1 UART Interface

This chapter describes the universal asynchronous receiver/transmitter (UART) serial ports included in the JZ4740 Processor. The JZ4740 processor has four UARTs: All UARTs use the same programming model. Each of the serial ports can operate in interrupt based mode or DMA-based mode.

The Universal asynchronous receiver/transmitter (UART) is compatible with the 16550 industry standard and can be used as slow infrared asynchronous interface that conforms to the Infrared Data Association (IrDA) serial infrared specification 1.1.

1.1 Overview

● Full-duplex operation
● 5-, 6-, 7- or 8-bit characters with optional no parity or even or odd parity and with 1, 1½, or 2 stop bits
● 16x8bit transmit FIFO and 16x11bit receive FIFO
● Independently controlled transmit, receive (data ready or timeout), line status interrupts
● Baud rate generation allows up to 230.4Kbps
● Internal diagnostic capability Loopback control and break, parity, overrun and framing-error is provided
● Separate DMA requests for transmit and receive data services in FIFO mode
● Modem Control Functions are provided
● Slow infrared asynchronous interface that conforms to IrDA specification

Jz4740 peripheral specification, Revision 1.0

1.2 Pin Description

Table 1-1 UART Pins Description

Name	Type	Description
RxD	Input	Receive data input
TxD	Output	Transmit data output
CTS_	Input	Clear to Send — Modem Transmission enabled
RTS_	Output	Request to Send — UART Transmission request

Note: Jz4730 has four UART, UART3 support RxD, TxD, RTS_, CTS_, UART2, UART1 and UART0 support only RxD, TxD.

1.3 Register Description

All UART register 32-bit access address is physical address. When ULCR.DLAB is 0, URBR, UTHR and UIER can be accessed; When ULCR.DLAB is 1, UDLLR and UDLHR can be accessed.
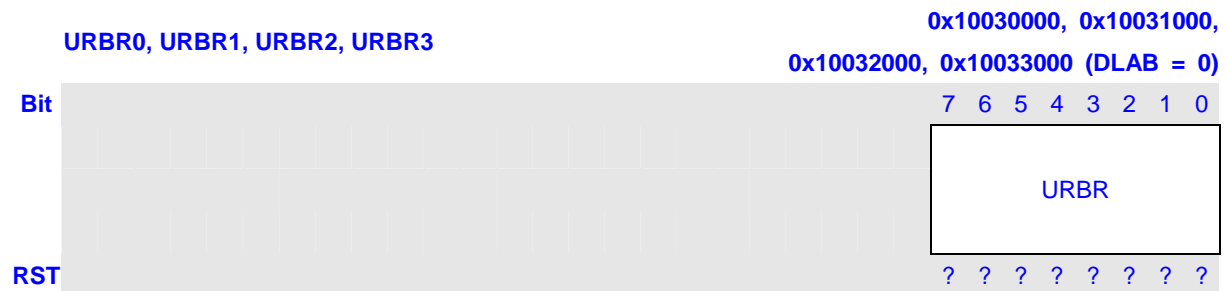
Table 1-2 UART Registers Description

Name	Description	RW	Reset Value	Address	Access Size
URBR0	UART Receive Buffer Register 0	R	0x??	0x10030000	8
UTHR0	UART Transmit Hold Register 0	W	0x??	0x10030000	8
UDLLR0	UART Divisor Latch Low Register 0	RW	0x00	0x10030000	8
UDLHR0	UART Divisor Latch High Register 0	RW	0x00	0x10030004	8
UIER0	UART Interrupt Enable Register 0	RW	0x00	0x10030004	8
UIIR0	UART Interrupt Identification Register 0	R	0x01	0x10030008	8
UFCR0	UART FIFO Control Register 0	W	0x00	0x10030008	8
ULCR0	UART Line Control Register 0	RW	0x00	0x1003000C	8
UMCR0	UART Modem Control Register 0	RW	0x00	0x10030010	8
ULSR0	UART Line Status Register 0	R	0x00	0x10030014	8
UMSR0	UART Modem Status Register 0	R	0x00	0x10030018	8
USPR0	UART ScratchPad Register 0	RW	0x00	0x1003001C	8
ISR0	Infrared Selection Register 0	RW	0x00	0x10030020	8
URBR1	UART Receive Buffer Register 1	R	0x??	0x10031000	8
UTHR1	UART Transmit Hold Register 1	W	0x??	0x10031000	8
UDLLR1	UART Divisor Latch Low Register 1	RW	0x00	0x10031000	8
UDLHR1	UART Divisor Latch High Register 1	RW	0x00	0x10031004	8
UIER1	UART Interrupt Enable Register 1	RW	0x00	0x10031004	8
UIIR1	UART Interrupt Identification Register 1	R	0x01	0x10031008	8
UFCR1	UART FIFO Control Register 1	W	0x00	0x10031008	8

ULCR1	UART Line Control Register 1	RW	0x00	0x1003100C	8
UMCR1	UART Modem Control Register 1	RW	0x00	0x10031010	8
ULSR1	UART Line Status Register 1	R	0x00	0x10031014	8
UMSR1	UART Modem Status Register 1	R	0x00	0x10031018	8
USPR1	UART ScratchPad Register 1	RW	0x00	0x1003101C	8
ISR1	Infrared Selection Register 1	RW	0x00	0x10031020	8
URBR2	UART Receive Buffer Register 2	R	0x??	0x10032000	8
UTHR2	UART Transmit Hold Register 2	W	0x??	0x10032000	8
UDLLR2	UART Divisor Latch Low Register 2	RW	0x00	0x10032000	8
UDLHR2	UART Divisor Latch High Register 2	RW	0x00	0x10032004	8
UIER2	UART Interrupt Enable Register 2	RW	0x00	0x10032004	8
UIIR2	UART Interrupt Identification Register 2	R	0x01	0x10032008	8
UFCR2	UART FIFO Control Register 2	W	0x00	0x10032008	8
ULCR2	UART Line Control Register 2	RW	0x00	0x1003200C	8
UMCR2	UART Modem Control Register 2	RW	0x00	0x10032010	8
ULSR2	UART Line Status Register 2	R	0x00	0x10032014	8
UMSR2	UART Modem Status Register 2	R	0x00	0x10032018	8
USPR2	UART ScratchPad Register 2	RW	0x00	0x1003201C	8
ISR2	Infrared Selection Register 2	RW	0x00	0x10032020	8
URBR3	UART Receive Buffer Register 3	R	0x??	0x10033000	8
UTHR3	UART Transmit Hold Register 3	W	0x??	0x10033000	8
UDLLR3	UART Divisor Latch Low Register 3	RW	0x00	0x10033000	8
UDLHR3	UART Divisor Latch High Register 3	RW	0x00	0x10033004	8
UIER3	UART Interrupt Enable Register 3	RW	0x00	0x10033004	8
UIIR3	UART Interrupt Identification Register 3	R	0x01	0x10033008	8
UFCR3	UART FIFO Control Register 3	W	0x00	0x10033008	8
ULCR3	UART Line Control Register 3	RW	0x00	0x1003300C	8
UMCR3	UART Modem Control Register 3	RW	0x00	0x10033010	8
ULSR3	UART Line Status Register 3	R	0x00	0x10033014	8
UMSR3	UART Modem Status Register 3	R	0x00	0x10033018	8
USPR3	UART ScratchPad Register 3	RW	0x00	0x1003301C	8
ISR3	Infrared Selection Register 3	RW	0x00	0x10033020	8

1.3.1 UART Receive Buffer Register (URBR)

The read-only URBR is corresponded to one level 11bit buffer in non-FIFO mode and a 16x11bit FIFO that holds the character(s) received by the UART. Bits in URBR are right-justified when being configured to use fewer than eight bits, and the rest of most significant data bits are zeroed and the most significant three bits of each buffer are the status for the character in the buffer. If ULSR.DRY is 0, don't read URBR, otherwise wrong operation may occur.

URBR0, URBR1, URBR2, URBR3

0x10030000, 0x10031000,
0x10032000, 0x10033000 (DLAB = 0)

Bit									7	6	5	4	3	2	1	0
												URBR				
RST									?	?	?	?	?	?	?	?

Bits	Name	Description	RW
7:0	URBR	8-bit UART receive read data	R

1.3.2　UART Transmit Hold Register (UTHR)

The write-only UTHR is corresponded to one leve 8 bit buffer in non-FIFO mode and a 16x8bit FIFO in FIFO mode that holds the data byte(s) to be transmitted next.

UTHR0, UTHR1, UTHR2, UTHR3

0x10030000, 0x10031000,
0x10032000, 0x10033000 (DLAB = 0)

Bit									7	6	5	4	3	2	1	0
												UTHR				
RST									?	?	?	?	?	?	?	?

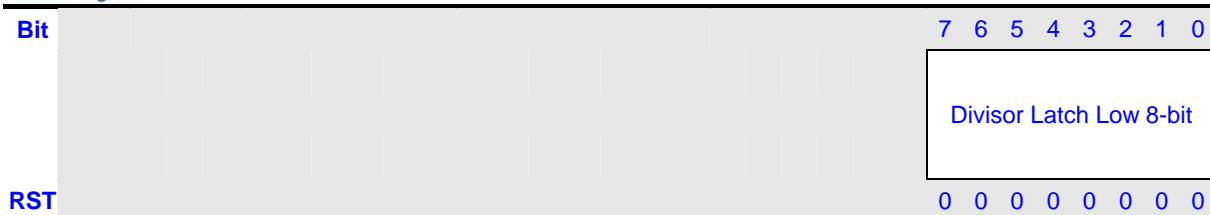| Bits	Name	Description	RW
7:0	UTHR	8-bit UART transmit write hold data	W

1.3.3　UART Divisor Latch Low/High Register (UDLLR / UDLHR)

UART Divisor Latch registers, UDLLR/UDLHR together compose the divisor for the programmable baud rate generator that can take the 3.6864-MHz fixed-input clock and divide it by 1 to $(2^{16} - 1)$. UDLHR/UDLLR stores the high/low 8-bit of the divisor respectively. Load these divisor latches during initialization to ensure that the baud rate generator operates properly. If both Divisor Latch registers are 0, the 16X clock stops. The maximum baud rate is 230.4kbps.

The relationship of baud rate and the value of Divisor is shown by the formula:
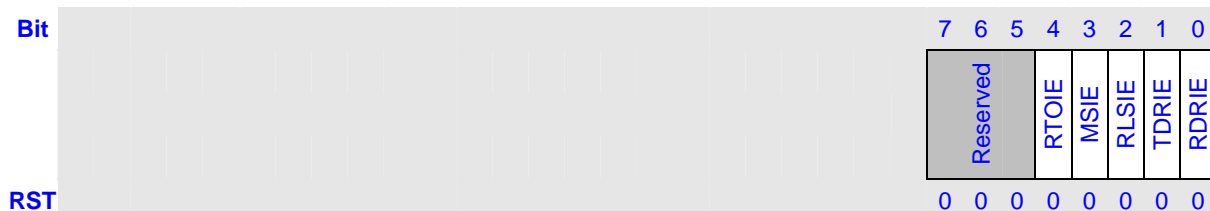　　　**Baud Rate** = 3.6864 Mhz / (16 * **Divisor**)

UDLLR0, UDLLR1, UDLLR2, UDLLR3

0x10030000, 0x10031000,
0x10032000, 0x10033000 (DLAB = 1)

Bit		7 6 5 4 3 2 1 0
		Divisor Latch Low 8-bit
RST		0 0 0 0 0 0 0 0

UDLHR0, UDLHR1, UDLHR2, UDLHR3
0x10030004, 0x10031004, 0x10032004, 0x10033004 (DLAB = 1)

Bit		7 6 5 4 3 2 1 0
		Divisor Latch High 8-bit
RST		0 0 0 0 0 0 0 0
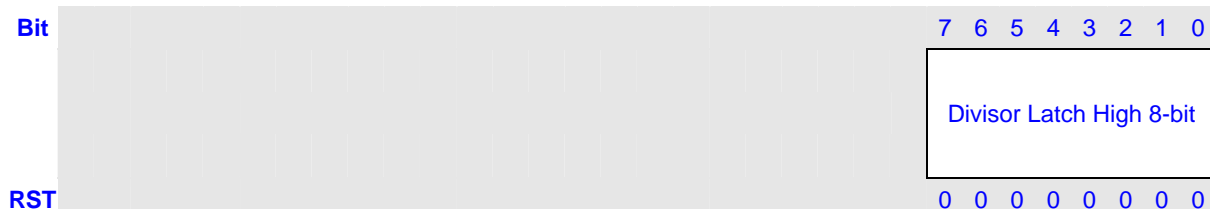
1.3.4 UART Interrupt Enable Register (UIER)

The UART Interrupt Enable Register (UIER) contains the interrupt enable bits for the five types of interrupts (receive data ready, timeout, line status, and transmit data request, and modem status) that set a value in UIIR.

UIER0, UIER1, UIER2, UIER3
0x10030004, 0x100301004, 0x10032004, 0x10033004 (DLAB = 0)

Bit		7 6 5 4 3 2 1 0
		Reserved RTOIE MSIE RLSIE TDRIE RDRIE
RST		0 0 0 0 0 0 0 0

Bits	Name	Description	RW
7:5	Reserved	Always read 0, write is ignored	R
4	RTOIE	**Receive Timeout Interrupt Enable** 0 = Disable the receive timeout interrupt 1 = Enable the receive timeout interrupt Timeout means the URDR (FIFO mode) is not empty but no character has received for a period of time T: T (bits) = 4 X Word length + 12	RW
3	MSIE	**Modem Status Interrupt Enable** 0 = Disable the modem status interrupt 1 = Enable the modem status interrupt	RW
2	RLSIE	**Receive Line Status Interrupt Enable** 0 = Disable receive line status interrupt 1 = Enable receive line status interrupt	RW

Jz4740 peripheral specification, Revision 1.0

1	TDRIE	**Transmit Data Request Interrupt Enable** 0 = Disable the transmit data request interrupt 1 = Enable the transmit data request interrupt	RW
0	RDRIE	**Receive Data Ready Interrupt Enable** 0 = Disable the receive data ready interrupt 1 = Enable the receive data ready interrupt	RW

1.3.5 UART Interrupt Identification Register (UIIR)

The read-only UART Interrupt Identification Register (UIIR) records the prioritized pending interrupt source information. Its initial value after power-on reset is 0x01.

UIIR0, UIIR1, UIIR2, UIIR3	0x10030008, 0x10031008, 0x10032008, 0x10033008

Bit	7	6	5	4	3	2	1	0
	FFMSEL		Reserved		INID			INPEND
RST	0	0	0	0	0	0	0	1

Bits	Name	Description	RW
7:6	FFMSEL	**FIFO Mode Select** 0b00 = Non-FIFO mode 0b01 = Reserved 0b10 = Reserved 0b11 = FIFO mode	R
5:4	Reserved	Always read 0, write is ignored	R
3:1	INID	**Interrupt Identifier** These bits identify the current highest priority pending interrupt. <table><tr><td>INID</td><td>Description</td></tr><tr><td>0b000</td><td>Modem Status</td></tr><tr><td>0b001</td><td>Transmit Data Request</td></tr><tr><td>0b010</td><td>Receive Data Ready</td></tr><tr><td>0b011</td><td>Receive Line Status</td></tr><tr><td>0b100</td><td>Reserved</td></tr><tr><td>0b101</td><td>Reserved</td></tr><tr><td>0b110</td><td>Receive Time Out</td></tr><tr><td>0b111</td><td>Reserved</td></tr></table> See Table 1-3 for details	R

0	INPEND	**Interrupt Pending**	R
		0 = interrupt is pending	
		1 = No interrupt pending	

Table 1-3 UART Interrupt Identification Register Description

UIIR.INID	Interrupt Set/Clear Cause			
	Priority	**Type**	**Source**	**Clear Condition**
0b0001	—	None	No pending interrupt	—
0b0110	1st Highest	Receive Line Status	Overrun, Parity, Frame Error, Break Interrupt, and FIFO Error (DMA mode only)	Reading ULSR or empty all the error characters in DMA mode
0b0100	2nd Highest	Receive Data Ready	FIFO mode: Trigger threshold was reached Non-FIFO mode: URBR full	FIFO mode: Reading URBR till below trigger threshold. Non-FIFO mode: Empty URBR
0b1100	2nd Highest	Receive Timeout	FIFO mode only: URBR not empty but no data read in for a period of time	Reset receive buffer by setting UFCR.RFRT to 1 or Reading URBR
0b0010	3rd Highest	Transmit Data Request	FIFO mode: Empty location in UTHR equal to half or more than half Non-FIFO mode: UTHR empty	FIFO mode: Data number in UTHR more than half Non-FIFO mode: Writing UTHR
0b0000	4th Highest	Modem Status	Modem CTS_ pin status change	Reading UMSR

1.3.6 UART FIFO Control Register (UFCR)

The write-only register UFCR contains the control bits for receive and transmit FIFO.

UFCR0, UFCR1, UFCR2, UFCR3	0x10030008, 0x10031008, 0x10032008, 0x1003300

Bit		7	6	5	4	3	2	1	0
		RDTR		Reserved	UME	DME	TFRT	RFRT	FME
RST		0	0	0	0	0	0	0	0

Bits	Name	Description	RW
7:6	RDTR	**Receive Buffer Data Number Trigger**	W
		These bits are used to select the trigger level for the receive data ready	

		interrupt in FIFO mode. 0b00 = 1 0b01 = 4 0b10 = 8 0b11 = 15	
5	Reserved	Always read 0, write is ignored	R
4	UME	**UART Module Enable** 0 = Disable UART 1 = Enable UART	W
3	DME	**DMA Mode Enable** 0 = Disable DMA mode 1 = Enable DMA mode	W
2	TFRT	**Transmit Holding Register Reset** 0 = Not reset 1 = Reset transmit FIFO	W
1	RFRT	**Receive Buffer Reset** 0 = Not reset 1 = Reset receive FIFO	W
0	FME	**FIFO Mode Enable** Set this bit before the trigger levels. 0 = non-FIFO mode 1 = FIFO mode	W

1.3.7 UART Line Control Register (ULCR)

The ULCR defines the format for UART data transmission.

ULCR0. ULCR1, ULCR2, ULCR3

0x1003000C, 0x1003100C
0x1003200C, 0x1003300C

Bit								7	6	5	4	3	2	1	0
								DLAB	SBK	STPAR	PARM	PARE	SBLS	WLS	
RST								0	0	0	0	0	0	0	0

Bits	Name	Description	RW
7	DLAB	**Divisor Latch Access Bit** 0 = Enable to access URBR, UTHR or UIER 1 = Enable to access UDLLR or UDLHR	W
6	SBK	**Set Break** Causes a break condition (at least one 0x00 data) to be transmitted to the receiving UART. Acts only on the TXD pin and has no effect on the	W

		transmit logic. 0 = No effect on TXD output 1 = Forces TXD output to 0	
5	STPAR	**Sticky Parity** Setting this bit forces parity location to be opposite of PARM bit when PARE is 1 (it is ignored when PARE is 0). 0 = Disable Sticky parity 1 = Enable Sticky parity (opposite o f PARM bit)	W
4	PARM	**Parity Odd/Even Mode Select** If PARE = 0, PARM is ignored 0 = Odd parity 1 = Even parity	W
3	PARE	**Parity Enable** Enables a parity bit to be generated on transmission or checked on reception. 0 = No parity 1 = Parity	W
2	SBLS	**Stop Bit Length Select** Specifies the number of stop bits transmitted and received in each character. When receiving, the receiver checks only the first stop bit. 0 = 1 stop bit 1 = 2 stop bits, except for 5-bit character then 1-1/2 bits	W
1:0	WLS	**Word Length Select** 0b00 = 5-bit character 0b01 = 6-bit character 0b10 = 7-bit character 0b11 = 8-bit character	W

1.3.8 UART Line Status Register (ULSR)

The read-only ULSR indicates status information during the data transfer. Receive error information in ULSR[4:1] remains set until software reads ULSR and it must be read before the error character is read.

ULSR0, ULSR1, ULSR2, ULSR3	0x10030014, 0x10031014, 0x10032014, 0x10033014

Bit	7	6	5	4	3	2	1	0
	FIFOE	TEMP	TDRQ	BI	FMER	PARER	OVER	DRY
RST	0	1	1	0	0	0	0	0

Bits	Name	Description	RW

Jz4740 peripheral specification, Revision 1.0

7	FIFOE	**FIFO Error Status (FIFO mode only)**	R
		FIFOE is set when there is at least one kind of receive error (parity, frame, overrun, break) for any of the characters in receive buffer. FIFOE is reset when all error characters are read out of the buffer.	
		During DMA transfer, the error interrupt generates when FIFOE is 1, and no receive DMA request generates even when data in receive buffer reaches the trigger threshold until all the error characters are read out. In non-DMA mode, FIFOE set does not generate error interrupt.	
		0 = No error data in receive buffer or non-FIFO mode	
		1 = One or more error character in receive buffer	
6	TEMP	**Transmit Holding Register Empty**	R
		Set when both UTHR and shift register are empty. It is cleared when either the UTHR or the shift register contains a data character.	
		0 = There is data in the transmit shifter and UTHR	
		1 = All the data in the transmit shifter and UTHR has been shifted out	
5	TDRQ	**Transmit Data Request**	R
		Set when UTHR has half or more empty location (FIFO mode) or empty (non-FIFO mode).	
		When both UIER.TDRIE and TDRQ are 1, transmit data request interrupt generates or during DMA transfer, DMA request to the DMA controller generates when UIER.TDRIE is 0 and TDRQ is 1.	
		0 = There is one (non-FIFO mode) or more than half data (FIFO mode) in UTHR	
		1 = None data (non-FIFO mode) or half or less than half data (FIFO mode) in UTHR	
4	BI	**Break Interrupt**	R
		BI is set when the received data input is held low for longer than a full-word transmission time (the total time of start bit + data bits + parity bit + stop bits). BI is cleared when the processor reads the ULSR. In FIFO mode, only one character equal to 0x00 is loaded into the FIFO regardless of the length of the break condition. BI shows the break condition for the character at the front of the FIFO, not the most recently received character.	
		0 = No break signal has been received	
		1 = Break signal received	
3	FMER	**Framing Error**	R
		Set when the bit following the last data bit or parity bit is detected to be 0. If the ULCR had been set for two or one and half stop bits, the other stop bits are not checked except the first one. In FIFO mode, FMER shows a	

		framing error for the character at the front of the receive buffer, not for the most recently received character. Cleared when the processor reads the ULSR. 0 = No framing error 1 = Invalid stop bit has been detected	
2	PARER	**Parity Error** Indicates that the received data character does not have the correct even or odd parity, as selected by the even parity select bit. PARER is set upon detection if a parity error and is cleared when the processor reads the ULSR. In FIFO mode, PARER shows a parity error for the character at the front of the FIFO, not the most recently received character. 0 = No parity error 1 = Parity error has occurred	R
1	OVER	**Overrun Error** Set when both receive buffer and shifter are full and new data is received which will be lost. Cleared when the processor reads the ULSR. 0 = No data has been lost 1 = Receive data has been lost	R
0	DRY	**Data Ready** Set when a complete incoming character has been received into the Receive Buffer registers. DRY is cleared when the receive buffer is read (non-FIFO mode) or when the buffer is empty or when the buffer is reset by setting UFCR.RFRT to 1. 0 = No data has been received 1 = Data is available in URBR	R

1.3.9 UART Modem Control Register (UMCR)

The UMCR uses the modem control pins RTS_ and CTS_ to control the interface with a modem or data set. UMCR also controls the loopback mode. Loopback mode must be enabled before the UART is enabled.

UMCR0, UMCR1, UMCR2, UMCR3   **0x10030010, 0x10031010, 0x10032010, 0x10033010**

Bit	7	6	5	4	3	2	1	0
	MDCE	Reserved	LOOP	Reserved	Reserved	RTS	Reserved	
RST	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
7	MDCE	**Modem Control Enable**	W

Jz4740 peripheral specification, Revision 1.0

		0 = Modem function is disabled	
		1 = Modem function is enabled	
6:5	Reserved	Always read 0, write is ignored	R
4	LOOP	**Loop Back** This bit is used for diagnostic testing of the UART. When LOOP is 1, TXD output pin is set to a logic 1 state, RXD is disconnected from the pin, and the output of the transmitter shifter register is looped back into the receiver shift register input internally, similar to CTS_ and RTS_ pins and the RTS bit of the UMCR is connected to CTS bit of UMSR respectively. Loopback mode must be selected before the UART is enabled. 0 = Normal operation mode 1 = Loopback-mode UART operation	W
3	Reserved	Always read 0, write is ignored	R
2	Reserved	Always read 0, write is ignored	R
1	RTS	**Request To Send** This bit can control the RTS_ output state. 0 = RTS_ force to high 1 = RTS_ force to low	W
0	Reserved	Always read 0, write is ignored	R

1.3.10 UART Modem Status Register (UMSR)

The read-only UMSR provides the current state of the control lines from the modem to the processor. They are cleared when the processor reads UMSR.

UMSR0, UMSR1, UMSR2, UMSR3   **0x10030018, 0x10031018, 0x10032018, 0x10033018**

Bit					7	6	5	4	3	2	1	0
					Reserved			CTS	Reserved			CCTS
RST					0	0	0	0	0	0	0	0

Bits	Name	Description	RW
7	Reserved	Always read 0, write is ignored	R
6	Reserved	Always read 0, write is ignored	R
5	Reserved	Always read 0, write is ignored	R
4	CTS	**Status of Clear To Send** When MDCE bit is 1, this bit is the complement of CTS_ input. If Loop bit of UMCR is 1, this bit is equivalent to RTS bit of UMCR. 0 = CTS_ pin is 1 1 = CTS_ pin is 0	R

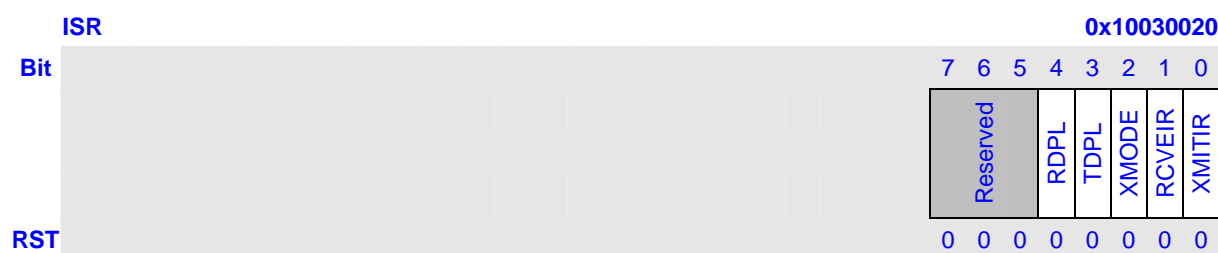| 3	Reserved	Always read 0, write is ignored	R
2	Reserved	Always read 0, write is ignored	R
1	Reserved	Always read 0, write is ignored	R
0	CCTS	**Change status of CTS_** When MDCE bit is 1, this bit indicates the state change on CTS_ pin. 0 = No state change on CTS_ pin since last read of UMSR 1 = A change occurs on the state of CTS_ pin	R

1.3.11 UART Scratchpad Register

This Scratchpad register is used as a scratch register for the programmer and has no effect on the UART.

USPR   **0x1003001C**

Bit   7 6 5 4 3 2 1 0

Scratch Data

RST   ? ? ? ? ? ? ? ?
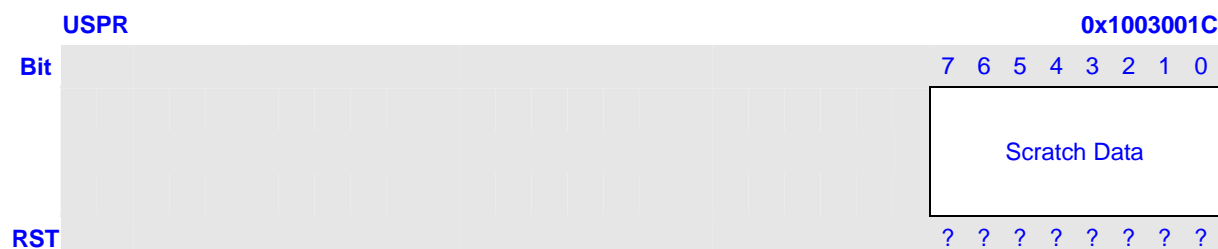
1.3.12 Infrared Selection Register (ISR)

The ISR is used to configure the slow-infrared (SIR) interface that is provided in each UART to support two-way wireless communication using infrared transmission that conforms to the IrDA serial infrared specification 1.1. The maximum frequency is up to 115.2kbps.

ISR   **0x10030020**

Bit   7 6 5 4 3 2 1 0

| Reserved | RDPL | TDPL | XMODE | RCVEIR | XMITIR |

RST   0 0 0 0 0 0 0 0

Bits	Name	Description	RW
7:5	Reserved	Always read 0, write is ignored	R
4	RDPL	**Receive Data Polarity** 0 = Slow-infrared (SIR) interface decoder takes positive pulses as zeros. 1 = SIR decoder takes negative pulses as zeros.	W
3	TDPL	**Transmit Data Polarity** 0 = SIR encoder generates a positive pulse for a data bit of zero. 1 = SIR encoder generates a negative pulse for a data bit of zero.	W

Jz4740 peripheral specification, Revision 1.0

2	XMODE	**Transmit Pulse Width Mode** Set when the transmit encoder needs to generate 1.6us pulses (that are 3/16 of a bit-time at 115.2 kbps). Cleared when the transmit encoder needs to generate 3/16 of a bit-time wide according to current baud rate. 0 = Transmit pulse width is 3/16 of a bit-time wide. 1 = Transmit pulse width is 1.6 us.	W
1	RCVEIR	**Receiver SIR Enable** This bit is used to select the signal from the RXD pin is processed by the IrDA decoder before it is fed to the UART (RCVEIR = 1) or bypass IrDA decoder and is fed directly to the UART (RCVEIR = 0). 0 = Receiver is in UART mode. 1 = Receiver is in SIR mode.	W
0	XMITIR	**Transmitter SIR Enable** This bit is used to select TXD output pin is processed by the IrDA encoder before it is fed to the device pin (XMITIR = 1) or bypass IrDA encoder and is fed directly to the device pin (XMITIR = 0). Note: disable infrared LED before XMITIR is set, otherwise a false start bit may occur. 0 = Transmitter is in UART mode. 1 = Transmitter is in SIR mode.	W

1.3.13 Uart M Register (UMR)

UMSR0, UMSR1, UMSR2, UMSR3   **0x10030024, 0x10031024, 0x10032024, 0x10033024**

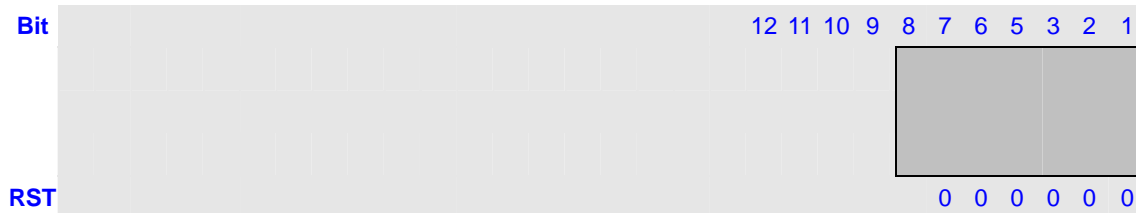| Bit	7	6	5	4	3	2	1	0
RST	0	0	0	0	0	0	0	0
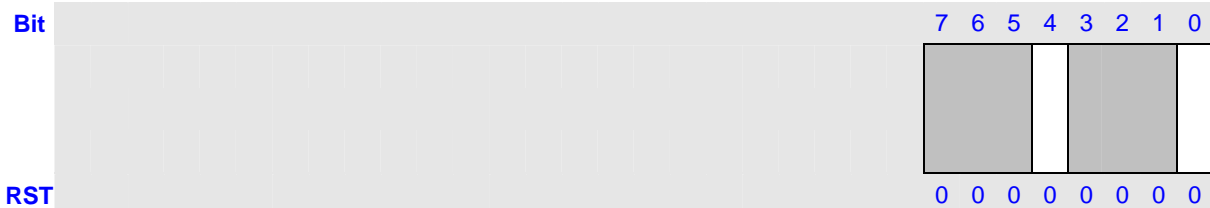
M is the value of UMR register.

It will take UART at least M cycles for transmitter to transmit one bit and receiver to receive one bit ,

It will take UART at most M+1 cycles for transmitter to transmit one bit and receiver to receive one bit,

1.3.14 Uart Add Cycle Register(UACR)

UMSR0, UMSR1, UMSR2, UMSR3

Bit						12	11	10	9	8	7	6	5	3	2	1
RST										0	0	0	0	0	0	

If Nth bit of the register is 1 ,It will take Uart M+1 cycles to transmit or receive the bit of date for transmit or receive.

If the register is 12'h0 ,UART will receive or transmit   a bit by M cycle .

If the register is 12'hfff ,UART will receive or transmit   a bit by M+1 cycle .

For the detail to see   1.4.8For any frequency clock to use the Uart

1.4   Operation

The following sections describe the UART operations that include flow of configuration, data transmission, data reception, and Infra-red mode.

1.4.1   UART Configuration

Before UART starts to transfer data or changing transfer format, configuration must be done to define the transfer format. The sample flow is as the following:

In FIFO mode, set FME bit of UFCR to 1, reset receive and transmit FIFO, then initialize the UART as described below.

1.   Clear UFCR.UME to 0
2.   Set value in UDLL/UDHR to generate the baud rate clock
3.   Set data format in ULCR
4.   If it is in FIFO MODE, set FME bit and other FIFO control in UFCR, reset receive and transmit FIFO, otherwise skip item 4
5.   Set each interrupt enable bit in UIER in interrupt-based transfer or set UFCR.DME in DMA-based transfer (DMA transfer is FIFO mode only), then set UFCR.UME

1.4.2   Data Transmission

After configuration, UART is ready for data transfer. For data transmission, refer to the following procedure:

1. Read ULSR.TDRQ (interrupt disable) or wait for transmit data request interrupt (interrupt enable), if TDRQ = 1 or transmit data request interrupt generates, that means there is enough empty location in UTHR for new data

2. If ULSR.TDRQ is 1 or get the transmit data request interrupt, write transmit data to UTHR to start transmission

3. Do item 1 and item 2 if there are more data waiting for transmit

4. After all necessary data are written to UTHR, wait ULSR.TEMP = 1, that means all data completely transmitted

5. If it is necessary to send break, set ULCR.SBK and at least wait for 1-bit interval time to send a valid break, then clear ULCR.SBK

6. Clear UME bit to finish UART transmission

1.4.3   Data Reception

After configuration, UART is ready for data transfer. For data reception, refer to the following sample procedure:

1. Read ULSR.DRY (interrupt disable) or wait for receive data request interrupt (interrupt enable), if ULSR.DRY =1 or receive data request interrupt generates, that means URBR has one data (non-FIFO mode) or data in URBR reaches the trigger value (FIFO mode)

2. If ULSR.DRY = 1 or receive data request interrupt generates, then read ULSR.FIFOE or see if there is error interrupt, if FIFOE = 1, it means received data has receive error, then go to error handler, other wise go to item 3

3. Read one received data in URBR (non-FIFO mode) or data equal to trigger value in URBR (FIFO mode)

4. Check whether all data received: check whether ULSR.DRY = 0, in FIFO mode and interrupt is enabled, timeout interrupt may generate, when timeout interrupt generates, read URBR till ULSR.DRY = 0

5. Clear UFCR.UME to end data reception when all data are received and ULSR.DRY = 0

1.4.4   Receive Error Handling

A sample error handling flow is as the following:

1. If ULSR.FIFOE = 1, it means there is receive error in received data, then check what error it is

2. If ULSR.OVER = 1, go to OVER error handling

3. If ULSR.BI = 1, go to Break handling

4. If ULSR.FMER = 1, go to Frame error handling

5. If PARER = 1, go to PARER error handling

1.4.5 Modem Transfer

When UMCR.MDCE = 1, modem control is enabled. Transfer flow can be stopped and restarted by software through RTS_ and CTS_ pin. When UART transmitter detects low level on CTS_ pin, it stops transmission and TxD pin goes to mark state after finishing transmitting the current character until it detects CTS_ pin goes back to high level. RTS_ pin is output to receiving UART and its state can be controlled by setting UMCR.RTS bit, that is, setting UMCR.RTS to 1, RTS_ pin is low level output that means UART is ready to receive data, on the contrary, it means UART currently can't receive more data.

1.4.6 DMA Transfer

UART can operate in DMA-based (UFCR.DME = 1, FIFO mode only), that is, dma request initiated by UART takes the place of interrupt request and transmission/reception is carried out using DMA instead of CPU. Be sure that software guarantee to disable transmit and receive interrupt except timeout and error interrupts.

During DMA transfer, if an interrupt occurs, software must first read the ULSR to see if an error interrupt exists, then check the UIIR for the source of the interrupt and if DMA channel is already halt because of the error indicator from UART, then disable DMA channel and read out all the error data from receive FIFO. Software re-set and re-enable DMA and data transfer by DMA will re-start.

1.4.7 Slow IrDA Asynchronous Interface

Each UART supports slow infra-red (SIR) transmission and reception by setting ISR.XMITIR and ISR.RCVEIR to 1 (make sure the two bits are not set to 1 at the same time because SIR can't operate full-duplex). According to the IrDA 1.1, data rate is limited at a maximum value of 115.2Kbps.

In SIR transmit mode, the transmit pulse comes out at a rate of 3/16 (when the transmit data bit is zero); in SIR receive mode, the receiver must detect the 3/16 pulsed period to recognize a zero value (an active high or low pulse is demodulation to 0, and no pulse is demodulation to 1).

Compared to normal UART, there are some limitations to SIR, that is, each character is fixed to 8-bit data width, no parity and 1 stop bit and modem function is ignored. The IrDA 1.1 specifies a minimum 10ms latency after an optical node ceases transmitting before its receiver recovers its receiving function and software must guarantee this delay.

In the IrDA 1.1 specification, communication must start up at the rate of 9600bps, but then allows the link to negotiate higher (or lower) data rates if supported by both ends. However, the communication rate will not automatically change. Change, if necessary, is performed by software.

1.4.8 For any frequency clock to use the Uart

Note: if you don't set M register and UACR the uart work at normal mode with the specified frequencies. To use other frequency you should to set Mregister and UACR to right value .

1.The Improving

Following changes are made

- One bit is composed by M CLK_{BR} cycles, which can be 4~1024
- Some extra CLK_{BR} cycles can be inserted in some bits in one frame, so that like M has fraction.

For instance:

$$CLK_{BR} = CLK_{DEV} / N \qquad N = 1, 2, …$$
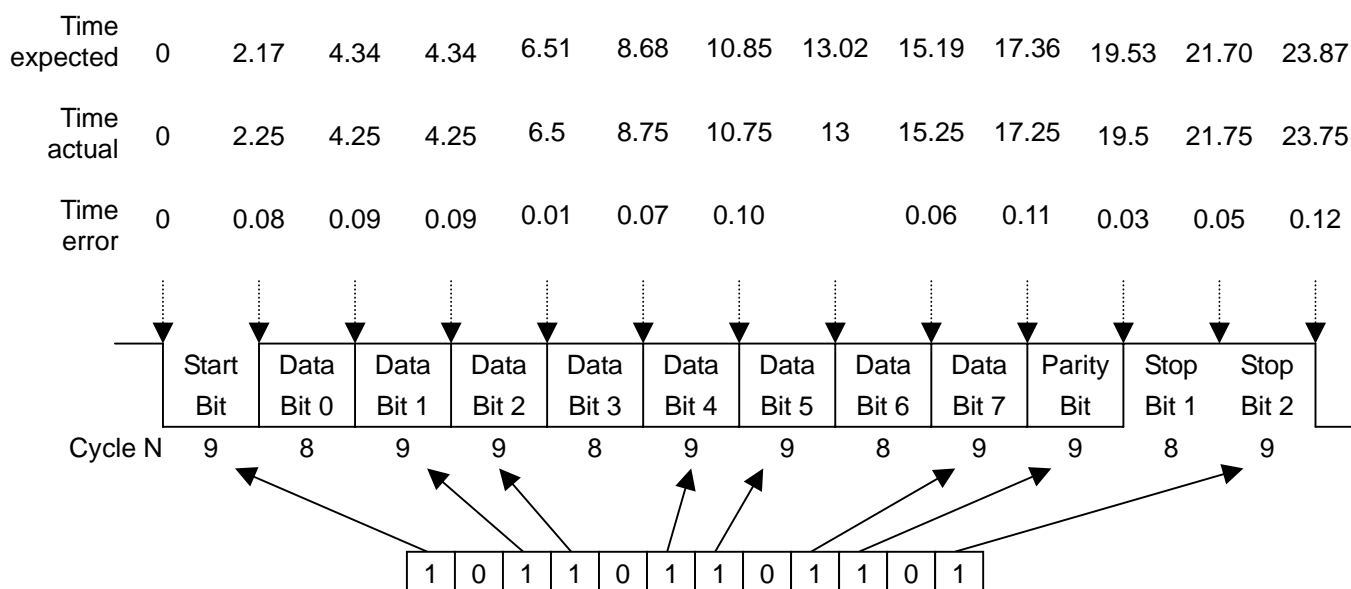$$CLK_{BR} = CLK_{DEV} = 4MHz$$
Band rate = 460800

In accurate

$$M_a = 8.681$$

We take

M = 8, with 8 extra cycles in every frame

A 12-bit register is used to indicate where to insert the extra cycles

Time expected	0	2.17	4.34	4.34	6.51	8.68	10.85	13.02	15.19	17.36	19.53	21.70	23.87
Time actual	0	2.25	4.25	4.25	6.5	8.75	10.75	13	15.25	17.25	19.5	21.75	23.75
Time error	0	0.08	0.09	0.09	0.01	0.07	0.10		0.06	0.11	0.03	0.05	0.12

Start Bit	Data Bit 0	Data Bit 1	Data Bit 2	Data Bit 3	Data Bit 4	Data Bit 5	Data Bit 6	Data Bit 7	Parity Bit	Stop Bit 1	Stop Bit 2

| Cycle N | 9 | 8 | 9 | 9 | 8 | 9 | 9 | 8 | 9 | 9 | 8 | 9 |

1	0	1	1	0	1	1	0	1	1	0	1

For transmission, in theory, the biggest error is half of CLK_{BR} cycle, which is 0.125us here.

2 To set UMR register

$CLK_{BR} = CLK_{DEV} / N$

$M_a = CLK_{BR}/band\ rate$

M is modem of M_a。

Write M to Mregister.

Considering the power and the robust quality, for M form 6 to 32 is you better select by set the UDLR.

The max error

$$\frac{0.5/CLK_{BR}}{M_a/CLK_{BR}} = 0.5/M_a < 0.5/M$$

M	4	8	16	32	64
error/W_{bit}	12.5%	6.25%	3.125%	1.56%	0.78%

Table 1

3. To set UACR value

for each bit of it means:

0 :means not to add additional cycle to the the bit that uart is prepare to transmit or receive , in another word ,you will to use M cycles to transmit or receive the bit .

1 :means to add additional cycle to the the bit that uart is prepare to transmit or receive,in another word, you will to use M+1 cycles to transmit or receive the bit .

To set UACR value you must ensure that the max error of each bit should be less than $0.5P_{BR}$.

For example：$M_a - M = 0.15$; $M+1 - M_a = 0.85$;

Write UMR 8

Write UMR 408

cycle/bit ：   M，M，M，M+1，M，M，M，M，M，M，M+1，M

UACR   ：   0   0   0   1   0   0   0   0   0   0   1   0