eSi-UART

1 Contents

2 Overview

The eSi-UART core can be used to implement asynchronous serial communications. It supports the following features:

- 7 or 8 data bits.
- 1 or 2 stop bits.
- Parity bit (None / Even / Odd / Mark / Space).
- Optional RTS/CTS flow control.
- Programmable bit rate.
- Optional ISO 7816-3 T=0 and T=1 support, with NACK and retry.
- Configurable TX and RX FIFO.
- Programmable receive timeout and transmit guard time.
- AMBA 3 APB slave interface.
- DMA flow-control interface.

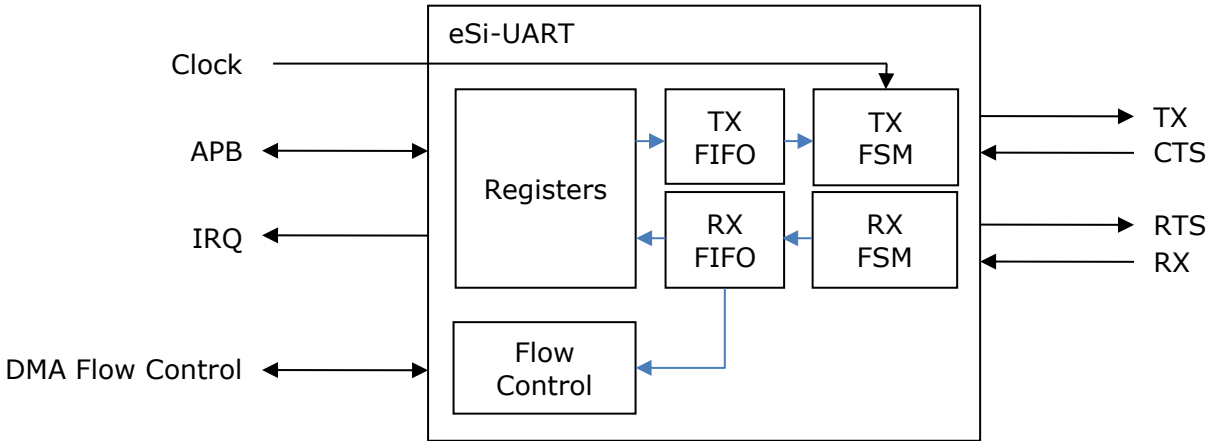


Figure 1: eSi-UART

3 Hardware Interface

Module Name	esi_apb_uart
HDL	Verilog
Technology	Generic
Source Files	esi_apb_uart.v, esi_fifo.v, esi_peripheral_flow_control.v, esi_global_include.v, esi_global_cfg_include.v, esi_i2c_cfg_include.v

Configuration Option	Values	Description
ISO_7816_ENABLED	TRUE, FALSE	Determines whether support for ISO 7816 functionality is implemented

Table 1: Configuration Options – Defined in esi_uart_cfg_include,v

Port	Type	Description
default_cycles_per_bit	Integer	Specifies the reset value of the cycles_per_bit register
tx_fifo_depth	Integer	Specifies the depth of the TX FIFO. Minimum of 2 and value must be a power of 2
rx_fifo_depth	Integer	Specifies the depth of the RX FIFO. Minimum of 2 and value must be a power of 2
apb_data_width	Integer	Width of APB data bus
apb_address_width	Integer	Width of APB address bus

Table 2: Parameters

Port	Direction	Width	Description
clk	Input	1	Clock used for serial data transmission and reception. This clock must be enabled when cactive is asserted. This clock will be divided by the value in the cycles_per_bit register. This clock must be the same frequency and synchronous to pclk.
pclk	Input	1	APB clock
presetn	Input	1	APB reset, active-low
paddr	Input	apb_address_width	APB address. Only 8 LSBs are used
psel	Input	1	APB slave select
penable	Input	1	APB enable
pwrite	Input	1	APB write
pdebug	Input	1	APB noninvasive debug read
pwdata	Input	apb_data_width	APB write data
rx	Input	1	Receive data
cts_n	Input	1	Clear to send, active-low
tx_ack	Input	1	Acknowledges tx_ready after transfer complete
rx_ack	Input	1	Acknowledges rx_ready after transfer complete
cactive	Output	1	Clock active
pready	Output	1	APB ready
prdata	Output	apb_data_width	APB read data
pslverr	Output	1	APB slave error
tx	Output	1	Transmit data
rts_n	Output	1	Ready to send, active-low
duplex	Output	1	Indicates full duplex (0) or half-duplex (1) operation. Only implemented if

			ISO_7816_ENABLED is TRUE.
interrupt_n	Output	1	Interrupt request, active-low
tx_ready	Output	1	Indicates device can accept new data
rx_ready	Output	1	Indicates device has data to be read

Table 3: I/O Ports

For complete details of the APB signals, please refer to the AMBA 3 APB Protocol v1.0 Specification available at:

http://www.arm.com/products/system-ip/amba/amba-open-specifications.php

The UART does not include internal synchronizing flip-flops. These should be implemented externally for the rx and cts_n ports if the transmitting clock domain is asynchronous to clk.

The UART does not include any internal filtering, other than for start detection. Should signal integrity issues be a potential concern, the rx input should be filtered externally to the UART.

4 Software Interface

4.1 Register Map

Register	Address offset	Access	Description
tx_data	0x00	W	Transmit data register
rx_data	0x04	R	Receive data register
status	0x08	R/W	Status register
control	0x0c	R/W	Control register
cycles_per_bit	0x10	R/W	Cycles per bit register
retries	0x14	R/W	Retries register
txae_thresh	0x18	R/W	Transmit FIFO almost empty threshold
rxaf_thresh	0x1c	R/W	Receive FIFO almost full threshold
tx_error_start	0x20	R/W	Transmit error window start register
tx_error_length	0x24	R/W	Transmit error window length register
rx_error_start	0x28	R/W	Receive error window start register
rx_error_length	0x2c	R/W	Receiver error window length register
tx_count	0x30	R	Count of entries used in TX FIFO
rx_count	0x34	R	Count of entries used in RX FIFO
tx_guard	0x38	R/W	Transmit guard time
rx_timeout	0x3c	R/W	Receive timeout
rx_timeout_cnt	0x40	R/W	Receive timeout counter

Table 4: Register Map

4.1.1 Transmit Data Register

Data to be transmitted over the serial interface should be written to the lower 8 bits of the transmit data register, in order to be written to the TX FIFO. The transmit data register should not be written to while the TXF bit in the status register is set, otherwise data loss may occur.

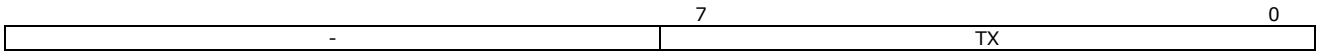


Figure 2: Format of the tx_data register

4.1.2 Receive Data Register

Data that is received over the serial interface and stored in the RX FIFO can be read in the lower 8 bits of the receive data register. A read from the received data register will remove the data from the RX FIFO, unless pdebug is high. Setting pdebug high allows a debugger to display the first element in the RX FIFO, without affecting the program being debugged.

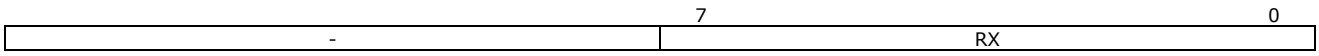
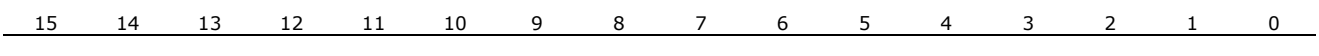


Figure 3: Format of the rx_data register

4.1.3 Status Register

The status register contains a selection of flags that indicate the current status of the UART. To clear a bit in the status register, write a 1 to it. Writing 0 will leave it unchanged.



RXAF	TXAE	-	RT	RLR	PE	FE	RXB	RXP	RXO	RXF	RXE	TXO	TXF	TXE

Figure 4: Format of the `status` register

Register	Values	Description
TXE	0 - Not empty 1 - Empty	Transmit FIFO empty
TXF	0 - Not full 1 - Full	Transmit FIFO full
TXO	0 - No overflow 1 - Overflow	Transmit FIFO overflow. Sticky
RXE	0 - Not empty 1 - Empty	Receive FIFO empty
RXF	0 - Not full 1 - Full	Receive FIFO full
RXO	0 - No overflow 1 - Overflow	Receive FIFO overflow. Sticky
RXP	0 - Parity bit was 0 1 - Parity bit was 1	Received parity bit for data at front of FIFO. This should be read before the corresponding data from the RX FIFO
RXB	0 - Break not received 1 - Break received	Received break. Sticky. No data is written to the RX FIFO when a break is received
FE	0 - No error 1 - Framing error	Framing error. Sticky. Received data is not written to the RX FIFO when a framing error occurs
PE	0 - No error 1 - Parity error	Parity error. Sticky
RLR	0 – Not reached 1 – Reached	Retry limited reached. Only implemented if `ISO_7816_ENABLED` is `TRUE`. Sticky
RT	0 – No timeout 1 – Timeout	Receive timeout. Only implemented if `ISO_7816_ENABLED` is `TRUE`. Sticky
TXAE	0 – Not almost empty 1 – Almost empty	Transmit FIFO almost empty
RXAF	0 – Not almost full 1 – Almost full	Receive FIFO almost full

Table 5: Fields of the `status` register

If an overflow or underflow occurs, the contents of the corresponding FIFO are undefined. The FIFOs should then be reset by writing `control.RF`.

4.1.4 Control Register

The control register contains a selection of flags that control the operation of the UART.

15	14	13	12	11	10	9	8	7	6	5 3	2	1	0
D	RTIE	RF	END	RLRIE	RXBIE	RXIE	TXIE	TXB	FC	PB	DB	SB	E

Figure 5: Format of the `control` register

Register	Values	Description
E	0 - Disabled 1 - Enabled	Enables the UART. When disabled, data will not be received or transmitted
SB	0 - 1 stop bit 1 - 2 stop bits	Number of stop bits
DB	0 - 8 data bits	Number of data bits

	1 - 7 data bits	
PB	0 - No parity bit 1 - Even parity 2 - Odd parity 3 - Mark 4 - Space 5 – ISO 7816 T=0 DC 6 – ISO 7816 T=0 IC	Parity bit control. Mode 5 and 6 are only implemented if ISO_7816_ENABLED is TRUE.
FC	0 - No flow control 1 - RTS/CTS	Flow control. With RTS/CTS flow control, the UART will not transmit until cts_n is low and rts_n will be set high when the RX FIFO is almost full
TXB	0 - Do not send break 1 - Send break	Transmit break. When set, the transmit data line will be held low, signalling a break. The break will be signalled until this bit is cleared
TXIE	0 - Disabled 1 - Enabled	Transmit interrupt enable
RXIE	0 - Disabled 1 - Enabled	Receive interrupt enable
RXBIE	0 - Disabled 1 - Enabled	Receive break interrupt enable
RLRIE	0 – Disabled 1 – Enabled	Retry limited reached interrupt enabled. Only implemented if ISO_7816_ENABLED is TRUE.
END	0 – LSB first 1 – MSB first	Endianness. Selects which bit is transmitted first. Only implemented if ISO_7816_ENABLED is TRUE.
RF	0 – Do not reset 1 – Reset FIFOs	Reset FIFOs. When written with a 1, transmit or receive FIFOs will be cleared. This bit clears automatically
RTIE	0 – Disabled 1 – Enabled	Receive timeout interrupt enable. Only implemented if ISO_7816_ENABLED is TRUE.
D	0 – Full duplex 1 – Half duplex	Duplex. Only implemented if ISO_7816_ENABLED is TRUE.

Table 6: Fields of the `control` register

4.1.5 Cycles Per Bit Register

The cycles per bit register is a 16-bit unsigned integer that specifies how many cycles of the clock, `clk`, each bit is transmitted for. The number of cycles is `cycles_per_bit` + 1. Use of a 16-bit register provides support for a wide range of clock frequencies and baud rates. A value of 0 is not supported.

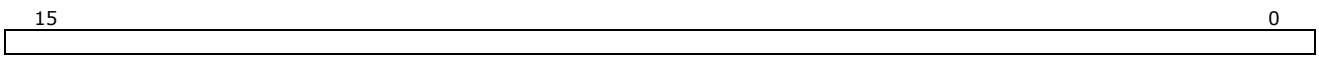


Figure 6: Format of the `cycles_per_bit` register

4.1.6 Retries Register

The retries register controls how the UART attempts retransmission of a character, when a parity error is signalled. Retransmission is only enabled when `control.PB` equals 5 or 6 (ISO 7816-3 T=0). The retries register is only implemented if `ISO_7816_ENABLED` is `TRUE`.

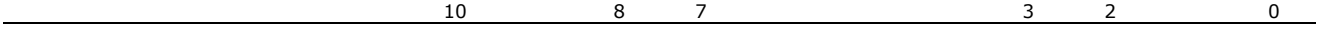


Figure 7: Format of the `retries` register

Register	Values	Description
RETIRES	0 – 7	3-bit unsigned integer that specifies how many times the UART should attempt retransmission of a character, when a parity error is signalled
DELAY	0 – 7	3-bit unsigned integer that specifies the delay in bit periods before the retransmission occurs

Table 7: Fields of the `retries` register

4.1.7 Transmit FIFO Almost Empty Threshold Register

The transmit FIFO almost empty threshold register sets the count of used entries in the transmit FIFO, below which, the `status.TXAE` flag and `tx_ready` will be set.

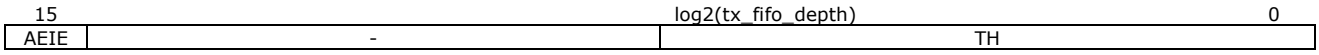


Figure 8: Format of the `txae_thresh` register

Register	Values	Description
TH	0 – `tx_fifo_depth`	Almost empty threshold
AEIE	0 – Interrupt disabled 1 – Interrupt enabled	Almost empty interrupt enable

Table 8: Fields of the `txae_thresh` register

4.1.8 Receive FIFO Almost Full Threshold Register

The receive FIFO almost full threshold register sets the count of used entries in the receive FIFO, above which, the `status.RXAF` flag and `rx_ready` will be set.

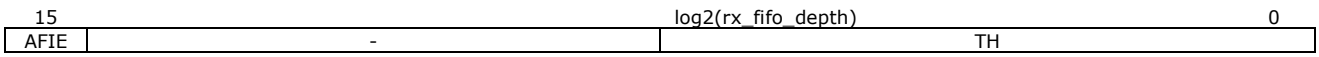


Figure 9: Format of the `rxaf_thresh` register

Register	Values	Description
TH	0 – `rx_fifo_depth`	Almost full threshold
AFIE	0 – Interrupt disabled 1 – Interrupt enabled	Almost full interrupt enable

Table 9: Fields of the `rxaf_thresh` register

4.1.9 Transmit Error Window Start Register

The transmit error window start register holds an unsigned integer that specifies how many cycles of the clock, `clk`, after the detection of the start bit, the transmit error window begins. The transmit error window is used after receiving a character, and determines when the receiver signals to the transmitter that a parity error was detected. The transmit error window start register is only implemented if `ISO_7816_ENABLED` is `TRUE`.

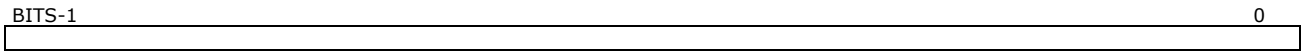


Figure 10: Format of the `tx_error_start` register

4.1.10 Transmit Error Window Length Register

The transmit error window length register holds an unsigned integer that specifies how many cycles of the clock, `clk`, the transmit error window lasts for. The transmit error window length register is only implemented if `ISO_7816_ENABLED` is `TRUE`.

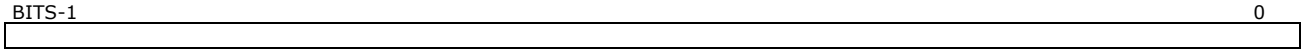


Figure 11: Format of the `tx_error_length` register

4.1.11 Receive Error Window Start Register

The receive error window start register holds an unsigned integer that specifies how many cycles of the clock, `clk`, after the transmission of the start bit, the receive error window begins. The receive error window is used after transmitting a character, and determines when the transmitter detects a signal sent by the receiver to indicate that a parity error was detected. The receive error window start register is only implemented if `ISO_7816_ENABLED` is `TRUE`.

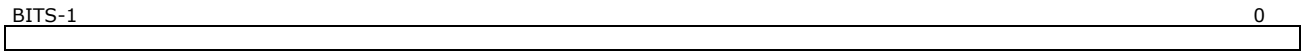


Figure 12: Format of the `rx_error_start` register

4.1.12 Receive Error Window Length Register

The receive error window length register holds an unsigned integer that specifies how many cycles of the clock, `clk`, the receive error window lasts for. The receive error window length register is only implemented if `ISO_7816_ENABLED` is `TRUE`.

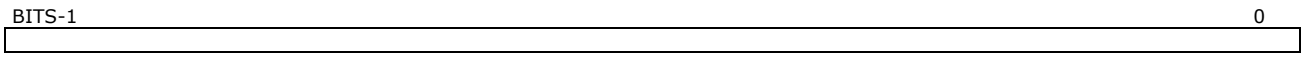


Figure 13: Format of the `rx_error_length` register

4.1.13 Transmit FIFO Count Register

The transmit FIFO count register indicates the count of used entries in the transmit FIFO.

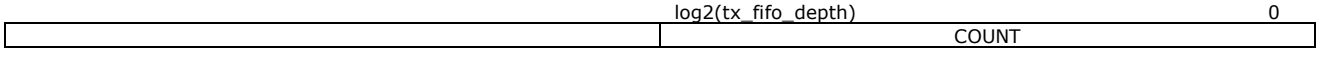


Figure 14: Format of the `tx_count` register

4.1.14 Receive FIFO Count Register

The receive FIFO count register indicates the count of used entries in the receive FIFO.

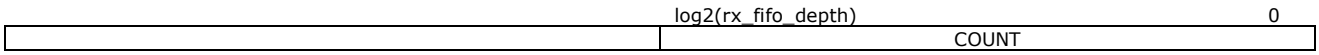


Figure 15: Format of the `rx_count` register

4.1.15 Transmit Guard Time Register

The transmit guard time register specifies the number of bit periods between each data transmission.

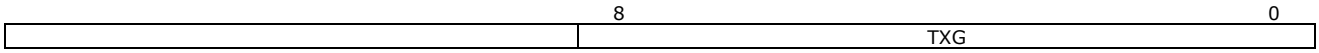


Figure 16: Format of the `tx_guard` register

4.1.16 Receive Timeout Register

The receive timeout register specifies the number of bit periods after data is received at which the receive timeout interrupt should be raised, if further data is not received. The receive timeout register is only implemented if `ISO_7816_ENABLED` is `TRUE`.

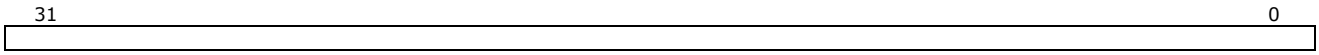


Figure 17: Format of the `rx_timeout` register

4.1.17 Receive Timeout Counter Register

The receive timeout counter register allows access to the receive timeout counter, which counts from 0 to the value in the receive timeout register. It is reset to 0 when data is received and written to the receive FIFO. The receive timeout counter register is only implemented if `ISO_7816_ENABLED` is `TRUE`.

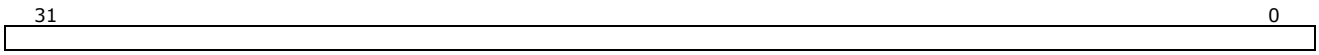


Figure 18: Format of the `rx_timeout_cnt` register

4.2 Interrupts

The UART supports the following interrupts.

- Transmit empty interrupt
- Transmit FIFO almost empty interrupt
- Receive interrupt
- Receive almost full interrupt
- Break interrupt
- Retry limit reached interrupt
- Receive timeout interrupt

The transmit empty interrupt will be raised when the transmit FIFO is empty and the `TXIE` flag in the `control` register is set to 1. This indicates that the transmitter has no data to transmit.

The transmit FIFO almost empty interrupt will be raised when the transmit FIFO is almost empty as determined by tx_thresh.TH and the AEIE flag in the tx_thresh register is set to 1.

The receive interrupt will be raised when the receiver FIFO is not empty and the RXIE flag in the control register is set to 1. This indicates that the receiver has received some data.

The receive FIFO almost full interrupt will be raised when the receive FIFO is almost full as determined by rx_thresh.TH and the AFIE flag in the rx_thresh register is set to 1.

The break interrupt will be raised then a break is detected on the receive data line and the RXBIE flag in the control register is 1. This interrupt can be acknowledged by writing a 1 to the RXB flag in the status register.

The retry limit reached interrupt will be raised when the parity mode is set to 5 (ISO 7816-3 T=0) and the UART has received a parity error signal for the transmission and each subsequent retransmission of a character. The number of times a character is attempted to be transmitted is determined by the retries register. The RLR flag in the status register will be set to 1 to indicate this limit being reached. The interrupt can be acknowledged by writing a 1 to the RLR flag in the status register. The retry limit reached interrupt is only implemented if ISO_7816_ENABLED is TRUE. The UART will not transmit while the RLR flag is set.

The receive timeout interrupt will be raised if the receive timeout counter reaches the receive timeout value before data is received. The interrupt can be acknowledged by clearing the rx_timeout_cnt register then writing a 1 to the RT flag in the status register.

4.3 ISO 7816-3 Operation

The eSi-UART supports both ISO 7816-3 T=0 and T=1 operation as a master when ISO_7816_ENABLED is TRUE.

- For T=0 operation, direct convention, the UART's control register should be configured as illustrated in Table 10: T=0 Direct Convention control Register Settings. The retries register should be set to the desired number of retries. The error window registers should be set according to the frequency of clk and the desired baud rate.

Field	Value	Description
E	1	Enabled
SB	1	2 stop bits
DB	0	8 data bits
PB	5	ISO 7816 T=0 DC parity
FC	0	No flow control
TXB	0	Do not send break
END	0	LSB first
D	1	Half duplex

Table 10: T=0 Direct Convention control Register Settings

- For T=0 operation, inverse convention, the UART's control register should be configured as illustrated in Table 11: T=0 Inverse Convention control Register SettingsTable 10: T=0 Direct Convention control Register Settings. The retries register should be set to the desired number of retries. The data to be transmitted / received must be inverted in software. The error window registers should be set according to the frequency of clk and the desired baud rate.

Field	Value	Description
E	1	Enabled
SB	1	2 stop bits
DB	0	8 data bits
PB	6	ISO 7816-3 T=0 IC parity
FC	0	No flow control
TXB	0	Do not send break
END	1	MSB first
D	1	Half duplex

Table 11: T=0 Inverse Convention `control` Register Settings

- For T=1 operation, direction convention, the UART's control register should be configured as illustrated in Table 12: T=1 Direct Convention `control` Register Settings. The `retries` register should be set to 0.

Field	Value	Description
E	1	Enabled
SB	0	1 stop bits
DB	0	8 data bits
PB	1	Even parity
FC	0	No flow control
TXB	0	Do not send break
END	0	LSB first
D	1	Half duplex

Table 12: T=1 Direct Convention `control` Register Settings

- For T=1 operation, direction convention, the UART's control register should be configured as illustrated in Table 13: T=1 Inverse Convention `control` Register Settings. The `retries` register should be set to 0. The data to be transmitted / received must be inverted in software.

Field	Value	Description
E	1	Enabled
SB	0	1 stop bits
DB	0	8 data bits
PB	2	Odd parity
FC	0	No flow control
TXB	0	Do not send break
END	1	MSB first
D	1	Half duplex

Table 13: T=1 Inverse Convention `control` Register Settings

To use the eSi-UART in an ISO 7816-3 application, where communication is half-duplex over a single bi-directional line, the eSi-UART should be connected as illustrated in Figure 19: Bi-directional I/O Interfacing.

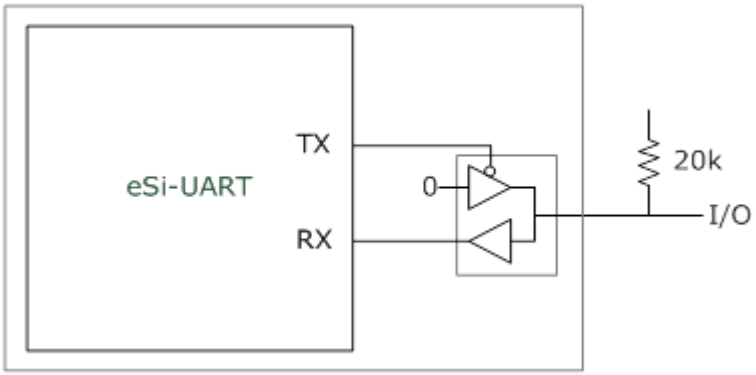


Figure 19: Bi-directional I/O Interfacing

5 Revision History

Hardware Revision	Software Release	Description
1	1.0.0	Initial release
2	2.4.0	Added ISO 7816-3 support. Added `status.RLR` register field. Added `control.PB` mode 5. Added `control.RLRIE` register field. Added `retries` register. Added retry limit reached interrupt. Added `tx_ack` and `rx_ack` ports. Added `duplex` output.
2	2.8.11	Added details of `ISO_7816_ENABLED` configuration option.
3	3.0.0	Added `control.END` register field.
4	3.2.12	Added `status.TXAE` field. Added `status.RXAF` field. Added `txae_thresh` register. Added `rxaf_thresh` register. Added `control.PB` mode 6.
5	3.3.0	Added `tx_error_start` register. Added `tx_error_length` register. Added `rx_error_start` register. Added `rx_error_length` register.
6	3.3.9	Added `tx_count` register. Added `rx_count` register.
7	3.3.16	Added `tx_guard` register. Added `retries.DELAY` register field. Added `rx_timeout` register. Added `rx_timeout_cnt` register. Added `status.RT` field. Added `control.RTIE` field.
8	4.1.15	`status.PE` is set when `control.PB` equals 5 or 6 and a retransmission is requested.
9	5.0.8	When `control.E` is cleared, the UART will finish transmission and reception of the current byte before `cactive` is deasserted. `rts_n` is asserted when RX FIFO is not almost full rather than not full.
10	6.0.2	Added `pdebug` input.
11	6.0.3	`tx_ready` is driven by TX FIFO almost empty, rather than not full. `rx_ready` is driven by RX FIFO almost full, rather than not empty.

Table 14: Revision History

