

UnTx output is received on the UnRx input. Note that the LBE bit should be set before the UART is enabled.

14.3.11 DMA Operation

The UART provides an interface to the µDMA controller with separate channels for transmit and receive. The DMA operation of the UART is enabled through the **UART DMA Control (UARTDMACTL)** register. When DMA operation is enabled, the UART asserts a DMA request on the receive or transmit channel when the associated FIFO can transfer data. For the receive channel, a single transfer request is asserted whenever any data is in the receive FIFO. A burst transfer request is asserted whenever the amount of data in the receive FIFO is at or above the FIFO trigger level configured in the **UARTIFLS** register. For the transmit channel, a single transfer request is asserted whenever there is at least one empty location in the transmit FIFO. The burst request is asserted whenever the transmit FIFO contains fewer characters than the FIFO trigger level. The single and burst DMA transfer requests are handled automatically by the µDMA controller depending on how the DMA channel is configured.

To enable DMA operation for the receive channel, set the RXDMAE bit of the **DMA Control (UARTDMACTL)** register. To enable DMA operation for the transmit channel, set the TXDMAE bit of the **UARTDMACTL** register. The UART can also be configured to stop using DMA for the receive channel if a receive error occurs. If the DMAERR bit of the **UARTDMACR** register is set and a receive error occurs, the DMA receive requests are automatically disabled. This error condition can be cleared by clearing the appropriate UART error interrupt.

If the µDMA is enabled, then the controller triggers an interrupt when the TX FIFO or RX FIFO has reached a trigger point as programmed in the **UARTIFLS** register. The interrupt occurs on the UART interrupt vector. Therefore, if interrupts are used for UART operation and DMA is enabled, the UART interrupt handler must be designed to handle the µDMA completion interrupt.

Note: To trigger an interrupt on transmit completion from the UART's serializer, the EOT bit must be set in the **UARTCTL** register. In this configuration, the transmit interrupt is generated once the FIFO is completely empty and all data including the stop bits have left the transmit serializer. In this case, setting the TXIFLSEL bit in the **UARTIFLS** register is ignored.

When transfers are performed from a FIFO of the UART using the µDMA, and any interrupt is generated from the UART, the UART module's status bit in the **DMA Channel Interrupt Status (DMACHIS)** register must be checked at the end of the interrupt service routine. If the status bit is set, clear the interrupt by writing a 1 to it.

See “Micro Direct Memory Access (µDMA)” on page 585 for more details about programming the µDMA controller.

14.4 Initialization and Configuration

To enable and initialize the UART, the following steps are necessary:

1. Enable the UART module using the **RCGCUART** register (see page 344).
2. Enable the clock to the appropriate GPIO module via the **RCGCGPIO** register (see page 340). To find out which GPIO port to enable, refer to Table 23-5 on page 1351.
3. Set the GPIO AFSEL bits for the appropriate pins (see page 671). To determine which GPIOs to configure, see Table 23-4 on page 1344.
4. Configure the GPIO current level and/or slew rate as specified for the mode selected (see page 673 and page 681).

5. Configure the `PMCn` fields in the **GPIOPCTL** register to assign the UART signals to the appropriate pins (see page 688 and Table 23-5 on page 1351).

To use the UART, the peripheral clock must be enabled by setting the appropriate bit in the **RCGCUART** register (page 344). In addition, the clock to the appropriate GPIO module must be enabled via the **RCGCGPIO** register (page 340) in the System Control module. To find out which GPIO port to enable, refer to Table 23-5 on page 1351.

This section discusses the steps that are required to use a UART module. For this example, the UART clock is assumed to be 20 MHz, and the desired UART configuration is:

- 115200 baud rate
- Data length of 8 bits
- One stop bit
- No parity
- FIFOs disabled
- No interrupts

The first thing to consider when programming the UART is the baud-rate divisor (BRD), because the **UARTIBRD** and **UARTFBRD** registers must be written before the **UARTLCRH** register. Using the equation described in “Baud-Rate Generation” on page 896, the BRD can be calculated:

$$\text{BRD} = 20,000,000 / (16 * 115,200) = 10.8507$$

which means that the `DIVINT` field of the **UARTIBRD** register (see page 914) should be set to 10 decimal or 0xA. The value to be loaded into the **UARTFBRD** register (see page 915) is calculated by the equation:

$$\text{UARTFBRD}[\text{DIVFRAC}] = \text{integer}(0.8507 * 64 + 0.5) = 54$$

With the BRD values in hand, the UART configuration is written to the module in the following order:

1. Disable the UART by clearing the `UARTEN` bit in the **UARTCTL** register.
2. Write the integer portion of the BRD to the **UARTIBRD** register.
3. Write the fractional portion of the BRD to the **UARTFBRD** register.
4. Write the desired serial parameters to the **UARTLCRH** register (in this case, a value of 0x0000.0060).
5. Configure the UART clock source by writing to the **UARTCC** register.
6. Optionally, configure the µDMA channel (see “Micro Direct Memory Access (µDMA)” on page 585) and enable the DMA option(s) in the **UARTDMACTL** register.
7. Enable the UART by setting the `UARTEN` bit in the **UARTCTL** register.

14.5 Register Map

Table 14-3 on page 904 lists the UART registers. The offset listed is a hexadecimal increment to the register's address, relative to that UART's base address:

- UART0: 0x4000.C000
- UART1: 0x4000.D000
- UART2: 0x4000.E000
- UART3: 0x4000.F000
- UART4: 0x4001.0000
- UART5: 0x4001.1000
- UART6: 0x4001.2000
- UART7: 0x4001.3000

The UART module clock must be enabled before the registers can be programmed (see page 344). There must be a delay of 3 system clocks after the UART module clock is enabled before any UART module registers are accessed.

The UART must be disabled (see the **UARTEN** bit in the **UARTCTL** register on page 918) before any of the control registers are reprogrammed. When the UART is disabled during a TX or RX operation, the current transaction is completed prior to the UART stopping.

Table 14-3. UART Register Map

Offset	Name	Type	Reset	Description	See page
0x000	UARTDR	RW	0x0000.0000	UART Data	906
0x004	UARTRSR/UARTECR	RW	0x0000.0000	UART Receive Status/Error Clear	908
0x018	UARTFR	RO	0x0000.0090	UART Flag	911
0x020	UARTILPR	RW	0x0000.0000	UART IrDA Low-Power Register	913
0x024	UARTIBRD	RW	0x0000.0000	UART Integer Baud-Rate Divisor	914
0x028	UARTFBRD	RW	0x0000.0000	UART Fractional Baud-Rate Divisor	915
0x02C	UARTLCRH	RW	0x0000.0000	UART Line Control	916
0x030	UARTCTL	RW	0x0000.0300	UART Control	918
0x034	UARTIFLS	RW	0x0000.0012	UART Interrupt FIFO Level Select	922
0x038	UARTIM	RW	0x0000.0000	UART Interrupt Mask	924
0x03C	UARTRIS	RO	0x0000.0000	UART Raw Interrupt Status	927
0x040	UARTMIS	RO	0x0000.0000	UART Masked Interrupt Status	930
0x044	UARTICR	W1C	0x0000.0000	UART Interrupt Clear	933
0x048	UARTDMACTL	RW	0x0000.0000	UART DMA Control	935
0x0A4	UART9BITADDR	RW	0x0000.0000	UART 9-Bit Self Address	936
0x0A8	UART9BITAMASK	RW	0x0000.00FF	UART 9-Bit Self Address Mask	937
0xFC0	UARTPP	RO	0x0000.0003	UART Peripheral Properties	938
0xFC8	UARTCC	RW	0x0000.0000	UART Clock Configuration	939
0xFD0	UARTPeriphID4	RO	0x0000.0000	UART Peripheral Identification 4	940
0xFD4	UARTPeriphID5	RO	0x0000.0000	UART Peripheral Identification 5	941
0xFD8	UARTPeriphID6	RO	0x0000.0000	UART Peripheral Identification 6	942

Table 14-3. UART Register Map (continued)

Offset	Name	Type	Reset	Description	See page
0xFDC	UARTPeriphID7	RO	0x0000.0000	UART Peripheral Identification 7	943
0xFE0	UARTPeriphID0	RO	0x0000.0060	UART Peripheral Identification 0	944
0xFE4	UARTPeriphID1	RO	0x0000.0000	UART Peripheral Identification 1	945
0xFE8	UARTPeriphID2	RO	0x0000.0018	UART Peripheral Identification 2	946
0xFEC	UARTPeriphID3	RO	0x0000.0001	UART Peripheral Identification 3	947
0xFF0	UARTPCellID0	RO	0x0000.000D	UART PrimeCell Identification 0	948
0xFF4	UARTPCellID1	RO	0x0000.00F0	UART PrimeCell Identification 1	949
0xFF8	UARTPCellID2	RO	0x0000.0005	UART PrimeCell Identification 2	950
0xFFC	UARTPCellID3	RO	0x0000.00B1	UART PrimeCell Identification 3	951

14.6 Register Descriptions

The remainder of this section lists and describes the UART registers, in numerical order by address offset.

Register 1: UART Data (UARTDR), offset 0x000

Important: This register is read-sensitive. See the register description for details.

This register is the data register (the interface to the FIFOs).

For transmitted data, if the FIFO is enabled, data written to this location is pushed onto the transmit FIFO. If the FIFO is disabled, data is stored in the transmitter holding register (the bottom word of the transmit FIFO). A write to this register initiates a transmission from the UART.

For received data, if the FIFO is enabled, the data byte and the 4-bit status (break, frame, parity, and overrun) is pushed onto the 12-bit wide receive FIFO. If the FIFO is disabled, the data byte and status are stored in the receiving holding register (the bottom word of the receive FIFO). The received data can be retrieved by reading this register.

UART Data (UARTDR)

UART0 base: 0x4000.C000

UART1 base: 0x4000.D000

UART2 base: 0x4000.E000

UART3 base: 0x4000.F000

UART4 base: 0x4001.0000

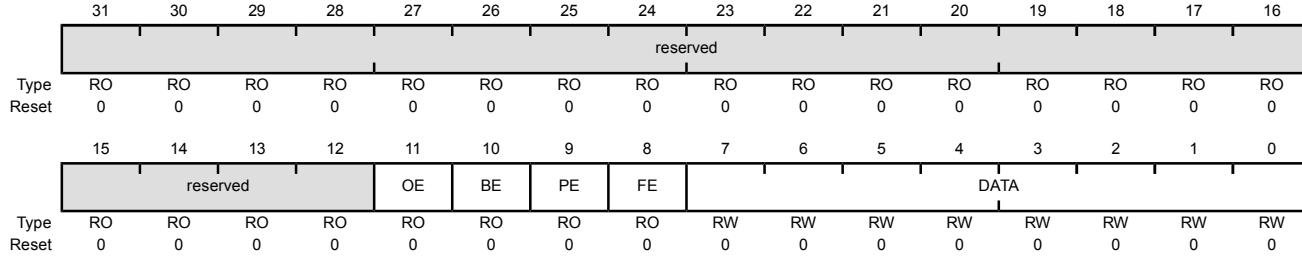
UART5 base: 0x4001.1000

UART6 base: 0x4001.2000

UART7 base: 0x4001.3000

Offset 0x000

Type RW, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:12	reserved	RO	0x0000.0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
11	OE	RO	0	UART Overrun Error
		Value	Description	
		0	No data has been lost due to a FIFO overrun.	
		1	New data was received when the FIFO was full, resulting in data loss.	

Bit/Field	Name	Type	Reset	Description						
10	BE	RO	0	<p>UART Break Error</p> <table> <thead> <tr> <th>Value</th><th>Description</th></tr> </thead> <tbody> <tr> <td>0</td><td>No break condition has occurred</td></tr> <tr> <td>1</td><td>A break condition has been detected, indicating that the receive data input was held Low for longer than a full-word transmission time (defined as start, data, parity, and stop bits).</td></tr> </tbody> </table> <p>In FIFO mode, this error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO. The next character is only enabled after the received data input goes to a 1 (marking state), and the next valid start bit is received.</p>	Value	Description	0	No break condition has occurred	1	A break condition has been detected, indicating that the receive data input was held Low for longer than a full-word transmission time (defined as start, data, parity, and stop bits).
Value	Description									
0	No break condition has occurred									
1	A break condition has been detected, indicating that the receive data input was held Low for longer than a full-word transmission time (defined as start, data, parity, and stop bits).									
9	PE	RO	0	<p>UART Parity Error</p> <table> <thead> <tr> <th>Value</th><th>Description</th></tr> </thead> <tbody> <tr> <td>0</td><td>No parity error has occurred</td></tr> <tr> <td>1</td><td>The parity of the received data character does not match the parity defined by bits 2 and 7 of the UARTLCRH register.</td></tr> </tbody> </table> <p>In FIFO mode, this error is associated with the character at the top of the FIFO.</p>	Value	Description	0	No parity error has occurred	1	The parity of the received data character does not match the parity defined by bits 2 and 7 of the UARTLCRH register.
Value	Description									
0	No parity error has occurred									
1	The parity of the received data character does not match the parity defined by bits 2 and 7 of the UARTLCRH register.									
8	FE	RO	0	<p>UART Framing Error</p> <table> <thead> <tr> <th>Value</th><th>Description</th></tr> </thead> <tbody> <tr> <td>0</td><td>No framing error has occurred</td></tr> <tr> <td>1</td><td>The received character does not have a valid stop bit (a valid stop bit is 1).</td></tr> </tbody> </table>	Value	Description	0	No framing error has occurred	1	The received character does not have a valid stop bit (a valid stop bit is 1).
Value	Description									
0	No framing error has occurred									
1	The received character does not have a valid stop bit (a valid stop bit is 1).									
7:0	DATA	RW	0x00	<p>Data Transmitted or Received</p> <p>Data that is to be transmitted via the UART is written to this field. When read, this field contains the data that was received by the UART.</p>						

Register 3: UART Flag (UARTFR), offset 0x018

The **UARTFR** register is the flag register. After reset, the TXFF, RXFF, and BUSY bits are 0, and TXFE and RXFE bits are 1. The CTS bit indicate the modem flow control. Note that the modem bits are only implemented on UART1 and are reserved on UART0 and UART2.

UART Flag (UARTFR)

UART0 base: 0x4000.C000

UART1 base: 0x4000.D000

UART2 base: 0x4000.E000

UART3 base: 0x4000.F000

UART4 base: 0x4001.0000

UART5 base: 0x4001.1000

UART6 base: 0x4001.2000

UART7 base: 0x4001.3000

Offset 0x018

Type RO, reset 0x0000.0090

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved								TXFE	RXFF	TXFF	RXFE	BUSY	reserved		CTS
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0

Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x0000.00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7	TXFE	RO	1	UART Transmit FIFO Empty The meaning of this bit depends on the state of the FEN bit in the UARTLCRH register.
		Value	Description	
	0		The transmitter has data to transmit.	
	1		If the FIFO is disabled (FEN is 0), the transmit holding register is empty. If the FIFO is enabled (FEN is 1), the transmit FIFO is empty.	
6	RXFF	RO	0	UART Receive FIFO Full The meaning of this bit depends on the state of the FEN bit in the UARTLCRH register.
		Value	Description	
	0		The receiver can receive data.	
	1		If the FIFO is disabled (FEN is 0), the receive holding register is full. If the FIFO is enabled (FEN is 1), the receive FIFO is full.	

Bit/Field	Name	Type	Reset	Description						
5	TXFF	RO	0	<p>UART Transmit FIFO Full</p> <p>The meaning of this bit depends on the state of the FEN bit in the UARTLCRH register.</p> <table> <thead> <tr> <th>Value</th><th>Description</th></tr> </thead> <tbody> <tr> <td>0</td><td>The transmitter is not full.</td></tr> <tr> <td>1</td><td>If the FIFO is disabled (FEN is 0), the transmit holding register is full. If the FIFO is enabled (FEN is 1), the transmit FIFO is full.</td></tr> </tbody> </table>	Value	Description	0	The transmitter is not full.	1	If the FIFO is disabled (FEN is 0), the transmit holding register is full. If the FIFO is enabled (FEN is 1), the transmit FIFO is full.
Value	Description									
0	The transmitter is not full.									
1	If the FIFO is disabled (FEN is 0), the transmit holding register is full. If the FIFO is enabled (FEN is 1), the transmit FIFO is full.									
4	RXFE	RO	1	<p>UART Receive FIFO Empty</p> <p>The meaning of this bit depends on the state of the FEN bit in the UARTLCRH register.</p> <table> <thead> <tr> <th>Value</th><th>Description</th></tr> </thead> <tbody> <tr> <td>0</td><td>The receiver is not empty.</td></tr> <tr> <td>1</td><td>If the FIFO is disabled (FEN is 0), the receive holding register is empty. If the FIFO is enabled (FEN is 1), the receive FIFO is empty.</td></tr> </tbody> </table>	Value	Description	0	The receiver is not empty.	1	If the FIFO is disabled (FEN is 0), the receive holding register is empty. If the FIFO is enabled (FEN is 1), the receive FIFO is empty.
Value	Description									
0	The receiver is not empty.									
1	If the FIFO is disabled (FEN is 0), the receive holding register is empty. If the FIFO is enabled (FEN is 1), the receive FIFO is empty.									
3	BUSY	RO	0	<p>UART Busy</p> <table> <thead> <tr> <th>Value</th><th>Description</th></tr> </thead> <tbody> <tr> <td>0</td><td>The UART is not busy.</td></tr> <tr> <td>1</td><td>The UART is busy transmitting data. This bit remains set until the complete byte, including all stop bits, has been sent from the shift register.</td></tr> </tbody> </table> <p>This bit is set as soon as the transmit FIFO becomes non-empty (regardless of whether UART is enabled).</p>	Value	Description	0	The UART is not busy.	1	The UART is busy transmitting data. This bit remains set until the complete byte, including all stop bits, has been sent from the shift register.
Value	Description									
0	The UART is not busy.									
1	The UART is busy transmitting data. This bit remains set until the complete byte, including all stop bits, has been sent from the shift register.									
2:1	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.						
0	CTS	RO	0	<p>Clear To Send</p> <table> <thead> <tr> <th>Value</th><th>Description</th></tr> </thead> <tbody> <tr> <td>0</td><td>The U1CTS signal is not asserted.</td></tr> <tr> <td>1</td><td>The U1CTS signal is asserted.</td></tr> </tbody> </table>	Value	Description	0	The U1CTS signal is not asserted.	1	The U1CTS signal is asserted.
Value	Description									
0	The U1CTS signal is not asserted.									
1	The U1CTS signal is asserted.									

Register 5: UART Integer Baud-Rate Divisor (UARTIBRD), offset 0x024

The **UARTIBRD** register is the integer part of the baud-rate divisor value. All the bits are cleared on reset. The minimum possible divide ratio is 1 (when **UARTIBRD**=0), in which case the **UARTFBRD** register is ignored. When changing the **UARTIBRD** register, the new value does not take effect until transmission/reception of the current character is complete. Any changes to the baud-rate divisor must be followed by a write to the **UARTLCRH** register. See “Baud-Rate Generation” on page 896 for configuration details.

UART Integer Baud-Rate Divisor (UARTIBRD)

UART0 base: 0x4000.C000

UART1 base: 0x4000.D000

UART2 base: 0x4000.E000

UART3 base: 0x4000.F000

UART4 base: 0x4001.0000

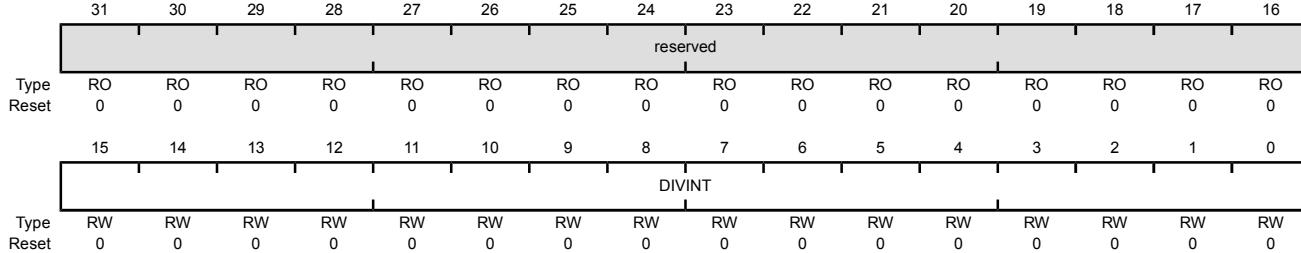
UART5 base: 0x4001.1000

UART6 base: 0x4001.2000

UART7 base: 0x4001.3000

Offset 0x024

Type RW, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:16	reserved	RO	0x0000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
15:0	DIVINT	RW	0x0000	Integer Baud-Rate Divisor

Register 6: UART Fractional Baud-Rate Divisor (UARTFBRD), offset 0x028

The **UARTFBRD** register is the fractional part of the baud-rate divisor value. All the bits are cleared on reset. When changing the **UARTFBRD** register, the new value does not take effect until transmission/reception of the current character is complete. Any changes to the baud-rate divisor must be followed by a write to the **UARTLCRH** register. See “Baud-Rate Generation” on page 896 for configuration details.

UART Fractional Baud-Rate Divisor (UARTFBRD)

UART0 base: 0x4000.C000

UART1 base: 0x4000.D000

UART2 base: 0x4000.E000

UART3 base: 0x4000.F000

UART4 base: 0x4001.0000

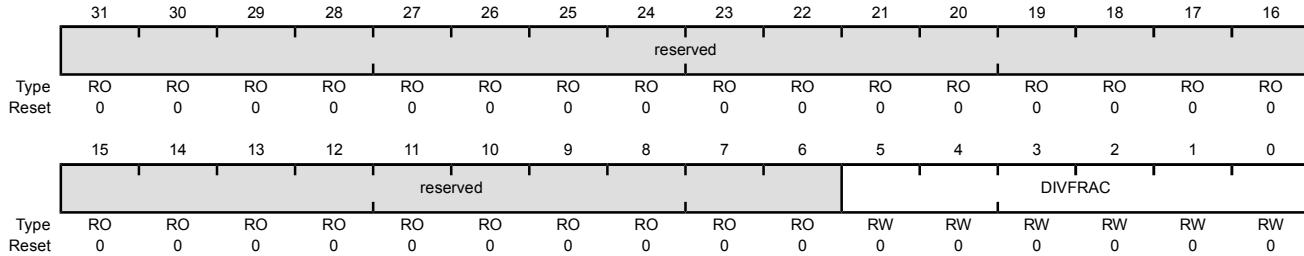
UART5 base: 0x4001.1000

UART6 base: 0x4001.2000

UART7 base: 0x4001.3000

Offset 0x028

Type RW, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:6	reserved	RO	0x0000.000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
5:0	DIVFRAC	RW	0x0	Fractional Baud-Rate Divisor

Register 7: UART Line Control (UARTLCRH), offset 0x02C

The **UARTLCRH** register is the line control register. Serial parameters such as data length, parity, and stop bit selection are implemented in this register.

When updating the baud-rate divisor (**UARTIBRD** and/or **UARTIFRD**), the **UARTLCRH** register must also be written. The write strobe for the baud-rate divisor registers is tied to the **UARTLCRH** register.

UART Line Control (UARTLCRH)

UART0 base: 0x4000.C000

UART1 base: 0x4000.D000

UART2 base: 0x4000.E000

UART3 base: 0x4000.F000

UART4 base: 0x4001.0000

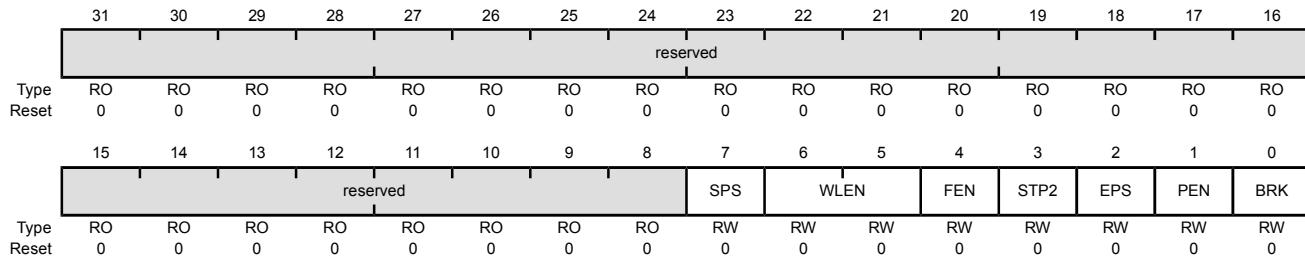
UART5 base: 0x4001.1000

UART6 base: 0x4001.2000

UART7 base: 0x4001.3000

Offset 0x02C

Type RW, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description										
31:8	reserved	RO	0x0000.00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.										
7	SPS	RW	0	UART Stick Parity Select When bits 1, 2, and 7 of UARTLCRH are set, the parity bit is transmitted and checked as a 0. When bits 1 and 7 are set and 2 is cleared, the parity bit is transmitted and checked as a 1. When this bit is cleared, stick parity is disabled.										
6:5	WLEN	RW	0x0	UART Word Length The bits indicate the number of data bits transmitted or received in a frame as follows: <table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0x0</td> <td>5 bits (default)</td> </tr> <tr> <td>0x1</td> <td>6 bits</td> </tr> <tr> <td>0x2</td> <td>7 bits</td> </tr> <tr> <td>0x3</td> <td>8 bits</td> </tr> </tbody> </table>	Value	Description	0x0	5 bits (default)	0x1	6 bits	0x2	7 bits	0x3	8 bits
Value	Description													
0x0	5 bits (default)													
0x1	6 bits													
0x2	7 bits													
0x3	8 bits													

Bit/Field	Name	Type	Reset	Description
4	FEN	RW	0	UART Enable FIFOs Value Description 0 The FIFOs are disabled (Character mode). The FIFOs become 1-byte-deep holding registers. 1 The transmit and receive FIFO buffers are enabled (FIFO mode).
3	STP2	RW	0	UART Two Stop Bits Select Value Description 0 One stop bit is transmitted at the end of a frame. 1 Two stop bits are transmitted at the end of a frame. The receive logic does not check for two stop bits being received. When in 7816 smartcard mode (the SMART bit is set in the UARTCTL register), the number of stop bits is forced to 2.
2	EPS	RW	0	UART Even Parity Select Value Description 0 Odd parity is performed, which checks for an odd number of 1s. 1 Even parity generation and checking is performed during transmission and reception, which checks for an even number of 1s in data and parity bits. This bit has no effect when parity is disabled by the PEN bit.
1	PEN	RW	0	UART Parity Enable Value Description 0 Parity is disabled and no parity bit is added to the data frame. 1 Parity checking and generation is enabled.
0	BRK	RW	0	UART Send Break Value Description 0 Normal use. 1 A Low level is continually output on the UnTx signal, after completing transmission of the current character. For the proper execution of the break command, software must set this bit for at least two frames (character periods).

Register 8: UART Control (UARTCTL), offset 0x030

The **UARTCTL** register is the control register. All the bits are cleared on reset except for the Transmit Enable (TXE) and Receive Enable (RXE) bits, which are set.

To enable the UART module, the **UARTEN** bit must be set. If software requires a configuration change in the module, the **UARTEN** bit must be cleared before the configuration changes are written. If the UART is disabled during a transmit or receive operation, the current transaction is completed prior to the UART stopping.

Note: The **UARTCTL** register should not be changed while the UART is enabled or else the results are unpredictable. The following sequence is recommended for making changes to the **UARTCTL** register.

1. Disable the UART.
2. Wait for the end of transmission or reception of the current character.
3. Flush the transmit FIFO by clearing bit 4 (FEN) in the line control register (**UARTLCRH**).
4. Reprogram the control register.
5. Enable the UART.

UART Control (UARTCTL)

UART0 base: 0x4000.C000

UART1 base: 0x4000.D000

UART2 base: 0x4000.E000

UART3 base: 0x4000.F000

UART4 base: 0x4001.0000

UART5 base: 0x4001.1000

UART6 base: 0x4001.2000

UART7 base: 0x4001.3000

Offset 0x030

Type RW, reset 0x0000.0300

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

Bit/Field	Name	Type	Reset	Description
31:16	reserved	RO	0x0000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
15	CTSEN	RW	0	Enable Clear To Send
	Value	Description		
	0	CTS hardware flow control is disabled.		
	1	CTS hardware flow control is enabled. Data is only transmitted when the U1CTS signal is asserted.		

Bit/Field	Name	Type	Reset	Description						
14	RTSEN	RW	0	<p>Enable Request to Send</p> <table> <thead> <tr> <th>Value</th><th>Description</th></tr> </thead> <tbody> <tr> <td>0</td><td>RTS hardware flow control is disabled.</td></tr> <tr> <td>1</td><td>RTS hardware flow control is enabled. Data is only requested (by asserting U1RTS) when the receive FIFO has available entries.</td></tr> </tbody> </table>	Value	Description	0	RTS hardware flow control is disabled.	1	RTS hardware flow control is enabled. Data is only requested (by asserting U1RTS) when the receive FIFO has available entries.
Value	Description									
0	RTS hardware flow control is disabled.									
1	RTS hardware flow control is enabled. Data is only requested (by asserting U1RTS) when the receive FIFO has available entries.									
13:12	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.						
11	RTS	RW	0	<p>Request to Send</p> <p>When RTSEN is clear, the status of this bit is reflected on the U1RTS signal. If RTSEN is set, this bit is ignored on a write and should be ignored on read.</p>						
10	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.						
9	RXE	RW	1	<p>UART Receive Enable</p> <table> <thead> <tr> <th>Value</th><th>Description</th></tr> </thead> <tbody> <tr> <td>0</td><td>The receive section of the UART is disabled.</td></tr> <tr> <td>1</td><td>The receive section of the UART is enabled.</td></tr> </tbody> </table> <p>If the UART is disabled in the middle of a receive, it completes the current character before stopping.</p> <p>Note: To enable reception, the UARTEN bit must also be set.</p>	Value	Description	0	The receive section of the UART is disabled.	1	The receive section of the UART is enabled.
Value	Description									
0	The receive section of the UART is disabled.									
1	The receive section of the UART is enabled.									
8	TXE	RW	1	<p>UART Transmit Enable</p> <table> <thead> <tr> <th>Value</th><th>Description</th></tr> </thead> <tbody> <tr> <td>0</td><td>The transmit section of the UART is disabled.</td></tr> <tr> <td>1</td><td>The transmit section of the UART is enabled.</td></tr> </tbody> </table> <p>If the UART is disabled in the middle of a transmission, it completes the current character before stopping.</p> <p>Note: To enable transmission, the UARTEN bit must also be set.</p>	Value	Description	0	The transmit section of the UART is disabled.	1	The transmit section of the UART is enabled.
Value	Description									
0	The transmit section of the UART is disabled.									
1	The transmit section of the UART is enabled.									
7	LBE	RW	0	<p>UART Loop Back Enable</p> <table> <thead> <tr> <th>Value</th><th>Description</th></tr> </thead> <tbody> <tr> <td>0</td><td>Normal operation.</td></tr> <tr> <td>1</td><td>The UnTx path is fed through the UnRx path.</td></tr> </tbody> </table>	Value	Description	0	Normal operation.	1	The UnTx path is fed through the UnRx path.
Value	Description									
0	Normal operation.									
1	The UnTx path is fed through the UnRx path.									
6	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.						

Bit/Field	Name	Type	Reset	Description						
5	HSE	RW	0	<p>High-Speed Enable</p> <table> <tr> <td>Value</td><td>Description</td></tr> <tr> <td>0</td><td>The UART is clocked using the system clock divided by 16.</td></tr> <tr> <td>1</td><td>The UART is clocked using the system clock divided by 8.</td></tr> </table> <p>Note: System clock used is also dependent on the baud-rate divisor configuration (see page 914) and page 915.</p> <p>The state of this bit has no effect on clock generation in ISO 7816 smart card mode (the SMART bit is set).</p>	Value	Description	0	The UART is clocked using the system clock divided by 16.	1	The UART is clocked using the system clock divided by 8.
Value	Description									
0	The UART is clocked using the system clock divided by 16.									
1	The UART is clocked using the system clock divided by 8.									
4	EOT	RW	0	<p>End of Transmission</p> <p>This bit determines the behavior of the TXRIS bit in the UARTRIS register.</p> <table> <tr> <td>Value</td><td>Description</td></tr> <tr> <td>0</td><td>The TXRIS bit is set when the transmit FIFO condition specified in UARTIFLS is met.</td></tr> <tr> <td>1</td><td>The TXRIS bit is set only after all transmitted data, including stop bits, have cleared the serializer.</td></tr> </table>	Value	Description	0	The TXRIS bit is set when the transmit FIFO condition specified in UARTIFLS is met.	1	The TXRIS bit is set only after all transmitted data, including stop bits, have cleared the serializer.
Value	Description									
0	The TXRIS bit is set when the transmit FIFO condition specified in UARTIFLS is met.									
1	The TXRIS bit is set only after all transmitted data, including stop bits, have cleared the serializer.									
3	SMART	RW	0	<p>ISO 7816 Smart Card Support</p> <table> <tr> <td>Value</td><td>Description</td></tr> <tr> <td>0</td><td>Normal operation.</td></tr> <tr> <td>1</td><td>The UART operates in Smart Card mode.</td></tr> </table> <p>The application must ensure that it sets 8-bit word length (WLEN set to 0x3) and even parity (PEN set to 1, EPS set to 1, SPS set to 0) in UARTLCRH when using ISO 7816 mode.</p> <p>In this mode, the value of the STP2 bit in UARTLCRH is ignored and the number of stop bits is forced to 2. Note that the UART does not support automatic retransmission on parity errors. If a parity error is detected on transmission, all further transmit operations are aborted and software must handle retransmission of the affected byte or message.</p>	Value	Description	0	Normal operation.	1	The UART operates in Smart Card mode.
Value	Description									
0	Normal operation.									
1	The UART operates in Smart Card mode.									
2	SIRLP	RW	0	<p>UART SIR Low-Power Mode</p> <p>This bit selects the IrDA encoding mode.</p> <table> <tr> <td>Value</td><td>Description</td></tr> <tr> <td>0</td><td>Low-level bits are transmitted as an active High pulse with a width of 3/16th of the bit period.</td></tr> <tr> <td>1</td><td>The UART operates in SIR Low-Power mode. Low-level bits are transmitted with a pulse width which is 3 times the period of the IrLPBaud16 input signal, regardless of the selected bit rate.</td></tr> </table> <p>Setting this bit uses less power, but might reduce transmission distances. See page 913 for more information.</p>	Value	Description	0	Low-level bits are transmitted as an active High pulse with a width of 3/16th of the bit period.	1	The UART operates in SIR Low-Power mode. Low-level bits are transmitted with a pulse width which is 3 times the period of the IrLPBaud16 input signal, regardless of the selected bit rate.
Value	Description									
0	Low-level bits are transmitted as an active High pulse with a width of 3/16th of the bit period.									
1	The UART operates in SIR Low-Power mode. Low-level bits are transmitted with a pulse width which is 3 times the period of the IrLPBaud16 input signal, regardless of the selected bit rate.									

Bit/Field	Name	Type	Reset	Description
1	SIREN	RW	0	UART SIR Enable
				Value Description
			0	Normal operation.
			1	The IrDA SIR block is enabled, and the UART will transmit and receive data using SIR protocol.
0	UARTEN	RW	0	UART Enable
				Value Description
			0	The UART is disabled.
			1	The UART is enabled.
				If the UART is disabled in the middle of transmission or reception, it completes the current character before stopping.

Register 18: UART Clock Configuration (UARTCC), offset 0xFC8

The **UARTCC** register controls the baud clock source for the UART module. For more information, see the section called “Communication Clock Sources” on page 222.

Note: If the PIOSC is used for the UART baud clock, the system clock frequency must be at least 9 MHz in Run mode.

UART Clock Configuration (UARTCC)

UART0 base: 0x4000.C000

UART1 base: 0x4000.D000

UART2 base: 0x4000.E000

UART3 base: 0x4000.F000

UART4 base: 0x4001.0000

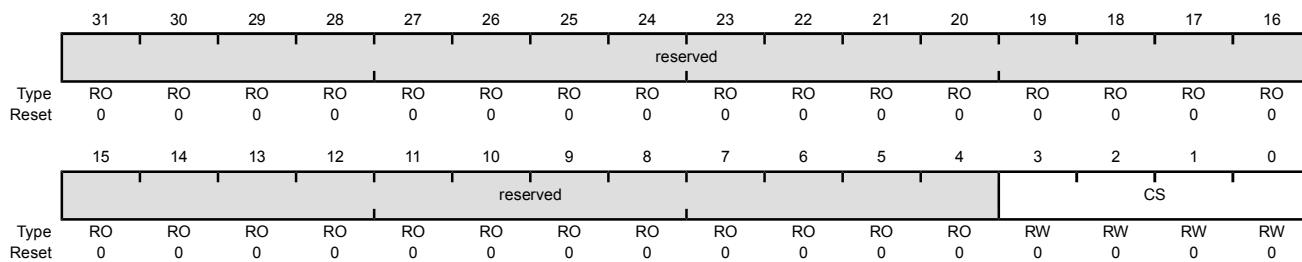
UART5 base: 0x4001.1000

UART6 base: 0x4001.2000

UART7 base: 0x4001.3000

Offset 0xFC8

Type RW, reset 0x0000.0000



Bit/Field Name Type Reset Description

31:4 reserved RO 0x0000.0000 Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

3:0 CS RW 0 UART Baud Clock Source

The following table specifies the source that generates for the UART baud clock:

Value	Description
0x0	System clock (based on clock source and divisor factor)
0x1-0x4	reserved
0x5	PIOSC
0x5-0xF	Reserved

Register 19: UART Peripheral Identification 4 (UARTPeriphID4), offset 0xFD0

The **UARTPeriphIDn** registers are hard-coded and the fields within the registers determine the reset values.

UART Peripheral Identification 4 (UARTPeriphID4)

UART0 base: 0x4000.C000

UART1 base: 0x4000.D000

UART2 base: 0x4000.E000

UART3 base: 0x4000.F000

UART4 base: 0x4001.0000

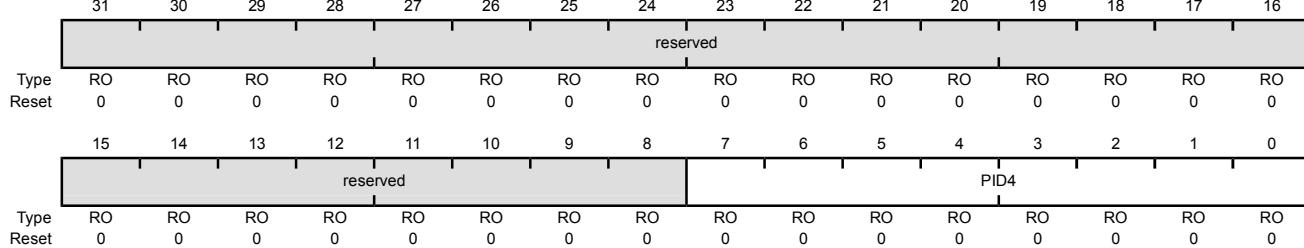
UART5 base: 0x4001.1000

UART6 base: 0x4001.2000

UART7 base: 0x4001.3000

Offset 0xFD0

Type RO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x0000.00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID4	RO	0x00	UART Peripheral ID Register [7:0] Can be used by software to identify the presence of this peripheral.