AHB-Lite UART Block

Overview

This is a simple asynchronous serial interface block, designed to be a slave on an AHB-Lite bus. The bit rate is fixed at 19 200 bit/s, with 8 data bits, no parity and one stop bit. Both transmit and receive paths have 16-byte FIFO buffers. The status of each buffer (empty, full) can be checked. The block can be configured to cause an interrupt based on these status bits.

Programmer's Model

The UART block presents four 8-bit registers to the processor, as shown in the table below. Only the rightmost 8 bits are used. Other bits will be ignored on write and will read as 0.

Register	Relative Address	Access	Description
receive data	0x0	R	8-bit data from the receive buffer. Only valid if the status bit indicates that receive data is available.
transmit data	0x4	R/W	Write: 8-bit data to be put into the transmit buffer. Read: 8-bit data at the output of the transmit buffer.
status	0x8	R	Bit 0: transmit FIFO full – unable to accept more data. Bit 1: transmit FIFO empty – transmission will stop. Bit 2: receive FIFO full – unable to accept more data. Bit 3: receive FIFO not empty – data available to read. Bits 7:4 are not used and read as 0.
control	0xC	R/W	Bits 3:0 enable the corresponding bit in the status register to cause an interrupt. Bits 7:4 are not used and read as 0.

An attempt to write transmit data while the transmit FIFO is full will be ignored. An attempt to read receive data while the receive FIFO is empty will return the previous receive data. If the receive FIFO is full, it should be read within 10 bit times, or receive data may be lost.

Implementation

The RTL diagram is shown below. This hardware is described in Verilog in the files `AHBuart.v`, `fifo.v` and `uart.v`. A simple testbench is provided as `TB_AHBuart.v`.
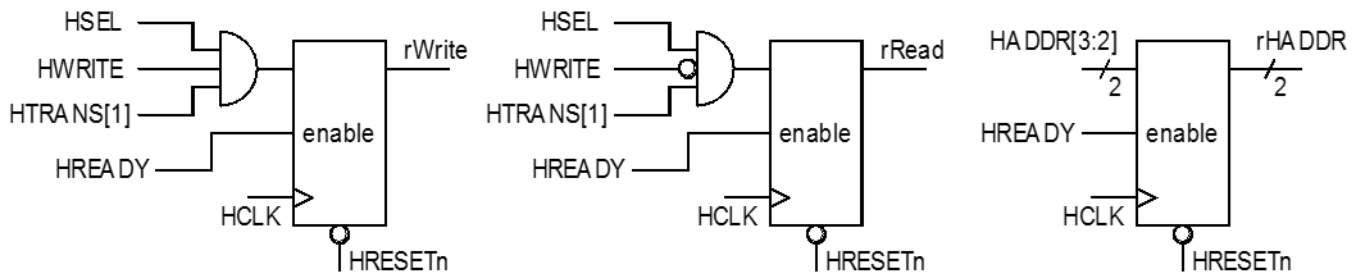
All registers operate on the bus clock, with synchronous active-low reset.

Bus Interface

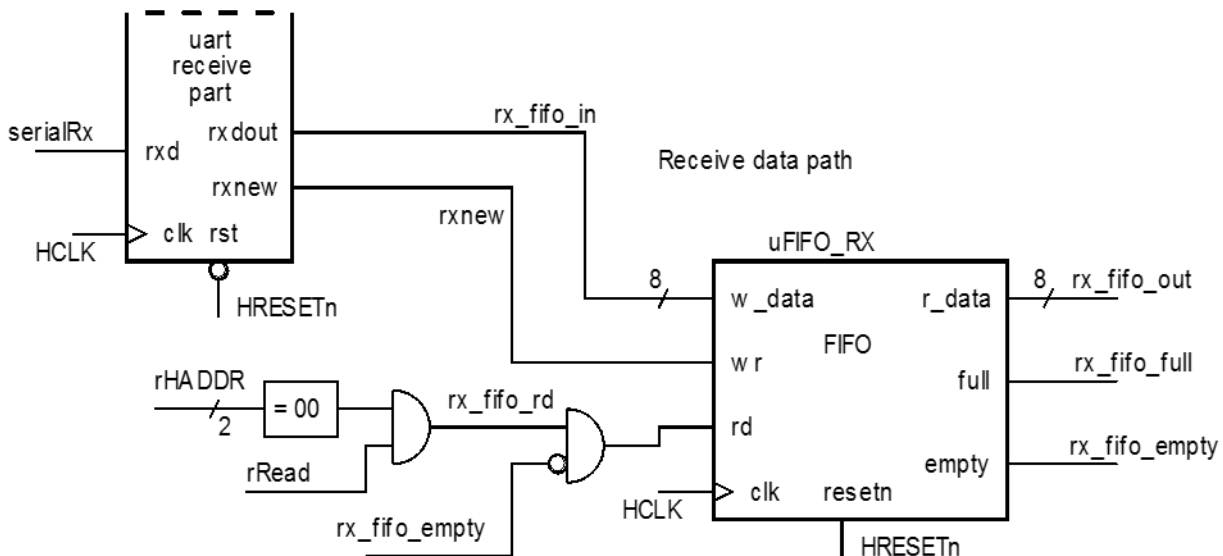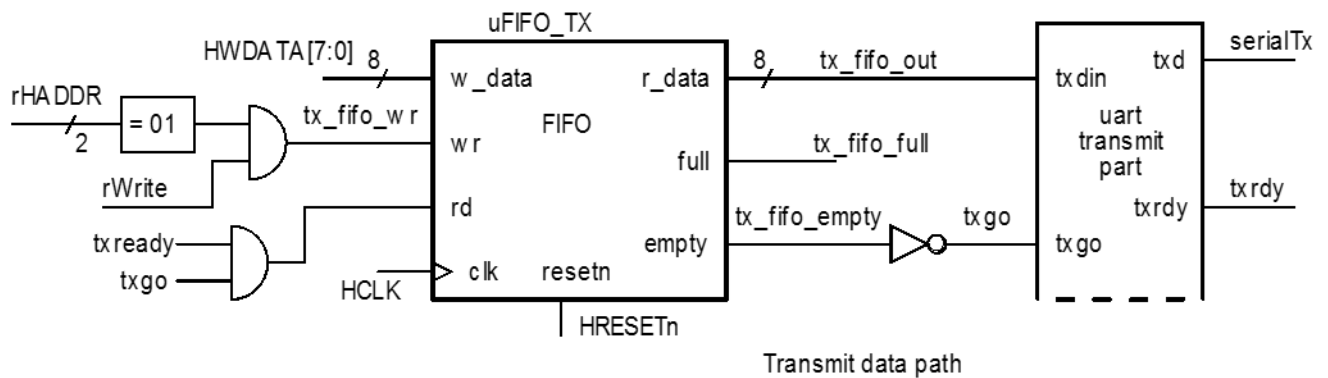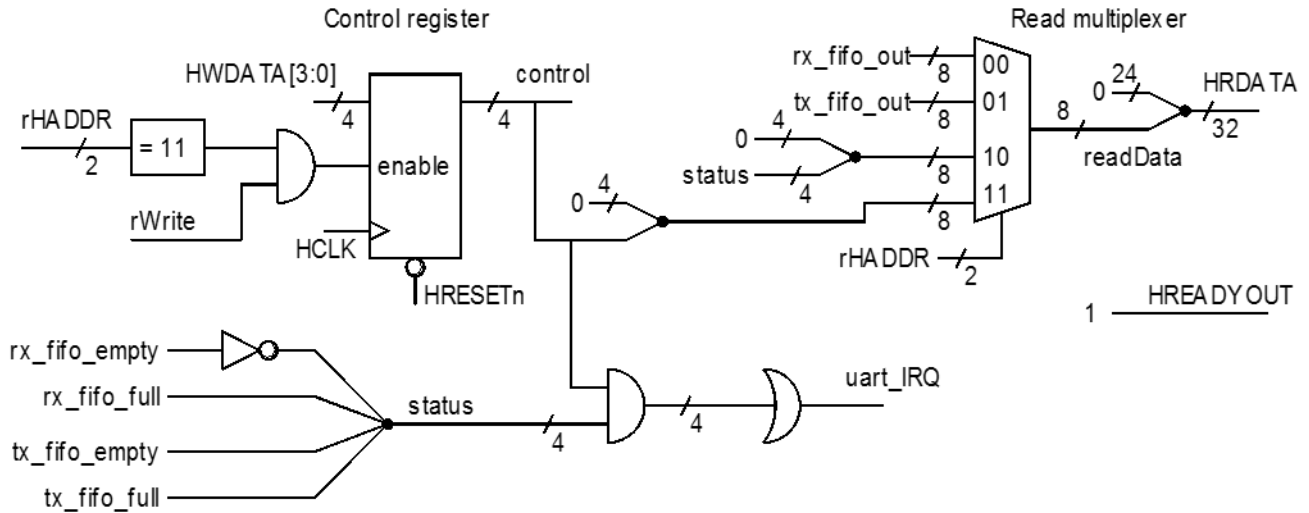The block ignores the `HSIZE` signal and has no port for it. One register holds HADDR[3:2] from the address phase of a bus transaction, for use in the data phase. As only there are only 4 relative addresses in use, and data width is ignored, no other bits are needed.

Write transactions and read transactions are identified during the address phase, and these signals are also held in flip-flops for use in the data phase.

Registers to hold signals from address phase

HSEL
HWRITE
HTRANS[1]
HREADY
enable
HCLK
HRESETn
rWrite

HSEL
HWRITE
HTRANS[1]
HREADY
enable
HCLK
HRESETn
rRead

HADDR[3:2]
2
HREADY
enable
HCLK
HRESETn
rHADDR
2

Control register

rHADDR
2
= 11
HWDATA[3:0]
4
rWrite
enable
HCLK
HRESETn
control
4

Read multiplexer

rx_fifo_out 8 00
tx_fifo_out 8 01
0 4
status 4
10 8
11 8
rHADDR 2
8
readData
0 24
HRDATA
32

HREADYOUT
1

rx_fifo_empty
rx_fifo_full
tx_fifo_empty
tx_fifo_full
status
4
uart_IRQ
0 4

uFIFO_TX

HWDATA[7:0] 8
rHADDR 2
= 01
rWrite
tx_fifo_wr
txready
txgo
HCLK
FIFO
w_data
wr
rd
clk   resetn
r_data 8 tx_fifo_out
full   tx_fifo_full
empty   tx_fifo_empty   txgo
HRESETn

uart
transmit
part
txdin   txd   serialTx
txrdy   txrdy
txgo

Transmit data path

uart
receive
part
serialRx
rxd
rxdout   rx_fifo_in
rxnew
clk rst
HCLK
HRESETn
rxnew

Receive data path

uFIFO_RX

rHADDR 2
= 00
rRead
rx_fifo_rd
rx_fifo_empty
HCLK
8
FIFO
w_data
wr
rd
clk   resetn
r_data 8 rx_fifo_out
full   rx_fifo_full
empty   rx_fifo_empty
HRESETn

2

The control register is implemented in the normal way, as a 4-bit register in the hardware, which captures write data from the bus on a write transaction to address 0xC.

A write to address 0x4 activates the write input to the transmit FIFO. If the FIFO is not full, this will cause the 8 least-significant bits of the HWRITE signal to be placed in the FIFO.

The four status bits from the FIFO buffers are combined into one status signal, and made available for reading. These status bits are also combined with the bits in the control register to generate the interrupt signal.

A read multiplexer selects the required 8-bit value for read transfers, based on the held address bits. This is combined with 24 zero bits to give a 32-bit read value.

No wait states are needed, so HREADYOUT is always 1.

FIFO buffers
Each FIFO has two control signals, which take effect on the rising edge of the clock:

wr – write, causes the data at the w_data port to be stored in the FIFO, if it is not full. The FIFO updates its internal write pointer to point to the next free storage location, or asserts the full signal if there is no free storage location.

rd – read, indicates that the data at the r_data port has been accepted, and the FIFO should present the next data in the queue. This is ignored if the FIFO is empty. Otherwise the FIFO updates its internal read pointer to point to the next data, or asserts the empty signal if there is no more data (read pointer and write pointer are equal).

UART
The UART transmitter has one control signal, txgo, to indicate that the data on the txdin port should be transmitted. This control signal is ignored if the transmitter is not ready. The ready state is reached as the stop bit of the previous transmission is being sent, and is indicated on the txrdy status output. The UART can transmit continuously if txgo is always asserted – txrdy will go high for one clock cycle at the end of each data byte, and this can be used as an indication that the data at the txdin port is being accepted for transmission.

The UART receiver has no input control signals – it responds to the serial received data. When a byte has been received, the data is presented at rxdout and the rxnew status output is asserted for one clock cycle to indicate that new data is available. The data remains valid until the next byte is received.

Transmit data path
The transmit data passes through a FIFO buffer and then to the transmit part of the UART block. Data is placed in the transmit FIFO on a bus write transaction to address 0x4.

Whenever the transmit FIFO is not empty, the txgo signal is asserted. If the UART txrdy signal is also asserted, the UART will accept the data at the FIFO output. This same combination of signals is used to drive the rd port on the FIFO, so that it presents new data, to be transmitted the next time the UART is ready.

Receive data path
The UART rxnew signal is used to write received data bytes into the receive FIFO. If the FIFO is full, this signal will be ignored and data will be lost.

A bus read from address 0x0, will return the output data of the receive FIFO. If the FIFO is not empty, the read transaction will also cause a pulse on the rd port of the FIFO, to indicate that the data has been read. The FIFO will then present the next data, if any.