gridconnect

DSTni-EX User Guide

Section Three

Part Number 900-335
Revision A 3/04

Copyright & Trademark

Lantronix and the Lantronix logo, and combinations thereof are registered trademarks of Lantronix, Inc. DSTni is a registered trademark of Lantronix, Inc. Ethernet is a registered trademark of Xerox Corporation. All other product names, company names, logos or other designations mentioned herein are trademarks of their respective owners.

- Am186 is a trademark of Advanced Micro Devices, Inc.
- Ethernet is a registered trademark of Xerox Corporation.
- SPI is a trademark of Motorola, Inc.

Lantronix
15353 Barranca Parkway
Irvine, CA 92618, USA
Phone: 949-453-3990
Fax: 949-453-3995

Technical Support
Phone: 630-245-1445
Fax: 630-245-1717

Master Distributor
Grid Connect
1841 Centre Point Circle, Suite 143
Naperville, IL 60563
Phone: 630-245-1445
www.gridconnect.com

Am186 is a trademark of Advanced Micro Devices, Inc.
Ethernet is a registered trademark of Xerox Corporation.
SPI is a trademark of Motorola, Inc.

REV	Changes	Released Date
A	Reformat. Add changes from Design Spec. 1.1	3-24-04

Contents

List of Figures

List of Tables

About This User Guide

This User Guide describes the technical features and programming interfaces of the Lantronix DSTni-EX chip (hereafter referred to as “DSTni”).

DSTni is an Application Specific Integrated Circuit (ASIC)-based single-chip solution (SCS) that integrates the leading-edge functionalities needed to develop low-cost, high-performance device server products. On a single chip, the DSTni integrates an x186 microprocessor, 16K-byte ROM, 256K-byte SRAM, programmable input/output (I/O), and serial, Ethernet, and Universal Serial Bus (USB) connectivity — key ingredients for device- server solutions. Although DSTni embeds multiple functions onto a single chip, it can be easily customized, based on the comprehensive feature set designed into the chip.

Providing a complete device server solution on a single chip enables system designers to build affordable, full-function solutions that provide the highest level of performance in both processing power and peripheral systems, while reducing the number of total system components. The advantages gained from this synergy include:

- Simplifying system design and increased reliability.
- Minimizing marketing and administration costs by eliminating the need to source products from multiple vendors.
- Eliminating the compatibility and reliability problems that occur when combining separate subsystems.
- Dramatically reducing implementation costs.
- Increasing performance and functionality, while maintaining quality and cost effectiveness.
- Streamlining development by reducing programming effort and debugging time.
- Enabling solution providers to bring their products to market faster.

These advantages make DSTni the ideal solution for designs requiring x86 compatibility; increased performance; serial, programmable I/O, Ethernet, and USB communications; and a glueless bus interface.

Intended Audience

This User Guide is intended for use by hardware and software engineers, programmers, and designers who understand the basic operating principles of microprocessors and their systems and are considering designing systems that utilize DSTni.

Conventions

This User Guide uses the following conventions to alert you to information of special interest.

The symbols # and n are used throughout this Guide to denote active LOW signals.

Notes: *Notes are information requiring attention.*

Navigating Online

The electronic Portable Document Format (PDF) version of this User Guide contains *hyperlinks*. Clicking one of these hyper links moves you to that location in this User Guide. The PDF file was created with Bookmarks and active links for the Table of Contents, Tables, Figures and cross-references.

Organization

This User Guide contains information essential for system architects and design engineers. The information in this User Guide is organized into the following chapters and appendixes.

- *Section 1: Introduction*
  Describes the DSTni architecture, design benefits, theory of operations, ball assignments, packaging, and electrical specifications. This chapter includes a DSTni block diagram.
- *Section 2: Microprocessor*
  Describes the DSTni microprocessor and its control registers.
- *Section 2: SDRAM*
  Describes the DSTni SDRAM and the registers associated with it.
- *Section 3: Serial Ports*
  Describes the DSTni serial ports and the registers associated with them.
- *Section 3: Programmable Input/Output*
  Describes DSTni's Programmable Input/ Output (PIO) functions and the registers associated with them.
- *Section 3: Timers*
  Describes the DSTni timers.
- *Section 4: Ethernet Controllers*
  Describes the DSTni Ethernet controllers.
- *Section 4: Ethernet PHY*
  Describes the DSTni Ethernet physical layer core.
- *Section 5: SPI Controller*
  Describes the DSTni Serial Peripheral Interface (SPI) controller.
- *Section 5: I2C Controller*
  Describes the DSTni I^2C controller.
- *Section 5: USB Controller*
  Describes the DSTni USB controller.
- *Section 5: CAN Controllers*
  Describes the DSTni Controller Area Network (CAN) bus controllers.
- *Section 6: Interrupt Controller*
  Describes the DSTni interrupt controller.
- *Section 6: Miscellaneous Registers*
  Describes DSTni registers not covered in other chapters of this Guide.
- *Section 6: Debugging In-circuit Emulator (Delce)*
- *Section 6: Packaging and Electrical*
  Describes DSTni's packaging and electrical characteristics.
- *Section 6: Applications*
  Describes DSTni's packaging and electrical characteristics.
- *Section 6: Instruction Clocks*
  Describes the DSTni instruction clocks.
- *Section 6: DSTni Sample Code*
- *Section 6: Baud Rate Calculations*
  Provides baud rate calculation tables.

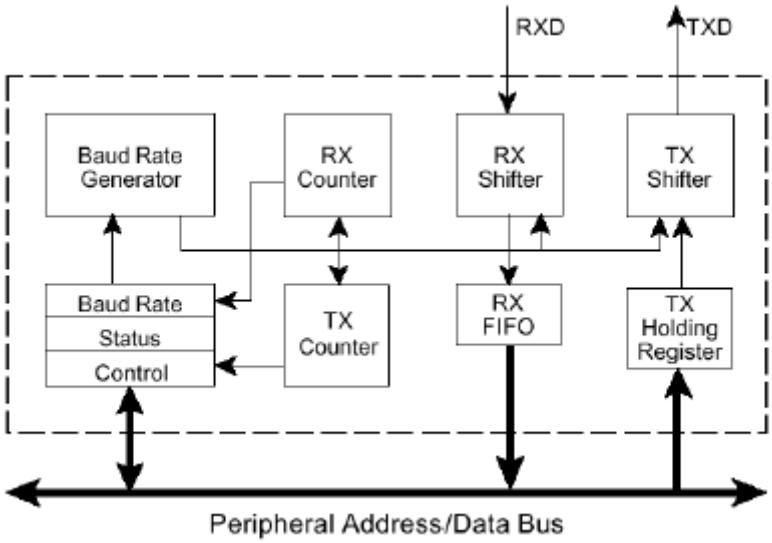
Serial Ports

Overview

This chapter describes the DSTni serial ports and the registers associated with them. Topics include:

Block Diagram

Figure 1. Serial Port Block Diagram



Theory of Operation

Features

DSTni provides four dentical asynchronous serial ports (see Figure 1). Each port operates independently and provides the following features:

- A read port and a write port for full-duplex operation with a programmable baud-rate generator.
- Supports 7-, 8-, and 9-bit operation, with even, odd, or no parity and one or two stop bits.
- Ability to transmit and detect break characters. Receive port can be programmed to generate interrupts when a break character is sent or received. It can also be programmed to generate interrupts when the next word of data can be sent or a valid word of data is received.
- Serial ports can be configured to use Direct Memory Access (DMA) for data transmission, reception, or both.

Flow Control

The DSTni serial ports support two hardware flow control protocols: DCE/DTE and Clear To Send/Ready To Receive (CTS/RTS). The hardware flow control protocol is selected using the ENRX and DTE bits in the System Auxiliary Control (AUXCON) register.

DCE/DTE Protocol

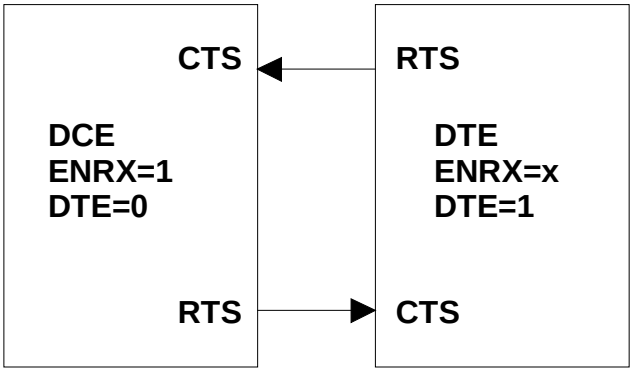
With the DCE/DTE protocol, the serial port is configured as either a Data Communication Equipment (DCE) or Data Terminal Equipment (DTE) device (see Figure 2).

- As a DTE device, the chip sends a Request To Send (RTS) when data is waiting to be sent. Then the DTE device waits for a CTS from the DCE device before it sends the frame.
- As a DCE device, the serial port receives a request to enable receiver (ENRX) from the DTE device. It then responds with an RTS signal.

To configure a serial port as a DTE device:

- Set both the associated DTE bit and the associated ENRX bit. The DTE and ENRX bits for both serial ports can be found in the System Auxiliary Control (AUXCON) register.
- Enable flow control by setting the FC bit in the Serial Port Control register.

Figure 2. DCE/DTE Protocol



Asymmetrical

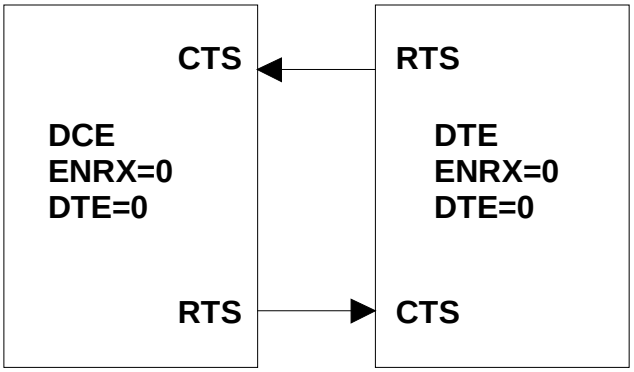
CTS/RTR Protocol

The Clear To Send/Ready To Receive (CTS/RTR) protocol supports full-duplex operation. When a device can receive data, it asserts the ready to receive output. A device with data to be sent waits for the Clear To Send input to be asserted before it sends the data. Figure 3. CTS/RTS Protocol show this protocol.

To configure a serial port to use the Clear To Send/Ready To Receive protocol:

- Clear both the ENRX and DTE for the associated serial port. The DTE and ENRX bits for both serial ports can be found in the System Auxiliary Control (AUXCON) register.
- Enable flow control by setting the FC bit in the Serial Port Control register.

Figure 3. CTS/RTS Protocol



Symmetrical

Flow Control Signals

Table 1 lists the flow control signals for the DSTni serial ports. It also shows the register settings associated with the flow control signals.

Table 1. Flow Control Signals

Flow Control Signal	Register Settings	Description
RTSx	**Auxiliary Configuration Register** RTS bit = 1 **Serial Port Configuration Register** FC bit = 1	Provides the RTS signal for the corresponding asynchronous serial port. The RTS signal is asserted when the associated Serial Port Transmit register contains data to transmit.
RTRx	**Auxiliary Configuration Register** RTS bit = 0 **Serial Port Configuration Register** FC bit = 1	Provides the RTR signal for the corresponding asynchronous serial port. The RTR signal is asserted when the associated Serial Port Receive register and FIFO do not contain data.
CTSx	**Auxiliary Configuration Register** ENRX bit = 0 **Serial Port Configuration Register** FC bit = 1	Provides the CTS signal for the corresponding asynchronous serial port. The CTS signal gates the transmission of data from the associated Serial Port Transmit register. When CTS is asserted, the transmitter transmits a frame of data, if any is available. If CTS is deasserted, the transmitter holds the data in the Serial Port Transmit register. The value of CTS is checked only at the beginning of the transmission of the frame.
ENRXx	**Auxiliary Configuration Register** ENRX bit = 1 **Serial Port Configuration Register** FC bit = 1	Provides the Enable Receiver Request for the corresponding asynchronous serial port. The ENRX signal enables the receiver for the associated serial port.

Data Transmission

To transmit a frame, the data is written to an empty Transmit Holding register. When the Transmit Shift register is empty, the frame transfers from the Holding register to the shift register, freeing the Holding register for the data for the next frame. In this way, the data for the next frame can be written while the previous frame is transmitting. The software can poll the Transmit Holding register empty bit in the Status register, or the asynchronous serial port can be configured to generate an interrupt when the Holding register becomes empty.

To minimize data-overrun errors, the receive side of the serial port contains a 4-word receive FIFO. When a frame is received, the data and error bits for that frame are placed in the FIFO and the data-ready bit in the Status register is asserted. Three more frames can be received and stored in the FIFO before data is lost, causing the data overrun error bit to be set. Even at high baud rates, this gives the software (or DMA) ample time to read the data before an overrun error occurs. If a fifth word of data is received before any data is read from the FIFO, the oldest data in the FIFO is overwritten and the overrun error bit in the Status register will be set. When the FIFO is empty, the data ready bit in the Status register clears.

DMA Transfers via Serial Ports

DSTni supports DMA transfers both to and from the serial ports. Any or all DMA channels and any or all serial ports can be used for DMA transmits or receives.

Serial Port Register Summary

Table 2 lists the programmable registers associated with the DSTni serial ports. The four DSTni Universal Asynchronous Receivers/Transmitters (UARTs) are similar, except as noted in the following sections. They are located at the following locations:

- SP0=80-8A
- SP1=10-1A
- SP2=00-0A
- SP3=E0-EA

Table 2. Serial Port Register Summary

Hex Offset	Mnemonic	Register Description	Page
Serial Port 0			
8E	///	///	///
8C	///	///	///
8A	SP0CNTRL	Serial Port 0 Auxiliary Control Register	10
88	SP0BAUD	Serial Port 0 Baud Rate Divisor Register	11
86	SP0RD	Serial Port 0 Receive Data Register	13
84	SP0TD	Serial Port 0 Transmit Data Register	14
82	SP0STS	Serial Port 0 Status Register	15
80	SP0CT	Serial Port 0 Control Register	17
Serial Port 1			
1E	///	///	///
1C	///	///	///
1A	SP1CNTRL	Serial Port 1 Auxiliary Control Register	10
18	SP1BAUD	Serial Port 1 Baud Rate Divisor Register	11
16	SP1RD	Serial Port 1 Receive Data Register	13
14	SP1TD	Serial Port 1 Transmit Data Register	14
12	SP1STS	Serial Port 1 Status Register	15
10	SP1CT	Serial Port 1 Control Register	17
Serial Port 2			
0E	///	///	///
0C	///	///	///
0A	SP2CNTRL	Serial Port 2 Auxiliary Control Register	10
08	SP2BAUD	Serial Port 2 Baud Rate Divisor Register	11
06	SP2RD	Serial Port 2 Receive Data Register	13
04	SP2TD	Serial Port 2 Transmit Data Register	14
02	SP2STS	Serial Port 2 Status Register	15
00	SP2CT	Serial Port 2 Control Register	17
Serial Port 3			
EE	///	///	///
EC	///	///	///
EA	SP3CNTRL	Serial Port 3 Auxiliary Control Register	10
E8	SP3BAUD	Serial Port 3 Baud Rate Divisor Register	11
E6	SP2RD	Serial Port 3 Receive Data Register	13
E4	SP3TD	Serial Port 3 Transmit Data Register	14
E2	SP3STS	Serial Port 3 Status Register	15
E0	SP3CT	Serial Port 3 Control Register	17

Serial Port Register Definitions

The following sections provide the serial port register definitions. In these sections, the initialization value shown is the register's initialization value at reset.

System Auxiliary Control (AUXCON) registers

The System Auxiliary Control (AUXCON) registers control both the transmit and receive sections of the serial ports.

Note: *The RTSZ, RTSF, RXM, CTSM, and RTSP bits manipulate the handshake capability of each serial port.*

Table 3. System Auxiliary Control (AUXCON) Registers

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET	Serial Port 0 = 8Ah Serial Port 1 = 1Ah Serial Port 2 = 0Ah Serial Port 3 = EAh															
FIELD	///								BRGO	FIFOD		RTSZ	RTSF	RXM	CTSM	RTSP
RESET	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
RW	R	R	R	R	R	R	R	R	RW	R W	RW	R W	R W	R W	R W	R W

Table 4. System Auxiliary Control (AUXCON) Register Definitions

Bits	Field Name	Description
15:8	///	**Reserved**
7	BRGO	**Baud Rate Generator Output** 1 = the corresponding RTS output connects to the baud rate generator output. The output signal will be 50% duty cycle if the baud rate is even. The duty cycle will be uneven by one clock if the baud rate is odd. Note: Baud Rate Generator Output cannot be used if the rate generator is using a count of 0000h or 0001h.
6:5	FIFOD	**FIFO Depth** These 2 bits set the FIFO depth (that is, the level at which the RTS is considered full. Values are from 1 to 4, default is 4. See Table 5.
4	RTSZ	**Ready to Send Zero** This bit allows the state of the RTS pin to be masked. 1 = RTS output is forced inactive. 0 = RTS output functions normally.
3	RTSF	**Ready To Send Force** This bit provides control over the state of the RTS pin. 1 = RTS output is forced active. 0 = RTS output functions normally.
2	RXM	**Receiver Mask** 1 = serial port is placed in half-duplex mode. 0 = serial port is in full-duplex mode, and the serial port receiver input is from the normal RXD pin. The receive input will be read as '0'.
1	CTSM	**CTS Mask** 1 = serial port CTS input is read inactive. 0 = serial port CTS input is from the normal CTS input pin.
0	RTSP	**RTS Polarity** 1 = RTS signal is inverted on the RTS pin. 0 = RTS signal is not inverted. This permits the RTS pin to control the TXEN pin of RS-485 drivers.

Table 5. FIFO Depth Settings

Bit [6]	Bit [5]	FIFO Depth
0	0	1
0	1	2
1	0	3 (*default*)
1	1	4

Serial Port Baud Rate Divisor Registers

All four asynchronous serial ports have a Baud Rate Divisor (BRD) register that allows the ports to operate at different rates. These registers specify a clock divisor for generating the serial clock that controls the associated serial port.

A general formula for the BRD is:

BRD = CPUCLK ÷ 16 ÷ BAUDRATE

The serial port receiver tolerates an overspeed baud rate deviation of 3.0% and an underspeed baud rate deviation of 2.5%. To help with your calculations, see Appendix B: Baud Rate Calculations for valid and invalid baud rate calculations at various microprocessor speeds.

Table 6. Serial Port Baud Rate Divisor Registers

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET	Serial Port 0 = 88h Serial Port 1 = 18h Serial Port 2 = 08h Serial Port 3 = E8h															
FIELD	BRD															
RESET	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 7. Serial Port Baud Rate Divisor Register Definitions

Bits	Field Name	Description
15:0	BRD	**Baud Rate Divisor** By setting the baud rate divisor to 0, the maximum baud rate for a given frequency (frequency/16) can be achieved (UART Clock/BRD).

Serial Port Receive Data Registers

The Serial Port Receive registers are Read Only registers that contain data received over the serial ports. The receiver is double-buffered; consequently, the receive section can be receiving a subsequent frame of data in the Receive Shift register, which is not accessible to software, while the Receive Data register is being read.

The Receive Data Ready (RDR) bit in the Serial Port Status register reports the current state of this register (see page 15). When the RDR bit is 1, the Receive register contains valid unread data. The RDR bit clears automatically when the Receive register is read.

If hardware handshaking is enabled, the CTS/ENRX signals are deasserted while the Receive register contains valid unread data. Reading the Receive register asserts the CTS/ENRX signals. This behavior prevents overrun errors, but can cause delays in character transmissions.

Table 8. Serial Port Receive Data Registers

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET	Serial Port 0 = 86h Serial Port 1 = 16h Serial Port 2 = 06h Serial Port 3 = E6h															
FIELD	///								Data							
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9. Serial Port Receive Data Register Definitions

Bits	Field Name	Description
15:8	///	**Reserved**
7:0	Data	**Received Data** After all data bits are received, and the data word has been checked for framing and parity errors, the data word is written to the receive data FIFO. The receive data ready bit in the Status register indicates that data is ready and has been transferred to the receive data FIFO. If another word is received after the FIFO is full and before this FIFO is read, the overrun error bit in the Status register is set.

Serial Port Transmit Data Registers

Each Transmit register is written by software with the value to be transmitted over the corresponding serial interface. The transmitter is double-buffered; consequently, before data is transmitted, it is copied from the Transmit register to the Transmit Shift register, which is not accessible to software. The state of the transmit and Transmit Shift registers is reflected in the TEMT and THRE bits for the associated Serial Port Status registers (see page 15).

If hardware handshaking is enabled, the transmitter does not transmit data while RTS/RTR inputs are deasserted. Data is held in the Transmit and Transmit Shift registers without affecting the transmit pin.

Table 10. Serial Port Transmit Data Registers

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET	Serial Port 0 = 84h Serial Port 1 = 14h Serial Port 2 = 04h Serial Port 3 = E4h															
FIELD	///								Data							
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Table 11. Serial Port Transmit Data Register Definitions

Bits	Field Name	Description
15:8	///	**Reserved**
7:0	Data	**Data to be Transmitted** Before the data can be written, the Transmit Holding Register Empty bit in the Serial Port Status register must be HIGH (see page 15); otherwise, the previous word of data is overwritten.

Serial Port Status Registers

The Serial Port Status registers provide information about the current status of the associated serial ports.

- The THRE and TEMT bits provide the software with information about the state of the transmitter.
- The BRK1, BRK0, RDR, FER, OER, and PER bits provide the software with information about the receiver.
- The HS0 bit reflects the value of the serial port's associated CTS signal.

The THRE, TEMT, and HS0 bits are updated during each microprocessor cycle.

The receive logic carries the framing error (FER), parity error (PER) and received character through the 4-deep FIFO. All the other bits set immediately.

Table 12. Serial Port Status Registers

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET	Serial Port 0 = 82h Serial Port 1 = 12h Serial Port 2 = 02h Serial Port 3 = E2h															
FIELD	///					BRK1	BRK0	RB8	RDR	THRE	FER	OER	PER	TEMT	HS0	// /
RESET	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
RW	R	R	R	R	R	RW	RW	RW	R	R	R W	R W	R W	R	R	R

Table 13. Serial Port Status Register Definitions

Bits	Field Name	Description
15:11	///	**Reserved**
10	BRK1	**Long Break Detected** This bit sets when a long break has been detected. A long break is defined as a '0' signal on the RXD pin for more than 2M+3 bit times. M = StartBit + #DataBits = ParityBit + StopBits To guarantee the reception of a break character it must not start in the middle of a frame, it must start outside of a frame. This bit must be cleared by software.
9	BRK0	**Short Break Detected** This bit sets when a short break has been detected. A short break is defined as a '0' signal on the RXD pin for more than M bit times. In order to guarantee the reception of a break character it must not start in the middle of a frame, it must start outside of a frame. This bit must be cleared by software.
8	RB8	**Receive Bit 8** This is the ninth data bit received in modes 2 and 3. This bit must be cleared by software.
7	RDR	**Receive Data Ready** This Read Only bit sets when a valid word of data is transferred to the receive data FIFO. When the receive FIFO becomes empty, this bit is automatically reset. The serial port can be configured to generate interrupts whenever this bit is set by the RXIE bit in the Serial Port Control register (see page 17).
6	THRE	**Transmit Holding Register Empty** This Read Only bit sets when a new word of data can be written to the Transmit Holding register. The bit is automatically reset when the Transmit Holding register is written. The serial port can be configured to generate interrupts when THRE is set by setting the TXIE bit in the Serial Port Control register to 1 (see page 17) and enabling the transmitter.
5	FER	**Framing Error** This bit indicates that a framing error has occurred. If receive status interrupts are enabled (RSIE bit in Serial Port Control register, see page 17) an interrupt is generated when this bit is set. The FER bit must be reset by the software.
4	OER	**Overrun Error** This bit is set if a word in the Receive Holding register was overwritten. If receive status interrupts are enabled (RSIE bit in Control register) an interrupt is generated when this bit is set. The OER bit must be reset by the software.
3	PER	**Parity Error** This bit indicates that a parity error has occurred. If receive status interrupts are enabled (RSIE bit in Control register) an interrupt is generated when this bit is set. The PER bit must be reset by the software.
2	TEMT	**Transmit Empty** This Read Only bit indicates that both the Transmit Holding register and Transmit Shift register are empty. When this bit is HIGH, it is safe to disable the transmitter.
1	HS0	**Handshake Signal 0** This Read Only bit reflects the value of the CTS pin (inverted). This bit is masked by the CTSM bit in the System Auxiliary Control (AUXCON) register (see page 10).
0	///	**Reserved** This Read Only bit is always 0.

Serial Port Control Registers

Table 14. Serial Port Control Registers

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET	Serial Port 0 = 80h Serial Port 1 = 10h Serial Port 2 = 00h Serial Port 3 = E0h															
FIELD	DMA			RSIE	BRK	TB8	FC	TXIE	RXIE	TMOD	RMOD	EVN	PE	MODE		
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 15. Serial Port Control Register Definitions

Bits	Field Name	Description
15:13	DMA	**Direct Memory Access** These bits configure the serial port for use with DMA according to Table 16 on page 18. DMA transfers to a serial port are destination synchronized. When the Transmit Holding register is empty, a new transmit is requested. When configured for transmit DMA, transmit interrupts are disabled regardless of the setting of the of TXIE bit. DMA transfers from the serial port (receive DMA) are source synchronized. A new transfer is requested when data is available in the receive FIFO. When a serial port is configured for receive DMA, receive interrupts are disabled regardless of the setting of the RXIE bit. However, receive status interrupts can still be enabled.
12	RSIE	**Receive Status Interrupt Enable** 1 = serial port generates an interrupt when it detects a break, parity, framing, or overrun error. 0 = no interrupt is generated.
11	BRK	**Send Break Enable** 1 = serial port generates a ‘0’ on the TXD output. 0 = no break is generated.
10	TB8	**Transmit Bit 8** In modes with nine data bits, this bit is transferred as the ninth data bit. This bit is cleared after a frame is transmitted. This bit must be set after the TEMT bit (bit [2] of the Serial Port Status register, page 15) is set.
9	FC	**Flow Control Enable** This bit enables hardware flow control. The type of flow control used is set by the System Auxiliary Control (AUXCON) register (see page 10) and the AUXCON register (see page **Error! Bookmark not defined.**). 1 = flow control enabled. 0 = flow control disabled.
8	TXIE	**Transmit Holding Register Empty Interrupt Enable** 1 = Transmit Holding register generates an interrupt when it goes empty. 0 = Transmit Holding register does not generate interrupt when it goes empty.
7	RXIE	**Receive Data Ready Interrupt Enable** 1 = Receive Data register generates an interrupt when a valid word is received. 0 = Receive Data register does not generate interrupt when a valid word is received.
6	TMOD	**Transmit Mode Enable** 1 = transmitter is enabled. 0 = transmitter will be disabled.
5	RMOD	**Receive Mode Enable** 1 = receiver is enabled. 0 = receiver will be disabled.

Bits	Field Name	Description
4	EVN	**Even Parity** If parity is enabled: 1 = even parity generation and checking. 0 = odd parity generation and checking.
3	PE	**Parity Enable** 1 = enables the serial port to insert a parity bit into each data character and check for the appropriate parity for each received data character. 0 = disables parity generation and checking.
2:0	MODE	**Operating Mode** Determines the operating mode for the serial port. See Table 17.

Table 16. DMA Bit Settings

DMA Bits			Receive	Transmit
Bit [15]	Bit]14]	Bit [13]		
0	0	0	No DMA	No DMA
0	0	1	DMA0 or 2 (See Note)	DMA1 or 3 (See Note)
0	1	0	DMA1 or 3 (See Note)	DMA0 or 2 (See Note)
0	1	1	Reserved	Reserved
1	0	0	DMA0 or 2 (See Note)	No DMA
1	0	1	DMA1 or 3 (See Note)	No DMA
1	1	0	No DMA	DMA0 or 2 (See Note)
1	1	1	No DMA	DMA1 or 3 (See Note)

Note: *Serial channels 0 and 1 use DMA 0 and 1, while serial channels 2 and 3 use DMA 2 and 3.*

Table 17. Serial Port Operating Mode Settings

Mode	Description	Parity Disable		Parity Enabled		Stop Bits
		Data Bits	Parity Bits	Data Bits	Parity Bits	
0	Reserved	///	///	///	///	///
1	Mode 1	8	0	7	1	1
2	Mode 2	9	0	N/A	N/A	1
3	Mode 3	9	0	8	1	1
4	Mode 4	7	0	N/A	N/A	1
5	Mode 5	8	0	7	1	2
6	Reserved	///	///	///	///	///
7	Reserved	///	///	///	///	///

Programmable Input/Output

Overview

This chapter describes the DSTni Programmable Input/Output (PIO) functions. Topics include:

Theory of Operation

PIO Functions

DSTni's PIO can be configured for input, output, or alternate pin sharing (normal operation). On reset, all pins default to PIO input with weak pull-ups. Table 18 lists the possible modes and directions. Weak internal pull-up resistors are roughly 56K to 122K Ohms.

Table 18. Parallel Port Functions

PIO Mode	PIO Direction	PIO Configuration
0	0	Normal output operation
0	1	PIO input with internal pull-up*
1	0	PIO output
1	1	PIO input with internal pull-up
* Reset state		

The 32 PIO signals can individually operate as open drain (collector) outputs by writing a 0 to the designated bit in the Data register and a 1 to the designated bit in the Mode register. Then:

- Writing a 1 in the Direction register disables the output.
- Writing a 0 in the Direction register drives the pin LOW.

PIO Sharing Designations

Error! Reference source not found. lists the PIO sharing designations.

Table 19. PIO Sharing Designations

PIO Number	Shared Function Name	Power-On Reset Function
0	TMRIN1	Input with pull-up
1	TMROUT1	Input with pull-up
2	PCS6#	Input with pull-up
3	PCS5#	Input with pull-up
4	PCS4#	Input with pull-up
5	PCS7#	Input with pull-up
6	ARDY	Input with pull-up
7	DRQ2	Input with pull-up
8	DRQ3	Input with pull-up
9	NMI	Input with pull-up
10	TMROUT0	Input with pull-up
11	TMRIN0	Input with pull-up
12	DRQ0	Input with pull-up
13	I^2CDTA	Input with pull-up
14	CAN1TX	Input with pull-up
15	CAN1RX	Input with pull-up
16	LOCK#/SRDYOUT	Input with pull-up
17	HOLD	Input with pull-up
18	CTS3#/SLVSEL#	Input with pull-up (see Note below)
19	RTS3#/SCK	Input with pull-up (see Note below)
20	RTS2#	Input with pull-up
21	CTS2#	Input with pull-up
22	TXD2	Input with pull-up
23	RXD2	Input with pull-up
24	MCS2#	Input with pull-up
25	MCS3#	Input with pull-up
26	A23	Input with pull-up
27	TXD3/SD0	Input with pull-up (see Note below)
28	RXD3/SDI	Input with pull-up (see Note below)
29	DRQ1	Input with pull-up
30	INT5#	Input with pull-up
31	I^2CCLK	Input with pull-up

Note: *PIO pins 18, 19, 27, and 28 are shared between serial port 2 and SPI, as selected by the SPIEN bit of the DSTni Configuration register.*

PIO Register Summary

Table 20. PIO Register Summary

Hex Offset	Mnemonic	Register Description	Page
7A	PDATA1	Parallel Port 1 Data	21
78	PDIR1	Parallel Port 1 Direction	22
76	PIOMODE1	Parallel Port 1 Mode	23
74	PDATA0	Parallel Port 0 Data	23
72	PDIR0	Parallel Port 0 Direction	24
70	PIOMODE0	Parallel Port 0 Mode	24

PIO Register Definitions

Parallel Port 1 Data Register

Note: *The reset value of this register's bits is indeterminate.*

Table 21. Parallel Port 1 Data Register

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET	7Ah															
FIELD	PIO DATA (31:16)															
RESET	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 22. Parallel Port 1 Data Register Definitions

Bits	Field Name	Description
15:0	PIO DATA (31:16)	**Data** When a PIO is configured as an output the corresponding value in the PIO Data register is driven out to the pin. For PIO pins configured as inputs, the value on the pin is reflected on the corresponding bit in the Data register.

Parallel Port 1 Direction Register

Table 23. Parallel Port 1 Direction Register

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET	78h															
FIELD	PIO DIRECTION (31:16)															
RESET	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
PIO SHARING DESIG-NATIONS	I^2CCLK	INT5#	DRQ1	RXD3\SDI	TXD3/SD0	A23	MCS3#	MCS2#	RXD2	TXD2	CTS2#	RTS2#	RTS3#/SCK	CTS3#/SLVSEL#	HOLD	LOCK#/SRDYOUT

Table 24. Parallel Port 1 Direction Register Definitions

Bits	Field Name	Description
15:0	PIO DIRECTION (31:16)	**Direction** Works with the PIO Mode register to determine the direction of each individual user-programmable IO. 1 = input (*default*). 0 = is output.

Parallel Port 1 Mode Register

Table 25. Parallel Port 1 Mode Register

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET	74h															
FIELD	PIO MODE (31:16)															
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PIO SHARING DESIG-NATIONS	I^2CCLK	INT5#	DRQ1	RXD3\SDI	TXD3/SD0	A23	MCS3#	MCS2#	RXD2	TXD2	CTS2#	RTS2#	RTS3#/SCK	CTS3#/SLVSEL#	HOLD	LOCK#/SRDYOUT

Table 26. Parallel Port 1 Mode Register Definitions

Bits	Field Name	Description
15:0	PIO MODE (31:16)	**Mode** Works with the PIO Direction register to determine the configuration of each individual user-programmable IO. Default mode is 0.

Parallel Port 0 Data Register

Note: *The reset value of this register's bits is indeterminate.*

Table 27. Parallel Port 0 Data Register

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET	74h															
FIELD	PIO DATA [15:0]															
RESET	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 28. Parallel Port 0 Data Register Definitions

Bits	Field Name	Description
15:0	PIO DATA [15:0]	**Data** When a PIO is configured as an output the corresponding value in the PIO Data register is driven out to the pin. For PIO pins configured as inputs, the value on the pin is reflected on the corresponding bit in the Data register.

Parallel Port 0 Direction Register

Table 29. Parallel Port 0 Direction Register

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET	72h															
FIELD	PIO DIRECTION [15:0]															
RESET	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PIO SHARING DESIG-NATIONS	CAN1RX	CAN1TX	I^2CDTA	DRQ0	TMRIN0	TMROUT 0	NMI	DRQ3	DRQ2	ARDY	PCS7#	PCS4#	PCS5#	PCS6#	TMROUT 1	TMRIN1

Table 30. Parallel Port 0 Direction Register Definitions

Bits	Field Name	Description
15:0	PIO DIRECTION [15:0]	**Direction** Works with the PIO Mode register to determine the direction of each individual user-programmable IO. 1 = input (*default*). 0 = is output.

Parallel Port 0 Mode Register

Table 31. Parallel Port 0 Mode Register

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET	70h															
FIELD	PIO MODE [15:0]															
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PIO SHARING DESIG-NATIONS	CAN1RX	CAN1TX	I^2CDTA	DRQ0	TMRIN0	TMROUT0	NMI	DRQ3	DRQ2	ARDY	PCS7#	PCS4#	PCS5#	PCS6#	TMROUT1	TMRIN1

Table 32. Parallel Port 0 Mode Register Definitions

Bits	Field Name	Description
15:0	PIO MODE [15:0]	**Mode** Works with the PIO Direction register to determine the configuration of each individual user-programmable IO. Default mode is 0.

Timers

This chapter describes the DSTni timers. Topics in this chapter include:

Timer Register Summary

Table 33. Timer Register Summary

Hex Offset	Mnemonic	Register Description	Page
66	T2CON	Timer 2 Mode/Control register	29
62	T2CMPA	Timer 2 Maxcount Compare A register	30
60	T2CNT	Timer 2 Count register	30
5E	T1CON	Timer 1 Mode/Control register	31
5C	T1CMPB	Timer 1 Maxcount Compare B register	33
5A	T1CMPA	Timer 1 Maxcount Compare A register	33
58	T1CNT	Timer 1 Count register	34
56	T0CON	Timer 0 Mode/Control register	34
54	T0CMPB	Timer 0 Maxcount Compare B register	36
52	T0CMPA	Timer 0 Maxcount Compare A register	36
50	T0CNT	Timer 0 Count register	37

Timer Register Definitions

Note: *Timer registers are all reset the same after a system reset or watchdog reset.*

Watchdog Timer Control Register

The Watchdog Timer Control register is open for one write following a reset. To open this register for writing, write a keyed sequence of 3333h to this register, followed by CCCCh. The register can then be written with the correct new values. After this first write, the Watchdog Timer Control register is locked and any further writes are ignored until a system reset. If the unlock bit is set during this first write, however, the Watchdog Timer Control register is not locked and can be written to again after writing another keyed sequence.

The count in the watchdog must be set before it counts down to zero and resets the chip or generates a nonmaskable interrupt (NMI). To set the count in the watchdog, issue a keyed sequence of AAAAh, followed by 5555h to the Watchdog Timer Control register.

Table 34. Watchdog Timer Control Register

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET	6Ch															
FIELD	ENA	WRST	///	NMIFLAG	UNLOCK	COUNT										
RESET	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0
RW	RW	RW	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 35. Watchdog Timer Control Register Definitions

Bits	Field Name	Description
15	ENA	**Watchdog Timer Enable** 1 = enables watchdog timer. 0 =disable watchdog timer.
14	WRST	**Watchdog Timer Reset** This bit selects whether the watchdog timer generates a system reset or an NMI when the timeout count is reached. 1 = a system reset is generated. 0 = a non-maskable interrupt is generated. Setting this bit to 0 and the NMIFLAG to 1 generates a system reset. This bit is a "1" after reset.
13	///	**Reserved** This Read Only bit always reads 0.
12	NMIFLAG	**WDT Non-maskable Interrupt Flag** This bit is asserted (1) after a watchdog timer NMI event occurs. If this bit is asserted and another watchdog timer timeout occurs, the watchdog timer system reset is generated, even if the WRST bit is not set. This bit is 0 after reset.
11	UNLOCK	**Unlock** This bit is written as 1 initially to keep the Watchdog register unlocked for further writes. When the unlock bit is written to 0, the register can no longer be accessed. This means the unlock cannot be set to 1 until a CPU reset occurs.
10:0	COUNT	**Timeout Count** The timeout count value determines the timeout interval of the watchdog timer. This value corresponds to an exponent. To calculate the timeout duration use the exponent from the formula: Duration = $2^{Exponent}$/Frequency ÷ 1,000,000 Duration is the watchdog timeout in seconds. The exponent comes from **Error! Reference source not found.**. The frequency is the CPU-programmed clock frequency, in MHz. provides a few examples. This byte is 80h after reset. ***Note:*** *The possible delay is an 11-bit exponent.*

Table 36. Watchdog Exponent Settings

Exponent	Count Bits										
	Bit [10]	Bit [9]	Bit [8]	Bit [7]	Bit [6]	Bit [5]	Bit [4]	Bit [3]	Bit [2]	Bit [1]	Bit [0]
N/A	0	N/A	0	N/A	0	N/A	0	N/A	0	N/A	0
10	0	10	0	10	0	10	0	10	0	10	0
20	0	20	0	20	0	20	0	20	0	20	0
21	0	21	0	21	0	21	0	21	0	21	0
22	0	22	0	22	0	22	0	22	0	22	0
23	0	23	0	23	0	23	0	23	0	23	0
24	0	24	0	24	0	24	0	24	0	24	0
25	0	25	0	25	0	25	0	25	0	25	0
26 Default	**0**	**26**	**0**	**26**	**0**	**26**	**0**	**26**	**0**	**26**	**0**
27	0	27	0	27	0	27	0	27	0	27	0
28	0	28	0	28	0	28	0	28	0	28	0
29	1	29	1	29	1	29	1	29	1	29	1

Table 37. Watchdog Timeout Examples

Exponent	1	10	24	48	75	100	115	
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	Seconds
20	1.05	0.10	0.04	0.02	0.01	0.01	0.01	Seconds
21	2.10	0.21	0.09	0.04	0.03	0.02	0.02	Seconds
22	4.19	0.42	0.17	0.09	0.06	0.04	0.03	Seconds
23	8.39	0.84	0.35	0.17	0.11	0.08	0.07	Seconds
24	16.78	1.68	0.70	0.35	0.22	0.17	0.13	Seconds
25	33.55	3.36	1.40	0.70	0.45	0.34	0.26	Seconds
26	67.11	6.71	**2.80 (Default)**	1.40	0.89	0.67	0.53	Seconds
27	134.22	13.42	5.59	2.80	1.79	1.34	1.06	Seconds
28	268.44	26.84	11.18	5.59	3.58	2.68	2.11	Seconds
29	536.87	53.69	22.37	11.18	7.16	5.37	4.23	Seconds

Timer 2 Mode/Control Register

Note: *Timer 2 does not support dual max count mode.*

Table 38. Timer 2 Mode/Control Register

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET	66h															
FIELD	EN	INH	INT	///							MC	///				CONT
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	R	R	R	R	R	R	R	RW	R	R	R	R	RW

Table 39. Timer 2 Mode/Control Register Definitions

Bits	Field Name	Description
15	EN	**Count Enable Bit for Timer** 1 = timer is enabled to increment. 0 = timer is disabled from counting. If CONT=0, the EN bit is automatically cleared upon a maximum count.
14	INH	**Write-protect for EN Bit** 1 during a write to the Control register = EN is modified by the write. 0 during a write = EN is not modified by the write. This bit is not stored in the Control register and is always 0 when read.
13	INT	**Interrupt Enable** The INT bit enables interrupts when a timer reaches its maximum count. The interrupt occurs on every max count reached. If set for dual max count, an interrupt occurs when max count A is reached and again when the second max count is reached. All interrupt requests from the timers are stored in the interrupt controller.
12:6	///	**Reserved**
5	MC	**Max Count** The MC bit is set when a timer reaches a maximum count. If set for dual max count, MC is set when max count A is reached and again when max count B is reached. This bit is always set regardless of the INT bit and must be cleared by software.
4:1	///	**Reserved**
0	CONT	**Continue** The CONT bit programs the timer for continuous operation or to halt upon reaching a max count. 1 = timer is set for continuous operation. 0 = timer stops when it reaches a maximum count. If the timer is set to dual max count mode the timer must count to the value in Max Count Register A and then to the value in Max Count Register B, then resets and halts. To stop the timer, clear the EN bit in the timer control word.

Timer 2 Max Count Compare A Register

Table 40. Max Count Compare A Register

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET	62h															
FIELD	MCCA [15:0]															
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 41. Max Count Compare A Register Definitions

Bits	Field Name	Description
15:0	MCCA [15:0]	**Max Count Compare Value** This register contains the value used to compare for max count before resetting the Count register.

Timer 2 Count Register

Table 42. Timer 2 Count Register

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET	60h															
FIELD	TC [15:0]															
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 43. Timer 2 Count Register Definitions

Bits	Field Name	Description
15:0	TC [15:0]	**Current Timer Value** This register contains the current value of the timer.

Timer 1 Mode/Control Register

Table 44. Timer 1 Mode Control Register

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET	5Eh															
FIELD	EN	INH	INT	RIU	///						MC	RTG	P	EXT	ALT	CONT
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
RW	RW	RW	RW	RW	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW

Table 45. Timer 1 Mode Control Register Definitions

Bits	Field Name	Description
15	EN	**Count Enable Bit for the Timer** 1 = timer is enabled to increment. 0 = timer is disabled from counting. If CONT=0 (bit [0] of the Timer 2 Mode/Control register), the EN bit is automatically cleared upon a maximum count.
14	INH	**Write-protect for the EN Bit** 1 during a write to the Control register = EN is modified by the write. 0 during a write, to the Control register = EN is not modified by the write. This bit is not stored in the Control register and is always 0 when read.
13	INT	**Interrupt Enable** The INT bit enables interrupts when a timer reaches its maximum count. The interrupt occurs on every max count reached. If set for dual max count, an interrupt occurs when max count A is reached and again when the second max count is reached. All interrupt requests from the timers are stored in the interrupt controller.
12	RIU	**Counter Comparison** The RIU bit indicates which Max Count register is being used for the counter comparison. 0 = Max Count Register A is being used. 1 = Max Count Register B is being used. When configured for single mode (ALT=0), this bit is always 0. This bit cannot be written by the CPU.
11:6	///	**Reserved**
5	MC	**Max Count** The max count bit is set when a timer reaches a maximum count. If set for dual max count, MC is set when max count A is reached and again when max count B is reached. This bit is always set, regardless of the INT bit and must be cleared by software.
4	RTG	**Retrigger** The retrigger bit is only valid when configured for internal clocking, EXT=0. EXT=0 = RTG determines the function of the timer input. EXT=0 and RTG=0 = the timer input is used as a count enable to the timer. When the timer input is HIGH, it enables the timer to count. If LOW, the timer is disabled. EXT=0 and RTG=1 = the input detects a LOW-to-HIGH transition. When it detects a transition, it causes the timer to reset to 0000h on the first clock and, if the input remains HIGH, the timer starts counting on the following clock transitions.
3	P	**Prescaler** The prescaler bit is only valid when configured for internal clocking (EXT=0). EXT=0 and P=0 = timers run at ¼ the CPU clock rate. EXT=0 and P=1 = output of Timer2 is used as a clock for the timers. Timer2 must be running to obtain the prescaled clock.

Bits	Field Name	Description
2	EXT	**External or Internal Clock Source** The external bit determines whether the clocking source for the timer will be an external or internal source. 1 = timer is configured for external clocking. The timer counts LOW-to-HIGH transitions on the timer input. 0 = timer is configured for internal clocking and the timer input pin is used for control. The RTG bit determines the control function of the input pin. Timer input to output delays can be as HIGH as four clocks due to internal synchronization. However, when using the external timer inputs, a clock's input transitions can be pipelined as closely as every eight clocks without losing clock pulses.
1	ALT	**Alternate** The alternate bit determines whether a timer operates in single or dual max count mode. 0 = timer is configured for single max count mode and only Max Count Register A is used. 1 = timer is configured for dual max count. In this mode, Max Count registers A and B are both used. Each Max Count register is used in an alternating fashion, and the current register in use is determined by the RIU bit. By using the ALT bit with the RIU bit, software can change one Max Count register while the other is being used. The ALT bit also determines the function of the timer output pin. 0 = timer out goes LOW for 1 clock, the clock after the maximum count has been reached. 1 = timer output pin is the inversion of the RIU bit (0/1 for B/A).
0	CONT	**Continue** The continue bit programs the timer for continuous operation or to halt upon reaching a max count. 1 = timer is set for continuous operation. 0 = timer stops when it reaches the maximum count. If the timer is set to dual max count mode the timer must count to the value in Max Count register A and then to the value in Max Count register B before resetting and halting. The timer is halted by clearing the EN bit in the timer control word.

Timer 1 Max Count Compare B Register

Table 46. Timer 1 Max Count Compare B Register

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET	5Ch															
FIELD	MCCB [15:0]															
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 47. Timer 1 Max Count Compare B Register Definitions

Bits	Field Name	Description
15:0	MCCB [15:0]	**Compare Value for Max Count** This register contains the value used to compare for max count before resetting the Count register.

Timer 1 Max Count Compare A Register

Table 48. Timer 1 Max Count Compare A Register

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET	5Ah															
FIELD	MCCA [15:0]															
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 49. Timer 1 Max Count Compare A Register Definitions

Bits	Field Name	Description
15:0	MCCA [15:0]	**Compare Value for Max Count** This register contains the value used to compare for max count before resetting the Count register.

Timer 1 Count Register

Table 50. Timer 1 Count Register

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET	58h															
FIELD	TC [15:0]															
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 51. Timer 1 Count Register Definitions

Bits	Field Name	Description
15:0	TC [15:0]	**Compare Timer Value** This register contains the current value of the timer.

Timer 0 Mode/Control Register

Table 52. Timer 0 Mode/Control Register

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET	56h															
FIELD	EN	INH	INT	RIU	///						MC	RTG	P	EXT	ALT	CONT
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
RW	RW	RW	RW	RW	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW

Table 53. Timer 0 Mode/Control Register Definitions

Bits	Field Name	Description
15	EN	**Count Enable Bit for the Timer** 1 = timer is enabled to increment. 0 = timer is disabled from counting. If CONT=0 (bit [0] of the Timer 2 Mode/Control register), the EN bit is automatically cleared upon a maximum count.
14	INH	**Write-protect for the EN Bit** 1 during a write to the Control register = EN is modified by the write. 0 during a write, to the Control register = EN is not modified by the write. This bit is not stored in the Control register and is always 0 when read.
13	INT	**Interrupt Enable** 1 = INT bit enables interrupts when a timer reaches its maximum count. The interrupt occurs on every max count reached. If set for dual max count, an interrupt occurs when max count A is reached and again when the second max count is reached. All interrupt requests from the timers are stored in the interrupt controller.

Bits	Field Name	Description
12	RIU	**Counter Comparison** The RIU bit indicates which Max Count register is being used for the counter comparison. 0 = Max Count Register A is being used. 1 = Max Count Register B is being used. When configured for single mode (ALT=0), this bit is always 0. This bit cannot be written by the CPU.
11:6	///	**Reserved**
5	MC	**Max Count** The max count bit is set when a timer reaches a maximum count. If set for dual max count, MC is set when max count A is reached and again when max count B is reached. This bit is always set, regardless of the INT bit and must be cleared by software.
4	RTG	**Retrigger** The retrigger bit is only valid when configured for internal clocking, EXT=0. EXT=0 = RTG determines the function of the timer input. EXT=0 and RTG=0 = the timer input is used as a count enable to the timer. When the timer input is HIGH, it enables the timer to count. If LOW, the timer is disabled. EXT=0 and RTG=1 = the input detects a LOW-to-HIGH transition. When it detects a transition, it causes the timer to reset to 0000h on the first clock and, if the input remains HIGH, the timer starts counting on the following clock transitions.
3	P	**Prescaler** The prescaler bit is only valid when configured for internal clocking (EXT=0). EXT=0 and P=0 = timers run at ¼ the CPU clock rate. EXT=0 and P=1 = output of Timer2 is used as a clock for the timers. Timer2 must be running to obtain the prescaled clock.
2	EXT	**External or Internal Clock Source** The external bit determines whether the clocking source for the timer will be an external or internal source. 1 = timer is configured for external clocking. The timer counts LOW-to-HIGH transitions on the timer input. 0 = timer is configured for internal clocking and the timer input pin is used for control. The RTG bit determines the control function of the input pin. Timer input to output delays can be as HIGH as four clocks due to internal synchronization. However, when using the external timer inputs, a clock's input transitions can be pipelined as closely as every eight clocks without losing clock pulses.
1	ALT	**Alternate** The alternate bit determines whether a timer operates in single or dual max count mode. 0 = timer is configured for single max count mode and only Max Count Register A is used. 1 = timer is configured for dual max count. In this mode, Max Count registers A and B are both used. Each Max Count register is used in an alternating fashion, and the current register in use is determined by the RIU bit. By using the ALT bit with the RIU bit, software can change one Max Count register while the other is being used. The ALT bit also determines the function of the timer output pin. 0 = timer out goes LOW for 1 clock, the clock after the maximum count has been reached. 1 = timer output pin is the inversion of the RIU bit (0/1 for B/A).
0	CONT	**Continue** The continue bit programs the timer for continuous operation or to halt upon reaching a max count. 1 = timer is set for continuous operation. 0 = timer stops when it reaches the maximum count. If the timer is set to dual max count mode the timer must count to the value in Max Count register A and then to the value in Max Count register B before resetting and halting. The timer is halted by clearing the EN bit in the timer control word.

Timer 0 Max Count Compare B Register

Table 54. Timer 0 Max Count Compare B Register

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET	54h															
FIELD	MCCB [15:0]															
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 55. Timer 0 Max Count Compare B Register Definitions

Bits	Field Name	Description
15:0	MCCB [15:0]	**Compare Value for Max Count** This register contains the value used to compare for max count before resetting the Count register.

Timer 0 Max Count Compare A Register

Table 56. Timer 0 Max Count Compare A Register

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET	52h															
FIELD	MCCA [15:0)															
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 57. Timer 0 Max Count Compare A Register Definitions

Bits	Field Name	Description
15:0	MCCA [15:0]	**Compare Value for Max Count** This register contains the value used to compare for max count before resetting the Count register.

Timer 0 Max Count Register

Table 58. Timer 0 Count Register

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET	50h															
FIELD	TC [15:0)															
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 59. Timer 0 Count Register Definitions

Bits	Field Name	Description
15:0	TC [15:0]	**Compare Timer Value** This register contains the current value of the timer.