Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

High Performance 1.62V To 3.6V Octal Uart with 64-Byte FIFO

Features

- 1.62V to 3.6V with 5V Tolerant Serial Inputs
- Single Interrupt output for all 8 UARTs
- A Global Interrupt Source Register for all 8 UARTs
- 5G "Flat" UART Registers for easier programming
- Simultaneous Initialization of all UART channels
- A General Purpose Command-driven 16-bit Timer/counter
- Sleep Mode with Wake-up Indication
- Highly Integrated Device for Space Saving
- Each UART is independently controlled with:
  - 16C550 Compatible 5G Register Set
  - 64-byte Transmit and Receive FIFOs
  - Transmit and Receive FIFO Level Counters
  - Programmable TX and RX FIFO Trigger Level
  - Automatic RTS/CTS or DTR/DSR Flow Control
  - Automatic Xon/Xoff Software Flow Control
  - RS485 HDX Control Output
  - RS485 auto address detection
  - Infrared (IrDA 1.0/1.1) Data Encoder/Decoder
  - Programmable Data Rate with Prescaler
- Up to 8 Mbps Serial Data Rate with 64MHz external clock input
- Crystal oscillator(up to 24MHz) or external clock(up to 80MHz) input
- Built in Power-On-Reset circuit

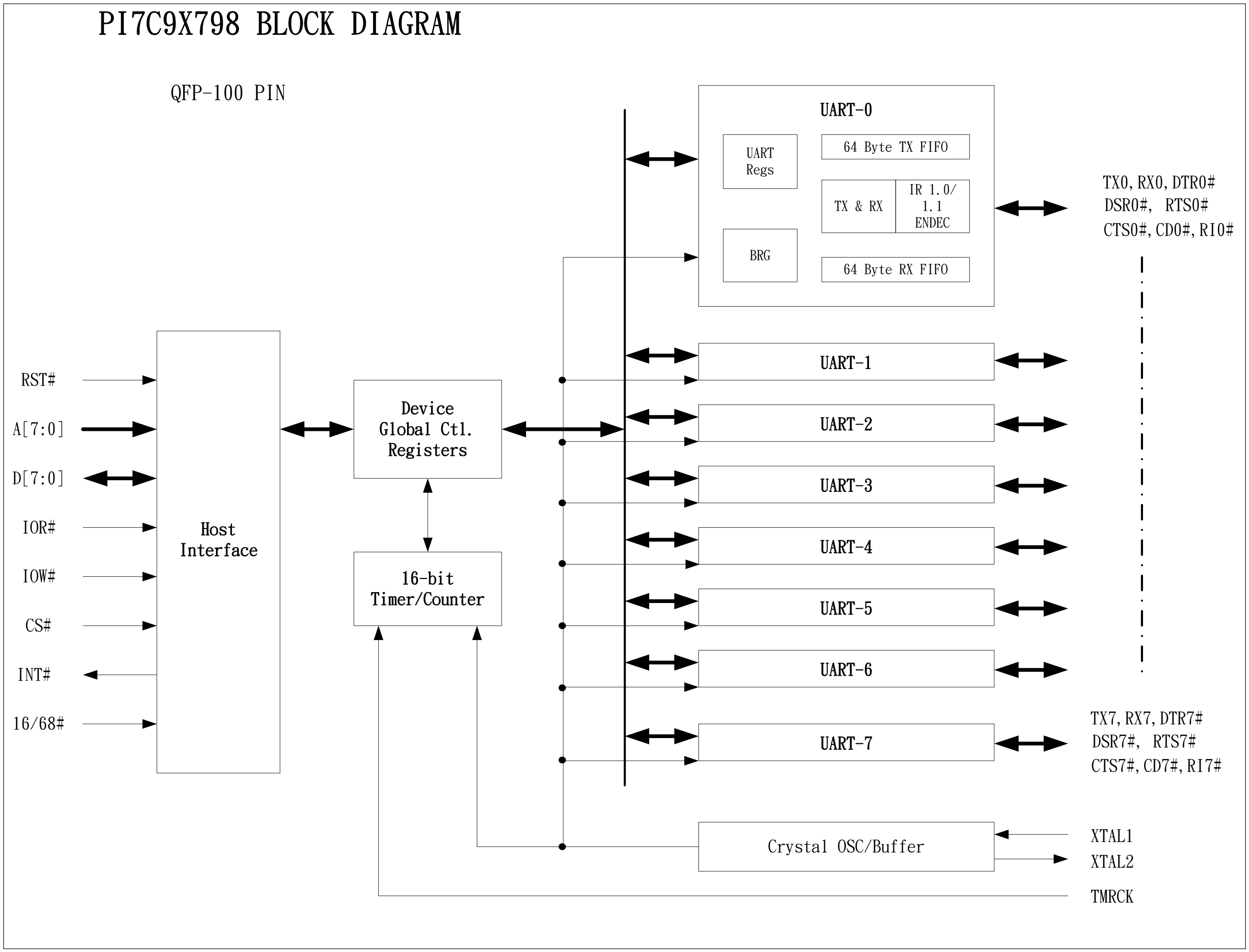
Application

- Remote Access Servers
- Ethernet Network to Serial Ports
- Network Management
- Factory Automation and Process Control
- Point-of-Sale Systems
- Multi-port RS-232/RS-422/RS-485 Cards

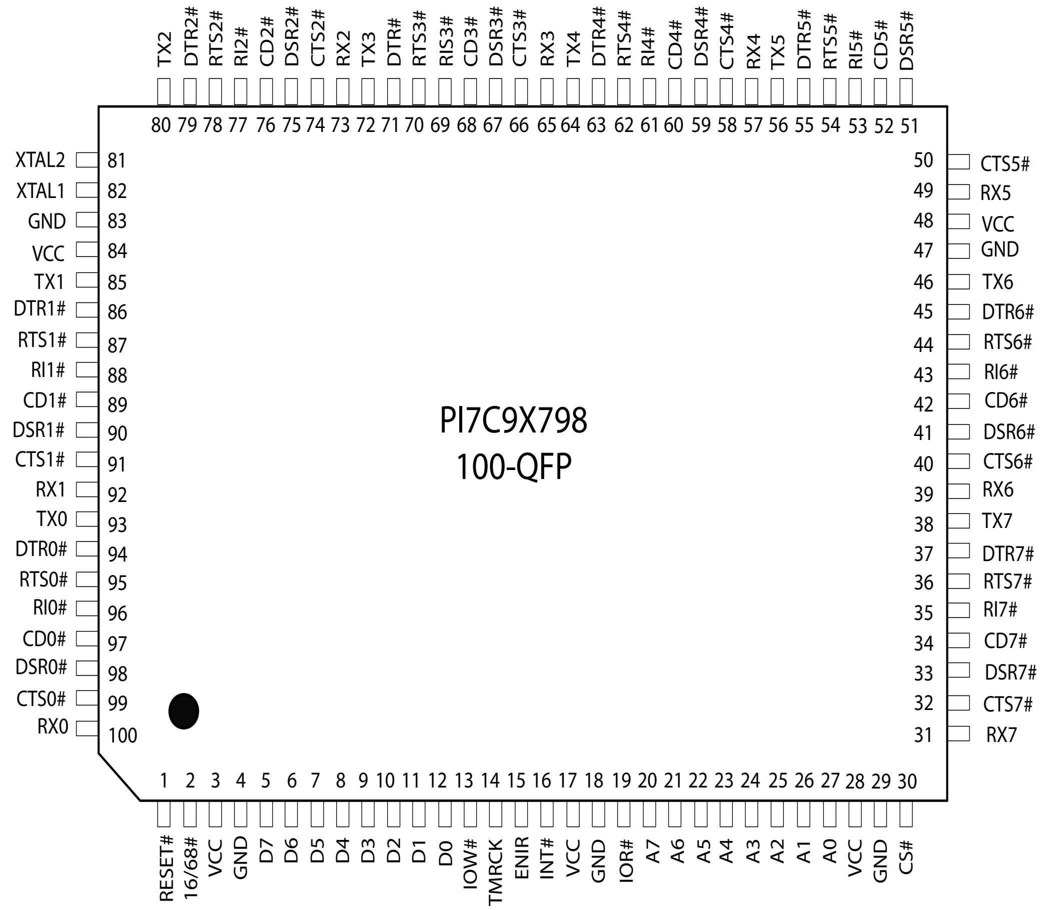
Description

The PI7C9X798 (798), is a 1.62V to 3.6V octal Universal Asynchronous Receiver and Transmitter (UART) with 5V tolerant serial (modem) inputs. The highly integrated device is designed for high bandwidth requirement in communication systems. The global interrupt source register provides a complete interrupt status indication for all 8 channels to speed up interrupt parsing. Each UART has its own 16C550 compatible set of configuration registers, TX and RX FIFOs of 64 bytes, fully programmable transmit and receive FIFO trigger levels, TX and RX FIFO level counters, automatic RTS/CTS or DTR/DSR hardware flow control with programmable hysteresis, automatic software (Xon/Xoff) flow control, RS-485 half-duplex direction control, and auto address detection, Intel or Motorola bus interface and sleep mode with a wake-up indicator.

Block Diagram



Pin Configuration 100-MQFP



Pin Description

Pin Name	100-MQFP Pin#	Type	Description
Data Bus Interface			
A7:A0	20-27	I	Address lines [7:0]. A0:A3 selects individual UART's 16 configuration registers, A4:A6 selects UART channel 0 to7, and A7 selects the global device configuration registers.
D7:D0	5-12	IO	Data bus lines [7:0] (bidirectional).
IOR#	19	I	When 16/68# pin is HIGH, it selects Intel bus interface and this input is read strobe (active LOW). The falling edge instigates an internal read cycle and retrieves the data byte from an internal register pointed by the address lines [A7:A0], places it on the data bus to allow the host processor to read it on the leading edge. When 16/68# pin is LOW, it selects Motorola bus interface and this input should be connected to VCC.
IOW# (R/W#)	13	I	When 16/68# pin is HIGH, it selects Intel bus interface and this input becomes write strobe (active LOW). The falling edge instigates the internal write cycle and the leading edge transfers the data byte on the data bus to an internal register pointed by the address lines. When 16/68# pin is LOW, it selects Motorola bus interface and this input becomes read (HIGH) and write (LOW) signal (R/W#).
CS#	30	I	When 16/68# pin is HIGH, this input is chip select (active LOW) to enable the XR16V798 device. When 16/68# pin is LOW, this input becomes the read and write strobe (active LOW) for the Motorola bus interface.
INT#	16	OD	Global interrupt output from XR16V798 (open drain, active LOW). This output requires an external pull-up resistor (47K-100K ohms) to operate properly. It may be shared with other devices in the system to form a single interrupt line to the host processor and have the software driver polls each device for the interrupt status.
Modem Or Serial I/O Interface			
TX0	93	O	UART channel 0 Transmit Data or infrared transmit data.
RX0	100	I	UART channel 0 Receive Data or infrared receive data. Normal RXD input idles HIGH. The infrared pulse can be inverted internally prior to decoding by setting FCTR bit-4.
RTS0#	95	O	UART channel 0 Request to Send or general purpose output (active LOW). This port may be used for one of two functions: 1) Auto hardware flow control, see EFR bit-6, MCR bits-1 & 2, FCTR bits 0-3 and IER bit-6 2) RS485 half-duplex direction control, see FCTR bit-5, MCR bit-2 and MSR bits 4-7.
CTS0#	99	I	UART channel 0 Clear to Send or general purpose input (active LOW). It can be used for auto hardware flow control, see EFR bit-7, MCR bit-2 and IER bit-7.
DTR0#	94	O	UART channel 0 Data Terminal Ready or general purpose output (active LOW). This port may be used forone of two functions. 1) auto hardware flow control, see EFR bit-6, FCTR bits-0 to 3, MCR bits-0 & 2, and IER bit-6 2) RS485 half-duplex direction control, see FCTR bit-5, MCR bit-2 and MSR bit 4-7.

Pin Name	100-MQFP Pin#	Type	Description
DSR0#	98	I	UART channel 0 Data Set Ready or general purpose input (active LOW). It can be used for auto hardware flow control, see EFR bit-7, MCR bit-2 and IER bit-7.
CD0#	97	I	UART channel 0 Carrier Detect or general purpose input (active LOW).
RI0#	96	I	UART channel 0 Ring Indicator or general purpose input (active LOW).
TX1	85	O	UART channel 1 Transmit Data or infrared transmit data.
RX1	92	I	UART channel 1 Receive Data or infrared receive data. Normal RXD input idles HIGH. The infrared pulse can be inverted internally prior to decoding by setting FCTR bit-4.
RTS1#	87	O	UART channel 1 Request to Send or general purpose output (active LOW). See description of RTS0# pin.
CTS1#	91	I	UART channel 1 Clear to Send or general purpose input (active LOW). See description of CTS0# pin.
DTR1#	86	O	UART channel 1 Data Terminal Ready or general purpose output (active LOW). See description of DTR0# pin.
DSR1#	90	I	UART channel 1 Data Set Ready or general purpose input (active LOW). See description of DSR0# pin.
CD1#	89	I	UART channel 1 Carrier Detect or general purpose input (active LOW).
RI1#	88	I	UART channel 1 Ring Indicator or general purpose input (active LOW).
TX2	80	O	UART channel 2 Transmit Data or infrared transmit data.
RX2	73	I	UART channel 2 Receive Data or infrared receive data. Normal RXD input idles HIGH. The infrared pulse can be inverted internally prior to decoding by setting FCTR bit-4.
RTS2#	78	O	UART channel 2 Request to Send or general purpose output (active LOW). See description of RTS0# pin.
CTS2#	74	I	UART channel 2 Clear to Send or general purpose input (active LOW). See description of CTS0# pin.
DTR2#	79	O	UART channel 2 Data Terminal Ready or general purpose output (active LOW). See description of DTR0# pin.
DSR2#	75	I	UART channel 2 Data Set Ready or general purpose input (active LOWactive LOW). See description of DSR0# pin.
CD2#	76	I	UART channel 2 Carrier Detect or general purpose input (active LOW).
RI2#	77	I	UART channel 2 Ring Indicator or general purpose input (active LOW).
TX3	72	O	UART channel 3 Transmit Data or infrared transmit data.
RX3	65	I	UART channel 3 Receive Data or infrared receive data. Normal RXD input idles HIGH. The infrared pulse can be inverted internally prior to decoding by setting FCTR bit-4.
RTS3#	70	O	UART channel 3 Request to Send or general purpose output (active LOW). See description of RTS0# pin.
CTS3#	66	I	UART channel 3 Clear to Send or general purpose input (active LOW). See description of CTS0# pin.

Pin Name	100-MQFP Pin#	Type	Description
DTR3#	71	O	UART channel 3 Data Terminal Ready or general purpose output (active LOW). See description of DTR0# pin.
DSR3#	67	I	UART channel 3 Data Set Ready or general purpose input (active LOW). See description of DSR0# pin.
CD3#	68	I	UART channel 3 Carrier Detect or general purpose input (active LOW).
RI3#	69	I	UART channel 3 Ring Indicator or general purpose input (active LOW).
TX4	64	O	UART channel 4 Transmit Data or infrared transmit data.
RX4	57	I	UART channel 4 Receive Data or infrared receive data. Normal RXD input idles HIGH. The infrared pulse can be inverted internally prior to decoding by setting FCTR bit-4.
RTS4#	62	O	UART channel 4 Request to Send or general purpose output (active LOW). See description of RTS0# pin.
CTS4#	58	I	UART channel 4 Clear to Send or general purpose input (active LOW). See description of CTS0# pin.
DTR4#	63	O	UART channel 4 Data Terminal Ready or general purpose output (active LOW). See description of DTR0# pin.
DSR4#	59	I	UART channel 4 Data Set Ready or general purpose input (active LOW). See description of DSR0# pin.
CD4#	60	I	UART channel 4 Carrier Detect or general purpose input (active LOW).
RI4#	61	I	UART channel 4 Ring Indicator or general purpose input (active LOW).
TX5	56	O	UART channel 5 Transmit Data or infrared transmit data.
RX5	49	I	UART channel 5 Receive Data or infrared receive data. Normal RXD input idles HIGH. The infrared pulse can be inverted internally prior to decoding by setting FCTR bit-4.
RTS5#	54	O	UART channel 5 Request to Send or general purpose output (active LOW). See description of RTS0# pin.
CTS5#	50	I	UART channel 5 Clear to Send or general purpose input (active LOW). See description of CTS0# pin.
DTR5#	55	O	UART channel 5 Data Terminal Ready or general purpose output (active LOW). See description of DTR0# pin.
DSR5#	51	I	UART channel 5 Data Set Ready or general purpose input (active LOW). See description of DSR0# pin.
CD5#	52	I	UART channel 5 Carrier Detect or general purpose input (active LOW).
RI5#	53	I	UART channel 5 Ring Indicator or general purpose input (active LOW).
TX6	46	O	UART channel 6 Transmit Data or infrared transmit data.
RX6	39	I	UART channel 6 Receive Data or infrared receive data. Normal RXD input idles HIGH. The infrared pulse can be inverted internally prior to decoding by setting FCTR bit-4.
RTS6#	44	O	UART channel 6 Request to Send or general purpose output (active LOW). See description of RTS0# pin.

Pin Name	100-MQFP Pin#	Type	Description
CTS6#	40	I	UART channel 6 Clear to Send or general purpose input (active LOW). See description of CTS0# pin.
DTR6#	45	O	UART channel 6 Data Terminal Ready or general purpose output (active LOW). See description of DTR0# pin.
DSR6#	41	I	UART channel 6 Data Set Ready or general purpose input (active LOW). See description of DSR0# pin.
CD6#	42	I	UART channel 6 Carrier Detect or general purpose input (active LOW).
RI6#	43	I	UART channel 6 Ring Indicator or general purpose input (active LOW).
TX7	38	O	UART channel 7 Transmit Data or infrared transmit data.
RX7	31	I	UART channel 7 Receive Data or infrared receive data. Normal RXD input idles HIGH. The infrared pulse can be inverted internally prior to decoding by setting FCTR bit-4.
RTS7#	36	O	UART channel 7 Request to Send or general purpose output (active LOW). See description of RTS0# pin.
CTS7#	32	I	UART channel 7 Clear to Send or general purpose input (active LOW). See description of CTS0# pin.
DTR7#	37	O	UART channel 7 Data Terminal Ready or general purpose output (active LOW). See description of DTR0# pin.
DSR7#	33	I	UART channel 7 Data Set Ready or general purpose input (active LOW). See description of DSR0# pin.
CD7#	34	I	UART channel 7 Carrier Detect or general purpose input (active LOW).
RI7#	35	I	UART channel 7 Ring Indicator or general purpose input (active LOW).
Ancillary Signals			
XTAL1	82	I	Crystal or external clock input. Caution: this input is not 5V tolerant.
XTAL2	81	O	Crystal or buffered clock output.
TMRCK	14	I	16-bit timer/counter external clock input.
ENIR	15	I	Infrared mode enable (active HIGH). This pin is sampled during power up, following a hardware reset (RESET#) or soft-reset (register RESET). It can be used to start up all 8 UARTs in the infrared mode. The sampled logic state is transferred to MCR bit-6 in the UART.
RESET#	1	I	Reset (active LOW). The configuration and UART registers are reset to default values, see Table 2.
16/68#	2	I	Intel or Motorola data bus interface select. The Intel bus interface is selected when this input is HIGH and the Motorola bus interface is selected when this input is LOW. This input affects the functionality of IOR#, IOW# and CS# pins.
VCC	3,17,28,48,84		1.62V to 3.6V supply with 5V tolerant serial (modem) inputs.
GND	4,18,29,47,83		Power supply common, ground.

Functional Description

The PI7C9X798 integrates the functions of 8 enhanced 16550 UARTs, a general purpose 16-bit timer/counter and an on-chip oscillator. The device configuration registers include a set of four consecutive interrupt source registers that provides interrupt-status for all 8 UARTs, timer/counter and a sleep wake up indicator.

Each UART channel has its own 16550 UART compatible configuration register set for individual channel control, status, and data transfer. Additionally, each UART channel has 64-byte of transmit and receive FIFOs, automatic RTS/CTS or DTR/DSR hardware flow control with hysteresis control, automatic Xon/Xoff and special character software flow control, programmable transmit and receive FIFO trigger levels, FIFO level counters, infrared encoder and decoder (IrDA ver. 1.0 and 1.1), programmable baud rate generator with a prescaler of divide by 1 or 4, and data rate up to 8Mbps with 8X sampling clock or 4Mbps with 16X sampling clock. The PI7C9X798 is a 1.62-3.6V device with 5 volt tolerant inputs (except XTAL1).

1. Trigger levels

The PI7C9X798 provides independent selectable and programmable trigger levels for both Receiver and transmitter interrupt generation. After reset, both transmitter and receiver FIFOs are disabled and so, in effect, the trigger level is the default value of one character. The selectable trigger levels are controlled via FIFO Control Register(FCR) and Feature Control

Register(FCTR). Refer to Table 1

Table 1. Transmit and Receive FIFO Trigger Table and Level Selection

Trigger	FCTR BIT-7	FCTR BIT-6	FCR BIT-7	FCR BIT-6	FCR BIT-5	FCR BIT-4	Receive Trigger Level	Transmit Trigger Level
Table-A	0	0			0	0		1(Default)
			0	0			1(Default)	
			0	1			4	
			1	0			8	
			1	1			14	
Table-B	0	1			0	0		16
					0	1		8
					1	0		24
					1	1		30
			0	0			8	
			0	1			16	
			1	0			24	
			1	1			28	
Table-C	1	0			0	0		8
					0	1		16
					1	0		32
					1	1		56
			0	0			8	
			0	1			16	
			1	0			56	
			1	1			60	
Table-D	1	1	X	X	X	X	Programmable via RX-TRG register	Programmable via TX-TRG register

2. Hardware flow control

Hardware flow control is comprised of Auto-$\overline{\text{CTS}}$ and Auto-RTS (see Figure 1). Auto-$\overline{\text{CTS}}$ and Auto-$\overline{\text{RTS}}$ can be enabled/disabled independently by programming EFR[7:6].

With Auto-$\overline{\text{CTS}}$, $\overline{\text{CTS}}$ must be active before the UART can transmit data.

Auto-$\overline{\text{RTS}}$ only activates the $\overline{\text{RTS}}$ output when there is enough room in the FIFO to receive data and de-activates the $\overline{\text{RTS}}$ output when the RX FIFO is sufficiently full. The halt and resume trigger levels is controlled by FCR and FCTR bits.

If both Auto-$\overline{\text{CTS}}$ and Auto-$\overline{\text{RTS}}$ are enabled, when $\overline{\text{RTS}}$ is connected to $\overline{\text{CTS}}$, data transmission does not occur unless the receiver FIFO has empty space. Thus, overrun errors are eliminated during hardware flow control. If not enabled, overrun errors occur if the transmit data rate exceeds the receive FIFO servicing latency.

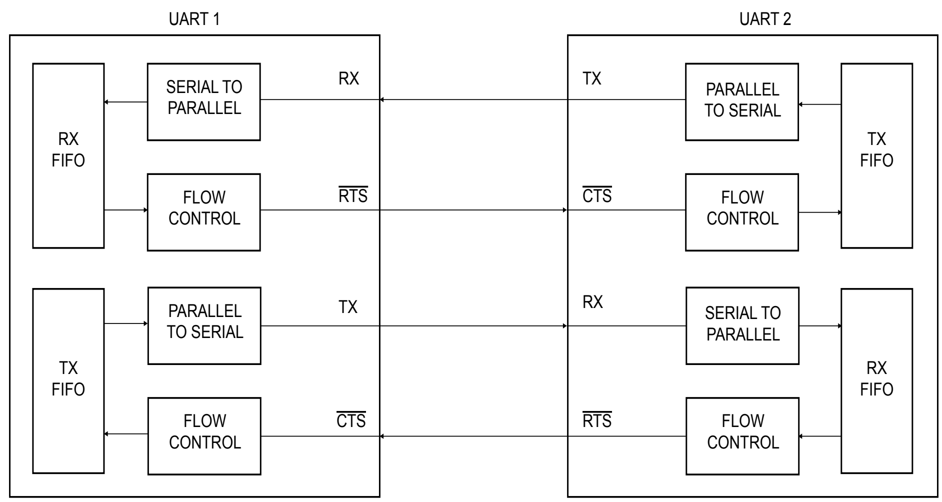


Figure 1. Auto flow control (Auto-$\overline{\text{RTS}}$ and Auto-$\overline{\text{CTS}}$) example

2.1 Auto RTS/DTR Hardware Flow Control Operation

Figure 2 shows RTS#/DTR# functional timing. The RTS#/DTR# output pin is used to request remote unit to suspend/resume data transmission. The flow control features are individually selected to fit specific application requirement:

•Select RTS (and CTS) or DTR (and DSR) through MCR bit-2.

•Enable auto RTS/DTR flow control using EFR bit-6.

•The auto RTS or auto DTR function must be started by asserting the RTS# or DTR# output pin (MCR

bit-1 or bit-0 to a logic 1, respectively) after it is enabled.

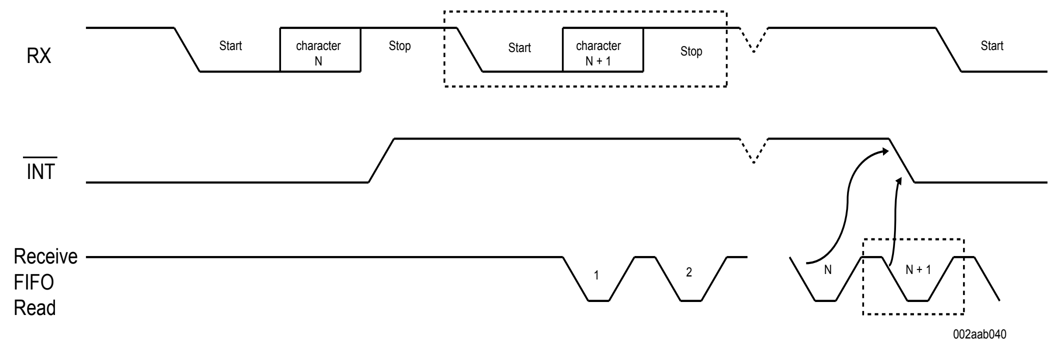
•If using programmable RX FIFO trigger levels, hysteresis levels can be selected via FCTR bits 3-0.

With the Auto RTS function enabled, the RTS# output pin will not be de-asserted (HIGH) when the receive FIFO reaches the programmed trigger level, but will be de-asserted when the FIFO reaches the next trigger level for Trigger Tables A-C (See Table 1). The RTS# output pin will be asserted (LOW) again after the FIFO is unloaded to the next trigger level below the programmed trigger level.

For Trigger Table D (or programmable trigger levels), the RTS# output pin is de-asserted when the the RX FIFO level reaches the RX trigger level plus the hysteresis level and is asserted when the RX FIFO level falls below the RX trigger level minus the hysteresis level.

However, even under these conditions, the 798 will continue to accept data until the receive FIFO is full if the remote UART transmitter continues to send data.

•If used, enable RTS/DTR interrupt through IER bit-6 (after setting EFR bit-4). The UART issues an interrupt when the RTS#/DTR# pin makes a transition: ISR bit-5 will be set to 1.



(1) N = receiver FIFO trigger level.
(2) The two blocks in dashed lines cover the case where an additional character is sent.

Figure 2. RTS functional timing

2.2 Auto CTS/DSR Flow Control

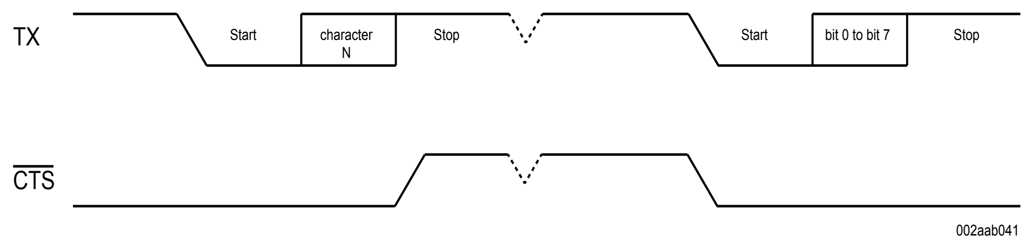
The CTS/DSR pin is monitored to suspend/restart local transmitter. The flow control features are individually selected to fit specific application requirement:

•Select CTS (and RTS) or DSR (and DTR) through MCR bit-2.

•Enable auto CTS/DSR flow control using EFR bit-7.

With the Auto CTS or Auto DTR function enabled, the UART will suspend transmission as soon as the stop bit of the character in the Transmit Shift Register has been shifted out. Transmission is resumed after the CTS#/DTR# input is re-asserted (LOW), indicating more data may be sent.

If used, enable CTS/DSR interrupt through IER bit-7 (after setting EFR bit-4). The UART issues an interrupt when the CTS#/DSR# pin makes a transition: ISR bit-5 will be set to a logic 1, and UART will suspend TX transmissions as soon as the stop bit of the character in process is shifted out. Transmission is resumed after the CTS#/DSR# input returns LOW, indicating more data may be sent.



(1) When $\overline{CTS}$ is LOW, the transmitter keeps sending serial data out.
(2) When $\overline{CTS}$ goes HIGH before the middle of the last stop bit of the current character, the transmitter finishes sending the current character, but it does not send the next character.
(3) When $\overline{CTS}$ goes from HIGH to LOW, the transmitter begins sending data again.

Figure 3. $\overline{CTS}$ functional timing

3 Software flow control

Software flow control is enabled through the Enhanced Features Register and the Modem Control Register. Different combinations of software flow control can be enabled by setting different combinations of EFR[3:0]. Table 1 shows software flow control options.

Table 2. Software flow control options (EFR[3:0])

EFR[3]	EFR[2]	EFR[1]	EFR[0]	TX, RX software flow control
0	0	x	x	no transmit flow control
1	0	x	x	transmit Xon1, Xoff1
0	1	x	x	transmit Xon2, Xoff2
1	1	x	x	transmit Xon1 and Xon2, Xoff1 and Xoff2
x	x	0	0	no receive flow control
x	x	1	0	receiver compares Xon1, Xoff1
x	x	0	1	receiver compares Xon2, Xoff2
1	0	1	1	transmit Xon1, Xoff1 receiver compares Xon1 or Xon2, Xoff1 or Xoff2
0	1	1	1	transmit Xon2, Xoff2 receiver compares Xon1 or Xon2, Xoff1 or Xoff2
1	1	1	1	transmit Xon1 and Xon2, Xoff1 and Xoff2 receiver compares Xon1 and Xon2, Xoff1 and Xoff2

There are two other enhanced features relating to software flow control:

• Xon Any function (MCR[5]): Receiving any character will resume operation after recognizing the Xoff character. It is possible that an Xon1 character is recognized as an Xon Any character, which could cause an Xon2 character to be written to the RX FIFO.

• Special character (EFR[5]): Incoming data is compared to Xoff2. Detection of the special character sets the Xoff interrupt (IIR[4]) but does not halt transmission. The Xoff interrupt is cleared by a read of the Interrupt Identification Register (IIR). The special character is transferred to the RX FIFO.

3.1 Receive flow control

When software flow control operation is enabled, UART will compare incoming data with Xoff1/Xoff2 programmed characters (in certain cases, Xoff1 and Xoff2 must be received sequentially). When the correct Xoff characters are received, transmission is halted after completing transmission of the current character. Xoff detection also sets IIR[4] (if enabled via IER[5]) and causes $\overline{\text{INT}}$to go LOW.

To resume transmission, an Xon1/Xon2 character must be received (in certain cases Xon1 and Xon2 must be received sequentially). When the correct Xon characters are received, IIR[4] is cleared, and the Xoff interrupt disappears.

3.2 Transmit flow control

Xoff1/Xoff2 character is transmitted after the receive FIFO crosses the programmed receiver trigger level (for all trigger tables A-D). Xon1/Xon2 character is transmitted as soon as receive FIFO is less than one trigger level below the programmed receiver trigger level (for Trigger Tables A, B, and C) or when receive FIFO is less than the trigger level minus the hysteresis value (for Trigger Table D). This hysteresis value is the same as the Auto RTS/DTR Hysteresis value in Table 3.

The transmission of Xoff/Xon(s) follows the exact same protocol as transmission of an ordinary character from the FIFO. This means that even if the word length is set to be 5, 6, or 7 bits, then the 5, 6, or 7 least significant bits of Xoff1/Xoff2, Xon1/Xon2 will be transmitted. (Note that the transmission of 5, 6, or 7 bits of a character is seldom done, but this functionality is included to maintain compatibility with earlier designs.)

It is assumed that software flow control and hardware flow control will never be enabled simultaneously. Figure 4 shows an example of software flow control.

Table 3. SELECTABLE HYSTERESIS LEVELS WHEN TRIGGER TABLE-D IS SELECTED

FCTR BIT-3	FCTR BIT-2	FCTR BIT-1	FCTR BIT-0	RTS/DTR Hysteresis (Characters)
0	0	0	0	0
0	0	0	1	+/- 4
0	0	1	0	+/- 6
0	0	1	1	+/- 8
0	1	0	0	+/- 8
0	1	0	1	+/- 16
0	1	1	0	+/- 24
0	1	1	1	+/- 32
1	1	0	0	+/- 12
1	1	0	1	+/- 20
1	1	1	0	+/- 28
1	1	1	1	+/- 36
1	0	0	0	+/- 40
1	0	0	1	+/- 44
1	0	1	0	+/- 48
1	0	1	1	+/- 52

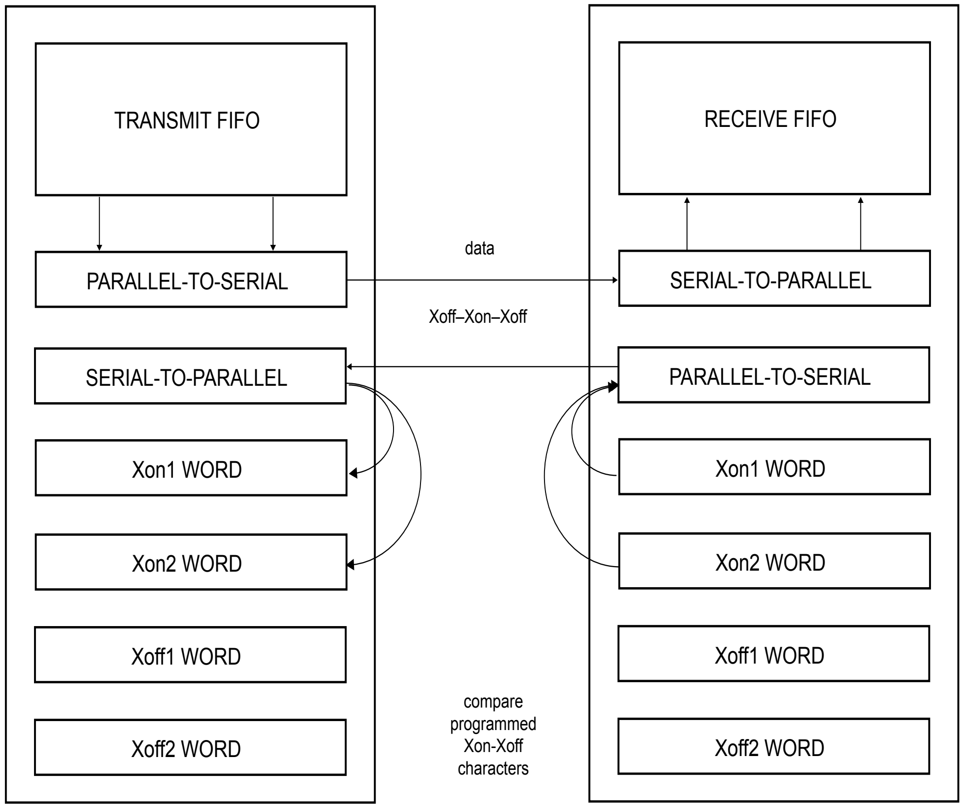


Figure 4. Example of software flow control

4. Hardware Reset, Power-On Reset (POR) and Software Reset

These three reset methods are identical and will reset the internal registers as indicated in Table 4.

Table 2 summarizes the state of register after reset.

Table 4. UART Reset Conditions

Register	Reset state
DLL	Bits 7-0 = 0x01
DLM	Bits 7-0 = 0x00
DLD	Bits 7-0 = 0x00
RHR	Bits 7-0 = 0xXX
THR	Bits 7-0 = 0xXX
IER	Bits 7-0 = 0x00
FCR	Bits 7-0 = 0x00
ISR	Bits 7-0 = 0x01
LCR	Bits 7-0 = 0x00
MCR	Bits 7-0 = 0x00
LSR	Bits 7-0 = 0x60
MSR	Bits 3-0 = logic 0 Bits 7-4 = logic level of the inputs
SPR	Bits 7-0 = 0xFF
FCTR	Bits 7-0 = 0x00
EFR	Bits 7-0 = 0x00
TFCNT	Bits 7-0 = 0x00
TFTRG	Bits 7-0 = 0x00
RFCNT	Bits 7-0 = 0x00
RFTRG	Bits 7-0 = 0x00
XCHAR	Bits 7-0 = 0x00
XON1	Bits 7-0 = 0x00
XON2	Bits 7-0 = 0x00
XOFF1	Bits 7-0 = 0x00
XOFF2	Bits 7-0 = 0x00

Table 5. Output signals after reset

Signal	Reset state
TX	HIGH
$\overline{\text{RTS}}$/$\overline{\text{DTR}}$	HIGH
$\overline{\text{INT}}$	HIGH by external pull-up

5 Interrupts

The UART has interrupt generation and prioritization (seven prioritized levels of interrupts) capability. The interrupt enable registers (IER and IOIntEna) enable each of the seven types of interrupts and the INT signal in response to an interrupt generation. When an interrupt is generated, the IIR indicates that an interrupt is pending and provides the type of interrupt through IIR[5:0]. Table 4 summarizes the interrupt control functions.

Table 6. Interrupt Source and Priority Level

IIR[5:0]	Priority level	Interrupt type	Interrupt source
00 0001	none	none	None
00 0110	1	receiver line status	Overrun Error (OE), Framing Error (FE), Parity Error (PE), or Break Interrupt (BI) errors occur in characters in the RX FIFO
00 1100	3	RX time-out	Stale data in RX FIFO
00 0100	2	RHR interrupt	Receive data ready (FIFO disable) or RX FIFO above trigger level (FIFO enable)
00 0010	4	THR interrupt	Transmit FIFO empty (FIFO disable) or TX FIFO passes above trigger level (FIFO enable)
00 0000	5	modem status	Change of state of modem input pins
01 0000	6	Xoff interrupt	Receive Xoff character(s)/special character
10 0000	7	$\overline{\text{CTS}}$, $\overline{\text{RTS}}$	$\overline{\text{RTS}}$ pin or $\overline{\text{CTS}}$ pin change state from active (LOW) to inactive (HIGH)

It is important to note that for the framing error, parity error, and break conditions, Line Status Register bit 7 (LSR[7]) generates the interrupt. LSR[7] is set when there is an error anywhere in the RX FIFO, and is cleared only when there are no more errors remaining in the FIFO. LSR[4:2] always represent the error status for the received character at the top of the RX FIFO. Reading the RX FIFO updates LSR[4:2] to the appropriate status for the new character at the top of the FIFO. If the RX FIFO is empty, then LSR[4:2] are all zeros.

For the Xoff interrupt, if an Xoff flow character detection caused the interrupt, the interrupt is cleared by an Xon flow character detection. If a special character detection caused the interrupt, the interrupt is cleared by a read of the IIR.

5.1 Interrupts Generation

- LSR is by any of the LSR bits 1, 2, 3, 4 and 7.
- RXRDY is by RX trigger level.
- RXRDY Time-out is by a 4-char delay timer.
- TXRDY is by TX trigger level or TX FIFO empty (or transmitter empty in auto RS-485 control).
- MSR is by any of the MSR bits 0, 1, 2 and 3.
- Receive Xoff/Special character is by detection of a Xoff or Special character.
- CTS# is when its transmitter toggles the input pin (from LOW to HIGH) during auto CTS flow control.
- RTS# is when its receiver toggles the output pin (from LOW to HIGH) during auto RTS flow control.

5.2 Interrupts Clearing

- LSR interrupt is cleared by reading all characters with errors out of the RX FIFO if it is Frame/Parity/Break Error, and is cleared by reading LSR if it is Overrun Error.
- RXRDY interrupt is cleared by reading data until FIFO falls below the trigger level.
- RXRDY Time-out interrupt is cleared by reading RHR.
- TXRDY interrupt is cleared by a read to the ISR register or writing to THR.
- MSR interrupt is cleared by a read to the MSR register.
- Xoff interrupt is cleared when Xon character(s) is received or reading ISR.
- Special character interrupt is cleared by a read to ISR or after next character is received
- RTS# and CTS# flow control interrupts are cleared by a read to the MSR register

5.3 Interrupt mode operation

In Interrupt mode (if any bit of IER[3:0] is 1) the host is informed of the status of the receiver and transmitter by an interrupt signal, $\overline{\text{INT}}$. Therefore, it is not necessary to continuously poll the Line Status Register (LSR) to see if any interrupt needs to be serviced. Figure 5 shows Interrupt mode operation.

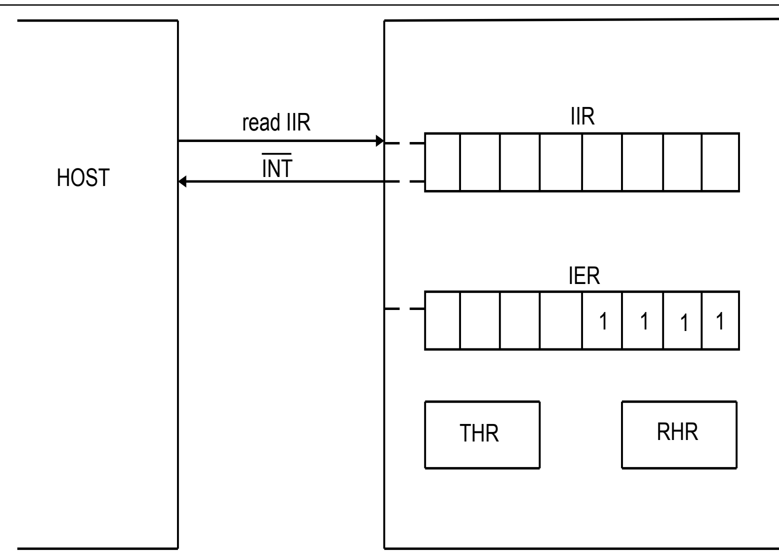


Figure 5. Interrupt mode operation

5.4 Polled mode operation

In Polled mode (IER[3:0] = 0000) the status of the receiver and transmitter can be checked by polling the Line Status Register (LSR). This mode is an alternative to the FIFO Interrupt mode of operation where the status of the receiver and transmitter is automatically known by means of interrupts sent to the CPU. Figure 6 shows FIFO Polled mode operation.

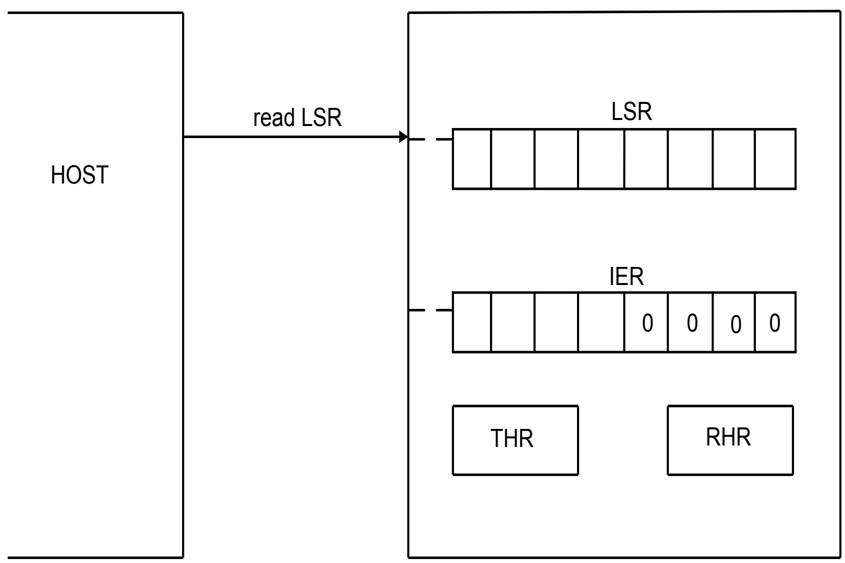


Figure 6. FIFO Polled mode operation

6 Sleep mode

Sleep mode is an enhanced feature of the UART. It is enabled when EFR[4], the enhanced functions bit, is set and when IER[4] is set. Sleep mode is entered when:

- The serial data input line, RX, is idle (see Section 7 "Break and time-out conditions").
- The TX FIFO and TX shift register are empty.
- There are no interrupts pending except THR.
- Sleep register = 0 xFF
- Modem inputs are not toggling

Remark: Sleep mode will not be entered if there is data in the RX FIFO.

In Sleep mode, the clock to the UART is stopped. Since most registers are clocked using these clocks, the power consumption is greatly reduced. The UART will wake up when any change is detected on the RX line, when there is any change in the state of the

modem input pins, or if data is written to the TX FIFO.

Remark: Writing to the divisor latches DLL and DLH to set the baud clock must not be done during Sleep mode. Therefore, it is advisable to disable Sleep mode using IER[4] before writing to DLL or DLH.

7 Break and time-out conditions

When the UART receives a number of characters and these data are not enough to set off the receive interrupt (because they do not reach the receive trigger level), the UART will generate a time-out interrupt instead, 4 character times after the last character is received. The time-out counter will be reset at the center of each stop bit received or each time the receive FIFO is read.

A break condition is detected when the RX pin is pulled LOW for a duration longer than the time it takes to send a complete character plus start, stop and parity bits. A break condition can be sent by setting LCR[6], when this happens the TX pin will be pulled LOW until LSR[6] is cleared by the software.

8 Programmable baud rate generator

The UART contains a programmable baud rate generator that takes any clock input and divides it by a divisor in the range between 1 and (2^{16} - 1). An additional divide-by-4 prescaler is also available and can be selected by MCR[7], as shown in Figure 7. The formula for the baud rate is:

$$\text{Baud rate} = \frac{\left(\frac{\text{XTAL1 crystal input frequency}}{\text{prescaler}}\right)}{\text{divisor x sample rate}}$$

where:

prescaler = 1, when MCR[7] is set to logic 0 after reset (divide-by-1 clock selected)

prescaler = 4, when MCR[7] is set to logic 1 after reset (divide-by-4 clock selected).

Divisor = {DLH, DLL}

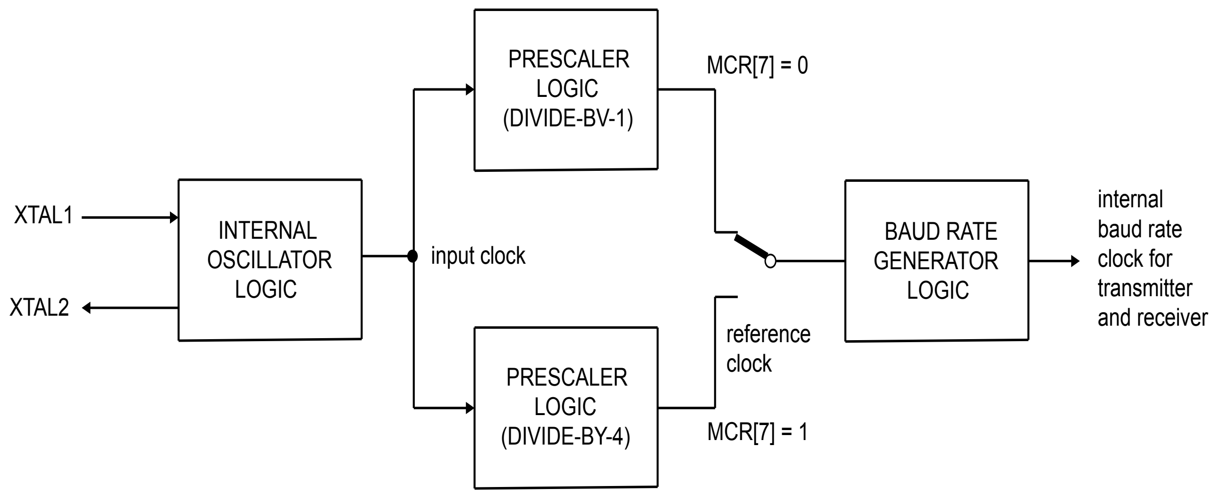


Figure 7. Prescaler and baud rate generator block diagram

Sample rate = 8 if MODE_8X = 1, ot = 16 - SCR + CPR if MODE _8X = 0

Remark: The default value of prescaler after reset is divide-by-1.

DLL and DLH must be written to in order to program the baud rate. DLL and DLH are the least significant and most significant byte of the baud rate divisor. If DLL and DLH are both zero, the UART is effectively disabled, as no baud clock will be generated.

Remark: The programmable baud rate generator is provided to select both the transmit and receive clock rates.

Table 5 to 8 show the baud rate and divisor correlation for crystal with frequency 1.8432 MHz, 3.072 MHz, 14.74926 MHz, and 24MHz respectively.

Table 7. Baud rates using a 1.8432 MHz crystal

Desired baud rate (bit/s)	Divisor used to generate 16x clock	Sample rate	Percent error difference between desired and actual
50	2304	16	0
75	1536	16	0
110	1047	16	0.026
134.5	857	16	0.058
150	768	16	0
300	384	16	0
600	192	16	0
1200	96	16	0
1800	64	16	0
2000	46	20	0.617
2400	48	16	0
3600	32	16	0
4800	24	16	0
7200	16	16	0
9600	12	16	0
19200	6	16	0
38400	3	16	0
56000	2	16	2.86

Table 8. Baud rates using a 3.072 MHz crystal

Desired baud rate (bit/s)	Divisor used to generate 16x clock	Sample rate	Percent error difference between desired and actual
50	2304	16	0
75	2560	16	0
110	1745	16	0.026
134.5	1428	16	0.034
150	1280	16	0
300	640	16	0
600	320	16	0
1200	160	16	0
1800	90	19	0.195
2000	96	16	0
2400	80	16	0
3600	45	19	0.195
4800	40	16	0
7200	25	17	0.392
9600	20	16	0
19200	10	16	0
38400	5	16	0

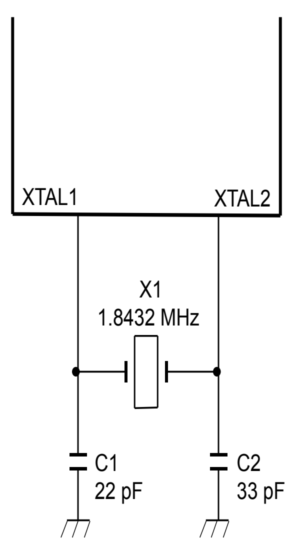


Figure 8. Crystal oscillator circuit reference

Table 9. Baud rates using a 14.74926 MHz crystal

Desired baud rate (bit/s)	Divisor used to generate 16x clock	Sample rate	Percent error difference between desired and actual
38400	24	16	0.025
56000	11	24	0.235
57600	16	16	0.025
115200	8	16	0.025
153600	6	16	0.025
921600	1	16	0.025

Table 10. Baud rates using a 24 MHz crystal

Desired baud rate (bit/s)	Divisor used to generate 16x clock	Sample rate	Percent error difference between desired and actual
4800	250	20	0
7200	159	21	0.17
25000	48	20	0
38400	25	25	0
57600	22	19	0.32
115200	8	26	0.16
225000	6	18	1.2
400000	3	20	0
921600	1	26	0.16
1000000	1	24	0

9. RS-485 features

9.1 Auto RS-485 $\overline{\text{RTS}}$ control

Normally the $\overline{\text{RTS}}$ pin is controlled by MCR bit 1, or if hardware flow control is enabled, the logic state of the $\overline{\text{RTS}}$ pin is controlled by the hardware flow control circuitry. FCTR register bit 5 will take the precedence over the other two modes; once this bit is set, the transmitter will control the state of the $\overline{\text{RTS}}$ pin. The transmitter automatically de-asserts the $\overline{\text{RTS}}$ pin (logic 1) once the host writes data to the transmit FIFO, and asserts $\overline{\text{RTS}}$ pin (logic 0) once the last bit of the data has been transmitted.

To use the auto RS-485 $\overline{\text{RTS}}$ mode the software would have to disable the hardware flow control function.

9.2 RS-485 $\overline{\text{RTS}}$ output inversion

RS485 register bit 5 reverses the polarity of the $\overline{\text{RTS}}$ pin if the UART is in auto RS-485 $\overline{\text{RTS}}$ mode. If RS485 bit 5 is and when the transmitter has data to be sent it asserts the $\overline{\text{RTS}}$ pin (logic 0), and when the last bit of the data has been sent out the transmitter de-asserts the $\overline{\text{RTS}}$ pin (logic 1).

9.3 Auto RS-485

RS458 register bit 0 is used to enable the RS-485 mode (multidrop or 9-bit mode). In this mode of operation, a 'master' station transmits an address character followed by data characters for the addressed 'slave' stations. The slave stations examine the received data and interrupt the controller if the received character is an address character (parity bit = 1).

To use the auto RS-485 RTS mode the software would have to disable the hardware flow control function.

9.3.1 Normal multidrop mode

The 9-bit mode in RS458 register bit 0 is enabled, but not Special Character Detect (EFR bit 5). The receiver is set to Force Parity 0 (LCR[5:3] = 111) in order to detect address bytes.

With the receiver initially disabled, it ignores all the data bytes (parity bit = 0) until an address byte is received (parity bit = 1). This address byte will cause the UART to set the parity error. The UART will generate a line status interrupt (IER bit 2 must be set to '1' at this time), and at the same time puts this address byte in the RX FIFO. After the controller examines the byte it must make a decision whether or not to enable the receiver; it should enable the receiver if the address byte addresses its ID address, and must not enable the receiver if the address byte does not address its ID address.

If the controller enables the receiver, the receiver will receive the subsequent data until being disabled by the controller after the controller has received a complete message from the 'master' station. If the controller does not disable the receiver after receiving a message from the 'master' station, the receiver will generate a parity error upon receiving another address byte. The controller then determines if the address byte addresses its ID address, if it is not, the controller then can disable the receiver. If the address byte addresses the 'slave' ID address, the controller take no further action; the receiver will receive the subsequent data.

9.3.2 Auto address detection

If Special Character Detect is enabled (EFR[5] is set and XOFF2 contains the address byte) the receiver will try to detect an address byte that matches the programmed character in XOFF2. If the received byte is a data byte or an address byte that does not match the programmed character in XOFF2, the receiver will discard these data. Upon receiving an address byte that matches the XOFF2 character, the receiver will be automatically enabled if not already enabled, and the address character is pushed into the RX FIFO along with the parity bit (in place of the parity error bit). The receiver also generates a line status interrupt (IER bit 2 must be set to 1 at this time). The receiver will then receive the subsequent data from the 'master' station until being disabled by the controller after having received a message from the 'master' station.

If another address byte is received and this address byte does not match XOFF2 character, the receiver will be automatically disabled and the address byte is ignored. If the address byte matches XOFF2 character, the receiver will put this byte in the RX FIFO along with the parity bit in the parity error bit (LSR[2]).

10. Host interface

The host interface is 8 data bits wide with 8 address lines and control signals to execute data bus read and write transactions. The PI7C9X798 data interface supports the Intel compatible types of CPUs and it is compatible to the industry standard 16C550 UART. No clock (oscillator nor external clock) is required for a data bus transaction. Each bus cycle is asynchronous using CS# IOR# and IOW# or CS#, R/W#, All eight UART channels share the same data bus for host operations. Please refer to pin description and host interface read/write timing(Figure 10 and Figure 11).

10.1 UART Channel Selection

A LOW on the chip select pin, CS#, allows the user to select one of the UART channels to configure, send transmit data and/or unload receive data to/from the UART. When address line A7 = 0, address lines A6:A4 are used to select one of the eight channels, while A3:A0 are used to select one of the register in one channel. See Table 11 below for UART channel selection.

Table 11. UART Channel Selection

A7	A6	A5	A4	Function
0	0	0	0	Channel 0 Selected
0	0	0	1	Channel 1 Selected
0	0	1	0	Channel 2 Selected
0	0	1	1	Channel 3 Selected
0	1	0	0	Channel 4 Selected
0	1	0	1	Channel 5 Selected
0	1	1	0	Channel 6 Selected
0	1	1	1	Channel 7 Selected

When A7=1, Uart device configuration registers are selected.These registers provide global controls and status of all 8 channel UARTs that include interrupt status, 16-bit general purpose timer control and status, 8X or 16X sampling clock, sleep mode control, soft-reset control, simultaneous UART initialization, and device identification and revision.

10.2 Simultaneous Write to All Channels

During a write cycle, the setting of the Device Configuration register REGB bit-0 to a logic 1 will override the channel selection of address A6:A4 and allow a simultaneous write to all 8 UART channels when any channel is written to. This functional capability allow the registers in all 8 UART channels to be modified concurrently, saving individual channel initialization time. Caution should be considered, however, when using this capability. Any in-process serial data transfer may be disrupted by changing an active channel's mode. Also, REGB bit-0 should be reset to a logic 0 before attempting to read from the UART

11. Infrared Mode

The UART includes the infrared encoder and decoder compatible to the IrDA (Infrared Data Association) version 1.0 and 1.1. The IrDA 1.0 standard that stipulates the infrared encoder sends out a 3/16 of a bit wide HIGH-pulse for each “0” bit in the transmit data stream with a data rate up to 115.2 Kbps. For the IrDA 1.1 standard, the infrared encoder sends out a 1/4 of a bit time wide HIGH-pulse for each "0" bit in the transmit data stream with a data rate up to 1.152 Mbps. This signal encoding reduces the on-time of the infrared LED, hence reduces the power consumption. See Figure 9 below.

The infrared encoder and decoder are enabled by setting MCR register bit-6 to a ‘1’. With this bit enabled, the infrared encoder and decoder is compatible to the IrDA 1.0 standard. For the infrared encoder and decoder to be compatible to the IrDA 1.1 standard, ASR bit-4 will also need to be set to a ’1’. When the infrared feature is enabled, the transmit data output, TX, idles LOW. Likewise, the RX input also idles LOW, see Figure 9.

The wireless infrared decoder receives the input pulse from the infrared sensing diode on the RX pin. Each time it senses a light pulse, it returns a logic 1 to the data bit stream.

The UART can be in the infrared mode upon power-up following a hardware reset (RESET#) or soft-reset if the ENIR pin is HIGH. After power-up, the infrared mode can be controlled via MCR bit-6.

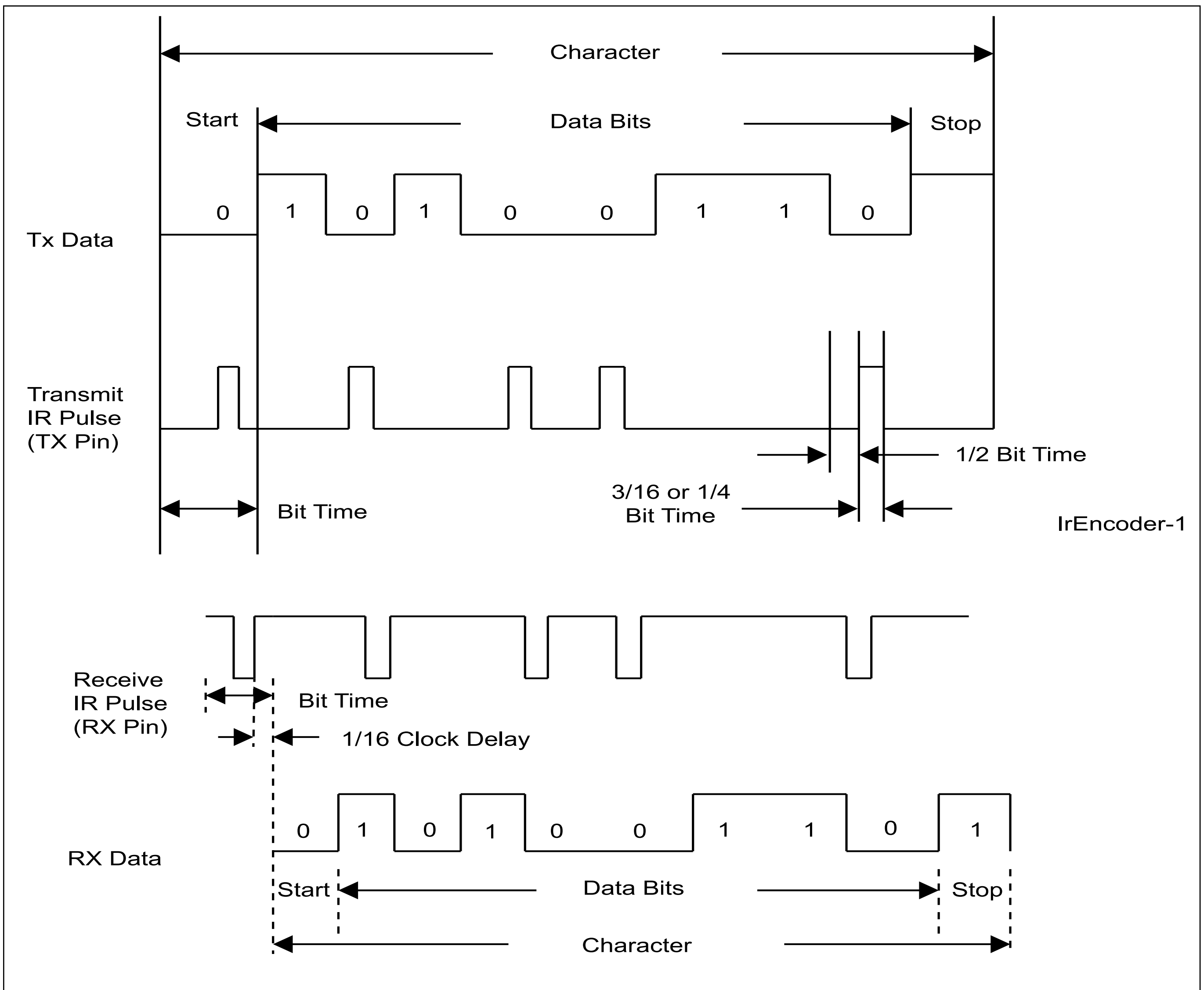


Figure 9. Infrared transmit data receive data deconding