Chapter 75
Universal Asynchronous Receiver/Transmitter (UART)

75.1 Overview

The UART block provides serial communication capability with external devices through an RS-232 cable or through use of external circuitry that converts infrared signals to electrical signals (for reception) or transforms electrical signals to signals that drive an infrared LED (for transmission) to provide low speed IrDA compatibility. The UART module supports NRZ encoding format and IrDA-compatible infrared slow data rate (SIR) format.

Figure 75-1 is the UART block diagram.

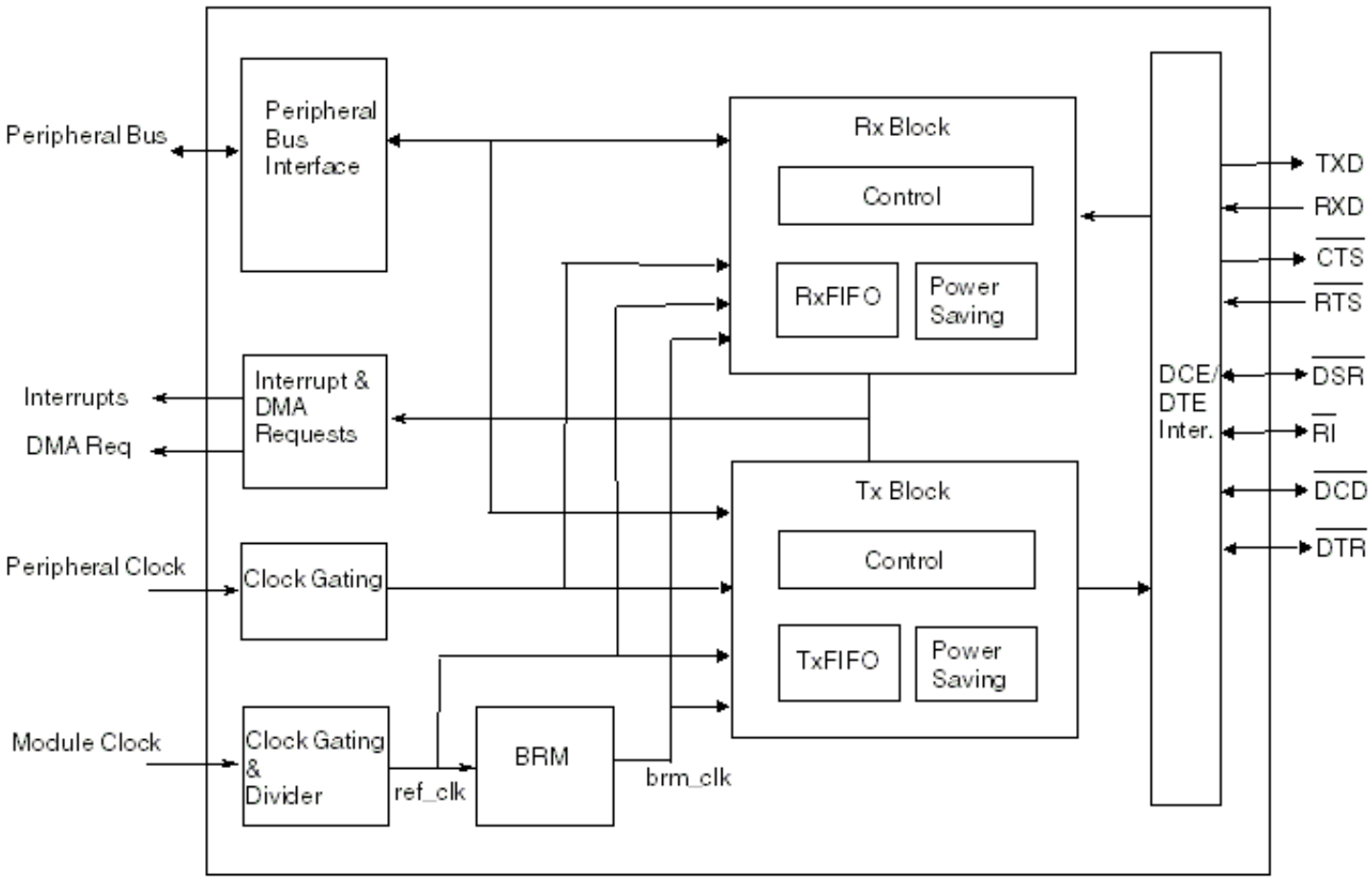


Figure 75-1. UART Block Diagram

75.1.1 Features

The UART includes the following features:

- High-speed TIA/EIA-232-F compatible, up to 4.0 Mbit/s
- Serial IR interface low-speed, IrDA-compatible (up to 115.2 Kbit/s)
- 7 or 8 data bits
- 1 or 2 stop bits
- Programmable parity (even, odd, and no parity)
- Hardware flow control support for request to send ($\overline{RTS}$) and clear to send ($\overline{CTS}$) signals
- Edge-selectable $\overline{RTS}$ and edge-detect interrupts
- Status flags for various flow control and FIFO states
- Voting logic for improved noise immunity (16x oversampling)
- Transmitter FIFO empty interrupt suppression

- UART internal clocks enable/disable
- Auto baud rate detection (up to 115.2 Kbit/s)
- Receiver and transmitter enable/disable for power saving
- DCE/DTE capability
- $\overline{\text{RTS}}$, IrDA asynchronous wake (AIRINT), receive asynchronous wake (AWAKE), $\overline{\text{RI}}$ (DTE only), $\overline{\text{DCD}}$ (DTE only), $\overline{\text{DTR}}$ (DCE only) and $\overline{\text{DSR}}$ (DTE only) interrupts wake the processor from STOP mode
- Maskable interrupts
- Two DMA Requests (TxFIFO DMA Request and RxFIFO DMA Request)
- Escape character sequence detection
- Software reset ($\overline{\text{SRST}}$)
- Two independent, 32-entry FIFOs for transmit and receive
- The peripheral clock can be totally asynchronous with the module clock. The module clock determines baud rate. This allows frequency scaling on peripheral clock (such as during DVFS mode) while remaining the module clock frequency and baud rate.

75.1.2 Modes of Operation

- Serial RS-232NRZ mode
- IrDA mode

75.2 External Signals

Conventions: Table 75-1 lists conventions for representing signals.

Table 75-1. Module Signal Conventions

Category	Convention	Example(s
Off-chip signal	Uppercase (all capital letters)	TXD
Internal signal[1]	Lowercase italics	*core_int*
Active low signal	_B (_b) suffix or overbar	RESET_EN_B or $\overline{\text{RESET\_EN}}$
Range of bussed or commonly named signals	Beginning and end points of the range are: • Separated by a colon. • Surrounded by square brackets.	ADDR[31:0] CSE_B[7:0] or $\overline{\text{CSE}}$[7:0]
Individual signal in a range of bussed or commonly named signals	Individual number in the range appears without a colon or square brackets	ADDR31 CSE0_B or $\overline{\text{CSE0}}$

1. Internal signals are for reference only in descriptions of internal module or SoC functionality.

Table 75-2 describes all UART signals that connect off-chip.

Table 75-2. Off-Chip Module Signals

Signal name	I/O	Active state	Description	Reset state
Serial / IrDA Signals				
RXD	I		Serial / infrared data receive	
TXD	O		Serial/infrared data transmit	High
Modem Control Signals				
$\overline{\text{CTS}}$	O	Low	Clear to send	High
$\overline{\text{RTS}}$	I	Low	Request to send	
$\overline{\text{DSR}}$	I/O	Low	Data set ready	High
$\overline{\text{DCD}}$	I/O	Low	Data carrier detected	High
$\overline{\text{DTR}}$	I/O	Low	Data terminal ready	
$\overline{\text{RI}}$	I/O	Low	Ring indicator	High
Interrupts				
interrupt_uart	O	Low	UART interrupt	High
DMA Requests				
dma_req_rx	O	Low	Receiver DMA request	High
dma_req_tx	O	Low	Transmitter DMA request	High
Clocks				
peripheral_clock	I		Peripheral clock	
module_clock	I		Clock source for the module's logic	
Special Signals				
stop_req	I	High	Module stop mode	
doze_req	I	High	Module doze mode	
debug_req	I	High	Module debug	

75.2.1 Detailed Signal Descriptions

75.2.1.1 Serial / IrDA Signals

75.2.1.1.1 RXD - Data Receive

Input asynchronous data receive in Serial and IrDA modes.

75.2.1.1.2 TXD - Data Transmit

Output asynchronous data transmit in Serial and IrDA modes.

75.2.1.2 Modem Control Signals

75.2.1.2.1 $\overline{\text{CTS}}$ - Clear To Send

Output in DCE and DTE mode. This signal informs the remote modem that UART is ready to receive data.

75.2.1.2.2 $\overline{\text{RTS}}$ - Request To Send

Input in DCE and DTE mode. This signal informs UART that remote modem is ready to receive data.

75.2.1.2.3 $\overline{\text{DSR}}$ - Data Set Ready

Input in DTE mode. Indicates to UART that remote modem is operational.

Output in DCE mode. Indicates to remote modem that UART is operational.

75.2.1.2.4 $\overline{\text{DCD}}$ - Data Carrier Detected

Input in DTE mode. Indicates to UART that a good carrier is being received from the remote modem.

Output in DCE mode. Indicates to remote device that a good carrier is being received from the UART.

75.2.1.2.5 $\overline{\text{DTR}}$ - Data Terminal Ready

Input in DCE mode. Indicates to UART (in DCE mode) that remote device (in DTE mode) is operational.

Output in DTE mode. Indicates to remote modem (in DCE mode) that UART (in DTE mode) is operational.

75.2.1.2.6 $\overline{\text{RI}}$ - Ring Indicator

Input in DTE mode. Indicates to UART that remote modem is detecting a ringing tone.

Output in DCE mode. Indicates to remote device that UART is detecting a ringing tone.

75.2.1.3 Interrupt Signals

75.2.1.3.1 *interrupt_uart* - UART Interrupt

Output interrupt request.

75.2.1.4 DMA Request Signals

75.2.1.4.1 *dma_req_rx* - Receiver DMA Request

Output DMA Request signal for receiver interface.

75.2.1.4.2 *dma_req_tx* - Transmitter DMA Request

Output DMA Request signal for transmitter interface. Set at 0 when TXDMAEN (UCR1[3]) is at 1 and TRDY (USR1[13]) is also at 1.

75.2.1.5 Clock Signals

75.2.1.5.1 *peripheral_clock* - Peripheral Clock

See Clocks for more information about *peripheral_clock* .

75.2.1.5.2 *module_clock* - Module Clock

See Clocks for more information about *module_clock* .

75.2.1.6 Special Signals

75.2.1.6.1 *stop_req* - Stop Mode

Input stop mode. Indicates to UART that ARM platform is going to enter in Stop Mode and clocks are going to stop running. See Low Power Modes for more information about Stop Mode.

75.2.1.6.2 *doze_req* - Doze Mode

Input doze mode. ARM platform requests UART to switch in doze mode (power saving mode). See Low Power Modes for more information about Doze Mode.

75.2.1.6.3 *debug_req* - Debug Mode

Input debug mode. Indicates UART it has to enter in debug mode. See UART Operation in System Debug State, for more information about Debug Mode.

75.3 Programmable Registers

UART supports 8-bit, 16-bit and 32-bit accesses to 32-bit memory-mapped addresses. Any access to unmapped memory location will yield a transfer error.

All registers except the ONEMS described in this section are 16-bit registers. The ONEMS register is a 24-bit register.

- For 32-bit write accesses, the upper two bytes will not be taken into account.
- For 32-bit read accesses the upper two bytes will return 0.

The ONEMS register is expanded from 16 bits to 24 bits in order to support the high frequency of the BRM internal clock *ref_clk* (*module_clock* after divider). The ONEMS register can be accessed as 8 bits, 16 bits or 32 bits.

- For 32-bit write accesses, the most significant byte of the ONEMS will be discarded.
- For 32-bit read accesses, the most significant byte of the ONEMS will be read as 0.

UART memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FB_C000	UART Receiver Register (UART-1_URXD)	32	R	0000_0000h	75.3.1/ 4414
53FB_C040	UART Transmitter Register (UART-1_UTXD)	32	W	0000_0000h	75.3.2/ 4416
53FB_C080	UART Control Register 1 (UART-1_UCR1)	32	R/W	0000_0000h	75.3.3/ 4417
53FB_C084	UART Control Register 2 (UART-1_UCR2)	32	R/W	0000_0001h	75.3.4/ 4420
53FB_C088	UART Control Register 3 (UART-1_UCR3)	32	R/W	0000_0700h	75.3.5/ 4422

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FB_C08C	UART Control Register 4 (UART-1_UCR4)	32	R/W	0000_8000h	75.3.6/ 4425
53FB_C090	UART FIFO Control Register (UART-1_UFCR)	32	R/W	0000_0801h	75.3.7/ 4427
53FB_C094	UART Status Register 1 (UART-1_USR1)	32	R/W	0000_2040h	75.3.8/ 4428
53FB_C098	UART Status Register 2 (UART-1_USR2)	32	R/W	0000_4028h	75.3.9/ 4431
53FB_C09C	UART Escape Character Register (UART-1_UESC)	32	R/W	0000_002Bh	75.3.10/ 4433
53FB_C0A0	UART Escape Timer Register (UART-1_UTIM)	32	R/W	0000_0000h	75.3.11/ 4434
53FB_C0A4	UART BRM Incremental Register (UART-1_UBIR)	32	R/W	0000_0000h	75.3.12/ 4434
53FB_C0A8	UART BRM Modulator Register (UART-1_UBMR)	32	R/W	0000_0000h	75.3.13/ 4435
53FB_C0AC	UART Baud Rate Count Register (UART-1_UBRC)	32	R	0000_0004h	75.3.14/ 4436
53FB_C0B0	UART One Millisecond Register (UART-1_ONEMS)	32	R/W	0000_0000h	75.3.15/ 4437
53FB_C0B4	UART Test Register (UART-1_UTS)	32	R/W	0000_0060h	75.3.16/ 4438
53FC_0000	UART Receiver Register (UART-2_URXD)	32	R	0000_0000h	75.3.1/ 4414
53FC_0040	UART Transmitter Register (UART-2_UTXD)	32	W	0000_0000h	75.3.2/ 4416
53FC_0080	UART Control Register 1 (UART-2_UCR1)	32	R/W	0000_0000h	75.3.3/ 4417
53FC_0084	UART Control Register 2 (UART-2_UCR2)	32	R/W	0000_0001h	75.3.4/ 4420
53FC_0088	UART Control Register 3 (UART-2_UCR3)	32	R/W	0000_0700h	75.3.5/ 4422
53FC_008C	UART Control Register 4 (UART-2_UCR4)	32	R/W	0000_8000h	75.3.6/ 4425
53FC_0090	UART FIFO Control Register (UART-2_UFCR)	32	R/W	0000_0801h	75.3.7/ 4427
53FC_0094	UART Status Register 1 (UART-2_USR1)	32	R/W	0000_2040h	75.3.8/ 4428
53FC_0098	UART Status Register 2 (UART-2_USR2)	32	R/W	0000_4028h	75.3.9/ 4431

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FC_009C	UART Escape Character Register (UART-2_UESC)	32	R/W	0000_002Bh	75.3.10/ 4433
53FC_00A0	UART Escape Timer Register (UART-2_UTIM)	32	R/W	0000_0000h	75.3.11/ 4434
53FC_00A4	UART BRM Incremental Register (UART-2_UBIR)	32	R/W	0000_0000h	75.3.12/ 4434
53FC_00A8	UART BRM Modulator Register (UART-2_UBMR)	32	R/W	0000_0000h	75.3.13/ 4435
53FC_00AC	UART Baud Rate Count Register (UART-2_UBRC)	32	R	0000_0004h	75.3.14/ 4436
53FC_00B0	UART One Millisecond Register (UART-2_ONEMS)	32	R/W	0000_0000h	75.3.15/ 4437
53FC_00B4	UART Test Register (UART-2_UTS)	32	R/W	0000_0060h	75.3.16/ 4438
53FC_4000	UART Receiver Register (UART-3_URXD)	32	R	0000_0000h	75.3.1/ 4414
53FC_4040	UART Transmitter Register (UART-3_UTXD)	32	W	0000_0000h	75.3.2/ 4416
53FC_4080	UART Control Register 1 (UART-3_UCR1)	32	R/W	0000_0000h	75.3.3/ 4417
53FC_4084	UART Control Register 2 (UART-3_UCR2)	32	R/W	0000_0001h	75.3.4/ 4420
53FC_4088	UART Control Register 3 (UART-3_UCR3)	32	R/W	0000_0700h	75.3.5/ 4422
53FC_408C	UART Control Register 4 (UART-3_UCR4)	32	R/W	0000_8000h	75.3.6/ 4425
53FC_4090	UART FIFO Control Register (UART-3_UFCR)	32	R/W	0000_0801h	75.3.7/ 4427
53FC_4094	UART Status Register 1 (UART-3_USR1)	32	R/W	0000_2040h	75.3.8/ 4428
53FC_4098	UART Status Register 2 (UART-3_USR2)	32	R/W	0000_4028h	75.3.9/ 4431
53FC_409C	UART Escape Character Register (UART-3_UESC)	32	R/W	0000_002Bh	75.3.10/ 4433
53FC_40A0	UART Escape Timer Register (UART-3_UTIM)	32	R/W	0000_0000h	75.3.11/ 4434
53FC_40A4	UART BRM Incremental Register (UART-3_UBIR)	32	R/W	0000_0000h	75.3.12/ 4434
53FC_40A8	UART BRM Modulator Register (UART-3_UBMR)	32	R/W	0000_0000h	75.3.13/ 4435

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FC_40AC	UART Baud Rate Count Register (UART-3_UBRC)	32	R	0000_0004h	75.3.14/ 4436
53FC_40B0	UART One Millisecond Register (UART-3_ONEMS)	32	R/W	0000_0000h	75.3.15/ 4437
53FC_40B4	UART Test Register (UART-3_UTS)	32	R/W	0000_0060h	75.3.16/ 4438
53FB_C000	UART Receiver Register (UART-4_URXD)	32	R	0000_0000h	75.3.1/ 4414
53FB_C040	UART Transmitter Register (UART-4_UTXD)	32	W	0000_0000h	75.3.2/ 4416
53FB_C080	UART Control Register 1 (UART-4_UCR1)	32	R/W	0000_0000h	75.3.3/ 4417
53FB_C084	UART Control Register 2 (UART-4_UCR2)	32	R/W	0000_0001h	75.3.4/ 4420
53FB_C088	UART Control Register 3 (UART-4_UCR3)	32	R/W	0000_0700h	75.3.5/ 4422
53FB_C08C	UART Control Register 4 (UART-4_UCR4)	32	R/W	0000_8000h	75.3.6/ 4425
53FB_C090	UART FIFO Control Register (UART-4_UFCR)	32	R/W	0000_0801h	75.3.7/ 4427
53FB_C094	UART Status Register 1 (UART-4_USR1)	32	R/W	0000_2040h	75.3.8/ 4428
53FB_C098	UART Status Register 2 (UART-4_USR2)	32	R/W	0000_4028h	75.3.9/ 4431
53FB_C09C	UART Escape Character Register (UART-4_UESC)	32	R/W	0000_002Bh	75.3.10/ 4433
53FB_C0A0	UART Escape Timer Register (UART-4_UTIM)	32	R/W	0000_0000h	75.3.11/ 4434
53FB_C0A4	UART BRM Incremental Register (UART-4_UBIR)	32	R/W	0000_0000h	75.3.12/ 4434
53FB_C0A8	UART BRM Modulator Register (UART-4_UBMR)	32	R/W	0000_0000h	75.3.13/ 4435
53FB_C0AC	UART Baud Rate Count Register (UART-4_UBRC)	32	R	0000_0004h	75.3.14/ 4436
53FB_C0B0	UART One Millisecond Register (UART-4_ONEMS)	32	R/W	0000_0000h	75.3.15/ 4437
53FB_C0B4	UART Test Register (UART-4_UTS)	32	R/W	0000_0060h	75.3.16/ 4438
63F9_0000	UART Receiver Register (UART-5_URXD)	32	R	0000_0000h	75.3.1/ 4414

Table continues on the next page...

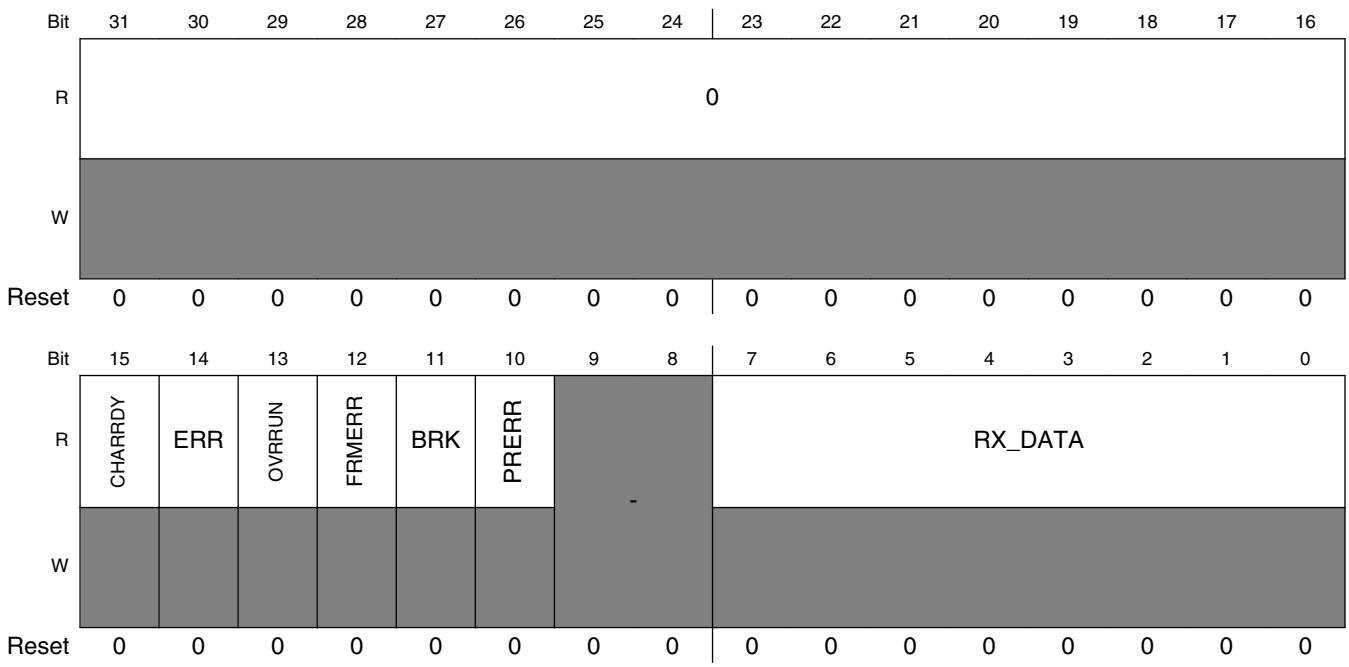
UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63F9_0040	UART Transmitter Register (UART-5_UTXD)	32	W	0000_0000h	75.3.2/ 4416
63F9_0080	UART Control Register 1 (UART-5_UCR1)	32	R/W	0000_0000h	75.3.3/ 4417
63F9_0084	UART Control Register 2 (UART-5_UCR2)	32	R/W	0000_0001h	75.3.4/ 4420
63F9_0088	UART Control Register 3 (UART-5_UCR3)	32	R/W	0000_0700h	75.3.5/ 4422
63F9_008C	UART Control Register 4 (UART-5_UCR4)	32	R/W	0000_8000h	75.3.6/ 4425
63F9_0090	UART FIFO Control Register (UART-5_UFCR)	32	R/W	0000_0801h	75.3.7/ 4427
63F9_0094	UART Status Register 1 (UART-5_USR1)	32	R/W	0000_2040h	75.3.8/ 4428
63F9_0098	UART Status Register 2 (UART-5_USR2)	32	R/W	0000_4028h	75.3.9/ 4431
63F9_009C	UART Escape Character Register (UART-5_UESC)	32	R/W	0000_002Bh	75.3.10/ 4433
63F9_00A0	UART Escape Timer Register (UART-5_UTIM)	32	R/W	0000_0000h	75.3.11/ 4434
63F9_00A4	UART BRM Incremental Register (UART-5_UBIR)	32	R/W	0000_0000h	75.3.12/ 4434
63F9_00A8	UART BRM Modulator Register (UART-5_UBMR)	32	R/W	0000_0000h	75.3.13/ 4435
63F9_00AC	UART Baud Rate Count Register (UART-5_UBRC)	32	R	0000_0004h	75.3.14/ 4436
63F9_00B0	UART One Millisecond Register (UART-5_ONEMS)	32	R/W	0000_0000h	75.3.15/ 4437
63F9_00B4	UART Test Register (UART-5_UTS)	32	R/W	0000_0060h	75.3.16/ 4438

75.3.1 UART Receiver Register (UARTx_URXD)

The UART will yield a transfer error on the peripheral bus when core is reading URXD register with receive interface disabled (RXEN=0 or UARTEN=0).

Addresses: UART-1_URXD is 53FB_C000h base + 0h offset = 53FB_C000h
UART-4_URXD is 53FB_C000h base + 0h offset = 53FB_C000h
UART-2_URXD is 53FC_0000h base + 0h offset = 53FC_0000h
UART-3_URXD is 53FC_4000h base + 0h offset = 53FC_4000h
UART-5_URXD is 63F9_0000h base + 0h offset = 63F9_0000h



UARTx_URXD field descriptions

Field	Description
31–16 Reserved	This read-only bitfield is reserved and always has the value zero. Reserved
15 CHARRDY	Character Ready. This read-only bit indicates an invalid read when the FIFO becomes empty and software tries to read the same old data. This bit should not be used for polling for data written to the RX FIFO. 0 Character in RX_DATA field and associated flags are invalid. 1 Character in RX_DATA field and associated flags valid and ready for reading.
14 ERR	**Error Detect.** Indicates whether the character present in the RX_DATA field has an error (OVRRUN, FRMERR, BRK or PRERR) status. The ERR bit is updated and valid for each received character. 0 No error status was detected 1 An error status was detected

Table continues on the next page...

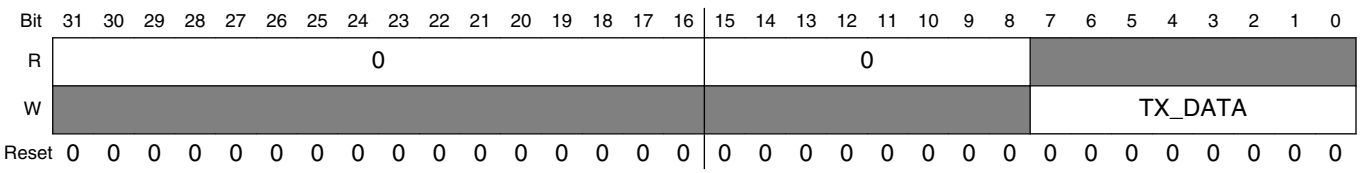
UARTx_URXD field descriptions (continued)

Field	Description
13 OVRRUN	**Receiver Overrun.** This read-only bit, when HIGH, indicates that the corresponding character was stored in the last position (32nd) of the Rx FIFO. Even if a 33rd character has not been detected, this bit will be set to '1' for the 32nd character. 0 No RxFIFO overrun was detected 1 A RxFIFO overrun was detected
12 FRMERR	**Frame Error.** Indicates whether the current character had a framing error (a missing stop bit) and is possibly corrupted. FRMERR is updated for each character read from the RxFIFO. 0 The current character has no framing error 1 The current character has a framing error
11 BRK	**BREAK Detect.** Indicates whether the current character was detected as a BREAK character. The data bits and the stop bit are all 0. The FRMERR bit is set when BRK is set. When odd parity is selected, PRERR is also set when BRK is set. BRK is valid for each character read from the RxFIFO. 0 The current character is not a BREAK character 1 The current character is a BREAK character
10 PRERR	**Parity Error flag**. Indicates whether the current character was detected with a parity error and is possibly corrupted. PRERR is updated for each character read from the RxFIFO. When parity is disabled, PRERR always reads as 0. 0 = No parity error was detected for data in the RX_DATA field 1 = A parity error was detected for data in the RX_DATA field
9–8 -	**Reserved**
7–0 RX_DATA	**Received Data**. Holds the received character. In 7-bit mode, the most significant bit (MSB) is forced to 0. In 8-bit mode, all bits are active.

75.3.2 UART Transmitter Register (UARTx_UTXD)

The UART will yield a transfer error on the peripheral bus when core is writing into UART_URXD register with transmit interface disabled (TXEN=0 or UARTEN=0). Memory space between UART_URXD and UART_UTXD registers is reserved. Any read or write access to this space will be considered as an invalid access and yield a transfer error.

Addresses: UART-1_UTXD is 53FB_C000h base + 40h offset = 53FB_C040h
UART-4_UTXD is 53FB_C000h base + 40h offset = 53FB_C040h
UART-2_UTXD is 53FC_0000h base + 40h offset = 53FC_0040h
UART-3_UTXD is 53FC_4000h base + 40h offset = 53FC_4040h
UART-5_UTXD is 63F9_0000h base + 40h offset = 63F9_0040h

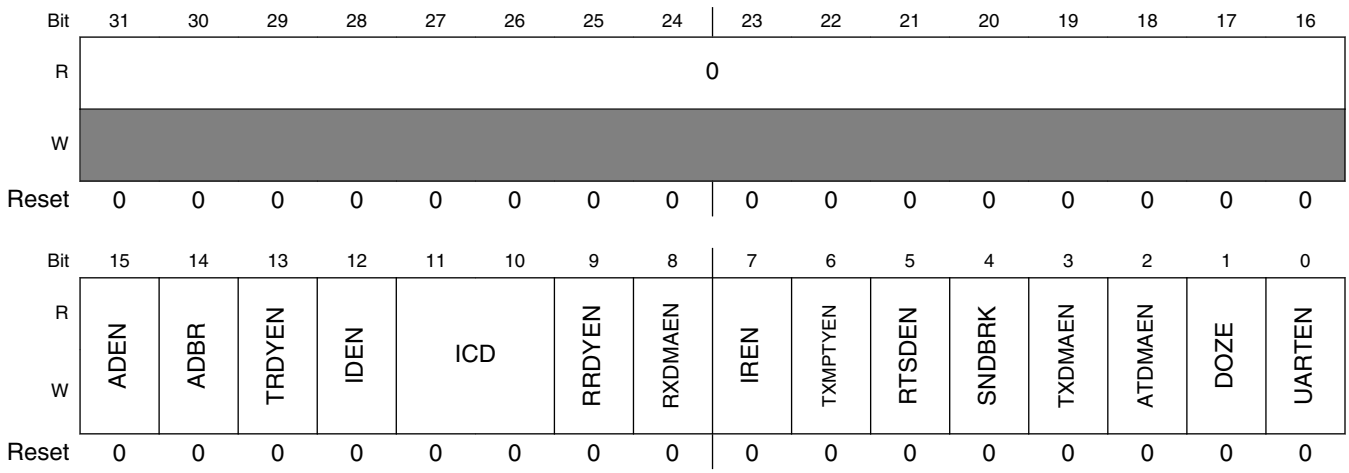


UARTx_UTXD field descriptions

Field	Description
31–16 Reserved	This read-only bitfield is reserved and always has the value zero. Reserved
15–8 Reserved	This read-only bitfield is reserved and always has the value zero. Reserved
7–0 TX_DATA	**Transmit Data**. Holds the parallel transmit data inputs. In 7-bit mode, D7 is ignored. In 8-bit mode, all bits are used. Data is transmitted least significant bit (LSB) first. A new character is transmitted when the TX_DATA field is written. The TX_DATA field must be written only when the TRDY bit is high to ensure that corrupted data is not sent.

75.3.3 UART Control Register 1 (UARTx_UCR1)

Addresses: UART-1_UCR1 is 53FB_C000h base + 80h offset = 53FB_C080h
UART-4_UCR1 is 53FB_C000h base + 80h offset = 53FB_C080h
UART-2_UCR1 is 53FC_0000h base + 80h offset = 53FC_0080h
UART-3_UCR1 is 53FC_4000h base + 80h offset = 53FC_4080h
UART-5_UCR1 is 63F9_0000h base + 80h offset = 63F9_0080h



UARTx_UCR1 field descriptions

Field	Description
31–16 Reserved	This read-only bitfield is reserved and always has the value zero. Reserved
15 ADEN	**Automatic Baud Rate Detection Interrupt Enable**. Enables/Disables the automatic baud rate detect complete (ADET) bit to generate an interrupt (*interrupt_uart* = 0). 0 Disable the automatic baud rate detection interrupt 1 Enable the automatic baud rate detection interrupt
14 ADBR	**Automatic Detection of Baud Rate**. Enables/Disables automatic baud rate detection. When the ADBR bit is set and the ADET bit is cleared, the receiver detects the incoming baud rate automatically. The ADET flag is set when the receiver verifies that the incoming baud rate is detected properly by detecting an ASCII character "A" or "a" (0x61 or 0x41). 0 Disable automatic detection of baud rate 1 Enable automatic detection of baud rate
13 TRDYEN	**Transmitter Ready Interrupt Enable**. Enables/Disables the transmitter Ready Interrupt (TRDY) when the transmitter has one or more slots available in the TxFIFO. The fill level in the TXFIFO at which an interrupt is generated is controlled by TxTL bits. When TRDYEN is negated, the transmitter ready interrupt is disabled. **NOTE:** An interrupt will be issued as long as TRDYEN and TRDY are high even if the transmitter is not enabled. In general, user should enable the transmitter before enabling the TRDY interrupt.

Table continues on the next page...

UARTx_UCR1 field descriptions (continued)

Field	Description
	0 Disable the transmitter ready interrupt 1 Enable the transmitter ready interrupt
12 IDEN	**Idle Condition Detected Interrupt Enable**. Enables/Disables the IDLE bit to generate an interrupt (*interrupt_uart* = 0). 0 Disable the IDLE interrupt 1 Enable the IDLE interrupt
11–10 ICD	**Idle Condition Detect**. Controls the number of frames RXD is allowed to be idle before an idle condition is reported. 00 Idle for more than 4 frames 01 Idle for more than 8 frames 10 Idle for more than 16 frames 11 Idle for more than 32 frames
9 RRDYEN	**Receiver Ready Interrupt Enable**. Enables/Disables the RRDY interrupt when the RxFIFO contains data. The fill level in the RxFIFO at which an interrupt is generated is controlled by the RXTL bits. When RRDYEN is negated, the receiver ready interrupt is disabled. 0 Disables the RRDY interrupt 1 Enables the RRDY interrupt
8 RXDMAEN	**Receive Ready DMA Enable**. Enables/Disables the receive DMA request *dma_req_rx* when the receiver has data in the RxFIFO. The fill level in the RxFIFO at which a DMA request is generated is controlled by the RXTL bits. When negated, the receive DMA request is disabled. 0 Disable DMA request 1 Enable DMA request
7 IREN	**Infrared Interface Enable**. Enables/Disables the IR interface. Refer to the IR interface description in Infrared Interface, for more information. 0 Disable the IR interface 1 Enable the IR interface
6 TXMPTYEN	**Transmitter Empty Interrupt Enable**. Enables/Disables the transmitter FIFO empty (TXFE) interrupt. *interrupt_uart*. When negated, the TXFE interrupt is disabled. **NOTE:** An interrupt will be issued as long as TXMPTYEN and TXFE are high even if the transmitter is not enabled. In general, user should enable the transmitter before enabling the TXFE interrupt. 0 Disable the transmitter FIFO empty interrupt 1 Enable the transmitter FIFO empty interrupt
5 RTSDEN	**RTS Delta Interrupt Enable**. Enables/Disables the RTSD interrupt. The current status of the $\overline{\text{RTS}}$ pin is read in the RTSS bit. 0 Disable RTSD interrupt 1 Enable RTSD interrupt
4 SNDBRK	**Send BREAK**. Forces the transmitter to send a BREAK character. The transmitter finishes sending the character in progress (if any) and sends BREAK characters until SNDBRK is reset. Because the transmitter samples SNDBRK after every bit is transmitted, it is important that SNDBRK is asserted high for a sufficient period of time to generate a valid BREAK. After the BREAK transmission completes, the

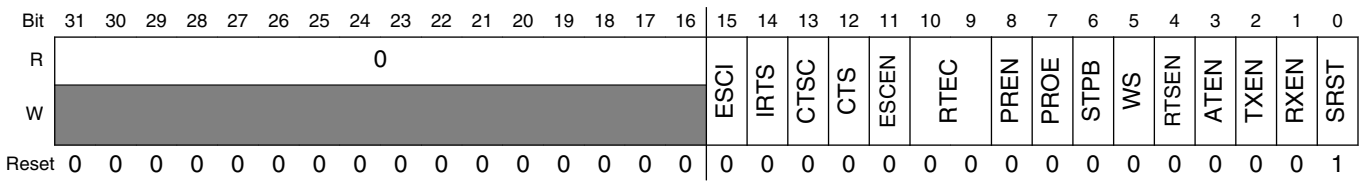
Table continues on the next page...

UARTx_UCR1 field descriptions (continued)

Field	Description
	UART transmits 2 mark bits. The user can continue to fill the TxFIFO and any characters remaining are transmitted when the BREAK is terminated. 0 Do not send a BREAK character 1 Send a BREAK character (continuous 0s)
3 TXDMAEN	**Transmitter Ready DMA Enable**. Enables/Disables the transmit DMA request *dma_req_tx* when the transmitter has one or more slots available in the TxFIFO. The fill level in the TxFIFO that generates the *dma_req_tx* is controlled by the TXTL bits. **NOTE:** A DMA request will be issued as long as TXDMAEN and TRDY are high even if the transmitter is not enabled. In general, user should enable the transmitter before enabling the transmit DMA request. 0 Disable transmit DMA request 1 Enable transmit DMA request
2 ATDMAEN	**Aging DMA Timer Enable**. Enables/Disables the receive DMA request *dma_req_rx* for the aging timer interrupt (triggered with AGTIM flag in USR1[8]). 0 Disable AGTIM DMA request 1 Enable AGTIM DMA request
1 DOZE	**DOZE**. Determines the UART enable condition in the DOZE state. When *doze_req* input pin is at '1', (the ARM Platform executes a doze instruction and the system is placed in the Doze State), the DOZE bit affects operation of the UART. While in the Doze State, if this bit is asserted, the UART is disabled. Refer to the description in Low Power Modes. 0 The UART is enabled when in DOZE state 1 The UART is disabled when in DOZE state
0 UARTEN	**UART Enable**. Enables/Disables the UART. If UARTEN is negated in the middle of a transmission, the transmitter stops and pulls the TXD line to a logic 1. UARTEN must be set to 1 before any access to UTXD and URXD registers, otherwise a transfer error is returned. This bit can be set to 1 along with other bits in this register. There is no restriction to the sequence of programing this bit and other control registers. 0 Disable the UART 1 Enable the UART

75.3.4 UART Control Register 2 (UARTx_UCR2)

Addresses: UART-1_UCR2 is 53FB_C000h base + 84h offset = 53FB_C084h
UART-4_UCR2 is 53FB_C000h base + 84h offset = 53FB_C084h
UART-2_UCR2 is 53FC_0000h base + 84h offset = 53FC_0084h
UART-3_UCR2 is 53FC_4000h base + 84h offset = 53FC_4084h
UART-5_UCR2 is 63F9_0000h base + 84h offset = 63F9_0084h



UARTx_UCR2 field descriptions

Field	Description
31–16 Reserved	This read-only bitfield is reserved and always has the value zero. Reserved
15 ESCI	**Escape Sequence Interrupt Enable**. Enables/Disables the ESCF bit to generate an interrupt. 0 Disable the escape sequence interrupt 1 Enable the escape sequence interrupt
14 IRTS	**Ignore RTS Pin**. Forces the RTS input signal presented to the transmitter to always be asserted (set to low), effectively ignoring the external pin. When in this mode, the RTS pin serves as a general purpose input. 0 Transmit only when the RTS pin is asserted 1 Ignore the RTS pin
13 CTSC	CTS **Pin Control**. Controls the operation of the $\overline{CTS}$ output pin. When CTSC is asserted, the $\overline{CTS}$ output pin is controlled by the receiver. When the RxFIFO is filled to the level of the programmed trigger level and the start bit of the overflowing character (TRIGGER LEVEL + 1) is validated, the $\overline{CTS}$ output pin is negated to indicate to the far-end transmitter to stop transmitting. When the trigger level is programmed for less than 32, the receiver continues to receive data until the RxFIFO is full. When the CTSC bit is negated, the $\overline{CTS}$ output pin is controlled by the CTS bit. On reset, because CTSC is cleared to 0, the $\overline{CTS}$ pin is controlled by the CTS bit, which again is cleared to 0 on reset. This means that on reset the $\overline{CTS}$ signal is negated. 0 The CTS pin is controlled by the CTS bit 1 The CTS pin is controlled by the receiver
12 CTS	**Clear to Send**. Controls the $\overline{CTS}$ pin when the CTSC bit is negated. CTS has no function when CTSC is asserted. 0 The CTS pin is high (inactive) 1 The CTS pin is low (active)
11 ESCEN	**Escape Enable**. Enables/Disables the escape sequence detection logic.

Table continues on the next page...

UARTx_UCR2 field descriptions (continued)

Field	Description
	0 Disable escape sequence detection 1 Enable escape sequence detection
10–9 RTEC	**Request to Send Edge Control**. Selects the edge that triggers the RTS interrupt. This has no effect on the RTS delta interrupt. RTEC has an effect only when RTSEN = 1 (see Table 75-106). 00 Trigger interrupt on a rising edge 01 Trigger interrupt on a falling edge 1X Trigger interrupt on any edge
8 PREN	**Parity Enable**. Enables/Disables the parity generator in the transmitter and parity checker in the receiver. When PREN is asserted, the parity generator and checker are enabled, and disabled when PREN is negated. 0 Disable parity generator and checker 1 Enable parity generator and checker
7 PROE	**Parity Odd/Even**. Controls the sense of the parity generator and checker. When PROE is high, odd parity is generated and expected. When PROE is low, even parity is generated and expected. PROE has no function if PREN is low. 0 Even parity 1 Odd parity
6 STPB	**Stop**. Controls the number of stop bits after a character. When STPB is low, 1 stop bit is sent. When STPB is high, 2 stop bits are sent. STPB also affects the receiver. 0 The transmitter sends 1 stop bit. The receiver expects 1 or more stop bits. 1 The transmitter sends 2 stop bits. The receiver expects 2 or more stop bits.
5 WS	**Word Size**. Controls the character length. When WS is high, the transmitter and receiver are in 8-bit mode. When WS is low, they are in 7-bit mode. The transmitter ignores bit 7 and the receiver sets bit 7 to 0. WS can be changed in-between transmission (reception) of characters, however not when a transmission (reception) is in progress, in which case the length of the current character being transmitted (received) is unpredictable. 0 7-bit transmit and receive character length (not including START, STOP or PARITY bits) 1 8-bit transmit and receive character length (not including START, STOP or PARITY bits)
4 RTSEN	**Request to Send Interrupt Enable.** Controls the RTS edge sensitive interrupt. When RTSEN is asserted and the programmed edge is detected on the $\overline{\text{RTS}}$ pin (the RTSF bit is asserted), an interrupt will be generated on the *interrupt_uart* pin. (See Table 75-106.) 0 Disable request to send interrupt 1 Enable request to send interrupt
3 ATEN	**Aging Timer Enable**. This bit is used to enable the aging timer interrupt (triggered with AGTIM) 0 AGTIM interrupt disabled 1 AGTIM interrupt enabled
2 TXEN	**Transmitter Enable**. Enables/Disables the transmitter. When TXEN is negated the transmitter is disabled and idle. When the UARTEN and TXEN bits are set the transmitter is enabled. If TXEN is negated in the middle of a transmission, the UART disables the transmitter immediately, and starts marking 1s. The transmitter FIFO cannot be written when this bit is cleared.

Table continues on the next page...

UARTx_UCR2 field descriptions (continued)

Field	Description
	0 Disable the transmitter 1 Enable the transmitter
1 RXEN	**Receiver Enable**. Enables/Disables the receiver. When the receiver is enabled, if the RXD input is already low, the receiver does not recognize BREAK characters, because it requires a valid 1-to-0 transition before it can accept any character. 0 Disable the receiver 1 Enable the receiver
0 SRST	**Software Reset**. Once the software writes 0 to $\overline{\text{SRST}}$, the software reset remains active for 4 *module_clock* cycles before the hardware deasserts $\overline{\text{SRST}}$. The software can only write 0 to $\overline{\text{SRST}}$. Writing 1 to $\overline{\text{SRST}}$ is ignored. 0 Reset the transmit and receive state machines, all FIFOs and register USR1, USR2, UBIR, UBMR, UBRC , URXD, UTXD and UTS[6-3]. 1 No reset

75.3.5 UART Control Register 3 (UARTx_UCR3)

Addresses: UART-1_UCR3 is 53FB_C000h base + 88h offset = 53FB_C088h
UART-4_UCR3 is 53FB_C000h base + 88h offset = 53FB_C088h
UART-2_UCR3 is 53FC_0000h base + 88h offset = 53FC_0088h
UART-3_UCR3 is 53FC_4000h base + 88h offset = 53FC_4088h
UART-5_UCR3 is 63F9_0000h base + 88h offset = 63F9_0088h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DPEC		DTREN	PARERREN	FRAERREN	DSR	DCD	RI	ADNIMP	RXDSEN	AIRINTEN	AWAKEN	DTRDEN	RXDMUXSEL	INVT	ACIEN
Reset	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0

UARTx_UCR3 field descriptions

Field	Description
31–16 Reserved	This read-only bitfield is reserved and always has the value zero. Reserved

Table continues on the next page...

UARTx_UCR3 field descriptions (continued)

Field	Description
15–14 DPEC	**DTR/DSR Interrupt Edge Control**. These bits control the edge of $\overline{\text{DTR}}$ (DCE) or $\overline{\text{DSR}}$ (DTE) on which an interrupt will be generated. An interrupt will only be generated if the DTREN bit is set. 00 interrupt generated on rising edge 01 interrupt generated on falling edge 1X interrupt generated on either edge
13 DTREN	**Data Terminal Ready Interrupt Enable**. When this bit is set, it will enable the status bit DTRF (USR2 [13]) (DTR/DSR edge sensitive interrupt) to cause an interrupt. 0 Data Terminal Ready Interrupt Disabled 1 Data Terminal Ready Interrupt Enabled
12 PARERREN	**Parity Error Interrupt Enable. Enables/Disables the interrupt. When asserted, PARERREN causes the PARITYERR bit to generate an interrupt.** 0 Disable the parity error interrupt 1 Enable the parity error interrupt
11 FRAERREN	**Frame Error Interrupt Enable. Enables/Disables the interrupt. When asserted, FRAERREN causes the FRAMERR bit to generate an interrupt.** 0 Disable the frame error interrupt 1 Enable the frame error interrupt
10 DSR	**Data Set Ready**. This bit is used by software to control the DSR/DTR output pin for the modem interface. In DCE mode it applies to $\overline{\text{DSR}}$ and in DTE mode it applies to $\overline{\text{DTR}}$. 0 DSR/ DTR pin is logic zero 1 DSR/ DTR pin is logic one
9 DCD	**Data Carrier Detect**. In DCE mode this bit is used by software to control the $\overline{\text{DCD}}$ output pin for the modem interface. In DTE mode, when this bit is set, it will enable the status bit DCDDELT (USR2 (6)) to cause an interrupt. 0 DCD pin is logic zero (DCE mode) 1 DCD pin is logic one (DCE mode) 0 DCDDELT interrupt disabled (DTE mode) 1 DCDDELT interrupt enabled (DTE mode)
8 RI	**Ring Indicator**. In DCE mode this bit is used by software to control the $\overline{\text{RI}}$ output pin for the modem interface. In DTE mode, when this bit is set, it will enable the status bit RIDELT (USR2 (10)) to cause an interrupt. 0 RI pin is logic zero (DCE mode) 1 RI pin is logic one (DCE mode) 0 RIDELT interrupt disabled (DTE mode) 1 RIDELT interrupt enabled (DTE mode)
7 ADNIMP	**Autobaud Detection Not Improved-. Disables new features of autobaud detection (refer to** Baud Rate Automatic Detection Protocol Improved**, for more details).** 0 Autobaud detection new features selected 1 Keep old autobaud detection mechanism
6 RXDSEN	Receive Status Interrupt Enable. Controls the receive status interrupt (*interrupt_uart*). When this bit is enabled and RXDS status bit is set, the interrupt *interrupt_uart* will be generated.

Table continues on the next page...

UARTx_UCR3 field descriptions (continued)

Field	Description
	0 Disable the RXDS interrupt 1 Enable the RXDS interrupt
5 AIRINTEN	Asynchronous IR WAKE Interrupt Enable. Controls the asynchronous IR WAKE interrupt. An interrupt is generated when AIRINTEN is asserted and a pulse is detected on the RXD pin. 0 Disable the AIRINT interrupt 1 Enable the AIRINT interrupt
4 AWAKEN	Asynchronous WAKE Interrupt Enable. Controls the asynchronous WAKE interrupt. An interrupt is generated when AWAKEN is asserted and a falling edge is detected on the RXD pin. 0 Disable the AWAKE interrupt 1 Enable the AWAKE interrupt
3 DTRDEN	**Data Terminal Ready Delta Enable**. Enables / Disables the asynchronous DTRD interrupt. When DTRDEN is asserted and an edge (rising or falling) is detected on $\overline{\text{DTR}}$ (in DCE mode) or on $\overline{\text{DSR}}$ (in DTE mode), then an interrupt is generated. 0 Disable DTRD interrupt 1 Enable DTRD interrupt
2 RXDMUXSEL	RXD Muxed Input Selected. Selects proper input pins for serial and Infrared input signal. **NOTE:** In this chip, UARTs are used in MUXED mode, so that this bit should always be set.
1 INVT	In **IrDA mode**, when INVT is cleared, the infrared logic block transmits a positive IR 3/16 pulse for all 0s and 0s are transmitted for 1s. When INVT is set (INVT = 1), the infrared logic block transmits an active low or negative infrared 3/16 pulse for all 0s and 1s are transmitted for 1s. — — 0 TXDActive low transmission 1 TXD Active high transmission
0 ACIEN	**Autobaud Counter Interrupt Enable.** This bit is used to enable the autobaud counter stopped interrupt (triggered with ACST (USR2[11]). 0 ACST interrupt disabled 1 ACST interrupt enabled

75.3.6 UART Control Register 4 (UARTx_UCR4)

Addresses: UART-1_UCR4 is 53FB_C000h base + 8Ch offset = 53FB_C08Ch
UART-4_UCR4 is 53FB_C000h base + 8Ch offset = 53FB_C08Ch
UART-2_UCR4 is 53FC_0000h base + 8Ch offset = 53FC_008Ch
UART-3_UCR4 is 53FC_4000h base + 8Ch offset = 53FC_408Ch
UART-5_UCR4 is 63F9_0000h base + 8Ch offset = 63F9_008Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CTSTL						INVR	ENIRI	WKEN	IDDMAEN	IRSC	LPBYP	TCEN	BKEN	OREN	DREN
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UARTx_UCR4 field descriptions

Field	Description
31–16 Reserved	This read-only bitfield is reserved and always has the value zero. Reserved
15–10 CTSTL	**CTS Trigger Level**. Controls the threshold at which the $\overline{\text{CTS}}$ pin is deasserted by the RxFIFO. After the trigger level is reached and the $\overline{\text{CTS}}$ pin is deasserted, the RxFIFO continues to receive data until it is full. The CTSTL bits are encoded as shown in the Settings column. All Other Settings Reserved 000000 0 characters received 000001 1 characters in the RxFIFO 100000 32 characters in the RxFIFO (maximum)
9 INVR	**In IrDA mode(UMCR[0] = 0), w**hen cleared, the infrared logic block expects an active low or negative IR 3/16 pulse for 0s and 1s are expected for 1s. When INVR is set (INVR 1), the infrared logic block expects an active high or positive IR 3/16 pulse for 0s and 0s are expected for 1s. — — 0 RXD active low detection 1 RXD active high detection
8 ENIRI	**Serial Infrared Interrupt Enable**. Enables/Disables the serial infrared interrupt. 0 Serial infrared Interrupt disabled 1 Serial infrared Interrupt enabled

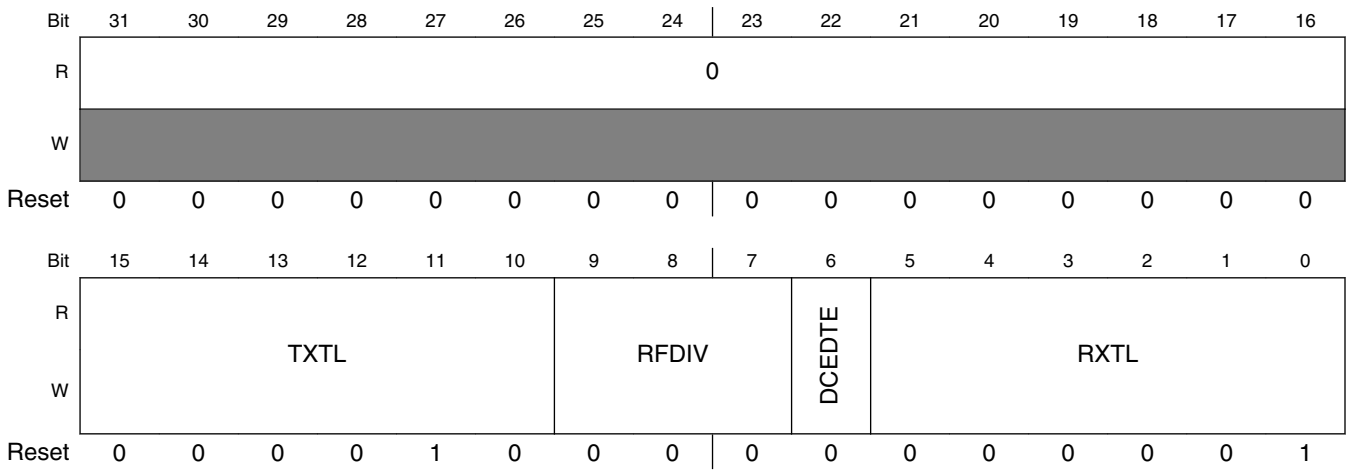
Table continues on the next page...

UARTx_UCR4 field descriptions (continued)

Field	Description
7 WKEN	**WAKE Interrupt Enable**. Enables/Disables the WAKE bit to generate an interrupt. The WAKE bit is set at the detection of a start bit by the receiver. 0 Disable the WAKE interrupt 1 Enable the WAKE interrupt
6 IDDMAEN	**DMA IDLE Condition Detected Interrupt Enable** Enables/Disables the receive DMA request *dma_req_rx* for the IDLE interrupt (triggered with IDLE flag in USR2[12]). 0 DMA IDLE interrupt disabled 1 DMA IDLE interrupt enabled
5 IRSC	**IR Special Case**. Selects the clock for the vote logic. When set, IRSC switches the vote logic clock from the sampling clock to the UART reference clock. The IR pulses are counted a predetermined amount of time depending on the reference frequency. Refer to InfraRed Special Case (IRSC) Bit. 0 The vote logic uses the sampling clock (16x baud rate) for normal operation 1 The vote logic uses the UART reference clock
4 LPBYP	**Low Power Bypass**. Allows to bypass the low power new features in UART. To use during debug phase. 0 Low power features enabled 1 Low power features disabled
3 TCEN	**TransmitComplete Interrupt Enable**. Enables/Disables the TXDC bit to generate an interrupt (*interrupt_uart* = 0) **NOTE:** An interrupt will be issued as long as TCEN and TXDC are high even if the transmitter is not enabled. In general, user should enable the transmitter before enabling the TXDC interrupt. 0 Disable TXDC interrupt 1 Enable TXDC interrupt
2 BKEN	**BREAK Condition Detected Interrupt Enable**. Enables/Disables the BRCD bit to generate an interrupt. 0 Disable the BRCD interrupt 1 Enable the BRCD interrupt
1 OREN	**Receiver Overrun Interrupt Enable**. Enables/Disables the ORE bit to generate an interrupt. 0 Disable ORE interrupt 1 Enable ORE interrupt
0 DREN	**Receive Data Ready Interrupt Enable**. Enables/Disables the RDR bit to generate an interrupt. 0 Disable RDR interrupt 1 Enable RDR interrupt

75.3.7 UART FIFO Control Register (UARTx_UFCR)

Addresses: UART-1_UFCR is 53FB_C000h base + 90h offset = 53FB_C090h
UART-4_UFCR is 53FB_C000h base + 90h offset = 53FB_C090h
UART-2_UFCR is 53FC_0000h base + 90h offset = 53FC_0090h
UART-3_UFCR is 53FC_4000h base + 90h offset = 53FC_4090h
UART-5_UFCR is 63F9_0000h base + 90h offset = 63F9_0090h



UARTx_UFCR field descriptions

Field	Description
31–16 Reserved	This read-only bitfield is reserved and always has the value zero. Reserved
15–10 TXTL	**Transmitter Trigger Level**. Controls the threshold at which a maskable interrupt is generated by the TxFIFO. A maskable interrupt is generated whenever the data level in the TxFIFO falls below the selected threshold. The bits are encoded as shown in the Settings column. All Other Settings Reserved 000000 Reserved 000001 Reserved 000010 TxFIFO has 2 or fewer characters 011111 TxFIFO has 31 or fewer characters 100000 TxFIFO has 32 characters (maximum)
9–7 RFDIV	Reference Frequency Divider. Controls the divide ratio for the reference clock. The input clock is *module_clock*. The output from the divider is *ref_clk* which is used by BRM to create the 16x baud rate oversampling clock (*brm_clk*). 000 Divide input clock by 6 001 Divide input clock by 5 010 Divide input clock by 4 011 Divide input clock by 3 100 Divide input clock by 2 101 Divide input clock by 1

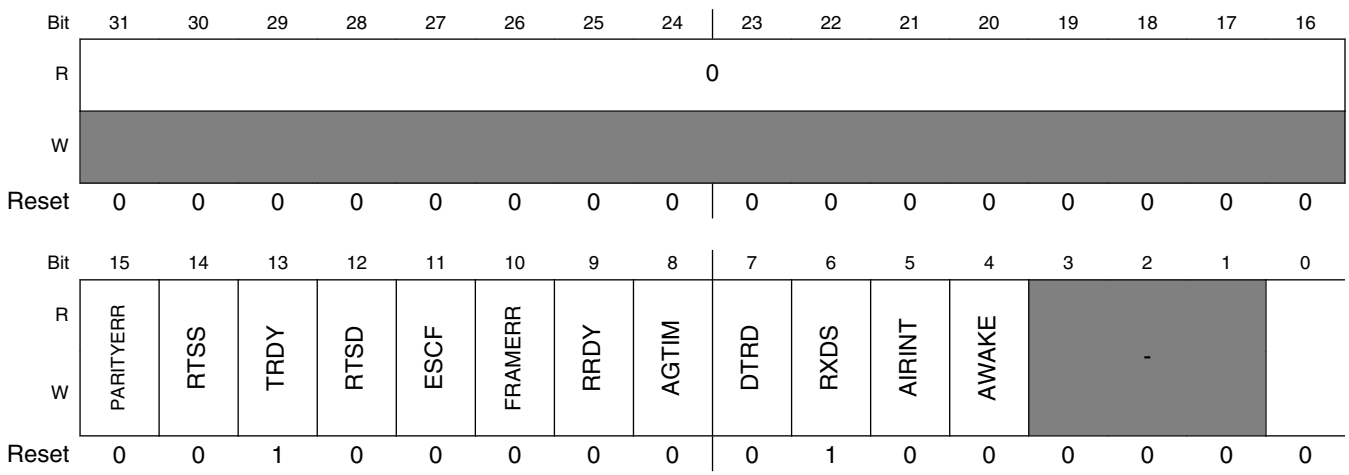
Table continues on the next page...

UARTx_UFCR field descriptions (continued)

Field	Description
	110 Divide input clock by 7 111 Reserved
6 DCEDTE	**DCE/DTE mode select**. Select UART as data communication equipment (DCE mode) or as data terminal equipment (DTE mode). 0 DCE mode selected 1 DTE mode selected
5–0 RXTL	**Receiver Trigger Level**. Controls the threshold at which a maskable interrupt is generated by the RxFIFO. A maskable interrupt is generated whenever the data level in the RxFIFO reaches the selected threshold. The RXTL bits are encoded as shown in the Settings column. All Other Settings Reserved 000000 0 characters received 000001 RxFIFO has 1 character 011111 RxFIFO has 31 characters 100000 RxFIFO has 32 characters (maximum)

75.3.8 UART Status Register 1 (UARTx_USR1)

Addresses: UART-1_USR1 is 53FB_C000h base + 94h offset = 53FB_C094h
UART-4_USR1 is 53FB_C000h base + 94h offset = 53FB_C094h
UART-2_USR1 is 53FC_0000h base + 94h offset = 53FC_0094h
UART-3_USR1 is 53FC_4000h base + 94h offset = 53FC_4094h
UART-5_USR1 is 63F9_0000h base + 94h offset = 63F9_0094h



UARTx_USR1 field descriptions

Field	Description
31–16 Reserved	This read-only bitfield is reserved and always has the value zero. Reserved
15 PARITYERR	**Parity Error Interrupt Flag**. Indicates a parity error is detected. PARITYERR is cleared by writing 1 to it. Writing 0 to PARITYERR has no effect. When parity is disabled, PARITYERR always reads 0. At reset, PARITYERR is set to 0. 0 No parity error detected 1 Parity error detected (write 1 to clear)
14 RTSS	$\overline{\text{RTS}}$ **Pin Status**. Indicates the current status of the $\overline{\text{RTS}}$ pin. A "snapshot" of the pin is taken immediately before RTSS is presented to the data bus. RTSS cannot be cleared because all writes to RTSS are ignored. At reset, RTSS is set to 0. 0 The RTS pin is high (inactive) 1 The RTS pin is low (active)
13 TRDY	**Transmitter Ready Interrupt / DMA Flag**. Indicates that the TxFIFO emptied below its target threshold and requires data. TRDY is automatically cleared when the data level in the TxFIFO exceeds the threshold set by TXTL bits. At reset, TRDY is set to 1. 0 The transmitter does not require data 1 The transmitter requires data (interrupt posted)
12 RTSD	**RTS Delta.** Indicates whether the $\overline{\text{RTS}}$ pin changed state. It (RTSD) generates a maskable interrupt. When in STOP mode, RTS assertion sets RTSD and can be used to wake the processor. The current state of the $\overline{\text{RTS}}$ pin is available on the RTSS bit. Clear RTSD by writing 1 to it. Writing 0 to RTSD has no effect. At reset, RTSD is set to 0. 0 RTS pin did not change state since last cleared 1 RTS pin changed state (write 1 to clear)
11 ESCF	**Escape Sequence Interrupt Flag**. Indicates if an escape sequence was detected. ESCF is asserted when the ESCEN bit is set and an escape sequence is detected in the RxFIFO. Clear ESCF by writing 1 to it. Writing 0 to ESCF has no effect. 0 No escape sequence detected 1 Escape sequence detected (write 1 to clear).
10 FRAMERR	**Frame Error Interrupt Flag**. Indicates that a frame error is detected. The *interrupt_uart* interrupt will be generated if a frame error is detected and the interrupt is enabled. Clear FRAMERR by writing 1 to it. Writing 0 to FRAMERR has no effect. 0 No frame error detected 1 Frame error detected (write 1 to clear)
9 RRDY	**Receiver Ready Interrupt / DMA Flag**. Indicates that the RxFIFO data level is above the threshold set by the RXTL bits. (See the RXTL bits description in UART FIFO Control Register (UART_UFCR) for setting the interrupt threshold.) When asserted, RRDY generates a maskable interrupt or DMA request. RRDY is automatically cleared when data level in the RxFIFO goes below the set threshold level. At reset, RRDY is set to 0. 0 No character ready 1 Character(s) ready (interrupt posted)
8 AGTIM	Ageing Timer Interrupt Flag. Indicates that data in the RxFIFO has been idle for a time of 8 character lengths (where a character length consists of 7 or 8 bits, depending on the setting of the WS bit in UCR2,

Table continues on the next page...

UARTx_USR1 field descriptions (continued)

Field	Description
	with the bit time corresponding to the baud rate setting) and FIFO data level is less than RxFIFO threshold level (RXTL in the UFCR). Clear by writing a 1 to it. 0 AGTIM is not active 1 AGTIM is active (write 1 to clear)
7 DTRD	**DTR Delta.** Indicates whether $\overline{\text{DTR}}$ (in DCE mode) or $\overline{\text{DSR}}$ (in DTE mode) pins changed state. DTRD generates a maskable interrupt if DTRDEN (UCR3[3]) is set. Clear DTRD by writing 1 to it. Writing 0 to DTRD has no effect. 0 DTR (DCE) or DSR (DTE) pin did not change state since last cleared 1 DTR (DCE) or DSR (DTE) pin changed state (write 1 to clear)
6 RXDS	Receiver IDLE Interrupt Flag. Indicates that the receiver state machine is in an IDLE state, the next state is IDLE, and the receive pin is high. RXDS is automatically cleared when a character is received. RXDS is active only when the receiver is enabled. 0 Receive in progress 1 Receiver is IDLE
5 AIRINT	Asynchronous IR WAKE Interrupt Flag. Indicates that the IR WAKE pulse was detected on the RXD pin. Clear AIRINT by writing 1 to it. Writing 0 to AIRINT has no effect. 0 No pulse was detected on the RXD IrDA pin 1 A pulse was detected on the RXD IrDA pin
4 AWAKE	Asynchronous WAKE Interrupt Flag. Indicates that a falling edge was detected on the RXD pin. Clear AWAKE by writing 1 to it. Writing 0 to AWAKE has no effect. 0 No falling edge was detected on the RXD Serial pin 1 A falling edge was detected on the RXD Serial pin
2–0 -	Reserved

75.3.9 UART Status Register 2 (UARTx_USR2)

Addresses: UART-1_USR2 is 53FB_C000h base + 98h offset = 53FB_C098h
UART-4_USR2 is 53FB_C000h base + 98h offset = 53FB_C098h
UART-2_USR2 is 53FC_0000h base + 98h offset = 53FC_0098h
UART-3_USR2 is 53FC_4000h base + 98h offset = 53FC_4098h
UART-5_USR2 is 63F9_0000h base + 98h offset = 63F9_0098h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADET	TXFE	DTRF	IDLE	ACST	RIDELT	RIIN	IRINT	WAKE	DCDDELT	DCDIN	RTSF	TXDC	BRCD	ORE	RDR
W																
Reset	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0

UARTx_USR2 field descriptions

Field	Description
31–16 Reserved	This read-only bitfield is reserved and always has the value zero. Reserved
15 ADET	**Automatic Baud Rate Detect Complete**. Indicates that an "A" or "a" was received and that the receiver detected and verified the incoming baud rate. Clear ADET by writing 1 to it. Writing 0 to ADET has no effect. 0 ASCII "A" or "a" was not received 1 ASCII "A" or "a" was received (write 1 to clear)
14 TXFE	**Transmit Buffer FIFO Empty**. Indicates that the transmit buffer (TxFIFO) is empty. TXFE is cleared automatically when data is written to the TxFIFO. Even though TXFE is high, the transmission might still be in progress. 0 The transmit buffer (TxFIFO) is not empty 1 The transmit buffer (TxFIFO) is empty
13 DTRF	**DTR edge triggered interrupt flag**. This bit is asserted, when the programmed edge is detected on the $\overline{\text{DTR}}$ pin (DCE mode) or on $\overline{\text{DSR}}$ (DTE mode). This flag can cause an interrupt if DTREN (UCR3[13]) is enabled. 0 Programmed edge not detected on DTR/DSR 1 Programmed edge detected on DTR/DSR (write 1 to clear)

Table continues on the next page...

UARTx_USR2 field descriptions (continued)

Field	Description
12 IDLE	**Idle Condition**. Indicates that an idle condition has existed for more than a programmed amount frame (refer to Idle Line Detect. An interrupt can be generated by this IDLE bit if IDEN (UCR1[12]) is enabled. IDLE is cleared by writing 1 to it. Writing 0 to IDLE has no effect. 0 No idle condition detected 1 Idle condition detected (write 1 to clear)
11 ACST	**Autobaud Counter Stopped**. In autobaud detection (ADBR=1), indicates the counter which determines the baud rate was running and is now stopped. This means either START bit is finished (if ADNIMP=1), or Bit 0 is finished (if ADNIMP=0). Refer to New Autobaud Counter Stopped bit and Interrupt, for more details. An interrupt can be flagged on *interrupt_uart* if ACIEN=1. 0 Measurement of bit length not finished (in autobaud) 1 Measurement of bit length finished (in autobaud). (write 1 to clear)
10 RIDELT	**Ring Indicator Delta**. This bit is used in DTE mode to indicate that the Ring Indicator input ($\overline{RI}$) has changed state. This flag can generate an interrupt if RI (UCR3[8]) is enabled. RIDELT is cleared by writing 1 to it. Writing 0 to RIDELT has no effect. 0 Ring Indicator input has not changed state 1 Ring Indicator input has changed state (write 1 to clear)
9 RIIN	**Ring Indicator Input**. This bit is used in DTE mode to reflect the status if the Ring Indicator input ($\overline{RI}$). The Ring Indicator input is used to indicate that a ring has occurred. In DCE mode this bit is always zero. 0 Ring Detected 1 No Ring Detected
8 IRINT	**Serial Infrared Interrupt Flag.** When an edge is detected on the RXD pin during SIR Mode, this flag will be asserted. This flag can cause an interrupt which can be masked using the control bit ENIRI: UCR4 [8]. 0 no edge detected 1 valid edge detected (write 1 to clear)
7 WAKE	**Wake**. Indicates the start bit is detected. WAKE can generate an interrupt that can be masked using the WKEN bit. Clear WAKE by writing 1 to it. Writing 0 to WAKE has no effect. 0 start bit not detected 1 start bit detected (write 1 to clear)
6 DCDDELT	**Data Carrier Detect Delta**. This bit is used in DTE mode to indicate that the Data Carrier Detect input ($\overline{DCD}$) has changed state. This flag can cause an interrupt if DCD (UCR3[9]) is enabled. When in STOP mode, this bit can be used to wake the processor. In DCE mode this bit is always zero. 0 Data Carrier Detect input has not changed state 1 Data Carrier Detect input has changed state (write 1 to clear)
5 DCDIN	**Data Carrier Detect Input**. This bit is used in DTE mode reflect the status of the Data Carrier Detect input ($\overline{DCD}$). The Data Carrier Detect input is used to indicate that a carrier signal has been detected. In DCE mode this bit is always zero. 0 Carrier signal Detected 1 No Carrier signal Detected
4 RTSF	**RTS Edge Triggered Interrupt Flag. Indicates if** a programmed edge is detected on the $\overline{RTS}$ pin. The RTEC bits select the edge that generates an interrupt (see Table 75-106). RTSF can generate an

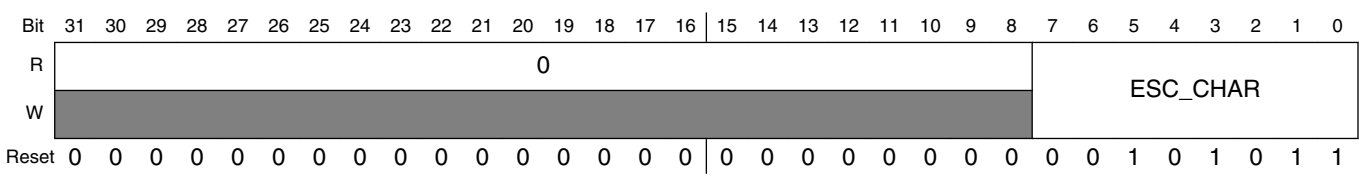
Table continues on the next page...

UARTx_USR2 field descriptions (continued)

Field	Description
	interrupt that can be masked using the RTSEN bit. Clear RTSF by writing 1 to it. Writing 0 to RTSF has no effect. 0 Programmed edge not detected on RTS 1 Programmed edge detected on RTS (write 1 to clear)
3 TXDC	**Transmitter Complete**. Indicates that the transmit buffer (TxFIFO) and Shift Register is empty; therefore the transmission is complete. TXDC is cleared automatically when data is written to the TxFIFO. 0 Transmit is incomplete 1 Transmit is complete
2 BRCD	**BREAK Condition Detected**. Indicates that a BREAK condition was detected by the receiver. Clear BRCD by writing 1 to it. Writing 0 to BRCD has no effect. 0 No BREAK condition was detected 1 A BREAK condition was detected (write 1 to clear)
1 ORE	**Overrun Error**. When set to 1, ORE indicates that the receive buffer (RxFIFO) was full (32 chars inside), and a 33rd character has been fully received. This 33rd character has been discarded. Clear ORE by writing 1 to it. Writing 0 to ORE has no effect. 0 No overrun error 1 Overrun error (write 1 to clear)
0 RDR	**Receive Data Ready**-Indicates that at least 1 character is received and written to the RxFIFO. If the URXD register is read and there is only 1 character in the RxFIFO, RDR is automatically cleared. 0 No receive data ready 1 Receive data ready

75.3.10 UART Escape Character Register (UARTx_UESC)

Addresses: UART-1_UESC is 53FB_C000h base + 9Ch offset = 53FB_C09Ch
UART-4_UESC is 53FB_C000h base + 9Ch offset = 53FB_C09Ch
UART-2_UESC is 53FC_0000h base + 9Ch offset = 53FC_009Ch
UART-3_UESC is 53FC_4000h base + 9Ch offset = 53FC_409Ch
UART-5_UESC is 63F9_0000h base + 9Ch offset = 63F9_009Ch



UARTx_UESC field descriptions

Field	Description
31–8 Reserved	This read-only bitfield is reserved and always has the value zero. Reserved
7–0 ESC_CHAR	**UART Escape Character**. Holds the selected escape character that all received characters are compared against to detect an escape sequence.

75.3.11 UART Escape Timer Register (UARTx_UTIM)

Addresses: UART-1_UTIM is 53FB_C000h base + A0h offset = 53FB_C0A0h
UART-4_UTIM is 53FB_C000h base + A0h offset = 53FB_C0A0h
UART-2_UTIM is 53FC_0000h base + A0h offset = 53FC_00A0h
UART-3_UTIM is 53FC_4000h base + A0h offset = 53FC_40A0h
UART-5_UTIM is 63F9_0000h base + A0h offset = 63F9_00A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TIM															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UARTx_UTIM field descriptions

Field	Description
31–12 Reserved	This read-only bitfield is reserved and always has the value zero. Reserved
11–0 TIM	UART Escape Timer. Holds the maximum time interval (in ms) allowed between escape characters. The escape timer register is programmable in intervals of 2 ms. Refer to Escape Sequence Detection and Table 75-111 for more information on the UART escape sequence detection. Reset value 0x000 = 2 ms up to 0xFFF = 8.192 s.

75.3.12 UART BRM Incremental Register (UARTx_UBIR)

This register can be written by both software and hardware. When enabling the automatic baud rate detection feature hardware can write 0x000F value into the UBIR after finishing detecting baud rate. Hardware has higher priority when both software and hardware try to write it at the same cycle[1].

Please note software reset will reset the register to its reset value.

Addresses: UART-1_UBIR is 53FB_C000h base + A4h offset = 53FB_C0A4h
UART-4_UBIR is 53FB_C000h base + A4h offset = 53FB_C0A4h
UART-2_UBIR is 53FC_0000h base + A4h offset = 53FC_00A4h
UART-3_UBIR is 53FC_4000h base + A4h offset = 53FC_40A4h
UART-5_UBIR is 63F9_0000h base + A4h offset = 63F9_00A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																INC															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UARTx_UBIR field descriptions

Field	Description
31–16 Reserved	This read-only bitfield is reserved and always has the value zero. Reserved
15–0 INC	Incremental Numerator. Holds the numerator value minus one of the BRM ratio (refer to Binary Rate Multiplier (BRM)). The UBIR register MUST be updated before the UBMR register for the baud rate to be updated correctly. If only one register is written to by software, the BRM will ignore this data until the other register is written to by software. Updating this field using byte accesses is not recommended and is undefined.

1. Note: The write priority in the new design is not same as the original UART. In the orginal design, software has higher priotiry than hardware when writing this register at the same time.

75.3.13 UART BRM Modulator Register (UARTx_UBMR)

This register can be written by both software and hardware. When enabling the automatic baud rate detection feature hardware can write a proper value into the UBMR based on detected baud rate. Hardware has higher priority when both software and hardware try to write it at the same cycle[1].

Please note software reset will reset the register to its reset value.

Addresses: UART-1_UBMR is 53FB_C000h base + A8h offset = 53FB_C0A8h
UART-4_UBMR is 53FB_C000h base + A8h offset = 53FB_C0A8h
UART-2_UBMR is 53FC_0000h base + A8h offset = 53FC_00A8h
UART-3_UBMR is 53FC_4000h base + A8h offset = 53FC_40A8h
UART-5_UBMR is 63F9_0000h base + A8h offset = 63F9_00A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																MOD															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UARTx_UBMR field descriptions

Field	Description
31–16 Reserved	This read-only bitfield is reserved and always has the value zero. Reserved
15–0 MOD	**Modulator Denominator.** Holds the value of the denominator minus one of the BRM ratio (refer to Binary Rate Multiplier (BRM)). The UBIR register MUST be updated before the UBMR register for the baud rate to be updated correctly. If only one register is written to by software, the BRM will ignore this data until the other register is written to by software. Updating this register using byte accesses is not recommended and undefined.

1. Note: The write priority in the new design is not same as the original UART. In the orginal design, software has higher priotiry than hardware when writing this register at the same time.

75.3.14 UART Baud Rate Count Register (UARTx_UBRC)

Addresses: UART-1_UBRC is 53FB_C000h base + ACh offset = 53FB_C0ACh
UART-4_UBRC is 53FB_C000h base + ACh offset = 53FB_C0ACh
UART-2_UBRC is 53FC_0000h base + ACh offset = 53FC_00ACh
UART-3_UBRC is 53FC_4000h base + ACh offset = 53FC_40ACh
UART-5_UBRC is 63F9_0000h base + ACh offset = 63F9_00ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																BCNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

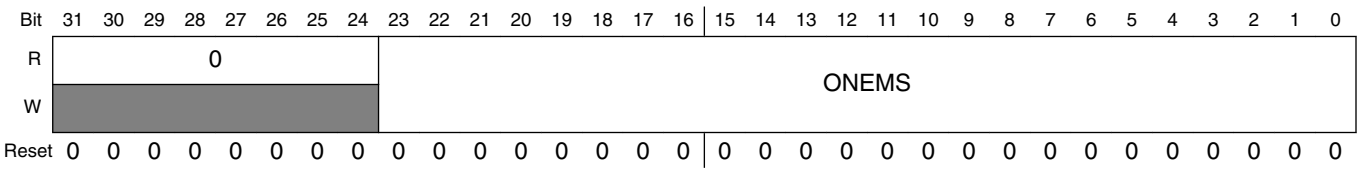
UARTx_UBRC field descriptions

Field	Description
31–16 Reserved	This read-only bitfield is reserved and always has the value zero. Reserved
15–0 BCNT	**Baud Rate Count Register.** This read only register is used to count the start bit of the incoming baud rate (if ADNIMP=1), or start bit + bit0 (if ADNIMP=0). When the measurement is done, the Baud Rate Count Register contains the number of UART internal clock cycles (clock after divider) present in an incoming bit. BCNT retains its value until the next Automatic Baud Rate Detection sequence has been initiated. The 16 bit Baud Rate Count register is reset to 4 and stays at hex FFFF in the case of an overflow.

75.3.15 UART One Millisecond Register (UARTx_ONEMS)

This register has been expanded from 16 bits to 24 bits. In previous versions, the 16-bit ONEMS can only support the maximum 65.535MHz (0xFFFFx1000) ref_clk. To support 4Mbps Bluetooth application with 66.5MHz module_clock, the value 0x103C4 (66.5M/1000) should be written into this register. In this case, the 16 bits are not enough to contain the 0x103C4. So this register was expanded to 24 bits to support high frequency of the ref_clk.

Addresses: UART-1_ONEMS is 53FB_C000h base + B0h offset = 53FB_C0B0h
UART-4_ONEMS is 53FB_C000h base + B0h offset = 53FB_C0B0h
UART-2_ONEMS is 53FC_0000h base + B0h offset = 53FC_00B0h
UART-3_ONEMS is 53FC_4000h base + B0h offset = 53FC_40B0h
UART-5_ONEMS is 63F9_0000h base + B0h offset = 63F9_00B0h

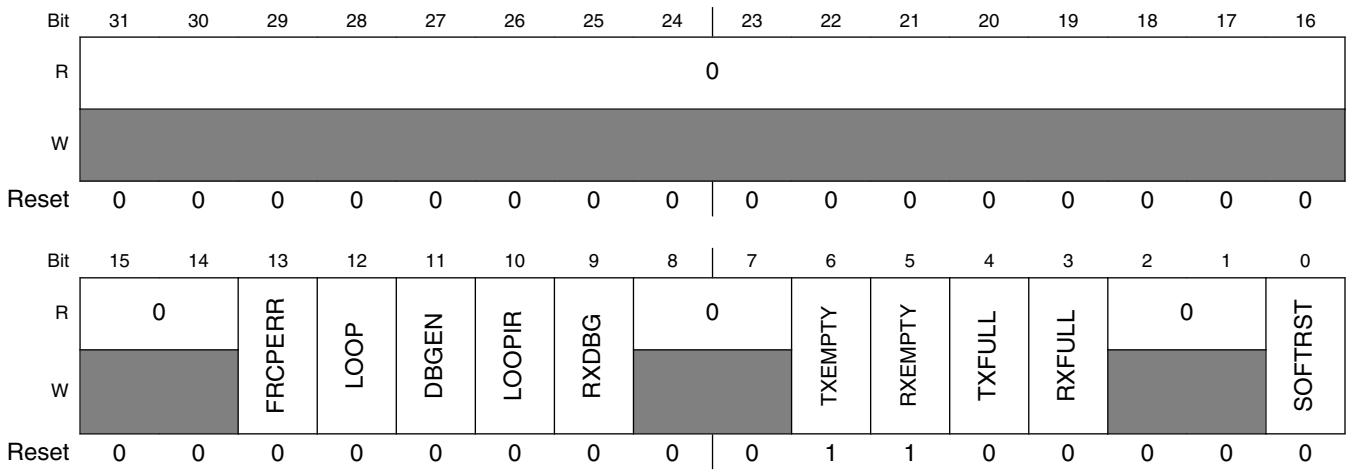


UARTx_ONEMS field descriptions

Field	Description
31–24 Reserved	This read-only bitfield is reserved and always has the value zero. Reserved
23–0 ONEMS	**One Millisecond Register.** This 24-bit register must contain the value of the UART internal frequency (*ref_clk* in Figure 75-1) divided by 1000. The internal frequency is obtained after the UART BRM internal divider (F (*ref_clk*) = F(*module_clock*) / RFDIV). In fact this register contains the value corresponding to the number of UART BRM internal clock cycles present in one millisecond. The ONEMS (and UTIM) registers value are used in the escape character detection feature (Escape Sequence Detection) to count the number of clock cycles left between two escape characters. The ONEMS register is also used in infrared special case mode (IRSC = UCR4[5] = 1'b1), refer to InfraRed Special Case (IRSC) Bit.

75.3.16 UART Test Register (UARTx_UTS)

Addresses: UART-1_UTS is 53FB_C000h base + B4h offset = 53FB_C0B4h
UART-4_UTS is 53FB_C000h base + B4h offset = 53FB_C0B4h
UART-2_UTS is 53FC_0000h base + B4h offset = 53FC_00B4h
UART-3_UTS is 53FC_4000h base + B4h offset = 53FC_40B4h
UART-5_UTS is 63F9_0000h base + B4h offset = 63F9_00B4h



UARTx_UTS field descriptions

Field	Description
31–14 Reserved	This read-only bitfield is reserved and always has the value zero. Reserved
13 FRCPERR	Force Parity Error. Forces the transmitter to generate a parity error if parity is enabled. FRCPERR is provided for system debugging. 0 Generate normal parity 1 Generate inverted parity (error)
12 LOOP	Loop TX and RX for Test. Controls loopback for test purposes. When LOOP is high, the receiver input is internally connected to the transmitter and ignores the RXD pin. The transmitter is unaffected by LOOP. If RXDMUXSEL (UCR3[2]) is set to 1, the loopback is applied on serial and IrDA signals. If RXDMUXSEL is set to 0, the loopback is only applied on serial signals. 0 Normal receiver operation 1 Internally connect the transmitter output to the receiver input
11 DBGEN	$\overline{\text{debug\_enable}}$**.** This bit controls whether to respond to the *debug_req* input signal. 0 UART will go into debug mode when debug_req is HIGH 1 UART will not go into debug mode even if debug_req is HIGH
10 LOOPIR	**Loop TX and RX for IR Test (LOOPIR)**. This bit controls loopback from transmitter to receiver in the InfraRed interface.

Table continues on the next page...

UARTx_UTS field descriptions (continued)

Field	Description
	0 No IR loop 1 Connect IR transmitter to IR receiver
9 RXDBG	**RX_fifo_debug_mode.** This bit controls the operation of the RX fifo read counter when in debug mode. 0 rx fifo read pointer does not increment 1 rx_fifo read pointer increments as normal
8–7 Reserved	This read-only bitfield is reserved and always has the value zero. Reserved
6 TXEMPTY	TxFIFO Empty. Indicates that the TxFIFO is empty. 0 The TxFIFO is not empty 1 The TxFIFO is empty
5 RXEMPTY	RxFIFO Empty. Indicates the RxFIFO is empty. 0 The RxFIFO is not empty 1 The RxFIFO is empty
4 TXFULL	TxFIFO FULL. Indicates the TxFIFO is full. 0 The TxFIFO is not full 1 The TxFIFO is full
3 RXFULL	RxFIFO FULL. Indicates the RxFIFO is full. 0 The RxFIFO is not full 1 The RxFIFO is full
2–1 Reserved	This read-only bitfield is reserved and always has the value zero. Reserved
0 SOFTRST	Software Reset. Indicates the status of the software reset ($\overline{\text{SRST}}$ bit of UCR2). 0 Software reset inactive 1 Software reset active

75.4 Functional Description

This section provides a complete functional description of the block.

75.4.1 Interrupts and DMA Requests

See Table 75-105 for the lists of all interrupt and DMA signals and associated interrupt and DMA sources of the UART. See register description section for explanation of interrupt/DMA enable and status.

Table 75-105. Interrupts and DMA

Interrupt/DMA Output	Interrupt/DMA Enable	Enable Register Location	Interrupt/DMA Flag	Flag Register Location
interrupt_uart	RRDYEN	UCR1 (bit 9)	RRDY	USR1 (bit 9)
	IDEN	UCR1 (bit 12)	IDLE	USR2 (bit 12)
	DREN	UCR4 (bit 0)	RDR	USR2 (bit 0)
	RXDSEN	UCR3 (bit 6)	RXDS	USR1 (bit 6)
	ATEN	UCR2 (bit 3)	AGTIM	USR1 (bit 8)
interrupt_uart	TXMPTYEN	UCR1 (bit 6)	TXFE	USR2 (bit 14)
	TRDYEN	UCR1 (bit 13)	TRDY	USR1 (bit 13)
	TCEN	UCR4 (bit 3)	TXDC	USR2 (bit 3)
interrupt_uart	OREN	UCR4 (bit 1)	ORE	USR2 (bit 1)
	BKEN	UCR4 (bit 2)	BRCD	USR2 (bit 2)
	WKEN	UCR4 (bit 7)	WAKE	USR2 (bit 7)
	ADEN	UCR1 (bit 15)	ADET	USR2 (bit 15)
	ACIEN	UCR3 (bit 0)	ACST	USR2 (bit 11)
	ESCI	UCR2 (bit 15)	ESCF	USR1 (bit 11)
	ENIRI	UCR4 (bit 8)	IRINT	USR2 (bit 8)
	AIRINTEN	UCR3 (bit 5)	AIRINT	USR1 (bit 5)
	AWAKEN	UCR3 (bit 4)	AWAKE	USR1 (bit 4)
	FRAERREN	UCR3 (bit 11)	FRAERR	USR1 (bit 10)
	PARERREN	UCR3 (bit 12)	PARITYERR	USR1 (bit 15)
	RTSDEN	UCR1 (bit 5)	RTSD	USR1 (bit 12)
	RTSEN	UCR2 (bit 4)	RTSF	USR2 (bit 4)
	DTREN (DCE)	UCR3 (bit 13)	DTRF	USR2 (bit 13)
	RI (DTE)	UCR3 (bit 8)	RIDELT	USR2 (bit 10)
	DCD (DTE)	UCR3 (bit 9)	DCDDELT	USR2 (bit 6)
	DTRDEN	UCR3 (bit 3)	DTRD	USR1 (bit 7)
dma_req_rx	RXDMAEN	UCR1 (bit 8)	RRDY	USR1 (bit 9)
	ATDMAEN	UCR1 (bit 2)	AGTIM	USR1 (bit 8)
	IDDMAEN	UCR4 (bit 6)	IDLE	USR2 (bit 12)
dma_req_tx	TXDMAEN	UCR1 (bit 3)	TRDY	USR1 (bit 13)

75.4.2 Clocks

This section describes clocks and special clocking requirements of the UART.

75.4.2.1 Clock requirements

UART module receives 2 clocks, *peripheral_clock* and *module_clock*. The *peripheral_clock* is used as write clock of the TxFIFO, read clock of the RxFIFO and synchronization of the modem control input pins. It must always be running when UART is enabled. There is an exception in stop mode (see Clocking in Low-Power Modes).

The *module_clock* is for all the state machines, writing RxFIFO, reading TxFIFO, etc. It must always be running when UART is sending or receiving characters.This clock is used in order to allow frequency scaling on *peripheral_clock* without changing configuration of baud rate (*module_clock* staying at a fixed frequency).

The constraints on *peripheral_clock* and *module_clock* are as follows:

- *peripheral_clock* and *module_clock* can totally be asynchronous. Off course, they can also be synchronous.
- Due to the 16x oversampling of the incoming characters, *module_clock* frequency must always be greater or equal to 16x the maximum baud rate. For example, if max baud rate is 4 Mbit/s, *module_clock* must be greater or equal to 4 M x 16 = 64MHz.

NOTE

The restriction that *peripheral_clock* frequency must be higher or equal to 16x baud rate has been removed. There is no limitation on *peripheral_clock* frequencyto baud rate.

75.4.2.2 Maximum Baud Rate

The max baud rate the UART can support is determined by the max frequency of the *module_clock* and logic synthesis results. For example, if the SoC can provide the fastest *module_clock* 66.5 MHz and the UART synthesis timing is acceptable under this constraint, the UART can transmit and receive serial data with the maximum baud rate 66.5M/16 = 4.15 Mbit/s.

The UART supports serial IR interface low speed. In the low speed IrDA mode, the max baud rate is 115.2Kbit/s. To support the 115.2Kbit/s, *module_clock* frequency must be higher or equal to 1.8432MHz.

75.4.2.3 Clocking in Low-Power Modes

The UART supports 2 low-power modes: DOZE and STOP.

In STOP mode (input pin *stop_req* is at '1'), the UART doesn't need any clock. In this mode the UART can wake-up the ARM platform with the asynchronous interrupts (refer to Low Power Modes). An application of this feature is when the system must be waken-up by the arrival of a frame of characters.

- If before entering in Stop mode the software has enabled RTSDEN interrupt, then when RTS will change state (put at '0' by external device started to send), the asynchronous interrupt will wake-up the system, *peripheral_clock* and *module_clock* will be provided to the UART before first start bit, so that no data will be lost.
- If RTS doesn't change state (already at '0' before entering in Stop mode), then wake-up interrupt (AWAKE) will be sent at the arrival of first Start bit (on falling edge). In this case, the UART must receive the *peripheral_clock* and *module_clock* during the first half of start bit to correctly receive this character (for example, at 115.2 Kbit/s, UART must receive *peripheral_clock* and *module_clock* at maximum 4.3 microseconds after falling edge of Start bit). If the UART receives *peripheral_clock* and *module_clock* too late, first character will be lost, and so should be dropped. Also, if autobaud detection is enabled, the first character won't be correctly received and another autobaud detection will need to be initiated.

In Doze mode, UART behavior is programmable through DOZE bit (UCR1[1]). If DOZE bit is set to '1', then UART is disabled in Doze mode, and in consequence, UART clocks can be switched-off (after being sure UART is not transmitting nor receiving). On the contrary, if DOZE bit is set to '0', UART is enabled and it must receive *peripheral_clock* and *module_clock* .

75.4.3 General UART Definitions

Definitions of terms that occur the following discussions are given in this section.

- Bit Time-The period of time required to serially transmit or receive 1 bit of data (1 cycle of the baud rate frequency).
- Start bit-The bit time of a logic 0 that indicates the beginning of a data frame. A start bit begins with a 1-to-0 transition, and is preceded by at least 1 bit time of logic 1.
- Stop bit-1 bit time of logic 1 that indicates the end of a data frame.
- BREAK-A frame in which all of the data bits, including the stop bit, are logic 0. This type of frame is usually sent to signal the end of a message or the beginning of a new message.
- Mark - When no data is being sent, the serial port's transmit pin's voltage is 1 and is said to be in a MARK state.
- Space - The serial port can also be forced to keep the transmit pin at a 0 and is said to be the SPACE or BREAK state.

- Frame-A start bit followed by a specified number of data or information bits and terminated by a stop bit. The number of data or information bits depends on the format specified and must be the same for the transmitting device and the receiving device. The most common frame format is 1 start bit followed by 8 data bits (least significant bit first) and terminated by 1 stop bit. An additional stop bit and a parity bit also can be included.
- Framing Error-An error condition that occurs when the stop bit of a received frame is missing, usually when the frame boundaries in the received bit stream are not synchronized with the receiver bit counter. Framing errors can go undetected if a data bit in the expected stop bit time happens to be a logic 1. A framing error is always present on the receiver side when the transmitter is sending BREAKs. However, when the UART is programmed to expect 2 stop bits and only the first stop bit is received, this is not a framing error by definition.
- Parity Error-An error condition that occurs when the calculated parity of the received data bits in a frame does not match the parity bit received on the RXD input. Parity error is calculated only after an entire frame is received.
- Idle-One in NRZ encoding format and selectable polarity in IrDA mode.
- Overrun Error-An error condition that occurs when the latest character received is ignored to prevent overwriting a character already present in the UART receive buffer (RxFIFO). An overrun error indicates that the software reading the buffer (RxFIFO) is not keeping up with the actual reception of characters on the RXD input.

75.4.3.1 $\overline{\text{RTS}}$ - UART Request To Send

The UART Request To Send input controls the transmitter. The modem or other terminal equipment signals the UART when it is ready to receive by setting '0' on the RTSpin. Normally, the transmitter waits until this signal is active (low) before transmitting a character, however when the Ignore RTS (IRTS) bit is set, the transmitter sends a character as soon as it is ready to transmit. An interrupt (RTSD) can be posted on any transition of this pin and can wake the ARM platform from STOP mode on its assertion. When $\overline{\text{RTS}}$ is set to '1' during a transmission, the UART transmitter finishes transmitting the current character and shuts off. The contents of the TxFIFO (characters to be transmitted) remain undisturbed. The operation of this input is the same regardless of whether the UART is in DTE or DCE mode.

75.4.3.2 RTS Edge Triggered Interrupt

The input to the $\overline{\text{RTS}}$ pin can be programmed to generate an interrupt on a selectable edge. See Table 75-106 for summary of the operation of the RTS edge triggered interrupt (RTSF).

To enable the $\overline{RTS}$ pin to generate an interrupt, set the request to send interrupt enable (RTSEN) bit (UCR2[4]) to 1. Writing 1 to the $\overline{RTS}$ edge triggered interrupt flag (RTSF) bit (USR2[4]) clears the interrupt flag. The interrupt can occur on the rising edge, falling edge, or either edge of the $\overline{RTS}$ input. The request to send edge control (RTEC) field (UCR2[10:9]) programs the edge that generates the interrupt. When RTEC is set to 0x00 and RTSEN = 1, the interrupt occurs on the rising edge (default). When RTEC is set to 0x01 and RTSEN = 1, the interrupt occurs on the falling edge. When RTEC is set to 0x1X and RTSEN = 1, the interrupt occurs on either edge. This is a synchronous interrupt. The RTSF bit is cleared by writing 1 to it. Writing 0 to RTSF has no effect.

Table 75-106. $\overline{RTS}$ Edge Triggered Interrupt Truth Table

$\overline{RTS}$	RTSEN	RTEC [1]	RTEC [0]	RTSF	Interrupt Occurs On	interrupt_uart
X	0	X	X	0	Interrupt disabled	1
1->0	1	0	0	0	Rising edge	1
0->1	1	0	0	1	Rising edge	0
1->0	1	0	1	1	Falling edge	0
0->1	1	0	1	0	Falling edge	1
1->0	1	1	X	1	Either edge	0
0->1	1	1	X	1	Either edge	0

There is another $\overline{RTS}$ interrupt that is not programmable. The status bit RTSD asserts the *interrupt_uart* interrupt when the $\overline{RTS}$ delta interrupt enable = 1. This is an asynchronous interrupt. The RTSD bit is cleared by writing 1 to it. Writing 0 to the RTSD bit has no effect.

75.4.3.3 $\overline{DTR}$ - Data Terminal Ready

This signal indicates the general readiness of the Data Terminal Equipment (DTE). This signal is an input in DCE mode and an output in DTE mode. If the connection between the DCE and the DTE is established once, the $\overline{DTR}$ signal must remain active throughout the whole connection time. In general the $\overline{DTR}$ and $\overline{DSR}$ signals are responsible for establishing the connection. $\overline{RTS}$ and $\overline{CTS}$ are responsible for the data transfer and the transfer direction in the case of a half-duplex configuration. The $\overline{DTR}$ signal is like a "main switch". If the $\overline{DTR}$ signal is inactive the $\overline{RTS}$ and $\overline{CTS}$ signals have no effect. In DCE mode, an interrupt (DTRD) can be posted on any transition of this pin and can wake the ARM platform from STOP mode on its assertion.

75.4.3.4 $\overline{\text{DSR}}$ - Data Set Ready

This signal indicates the general readiness of the DCE. This signal is an output in DCE mode and an input in DTE mode. The DCE uses this signal to inform the DTE that it is switched on, has completed all preparations and can communicate with the DTE. In DTE mode, an interrupt (DTRD) can be posted on any transition of this pin and can wake the ARM platform from STOP mode on its assertion.

75.4.3.5 $\overline{\text{DTR/DSR}}$ Edge Triggered Interrupt

The $\overline{\text{DTR}}$ input pin (DCE mode) or $\overline{\text{DSR}}$ input pin (DTE mode) can be configured to cause an interrupt on a selectable edge. See Table 75-107 for summary of the operation of the DTR/DSR edge triggered interrupt. To enable the interrupt, set the DTREN bit (UCR3[13]) to '1'. Write a "one" to the DTRF bit (USR2[13]) to clear the interrupt flag.

The interrupt can be configured to occur on either the rising, falling, or either edge of the $\overline{\text{DTR/DSR}}$ input. Write to the DPEC[1:0] bits (UCR3[15:14]) to program which edge will cause an interrupt. If the bits are set to 00b and DTREN = 1, the interrupt will occur on the rising edge (default). If the bits are set to 01b and DTREN = 1, the interrupt will occur on the falling edge. If the bits are set to 1Xb and DTREN = 1, the interrupt will occur on either edge.

Table 75-107. $\overline{\text{DTR/DSR}}$ Edge Triggered Interrupt Truth Table

$\overline{\text{DTR}}$ / $\overline{\text{DSR}}$	DTREN	DPEC[1]	DPEC[0]	DTRF	Interrupt occurs on:	interrupt_uart
X	0	X	X	0	turned off	1
1->0	1	0	0	0	rising edge	1
0->1	1	0	0	1	rising edge	0
1->0	1	0	1	1	falling edge	0
0->1	1	0	1	0	falling edge	1
1->0	1	1	X	1	either edge	0
0->1	1	1	X	1	either edge	0

75.4.3.6 $\overline{\text{DCD}}$ - Data Carrier Detect

This signal is an output in DCE mode and an input in DTE mode. If used, the DCE device uses this signal to inform the DTE it has detected the carrier signal and the connection will be set up. This signal remains active while the connection remains established. In DTE mode this input can trigger an interrupt on changing state. This is

achieved by setting to '1' the interrupt enable bit (DCD, UCR3[9]). The change state is reflected in DCDDELT (USR2[6]). Also, the state of the Data Carrier Detect input is mirrored in the status register DCDIN (USR2[5]).

75.4.3.7 $\overline{\text{RI}}$ - Ring Indicator

This signal is an output in DCE mode and an input in DTE mode. If used, the DCE device uses this signal to inform the DTE that a ring just occurred. In DTE mode this input can trigger an interrupt on changing state. This is achieved by setting to '1' the interrupt enable bit (RI, UCR3[8]). The change state is reflected in RIDELT (USR2[10]). Also, the state of the Ring Indicator input is mirrored in the status register RIIN (USR2[9]).

75.4.3.8 $\overline{\text{CTS}}$ - Clear To Send

This output pin serves two purposes. Normally, the receiver indicates that it is ready to receive data by asserting this pin (low). When the $\overline{\text{CTS}}$ trigger level is programmed to trigger at 32 characters received and the receiver detects the valid start bit of the 33 character, it de-asserts this pin. The operation of this output is the same regardless of whether the UART is in DTE or DCE mode.

75.4.3.9 Programmable $\overline{\text{CTS}}$ Deassertion

The $\overline{\text{CTS}}$ output can also be programmed to deassert when the RxFIFO reaches a certain level. Setting the CTS trigger level (UCR4[15:10]) at any value less than 32 deasserts the $\overline{\text{CTS}}$ pin on detection of the valid start bit of the N + 1 character (where N is the trigger level setting). However, the receiver continues to receive characters until the RxFIFO is full.

75.4.3.10 TXD - UART Transmit

This is the transmitter serial output. When operating in RS-232 mode, NRZ encoded data is . When operating in infrared mode, a 3/16 bit-period pulse is output for each 0 bit transmitted, and no pulse is output for each 1 bit transmitted. For RS-232 applications, this pin must be connected to an RS-232 transmitter. The operation of this output is the same regardless of whether the UART is in DTE or DCE mode. See Figure 75-98.

75.4.3.11 RXD - UART Receive

This is the receiver serial input. When operating in RS-232 mode, NRZ encoded data is expected. When operating in infrared mode, a narrow pulse is expected for each 0 bit received and no pulse is expected for each 1 bit received. External circuitry must convert the IR signal to an electrical signal. RS-232 applications require an external RS-232 receiver to convert voltage levels. The operation of this input is the same regardless of whether the UART is in DTE or DCE mode. See Figure 75-98.

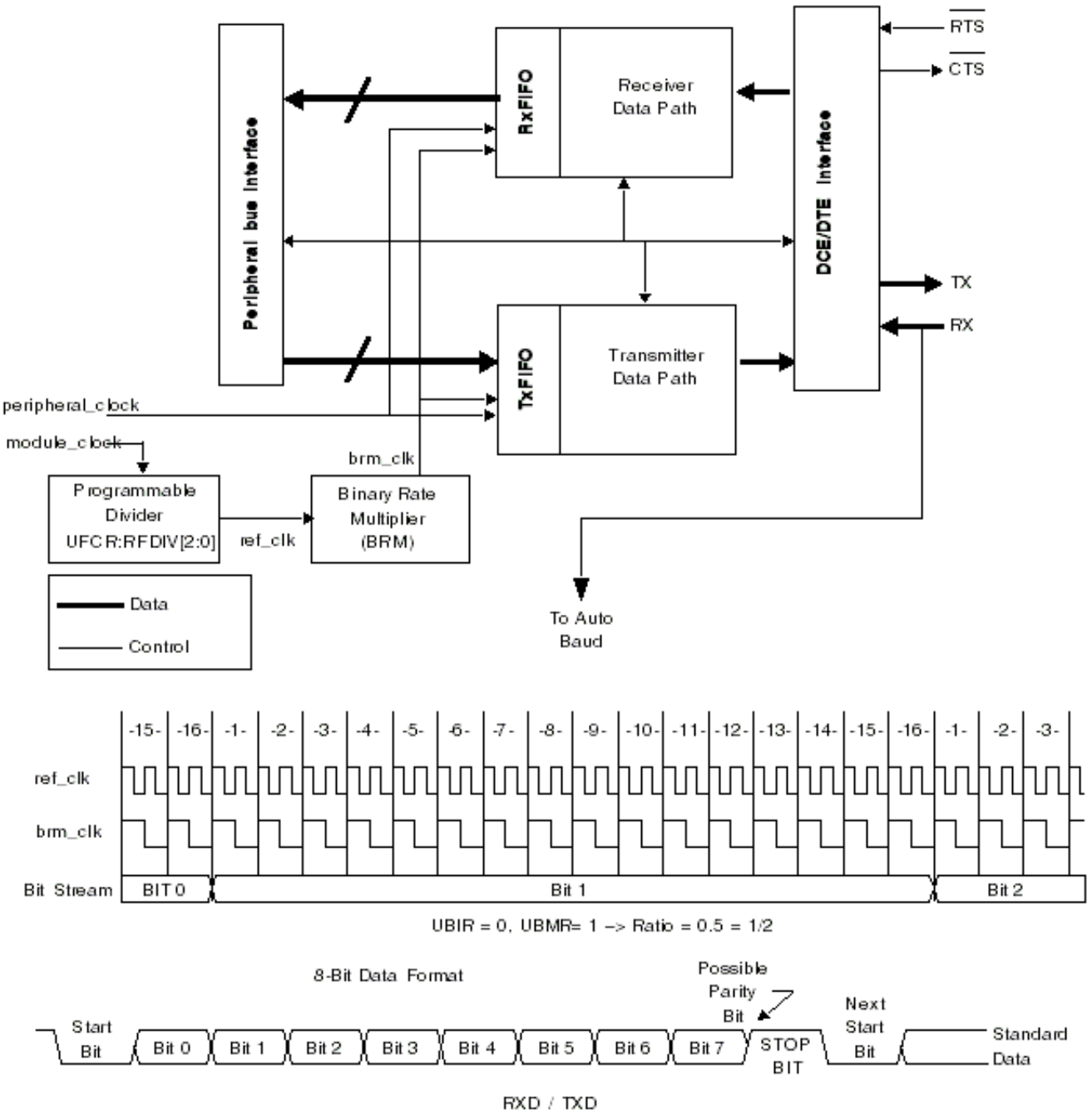


Figure 75-98. UART Simplified Block and Clock Generation Diagrams

75.4.4 Transmitter

The transmitter accepts a parallel character from the ARM platform and transmits it serially. The start, stop, and parity (when enabled) bits are added to the character. When the ignore RTS bit (IRTS) is set, the transmitter sends a character as soon as it is ready to transmit. $\overline{\text{RTS}}$ can be used to provide flow-control of the serial data. When $\overline{\text{RTS}}$ is set to '1', the transmitter finishes sending the character in progress (if any), stops, and waits for $\overline{\text{RTS}}$ to be set to '0' again. Generation of BREAK characters and parity errors (for debugging purposes) is supported. The transmitter operates from the clock provided by the BRM. Normal NRZ encoded data is transmitted when the IR interface is disabled.

The transmitter FIFO (TxFIFO) contains 32 bytes. The data is written to TxFIFO by writing to the UTXD register with the byte data to the [7:0] bits. The data is written consecutively if the TxFIFO is not full. It is read (internally) consecutively if the TxFIFO is not empty. TXFULL bit (UTS[4]) can be used to control whether TXFIFO is full or not. The TxFIFO can be written regardless of the transmitter is disabled or enabled. If the UART is disabled, user can still write data into the TxFIFO correctly. But in this case the write access will yield to a transfer error.

75.4.4.1 Transmitter FIFO Empty Interrupt Suppression

The transmitter FIFO empty interrupt suppression logic suppresses the TXFE interrupt between writes to the TxFIFO. When TxFIFO is empty, the software can either send one or several characters. If the software sends one character, it would write the character into the UTXD register, then that character is immediately transferred to the transmitter shift register, assuming the transmitter is already enabled. Without interrupt suppression logic, the TXFE interrupt flag would be set immediately. But, with this logic, the interrupt flag is set when the last bit of the character has been transmitted, for example, before the transmission of the parity bit (if exists) and the stop bit(s).

So, the suppression logic doesn't immediately send the TXFE interrupt flag. It allows the software to write another character to the TxFIFO before the interrupt flag is asserted.

When the transmitter shift register empties before another character is written to the TxFIFO, the interrupt flag is asserted. Writing data to the TxFIFO would release the interrupt flag. The interrupt flag is asserted on the following conditions:

- System Reset
- UART software reset

- When a single character has been written to Transmitter FIFO and then the Transmitter FIFO and the Transmitter Shift Register become empty until another character is written to the Transmitter FIFO
- The last character in the TxFIFO is transferred to the shift register, when TxFIFO contains two or more characters. See Figure 75-99.

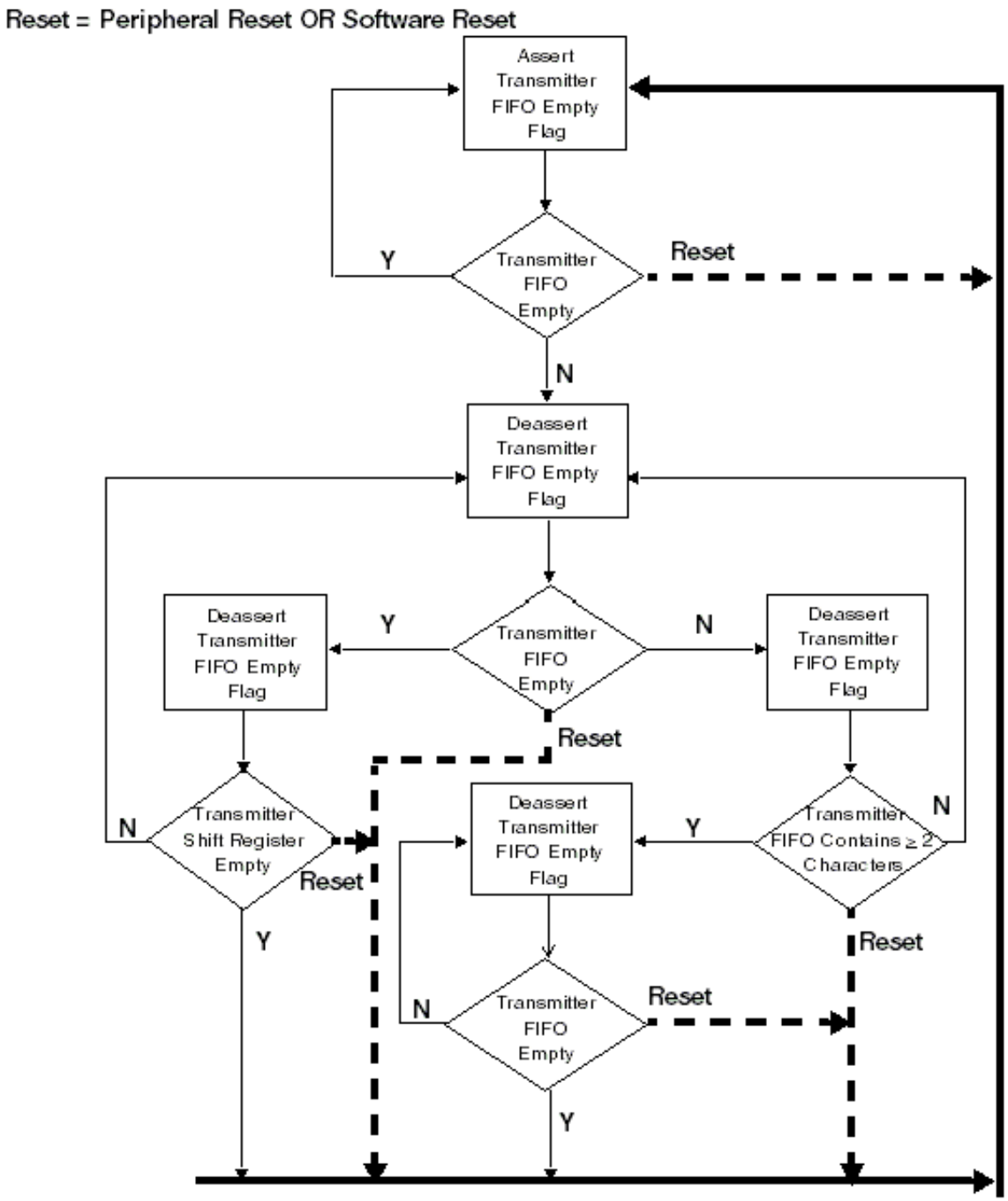


Figure 75-99. Transmitter FIFO Empty Interrupt Suppression Flow Chart

75.4.4.2 Transmitting a Break Condition

Asserting SNDBRK bit of the UCR1 Register forces the transmitter to send a break character (continuous zeros). The transmitter will finish sending the character in progress (if any) before sending break until this bit is reset. The user is responsible to ensure that this bit is high for long enough to generate a valid BREAK. The transmitter samples SNDBRK after every bit is transmitted. Following completion of the BREAK transmission, the UART will transmit two mark bits. The user can continue to fill the FIFO and any character remaining will be transmitted when the break is terminated.

75.4.5 Receiver

See Figure 75-100 for the receiver flow chart. The receiver accepts a serial data stream and converts it into parallel characters. When enabled, it searches for a start bit, qualifies it, and samples the following data bits at the bit-center. Jitter tolerance and noise immunity are provided by sampling at a 16x rate and using voting techniques to clean up the samples. Once the start bit is found, the data bits, parity bit (if enabled), and stop bits (either 1 or 2 depending on user selection) are shifted in. Parity is checked and its status reported in the URXD register when parity is enabled. Frame errors and BREAKs are also checked and reported. When a new character is ready to be read by the ARM platform from the RxFIFO, the receive data ready (RDR = USR2[0]) bit is asserted and an interrupt is posted (if DREN = UCR4[0] = 1). If the receiver trigger level is set to 2 (RXTL[5:0] = UFCR[5:0] = 2), and 2 chars have been received into RxFIFO, the receiver ready interrupt flag (RRDY = USR1[9]) is asserted and an interrupt is posted if the receiver ready interrupt enable bit is set (RRDYEN = UCR1[9] = 1). If the UART Receiver Register (URXD) is read once, and in consequence there is only 1 character in the RxFIFO, the interrupt generated by the RDR bit is automatically cleared. The RRDY bit is cleared when the data in the RxFIFO falls below the programmed trigger level.

Normal NRZ encoded data is expected when the IR interface is disabled.The RxFIFO contains 32 half-word entries. Characters received are written consecutively into this FIFO. If the FIFO is full and a 33rd characters is received, this character will be ignored and the USR2[ORE] bit will be set.

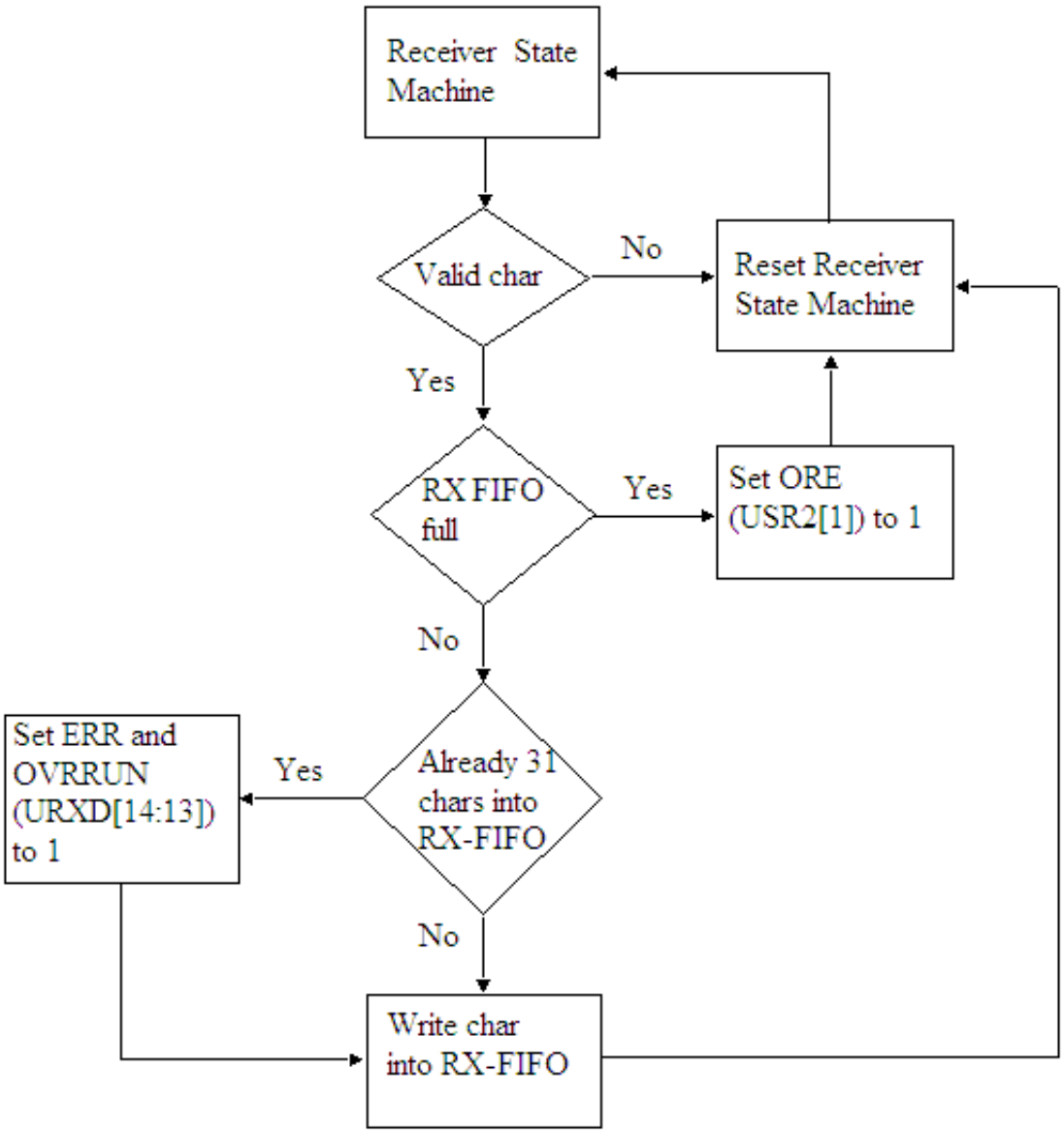


Figure 75-100. Receiver Flow Chart

75.4.5.1 Idle Line Detect

The receiver logic block includes the ability to detect an idle line. Idle lines indicate the end or the beginning of a message.

For an idle condition to occur:

- RxFIFO must be empty and
- RXD pin must be idle for more than a configured number of frames (ICD[1:0] = UCR1[11:10]).

When the idle condition detected interrupt enable (IDEN = UCR1[12]) is set and the line is idle for 4 (default), 8, 16, or 32 (maximum) frames, the detection of an idle condition flags an interrupt (see Table 75-108). When an idle condition is detected, the IDLE (USR2[12]) bit is set. Clear the IDLE bit by writing 1 to it. Writing 0 to the IDLE bit has no effect.

Table 75-108. Detection Truth Table

IDEN	ICD [1]	ICD [0]	IDLE	*interrupt_uart*
0	X	X	0	1
1	0	0	asserted after 4 idle frames	asserted after 4 idle frames
1	0	1	asserted after 8 idle frames	asserted after 8 idle frames
1	1	0	asserted after 16 idle frames	asserted after 16 idle frames
1	1	1	asserted after 32 idle frames	asserted after 32 idle frames
NOTE: This table assumes that no other interrupt is set at the same time this interrupt is set for the *interrupt_uart* signal. This table shows how this interrupt affects the *interrupt_uart* signal.				

During a normal message there is no idle time between frames. When all of the information bits in a frame are logic 1s, the start bit ensures that at least one logic 0 bit time occurs for each frame so that the IDLE bit is not asserted.

75.4.5.2 Aging Character Detect

The receiver block also includes the possibility to detect when at least one character has been sitting into the RxFIFO for a time corresponding to 8 characters. This aging character capability allows the UART to inform the ARM platform that there is less character into the RxFIFO than the Rx trigger and, no new character has been detected on the RXD line. The agiing capability is a timer which starts to count as soon as the RxFIFO is not empty and its trigger level is not reached (RRDY=0). This counter is reset when either a RxFIFO read is performed or another character starts to present on the RXD line. If none of those two events occurs, the bit AGTIM (USR1[8]) is set when the counter has measured a time corresponding to 8 characters. AGTIM is cleared by writing a 1 to it. AGTIM can flag an interrupt to ARM platform on *interrupt_uart* if ATEN (UCR2[3]) has been set.

To summarize, AGTIM is set when:

- There is at least one character into RxFIFO.
- No read has occurred on RxFIFO and RXD line has stayed high, for a time corresponding to 8 characters.
- The RxFIFO trigger is not reached (RRDY=0)

75.4.5.3 Receiver Wake

The WAKE bit (USR2[7]) is set when the receiver detects a qualified Start bit. For this, two conditions must be fulfilled, firstly a falling edge on RXD line must be detected and secondly the RXD line must stays at low level for more than a half-bit duration. When the wake interrupt enable WKEN (UCR4[7]) bit is enabled, the receiver flags an interrupt (*interrupt_uart*) if the WAKE status bit is set. The WAKE bit is cleared by writing 1 to it. Writing 0 to the WAKE bit has no effect. The WAKE status bit can be asserted in either serial RS-232 mode or IR mode. The generation of the WAKE interrupt needs the clock *module_clock* .

When the asynchronous wake interrupt (AWAKE) is enabled (AWAKEN = UCR3[4] = 1), and the ARM platform is in STOP mode, and UART clocks have been shut-off, then a falling edge detected on the receive pin (RXD) asserts the AWAKE bit (USR1[4]) and the *interrupt_uart* interrupt to wake the ARM platform from STOP mode. Re-enable UART clocks and clear the AWAKE bit by writing 1 to it. Writing 0 to the AWAKE bit has no effect. When IR interface is enabled (UCR1[7]=1), the AWAKE bit is always not asserted. The generation of the asynchronous AWAKE interrupt does not need any clocks.

In IR mode, if the asynchronous IR WAKE interrupt is enabled (AIRINTEN = UCR3[5] = 1), and if the ARM platform is in STOP mode (UART clocks are off when ARM platform in STOP mode), then the detection of a falling edge on the receive pin (RXD_IR), asserts the AIRINT bit (USR1[5]), and the *interrupt_uart* interrupt. This interrupt wakes the ARM platform from STOP mode. Software re-enables UART clocks and clear the AIRINT bit by writing 1 to it. Writing 0 to the AIRINT bit has no effect. When IR interface is disabled (UCR1[7]=0), the AIRINT bit is always not asserted. The generation of the asynchronous AIRINT interrupt does not need any clocks.

Recommended procedure for programming the asynchronous interrupts is to first clear them by writing 1 to the appropriate bit in the UART Status Register 1 (USR1). Poll or enable the interrupt for the Receiver IDLE Interrupt Flag (RXDS) in the USR1. When asserted, the RXDS bit indicates to the software that the receiver state machine is in the idle state, the next state is idle, and the RXD pin is idle (high). After following this procedure, enable the asynchronous interrupt and enter STOP mode.

75.4.5.4 Receiving a BREAK Condition

A BREAK condition is received when the receiver detects all 0s (including a 0 during the bit time of the stop bit) in a frame. The BREAK condition asserts the BRCD bit (USR2[2]) and writes only the first BREAK character to the RxFIFO. Clear the BRCD bit by writing 1 to it. Writing 0 to the BRCD bit has no effect.

Asserting BRCD would generate an interrupt on *interrupt_uart*. The interrupt generation can be masked using the control bit BKEN (UCR4[2]). Receiving a break condition will also effect the following bits in the receiver register URXD:

URXD(11) = BRK. While high this bit indicates that the current char was detected as a break.

URXD(12) = FRMERR. The frame error bit will always be set when BRK is set.

URXD(10) = PRERR. If odd parity was selected the parity error bit will also be set when BRK is set.

URXD(14) = ERR. The error detect bit indicates that the character present in the rx data field has an error status. This can be asserted by a break.

75.4.5.5 Vote Logic

The vote logic block provides jitter tolerance and noise immunity by sampling with respect to a 16x clock (*brm_clk*) and using voting techniques to clean up the samples. The voting is implemented by sampling the incoming signal constantly on the rising edge of the *brm_clk*. See Figure 75-101. The receiver is provided with the majority vote value, which is 2 out of the 3 samples. For examples of the majority vote results of the vote logic, see Table 75-109.

Table 75-109. Majority Vote Results

Samples	Vote
000	0
101	1
001	0
111	1

The vote logic captures a sample on every rising edge of *brm_clk*, however the receiver uses 16x oversampling to take its value in the middle of the sample character.

The receiver starts to count when the Start bit is set however it does not capture the contents of the RxFIFO at the time the Start bit is set. The start bit is validated when 0s are received for 7 consecutive 1/16 of bit times following the 1-to-0 transition. Once the counter reaches 0xF, it starts counting on the next bit and captures it in the middle of the sampling frame (see Table 75-109). All data bits are captured in the same manner. Once the stop bit is detected, the receiver shift register (SIPO_OUT) data is parallel shifted to the RxFIFO.

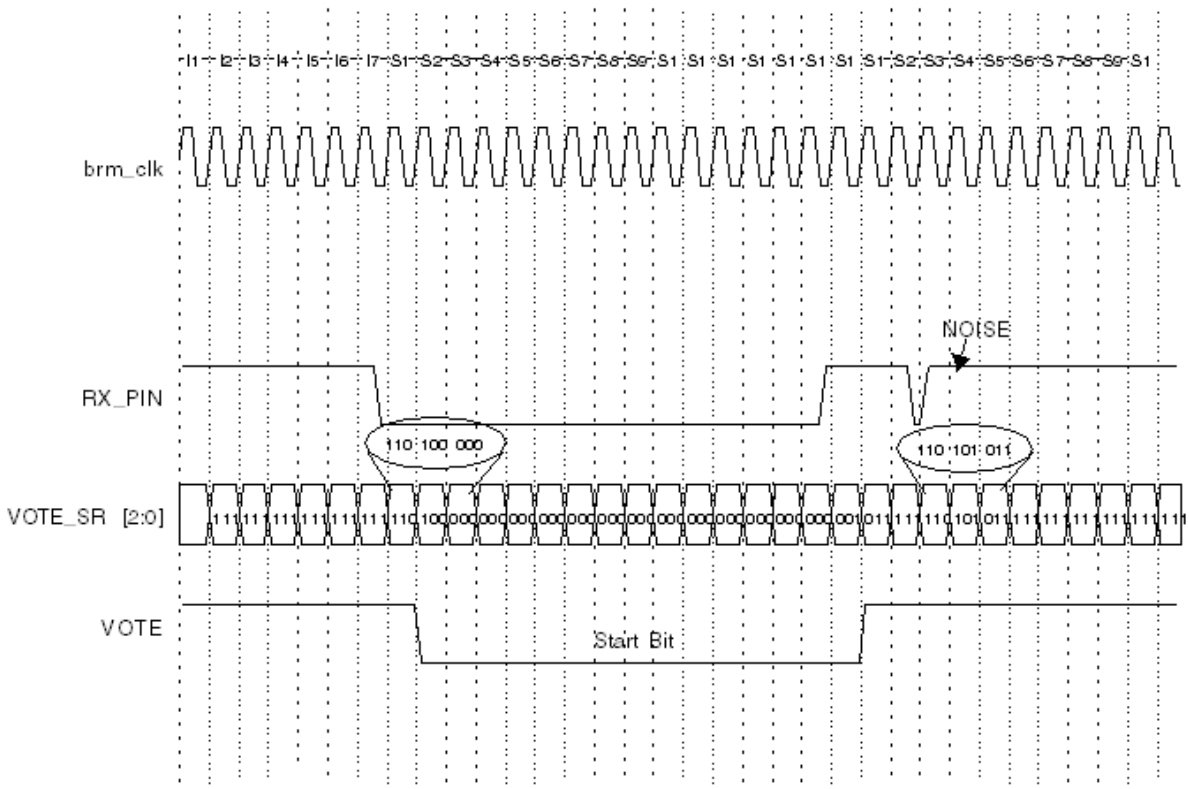


Figure 75-101. Majority Vote Results

A new feature has been recently implemented, it allows to re-synchronize the counter on each edge of RXD line. This is automatic and allows to improve the immunity of UART against signal distortion.

There is a special case when the *brm_clk* frequency is too low and is unable to capture a 0 pulse in IrDA. In this case, the software must set the IRSC (UCR4[5]) bit so that the reference clock (after internal divider) is used for the voting logic. The pulse is validated by counting the length of the pulse.

Refer to Infrared Interface for more details.

75.4.5.6 Baud Rate Automatic Detection Logic

When the baud rate automatic detection logic is enabled, the UART locks onto the incoming baud rate. To enable this feature, set the automatic detection of baud rate bit (ADBR = UCR1[14] = 1) and write 1 to the ADET bit (USR2[15]) to clear it. When ADET=0 and ADBR =1, the detection starts. Then, once the beginning of start bit (transition from 1-to-0 of RXD) has been detected, UART start a counter (UBRC) working at reference frequency. Once the end of start bit is detected (transition from 0-to-1 of RXD), the value of UBRC - 1 is directly copied into UBMR register. UBIR register is filled with 0x000F.

So, at the end of start bit, registers gets following values:

```
UBRC = number of reference clock periods (after divider) during Start bit.
UBIR = 0x000F
UBMR = UBRC - 1
```

The updated values of the 3 registers can be read.

See Table 75-110 for list of parameters for baud rate detection and Figure 75-102 for baud rate detection protocol diagram.

If any of the UART BRM registers are simultaneously written by the baud rate automatic detection logic and by the peripheral data bus, the peripheral data bus would have lower priority.

Table 75-110. Baud Rate Automatic Detection

ADBR	ADET	Baud Rate Detection	*interrupt_uart*
0	X	Manual Configuration	1
1	0	Auto Detection Started	1
1	1	Auto Detection Complete	0
NOTE: This table assumes that no other interrupt is set at the same time this interrupt is set for the *interrupt_uart* signal.			

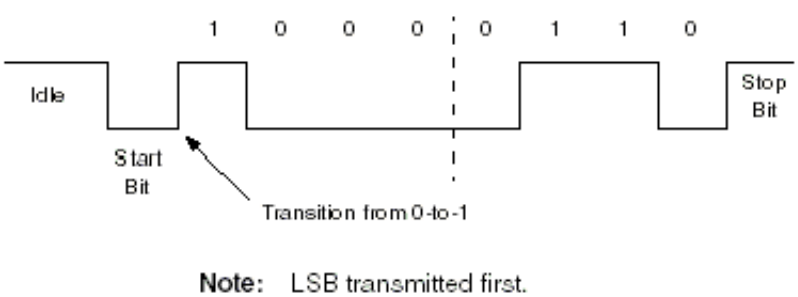


Figure 75-102. Baud Rate Detection Protocol Diagram

75.4.5.6.1 Baud Rate Automatic Detection Protocol

The receiver must receive an ASCII character "A" or "a" to verify proper detection of the incoming baud rate. When an ASCII character "A" (0x41) or "a" (0x61) is received and no error occurs, the Automatic Detect baud rate bit is set (ADET=1) and if the interrupt is enabled (ADEN=UCR1[15]=1), an interrupt *interrupt_uart* is generated.

When an ASCII character "A" or "a" is not received (because of a bit error or the reception of another character), the auto detection sequence restarts and waits for another 1-to-0 transition.

As long as ADET = 0 and ADBR = 1, the UART continues to try to lock onto the incoming baud rate. Once the ASCII character "A" or "a" is detected and the ADET bit is set, the receiver ignores the ADBR bit and continues normal operation with the calculated baud rate.

The UART interrupt is active (*interrupt_uart* = 0) as long as ADET = 1 and ADBR = 1. This can be disabled by clearing the automatic baud rate detection interrupt enable bit (ADEN = 0). Before starting an automatic baud rate detection sequence, set ADET = 0 and ADBR = 1.

The RxFIFO must contain the ASCII character "A" or "a" following the automatic baud rate detection interrupt.

The 16-bit UART Baud Rate Count Register (UBRC) is reset to 4 and stays at 0xFFFF when an overflow occurs. The UBRC register counts (measures) the duration of start bit. When the start bit is detected and counted, the UART Baud Rate Count Register retains its value until the next automatic baud rate detection sequence is initiated.

The Baud Rate Count Register counts only when auto detection is enabled.

75.4.5.6.2 Baud Rate Automatic Detection Protocol Improved

NOTE

Several issues have been reported for ICs using the existing autobaud protocol, especially for 57.6 Kbit/s and 115.2 Kbit/s. As a consequence, this protocol has been improved. The old one is still available in the current UART block, but several modifications can also be used in order to make this autobaud detection more reliable. If the user wants to keep with the old method, he has to set the bit ADNIMP (UCR3[7]) to 1. If this bit isn't set (default), the autobaud improvements will be used. Those improvements are mainly grouped in two categories: the new baud rate measurement and the new ACST bit (and associated interrupt).

75.4.5.6.3 New Baud Rate Determination

In order to fight against the problems caused by the distortion and the noise on the RXD line, the duration of the baud rate measurement has been extended. Previously, as described above, this determination was based on the measurement of the START bit duration. Now, this measurement is based on the duration of START bit + bit0. Bit0 is the first bit following the START bit. In fact, the counter which is started at the falling edge of START bit is no longer stopped at next rising edge (end of START bit), but it is stopped at the next falling edge (end of bit0). As the character sent is always a "A" (41h) or a "a" (61h), this second falling edge will always be present and it will indicate the end of bit0. Once this counter is stopped, the result is divided by 2 and used by the BRM to determine the incoming baud rate.

NOTE

UBRC register contains the result of this division by two, in consequence it reflects the measurement of the duration of one bit.

75.4.5.6.3.1 New Autobaud Counter Stopped bit and Interrupt

A new bit has been added in USR2 register: ACST (USR2[11]). This bit is set immediately after the determination of the baud rate, So,

- if ADNIMP is not set (default), ACST is set to 1 after the end of bit0,
- If ADNIMP is set to 1, ACST is set to 1 at the end of START bit.

If ACIEN (UCR3[0]) is set to 1, ACST will flag an interrupt on *interrupt_uart* signal. This interrupt informs the ARM platform the BRM has just been set with the result of the bit length measurement. If needed, the ARM platform can perform a read of UBMR (or UBRC) register and determine by itself the baud rate measured. Then the ARM platform has the possibility to correct the BRM registers with the nearest standardized baud rate.

NOTE

ACST is set only if ADBR is set to 1, for example, the UART is autobauding.

Clear the ACST bit by writing 1 to it. Writing 0 to the ACST bit has no effect.

75.4.5.7 Escape Sequence Detection

An escape sequence typically consists of 3 characters entered in rapid succession (such as +++). Because these are valid characters by themselves, the time between characters determines if it is a valid escape sequence. Too much time between two of the "+" characters is interpreted as two "+" characters, and not part of an escape sequence.

The software chooses the escape character and writes its value to the UART Escape Character Register (UESC). The software must also enable escape detection feature by setting ESCEN (UCR2[11]) to 1. The hardware compares this value to incoming characters in the RxFIFO. When an escape character is detected, the internal escape timer starts to count. The software specifies a time-out value for the maximum allowable time between 2 successive escape characters (see Table 75-111). The escape timer is programmable in intervals of 2 ms to a maximum interval of 8.192 seconds.

Table 75-111. Escape Timer Scaling

UTIM Register	Maximum Time Between Specified Escape Characters
0x000	2 ms
0x001	4 ms
0x002	6 ms
0x003	8 ms
0x004	10 ms
...	...
0F8	498 ms
0F9	500 ms
...	...
9C3	5 s
...	...
FFD	8.188 s
FFE	8.190 s
FFF	8.192 s
NOTE: To calculate the time interval: (UTIM_Value + 1) x 0.002 = Time_Interval Example: (09C3 + 1) x 0.002 = 5 s.	

The escape sequence detection feature is available for all the reference frequencies. Before using Escape Sequence Detection, the user must fill the ONEMS register. This 24-bit register must contain the value of the UART internal frequency divided by 1000. The internal frequency is obtained after the UART internal divider which is applied on *module_clock* clock.

Example I:

- If the input clock *module_clock* frequency is 66.5 MHz.
- And if the input clock *module_clock* is divided by 2 with the internal divider: UFCR[9:7] = 3'b100

$$ONEMS = \frac{66.5 \times 10^6}{2 \times 1000} = 33250 = 81E2h$$

Figure 75-103. Calculation of Frequency for ONEMS Register

Example II:

- If the input clock *module_clock* frequency is 66.5 MHz.
- And if the input clock *module_clock* is divided by 1with the internal divider: UFCR[9:7] = 3'b101

$$ONEMS = \frac{66.5 \times 10^6}{1000} = 66500 = 103C4h$$

Figure 75-104. Calculation of Frequency for ONEMS Register

The escape sequence detection interrupt is asserted when the escape sequence interrupt enable (ESCI) bit is set and an escape sequence is detected (ESCF set). Clear the ESCF bit by writing 1 to it. Writing 0 to the ESCF bit has no effect.

75.4.6 Binary Rate Multiplier (BRM)

The BRM sub-block receives *ref_clk* (*module_clock* clock after divider). From this clock, and with integer and non-integer division, BRM generates a 16x baud rate clock whose frequency is 16 times of baud rate. The uart transmitter will shift data out based on this 16x baud rate clock. The uart receiver will sample the serial data line based on this 16x baud rate clock. The input and output frequency ratio is programmed in the UART BRM Incremental Register (UBIR) and UART BRM MOD Register (UBMR). The output frequency is divided by the input frequency to produce this ratio. For integer division, set the UBIR = 0x000F and write the divisor to the UBMR register. All values written to these registers must be one less than the actual value to eliminate division by 0 (undefined), and to increase the maximum range of the registers.

Updating the BRM registers requires writing to both registers. The UBIR register must be written before writing to the UBMR register. If only one register is written to by the software, the BRM continues to use the previous values.

The following examples show how to determine what values are to be programmed into UBIR and UBMR for a given reference frequency and desired baud rate. The following equation can be used to help determine these values:

$$\text{BaudRate} = \frac{\text{RefFreq}}{\left(16 \times \frac{\text{UBMR} + 1}{\text{UBIR} + 1}\right)}$$

Figure 75-105. Frequency and Baud Rate for UBIR and UBMR

With:

Reference Frequency (Hz): UART Reference Frequency (*module_clock* after RFDIV divider)

Baud Rate (bit/s): Desired baud rate.

Integer Division ÷ 21

Reference Frequency = 19.44 MHz

UBIR = 0x000F

UBMR = 0x0014

Baud Rate = 925.7 kbit/s

NOTE

Observe that each value written to the registers is one less than the actual value.

Non-Integer Division

```
Reference Frequency = 16 MHz
Desired Baud Rate = 920 Kbits/s
```

$$\frac{UBMR+1}{UBIR+1} = \frac{RefFreq}{16 \times BaudRate} = \frac{16 \times 10^6}{16 \times 920 \times 10^3} = 1.087$$

```
Ratio = 1.087 = 1087 / 1000
UBIR = 999 (decimal)= 0x3E7
UBMR = 1086 (decimal)= 0x43E
Non-Integer Division
Reference Frequency = 25 MHz
Desired Baud Rate = 920 kbit/s
Ratio = 1.69837 = 625 / 368
UBIR = 367 (decimal)= 0x16F
UBMR = 624 (decimal)= 0x270
```

Non-Integer Division

```
Reference Frequency: 30 MHz
Desired Baud Rate = 115.2 kbit/s
Ratio = 16.276043 = 65153 / 4003
UBIR = 4002 (decimal) = 0x0FA2
UBMR = 65152 (decimal) = 0xFE80
```

75.4.7 Infrared Interface

75.4.7.1 Generalities-Infrared

The Infrared interface is selected when IREN (UCR1[7]) is set to 1.

The Infrared Interface is compatible with IrDA Serial Infrared Physical Layer Specification. In this specification, a "zero" is represented by a positive pulse, and a "one" is represented by no pulse (line remains low).

In the UART:

In TX: For each "zero" to be transmitted, a narrow positive pulse which is 3/16 of a bit time is generated. For each "one" to be transmitted no pulse is generated (output is low). External circuitry has to be provided to drive an Infrared LED.

In RX: When receiving, a narrow negative pulse is expected for each "zero" transmitted while no pulse is expected for each "one" transmitted (input is high).

NOTE

Rx part of IR block expects to receive an inverted signal compared to IrDA specification. Circuitry external to the IC transforms the Infrared signal to an electrical signal.

The IR interface has an edge triggered interrupt (IRINT). This interrupt validates a zero bit being received. This interrupt is enabled by writing a "one" to ENIRI bit.

The behavior of Infrared Interface is determined by 3 bits INVT (UCR3[1]), INVR (UCR4[9]) and IRSC (UCR4[5]).

75.4.7.2 Inverted Transmission and Reception bits (INVT & INVR)

The values of INVT and INVR depend of the IrDA transceiver connected on the TXD_IR and RXD_IR pins of the UART. If this transceiver is not inverting on both paths Tx and Rx, a Zero is represented by a positive pulse and a One is represented by no pulse (line remains low). In this case, the bit INVT must be set to 0 and the bit INVR must be set to 1 (because Rx IR block expects an inverted signal).

On the contrary user must set INVT=1 and INVR=0 if both paths of the transceiver are inverting, that is, a Zero is represented as a negative pulse and a One is represented by no pulse (line remains high). The transceiver can also be inverting on only one path (Tx or Rx), in this case INVT and INVR must be together equal to 1 or to 0, depending on which path is inverted.

75.4.7.3 InfraRed Special Case (IRSC) Bit

The value to apply to IRSC bit is based on 2 parameters: the baud rate and the Minimum Pulse Duration (MPD) of the transceiver. According to IrDA Standard Specification, for SIR (Serial IR) baud rates from 2.4 Kbit/s to 115.2 Kbit/s this nominal pulse duration is equal to 3/16 of a bit duration (at the selected baud rate). But, for all the baud rates a Minimum Pulse Duration is also specified. According to IrDA Standard, a Zero is represented by a light pulse, so the IrDA transceiver can't emit a light pulse shorter than the MPD. For SIR, the MPD is constant and equal to 1.41 us.

But user must take into account the electrical MPD associated to the transceiver on the receiver path. Typically this value is 2.0 us, but for some manufacturers MPD can go down to 1.0 us.

In order to understand the meaning of IRSC bit, one must understand how the RX path work in IrDA mode.

When UART is in IrDA mode, a Zero is not only detected by the state of the RXD_IR line, but also with the duration of the pulse. This pulse duration can be measured with 2 different clocks. In this case, clock is selected with the IRSC bit.

- If IRSC = 0, the clock used is the BRM clock.
- If IRSC = 1, the clock used is the UART internal clock (UART clock after the divider (RFDIV)).

In normal operation, IRSC=0. This means at any time, the user must insure the frequency of BRM_clock is high enough to measure the pulse. In the UART and for IRSC=0, the pulse must last at least 2 BRM clock cycles.

If this condition is not fulfilled, IRSC must be set to 1.

Let's take 2 examples, with the Minimum Pulse Duration equals to the MPD of the IrDA specification (in SIR).

Calculation of BRM Clock Period (Clock Period < 1.41 µs)

The user wants to receive IrDA data at 115.2 Kbit/s. The UBIR and UBMR registers are set in order to create the BRM_clock with a frequency of 16*baud rate = 16 * 115.2 = 1.843 MHz. But at the same time, in order to correctly detect the pulse, the user must be sure that 2* BRM_clock period is lower than 1.41 us. Lets check:

BRM_clock period = 1/1843000 = 542 ns

So 2*BRM_clock period = 1.09 us < 1.41 us. It is fine.

Calculation of BRM Clock Period (Clock Period > 1.41 µs)

This time the user wants to receive at 19.2 Kbit/s. So, the BRM_clock is set to 16*19200 = 307.2 kHz. Let's check if 2* BRM_clock period < 1.41 us:

1. BRM_clock period =1/307200 = 3.25 us

   So 2*BRM_clock period = 6.50 us >> 1.41 us. It doesn't work.

   So, in this case, the BRM clock can't be used to measure the pulse duration and the user must select the UART internal clock by setting IRSC =1.

NOTE

Like for Escape character detection, when IR Special Case is enabled (IRSC=1), the UART must measure a duration. In order to do that, the user must fill the ONEMS register. Refer to Escape Sequence Detection.

75.4.7.4 IrDA interrupt

Serial infrared mode (SIR) uses an edge triggered interrupt flag IRINT (USR2[8]). When INVR =0, detection of a falling edge on the RXD pin asserts the IRINT bit. When INVR=1, detection of a rising edge on the RXD pin asserts the IRINT bit. When IRINT and ENIRI bits are both asserted, the *interrupt_uart* interrupt is asserted. Clear the IRINT bit by writing 1 to it. Writing 0 to the IRINT bit has no effect.

75.4.7.5 Conclusion about IrDA

Before using the UART in IrDA, the baud rate limit must be calculated. This baud rate limit will inform the user if IRSC bit has to be set or not.

Let's determine this limit:

As already described, if IRSC = 0, the following condition must always be fulfilled

$$2 \times BRMClockPeriod < MinPulseDuration$$

Figure 75-106. Calculation of Baud Rate

So,

$$BRMClockFrequency > \frac{2}{MPD}$$

So, knowing BRM_clock frequency = 16 * Baud Rate, we get:

$$BaudRate > \frac{1}{8 \times MinPulseDuration}$$

So, the user needs to set IRSC = 0 when:

- If Minimum Pulse Duration = 2.5 us and Baud Rate > 50 Kbit/s.
- If Minimum Pulse Duration = 2.0 us and Baud Rate > 62.5 Kbit/s.
- If Minimum Pulse Duration = 1.41 us and Baud Rate > 88.6 Kbit/s.

NOTE

For baud rates lower than the limit, IRSC must be set to1.

75.4.7.6 Programming IrDA Interface

75.4.7.6.1 High Speed

As an example, the following sequence can be used to program the IrDA interface in order to send and receive characters at 115.2 Kbit/s.

Assumptions:

- Input UART clock = 90 MHz
- Internal clock divider = 3 (divide Input UART clock by 3)
- Baud rate = 115.2 Kbit/s
- IrDA transceiver is not inverting on both channels: for Tx and Rx, a Zero is represented by a positive pulse, and a One is represented by no pulse (line stays low).
- Interrupt: Sent to ARM platform when 1 char is received into the Rx FIFO (RDR)

Registers values and Programming orders:

```
UCR1 = 0x0085
UCR1[7] = IREN = 1: Enable IR interface
UCR1[0] = UARTEN = 1: Enable UART
UTS = 0x0000
UFCR = 0x0981
TXTL[5:0] = 0x02: Default value
RFDIV[2:0] = 0x3: Divide Input UART clock by 3 (resulting internal clock  is 30 MHz)
RXTL[5:0] = 0x01: Default value
UBIR = 0x0202
UBMR = 0x20BE Baud rate = 115.2 kbit/s with internal clock = 30 MHz
UCR2 = 0x4027
UCR2[14] = IRTS = 1: Ignore level of RTS input signal
UCR2[5] = WS = 1: Characters are 8-bit length
UCR2[2] = TXEN = 1: Enable Rx path
UCR2 [1] = RXEN = 1: Enable Tx path
UCR2[0] = SRST_B = 1: No software reset
UCR3 = 0x0000

         UCR4 = 0x8201
CTSTL[5:0] = 0x20: Default value
UCR4[9] = INVR = 1: Inverted Infrared Reception (because IrDA transceiver is not inverting)
UCR4[1] = DREN = 1: To enable RDR interrupt (sent when one char is received)
```

The UART is ready to send a character as soon as there is a write into UTXD register. And an interrupt will be sent to ARM platform when a character is received.

75.4.7.6.2 Low Speed

This time, we keep the same assumptions but the speed is now 9.6 Kbit/s. So, this baud rate is below the limit (even with a Min. Pulse Duration of 2.5 us) and thus IRSC must be set to 1.

Assumptions:

- Input UART clock = 90 MHz
- Internal clock divider = 3 (divide Input UART clock by 3)

- Baud rate = 9.6 Kbit/s
- IrDA transceiver is not inverting on both channels: for Tx and Rx, a Zero is represented by a positive pulse, and a One is represented by no pulse (line stays low).
- Interrupt: Sent to ARM platform when 1 char is received into the Rx FIFO (RDR).

Registers values and Programming orders:

```
UCR1 = 0x0085
UCR1[7] = IREN = 1: Enable IR interface
UCR1[0] = UARTEN = 1: Enable UART
UFCR = 0x0981
UFCR[15:10] = TXTL[5:0] = 0x02: Default value
RFDIV[2:0] = 0x3: Divide Input UART clock by 3 (resulting internal clock  is 30 MHz)
UFCR[5:0] = RXTL[5:0] = 0x01: Default value
UBIR = 0x00FF
UBMR = 0xC354 Baud rate = 9.6 kbit/s with internal clock = 30 MHz
UCR2 = 0x4027
UCR2[14] = IRTS = 1: Ignore level of RTS input signal
UCR2[5] = WS = 1: Characters are 8-bit length
UCR2[2] = TXEN = 1: Enable Rx path
UCR2 [1] = RXEN = 1: Enable Tx path
UCR2[0] = SRST_B = 1: No software reset
UCR3 = 0x0000
UCR3[1] = INVT = 0: Positive pulse represents 0.
UCR4 = 0x8221
UCR4[15:10] = CTSTL[5:0] = 0x20: Default value
UCR4[9] = INVR = 1: Inverted Infrared Reception (because IrDA transceiver is not inverting)
UCR4[5] = IRSC = 1: Because data rate is below the limit and thus the UART internal clock is
used to measure the pulse duration.
UCR4[1] = DREN = 1: To enable RDR interrupt (sent when one char is received)
```

The UART is now ready to send a character as soon as there is a write into UTXD register. An interrupt will be sent to ARM platform when a character is received.

75.4.8 Low Power Modes

These modes are controlled by the signals *doze_req* and *stop_req*. The control/status/data registers won't change when getting into/out of low power modes.

Table 75-112. UART Low Power State Operation

	Normal State (*doze_req* = 1'b0 & *stop_req* = 1'b0)	Doze State (*doze_req* = 1'b1)		Stop State (*stop_req* = 1'b1)
		DOZE bit = 0	DOZE bit = 1	
UART-Clock	ON	ON	ON	OFF
UART Serial / IrDA	ON	ON	OFF	OFF

75.4.8.1 UART Operation in System Doze Mode

While in Doze State (when *doze_req* input pin is set to 1'b1), the UART behavior depends on the DOZE (UCR1[1]) control bit. While the DOZE bit is negated, the UART serial interface is enabled. While the system is in the Doze State, and the DOZE bit is asserted, the UART is disabled. If the Doze State is entered with the DOZE bit asserted while the UART serial interface was receiving or transmitting data, it will complete the receive/transmit of the current character and signal to the far-end transmitter/receiver to stop sending/receiving.

75.4.8.2 UART Operation in System Stop Mode

The internal baud rate clocks of the transmitter and receiver are gated off if the *stop_req* signal to UART is asserted. Even though the clocks at the input of the UART continue to run during system Stop mode, the UART will not do any transmission or reception.

The following UART interrupts wake the ARM platform processor from STOP mode:

- RTS (RTSD)
- IrDA Asynchronous WAKE (AIRINT)
- Asynchronous WAKE (AWAKE)
- RI (RIDELT in DTE mode only)
- DCD (DCDDELT in DTE mode only)
- DTR (DTRD in DCE mode only)
- DSR (DTRD in DTE mode only)

When an asynchronous WAKE (awake) interrupt exits the ARM platform from STOP mode, make sure that a dummy character is sent first because the first character may not be received correctly.

75.4.8.3 Power Saving Method in UART

The RXEN (UCR2[1]), TXEN (UCR2[2]) and UARTEN (UCR1[0]) bits are set by the user and provide software control of low-power modes.

Setting the UARTEN (UCR1[0]) bit to 0 shuts off the receiver and transmitter logic and the associated clocks.

If the UART is used only in transmit mode, UARTEN and TXEN must be set to 1. If the UART is used only in receive mode, UARTEN and RXEN must be set to 1. Setting TXEN or RXEN to 0 allows to save a lot of power.

75.4.9 UART Operation in System Debug State

The bit UTS [11] controls whether the UART will respond to the input signal *debug_req*, or whether it will continue to run as normal.

If the UART is programmed to respond to *debug_req*:

1. The UART will halt all operations upon detecting the *debug_req* input.
2. A transfer in progress, either to/from a core (using the IP Bus interface) or to/from an external device, will be completed before halting. This means a single byte/word transfer, not an entire FIFO. Reception of any further data from an external device will be disabled.
3. Internal registers will continue to be writable and readable using the IP Bus interface. A read will leave the contents unaffected.
4. The RX FIFO is affected in debug mode in the following way:
   - All writes into the RX FIFO are prevented.
   - The bit RXDBG (UTS[9]) is used to select the readability of the RX FIFO during debug mode:

     RXDBG = 0: hold the read pointer at the location it had upon entering debug mode, and URXD register returns only the data value at that location, no matter how many reads attempted.

     RXDBG = 1, selectable at any time: Allow to read the characters received in Rx FIFO. It will not be possible to re-read previously read locations, nor will it be possible to readjust the read pointer to the value it had prior to entering debug mode.

75.4.10 Reset

This section describes how to reset the block and explains special requirements related to reset.

75.4.10.1 Hardware reset

All of registers, FIFOs, state machines and sequential elements can be reset to their initial values by hardware reset or power on reset.

75.4.10.2 Software reset

The status registers USR1 and USR2, BRM registers UBIR and UBMR, TxFIFO and RxFIFO, and transmitter and receiver state machines can be reset by software reset. Internal logic will remain the software reset signal at active for about 4 *module_clock* cycles. Programmer can follow the following software reset sequence:

1. Clear the $\overline{\text{SRST}}$ bit (UCR2[0])
2. Wait for software reset complete: poll SOFTRST bit (UTS[0]) until it is 0.
3. Re-program baud rate registers: Re-write UBIR and UBMR.

75.4.11 Transfer Error

The UART can generate a transfer error on the peripheral bus in the following cases:

- Core is writing into a read-only register.
- Core is accessing (read or write) an unused location within the assigned address space reserved to UART.
- Core is writing into UTXD register with transmit interface disabled (TXEN=0 or UARTEN=0)
- Core is reading URXD register with receive interface disabled (RXEN=0 or UARTEN=0)

75.4.12 Functional Timing

This section includes timing diagrams for functional signaling.

75.4.12.1 IrDA Mode

According to IrDA specification, the low speed (115.2Kbit/s and below) IR frame format is compatible with UART frame. In this figure, an example data 0x65 is used.

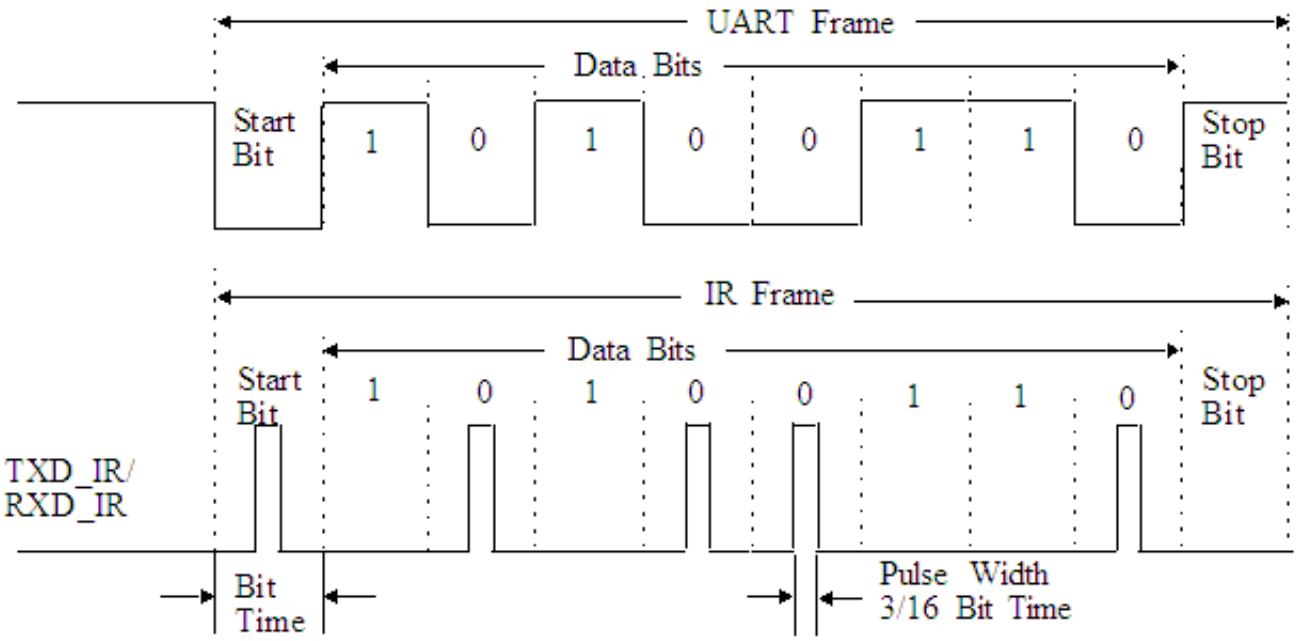


Figure 75-107. Timing diagram of Low Speed IR (<=115.2 Kbit/s) Data Line

75.5 Initialization

75.5.1 Programming the UART in RS-232 mode

As an example, the following sequence can be used to program the UART in order to send and receive characters in RS-232 mode.

Assumptions:

- Input uart clock = 100 MHz
- Baud rate = 921.6Kbps
- Data bits = 8 bits
- Parity = Even
- Stop bits = 1 bit
- Flow control = Hardware

Main program:

1. UCR1 = 0x0001

   Enable the UART.

2. UCR2 = 0x2127

   Set hardware flow control, data format and enable transmitter and receiver.

3. UCR3 = 0x0704

   Set UCR3[RXDMUXSEL] = 1.

4. UCR4 = 0x7C00

   Set CTS trigger level to 31,

5. UFCR = 0x089E

   Set internal clock divider = 5 (divide input uart clock by 5). So the reference clock is 100MHz/5 = 20MHz.

   Set TXTL = 2 and RXTL = 30.

6. UBIR = 0x08FF
7. UBMR = 0x0C34

   In the above two steps, set baud rate to 921.6Kbps based on the 20MHz reference clock.

8. UCR1 = 0x2201>

   Enable the TRDY and RRDY interrupts.

Interrupt service routine for the transmitter:

- Write characters into UTXD

The TRDY interrupt will be automatically de-asserted when the data level of the TxFIFO exceeds the TXTL=2. Note: For the first time the interrupt may be de-asserted after 4 characters are written into the TxFIFO because of the shift register.

Interrupt service routine for the receiver:

- Read characters from URXD

The RRDY interrupt will be automatically de-asserted when the data level of the RxFIFO is below the RXTL=30.

75.5.1.1 References:

- EIA/TIA-232-F Interface Standard

http://www.eia.org, http://www.tiaonline.org/standards

- IrDA Standard

  http://www.irda.org