ext. TxCB in | to TxCB pin | = 265.0 nS |
| From Ext. TxCB in | to RxCB pin | = 265.0 nS |
| From Ext. RxCB in | to TxCB pin | = 265.0 nS |
| From Ext. RxCB in | to RxCB pin | = 265.0 nS |
| From Ext. TxCB in | to RxCB pin | = 265.0 nS |
| From TxCB pin | to Ext. TxCB out | = 7.2 nS |
| From TxCA pin | to Ext. TxCA out | = 7.4 nS |

Synchronous Clock/Data Skew -- Option 95121 on 32 MHz IP Bus: (Rev. C):

| | | |
|---|---|---|
| From Ext.TxCA in | to TxCA pin | = 69.0 nS |
| From Ext. TxCA in | to RxCA pin | = 69.0 nS |
| From Ext. RxCA in | to TxCA pin | = 69.0 nS |
| From Ext. RxCA in | to RxCA pin | = 69.0 nS |
| From Ext. TxCB in | to TxCB pin | = 69.0 nS |
| From Ext. TxCB in | to RxCB pin | = 69.0 nS |
| From Ext. RxCB in | to TxCB pin | = 69.0 nS |
| From Ext. RxCB in | to RxCB pin | = 69.0 nS |
| From Ext. TxCB in | to RxCB pin | = 69.0 nS |
| From TxCB pin | to Ext. TxCB out | = 7.2 nS |
| From TxCA pin | to Ext. TxCA out | = 7.4 nS |

Synchronous Clock/Data Skew (Rev. D): in EIA-422/485 mode:

| | | |
|---|---|---|
| From Ext.TxCA in | to TxCA pin | = 22.0 nS |
| From Ext. TxCA in | to RxCA pin | = 24.4 nS |
| From TxCA pin | to Ext. TxCA out | = 15.7 nS |
| From Ext. RxCA in | to TxCA pin | = 23.6 nS |
| From Ext. RxCA in | to RxCA pin | = 23.6 nS |
| From Ext. TxCB in | to TxCB pin | = 23.9 nS |
| From Ext. TxCB in | to RxCB pin | = 23.9 nS |
| From TxCB pin | to Ext. TxCB out | = 18.7 nS |
| From Ext. RxCB in | to TxCB pin | = 25.5 nS |
| From Ext. RxCB in | to RxCB pin | = 25.5 nS |

| | | | |
|---|---|---|---|
| | in EIA-232 mode: | | |
| | From Ext.TxCA in | to TxCA pin | = 17.0 nS |
| | From Ext. TxCA in | to RxCA pin | = 17.4 nS |
| | From TxCA pin | to Ext. TxCA out | = 15.7 nS |
| | From Ext. RxCA in | to TxCA pin | = 15.5 nS |
| | From Ext. RxCA in | to RxCA pin | = 15.5 nS |
| | From Ext. TxCB in | to TxCB pin | = 17.8 nS |
| | From Ext. TxCB in | to RxCB pin | = 17.8 nS |
| | From TxCB pin | to Ext. TxCB out | = 18.7 nS |
| | From Ext. RxCB in | to TxCB pin | = 17.3 nS |
| | From Ext. RxCB in | to RxCB pin | = 17.3 nS |

| | |
|---|---|
| Power Requirements (Rev. C): | +5 VDC, 340 mA typical<br>+12 VDC, 60mA typical<br>–12 VDC, 70 mA typical |
| Power Requirements (Rev. D):<br>(RS-422 interface, 32 MHz IP bus, Bysync mode, 3.6864MHz Tx and Rx Clock, all ports are sending and receiving data, default jumper setting) | +5 VDC, 390 mA typical<br>+12 VDC, 0mA typical<br>–12 VDC, 20 mA typical |
| Environmental: | Operating temperature: 0° to 70° C<br>Humidity: 5% - 95% non-condensing<br>Storage temperature: –40° to +85° C |
| Interface Options: | 50 pin flat cable<br>50 screw terminal block interface User cable |
| Dimensions: | Standard Single IndustryPack width and length. 1.8 x 3.9 inches |
| Construction: | 6 Layer Printed Circuit,<br>Through Hole and Surface Mount Components. |
| Temperature Coefficient: | 0.89 W/°C for uniform heat across IP |
| Test conditions: | 20°C, typical |

# Appendix A

The following table summarizes the differences between IP-MP-SERIAL Revision C and IP-MP-SERIAL Revision D:

| | **Revision C** | **Revision D** |
|---|---|---|
| Physical appearance | Single sided board with paper sticker on the back (Type I) | Components on both sides (Type II IndustryPack) |
| Shunts | Four shunts in a group to select 422/485 bias resistors for each channel | Six shunts in a group to select 422/485 bias resistors and 232 mode for each channel |
| ID PROM revision word<br>IP-MP SERIAL<br>IP-MP SERIAL-423 | <br>0xC2<br>0xB1 | <br>0xD1<br>0xC1 |
| Resistor networks | Five in a group to select 422/485 termination and 232 receiver pull-ups. Factory default: All Resistor Networks installed. | Four in a group to select 422/485 termination. Factory default: Only two resistors are installed. Installation by user required. |
| EIA-232 input for the following signals: transmit clock (TxC), receive data (RxD), receive clock (RxC), CTS, DSR, DCD | Achieved via 422/485 drivers | Achieved via 232 drivers |
| Option 95121 (clock noise fix) | Available | Not available, not required |
| Differential termination of transmit data (TxD) line | Not available | By default, TxD lines are terminated. Removing termination is optional. |
| User oscillator location | Provided, though-hole | Provided, surface-mount |
| EIA-422 Nomenclature in this manual | Non-inverting signals are labeled with '+'. Inverting signals are labeled with '– ', for example TxD+ and TxD– | Non-inverting signals are labeled with '$_A$'. Inverting signals are labeled with '$_B$', for example $TxD_A$ and $TxD_B$ |