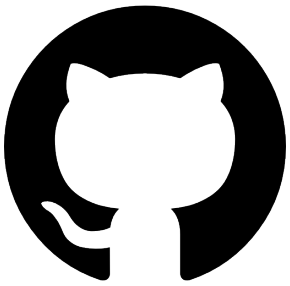

YetAnotherUART

Kirill Liubavin
github.com/kenezoer

12th October 2022

Version: 1.0

Contents

Part I

License

Part II

Project Overview

The Project contains yet another UART Soft-IP module. TODO

Part III

Register's Map

All presented configuration registers of the device are 4 bytes in size. Total size of the regmap is 40 bytes.

The map presented below:

#	Offset	Name	Access Type	Description
1	0x0	DFIFO	RW	Downstream FIFO input
2	0x4	CTRL	RW	UART Controller settings
3	0x8	UART_BIT_LENGTH	RW	Bit Period Setup
4	0xC	IRQ_EN	RW	IRQ Events Enable
5	0x10	IRQ_MASK	RW	IRQ Events Masking
6	0x14	IRQ_EVENT	W1C	IRQ Events Status
7	0x18	UFIFO	RO	Upstream FIFO output
8	0x1C	STATS	RO	UART Stats
9	0x20	HWINFO	RO	Hardware Parameters Stats

1 Downstream FIFO Input (DFIFO) @ 0x0

Bits	Field	Reset value	Description
31:8	-	0x0	-
7:0	DATA_IN	0x0	Data to send (input data of DFIFO)

2 UART Controller Settings (CTRL) @ 0x4

Bits	Field	Reset value	Description
31:9	-	0x0	-
8	SEND_PARITY	0x0	Enable parity bit sending/receiving. Active if set to 1.
7	UFIFO_RST	0x0	Upstream FIFO manual reset. Active if set to 1.
6	DFIFO_RST	0x0	Downstream FIFO manual reset. Active if set to 1.
5	MSB_FIRST	0x0	Most Significant Bit send first mode. Active if set to 1.
4	HW_FLOW_CTRL_EN	0x0	Hardware Flow Control Enable. Active if set to 1.
3:2	STOP_BIT_VALUE	0x0	Stop Bit value select. The value puts into data flow 'as is', i.e. the first stop bit to send is upper bit ([1]) of the field.
1:0	STOP_BIT_MODE	0x0	Stop Bit mode select. Determines the total length of stop bit. Available values: [0]: Half Period; [1]: One Period; [2]: One and Half Period; [3]: Two Periods.

3 UART Bit Period (UART_BIT_LENGTH) @ 0x8

Bits	Field	Reset value	Description
31:0	BIT_PERIOD	0x0	Determines Bit Period (in clock cycles)

4 IRQ Events Enable (IRQ_EN) @ 0xC

Bits	Field	Reset value	Description
31:10	-	0x0	-
9	UART_BAD_FRAME	0x0	Enables detection of the frame error event while receiving a new packet
8	UART_PARITY_ERR	0x0	Enables detection of the parity error event while receiving a new packet
7	UFIFO_EMPTY	0x0	Enables detection of the UFIFO full flag assertion event
6	UFIFO_ERROR	0x0	Enables detection of the UFIFO parity error event
5	DFIFO_EMPTY	0x0	Enables detection of the DFIFO empty flag assertion event
4	DFIFO_ERROR	0x0	Enables detection of the DFIFO parity error event
3	RX_DONE	0x0	Enables detection of the packet receiving done event
2	RX_STARTED	0x0	Enables detection of the packet receiving start event
1	TX_DONE	0x0	Enables detection of the packet transmission done event
0	TX_STARTED	0x0	Enables detection of the packet transmission start event

5 IRQ Events Masking (IRQ_MASK) @ 0x10

NOTE: The IRQ output will be asserted only in case if for selected event regfields of IRQ_EN is asserted, regfield of the IRQ_MASK is deasserted and the event occured (the event occuring can be seen in IRQ_EVENTS register).

Bits	Field	Reset value	Description
31:10	-	0x0	-
9	UART_BAD_FRAME	0x0	Enables masking of the frame error event while receiving a new packet
8	UART_PARITY_ERR	0x0	Enables masking of the parity error event while receiving a new packet
7	UFIFO_EMPTY	0x0	Enables masking of the UFIFO full flag assertion event
6	UFIFO_ERROR	0x0	Enables masking of the UFIFO parity error event
5	DFIFO_EMPTY	0x0	Enables masking of the DFIFO empty flag assertion event
4	DFIFO_ERROR	0x0	Enables masking of the DFIFO parity error event
3	RX_DONE	0x0	Enables masking of the packet receiving done event
2	RX_STARTED	0x0	Enables masking of the packet receiving start event
1	TX_DONE	0x0	Enables masking of the packet transmission done event
0	TX_STARTED	0x0	Enables masking of the packet transmission start event

6 IRQ Events Status (IRQ_EVENTS) @ 0x14

NOTE: The register shows if the enabled event is occured. The register has a W1C (Write 1 to Clear) access type. In case of event occured appropriated event field is asserted and output IRQ signal will be asserted (only if the event is not masked) as well. To deassert the output IRQ signal, user should write a logical '1' into necessary field of the IRQ_EVENTS register.

Bits	Field	Reset value	Description
31:10	-	0x0	-
9	UART_BAD_FRAME	0x0	Frame error event while receiving a new packet status
8	UART_PARITY_ERR	0x0	Parity error event while receiving a new packet status
7	UFIFO_EMPTY	0x0	UFIFO full flag assertion event status
6	UFIFO_ERROR	0x0	UFIFO parity error event status
5	DFIFO_EMPTY	0x0	DFIFO empty flag assertion event status
4	DFIFO_ERROR	0x0	DFIFO parity error event status
3	RX_DONE	0x0	Receiving done event status
2	RX_STARTED	0x0	Receiving start event status
1	TX_DONE	0x0	Transmission done event status
0	TX_STARTED	0x0	Transmission start event status

7 Upstream FIFO Output (UFIFO) @ 0x18

Bits	Field	Reset value	Description
31:8	-	0x0	-
7:0	DATA_OUT	0x0	Received data (output data of UFIFO)

8 UART Stats (STATS) @ 0x1C

Bits	Field	Reset value	Description
31:27	-	0x0	-
26	RX_STATUS	0x0	Immediate value of RX Status
25	UFIFO_FULL	0x0	Immediate value of UFIFO 'full' flag
24	UFIFO_EMPTY	0x0	Immediate value of UFIFO 'empty' flag
23:16	UFIFO_USED	0x0	Immediate value of UFIFO used words
15:11	-	0x0	-
10	TX_STATUS	0x0	Immediate value of TX Status
9	DFIFO_FULL	0x0	Immediate value of DFIFO 'full' flag
8	DFIFO_EMPTY	0x0	Immediate value of DFIFO 'empty' flag
7:0	DFIFO_USED	0x0	Immediate value of DFIFO used words

9 Hardware Paremeters Info (HWINFO) @ 0x20

Bits	Field	Reset value	Description
31	PARITY_CHECK_EN	0x0	FIFO Parity Check Enable
30:24	-	0x0	-
23:16	UFIFO_DEPTH	0x0	UFIFO Depth Value
15:8	DFIFO_DEPTH	0x0	DFIFO Depth Value
7:4	IP_VERSION_MJR	0x0	Major IP Version
3:0	IP_VERSION_MNR	0x0	Minor IP Version

Part IV

Driver part

The current version of driver part is made to use with Syntacore's SCRx CPU Cores and theirs BSP. Allows to use standard 'printf' function in C language.

Syntacore: www.github.com/syntacore

Part V

Module Parametrizing

#	Parameter	Default value	Supported Values	Description
1	APB_ADDR_WIDTH	32	16 - 256	APB Bus Address Width
2	APB_DATA_WIDTH	32	8, 16, 32, 64	APB Bus Data Width
3	FIFO_PARITY_CHECK_EN	TRUE	FALSE / TRUE	Enables Parity check enable in FIFO