| Signal Name | Direction | Width | Description |
|---|---|---|---|
| pclk_i | Input | 1 | Input clock |
| prst_ni | Input | 1 | Active low reset |
| pwdata_i | Input | 32 | Write Data |
| paddr_i | Input | 8 | Input Address |
| psel_i | Input | 1 | Selects the peripheral |
| pwrite_i | Input | 1 | When set 1 (writes the data) and set 0 (reads the data). |
| penable_i | Input | 1 | Enables the peripheral |
| prdata_o | Output | 32 | Read Data |
| tx_o | Output | 1 | Transmit pin |
| rx_i | Input | 1 | Receiver pin |
| intr_tx | Output | 1 | When transfer is done |
| intr_rx | Output | 1 | When data is received |
| intr_tx_level | Output | 1 | When desired level of buffer reached |
| intr_rx_timeout | Output | 1 | When all bytes are received |
| intr_tx_full | Output | 1 | When tx buffer is full |
| intr_tx_empty | Output | 1 | When tx buffer is empty |
| intr_rx_full | Output | 1 | When rx buffer is full |
| intr_rx_empty | Output | 1 | When rx buffer is empty |

For transmitting the data following steps will be done:

- First we have to set configurable registers which are baud, tx_level, tx_data, tx_en_fifo, rd_en_fifo
- To set the baud rate, at address 0x0 we can assign the desired baud rate we have to set.
- Then set tx_level, at address 0x18 we can assign the level of the buffer.
- Then data to be transferred is sent at address 0x4.
- Then the fifo’s write is enabled at address 0x14.
- Then transfer is enabled by writing at address 0x1c.

For receiving the data:

- Baud rate will be configured at address 0x0.
- Then to start receiving the data,rx will be enabled at address 0xc.
- At address 0x8, we can read the data that is received.

## Register Map:

| Register | Width | Description |
|---|---|---|
| Baud[0:15] | 16 | Contains the baud |
| tx_data[0:7] | 8 | Contains the data to transfer |
| rx_en | 1 | If set 1(receiving enable) |
| tx_en_fifo | 1 | If set 1(tx fifo enabled to write) |
| tx_level[0:2] | 3 | Level of tx fifo to be filled. |
| rd_en_fifo | 1 | When set 1(transfer is enabled) |

There are 5 modules in this RTL:

- Uart_core
- Uart_rx
- Uart_tx
- Uart_fifo_tx
- Uart_fifo_rx
- timer_rx

Uart_rx:

Uart_rx is module of receiver, the data receiving from receiver pin(rx_i) is the input in this module when rx is enabled and start bit is detected i.e. 0 the data start to store in the register and when the data is completely received (when the stop bit is detected i.e. 1) the received data is then output from rx_o register and an rx_done signal is set high showing that the data is received.

| Register | Width | Description |
|---|---|---|
| clk_i | 1 | Input clock |

| | | |
|---|---|---|
| rst_ni | 1 | Resets the receiver(active low) |
| rx_i | 1 | Input the data receiving |
| rx_en_i | 1 | Receiver enable |
| clks_per_bit | 16 | Input baud rate |
| rx_o | 8 | Data received is stored |
| c_START | 1 | Start bit detected |
| rx_done_o | 1 | When one byte is received |

Uart_tx:
Uart_tx is a module of the receiver, the data that we have to transmit is taken as input in this module. When the transfer is enabled the data start to transmit from the output pin tx_o. The data is transferred bit by bit. First, the start bit is transferred i.e. 0 and then the data is transferred and after the data is transferred a stop bit is sent.When the data is transferred tx_done signal is also set high. The data is transferred according to the baud rate set.

| **Register** | **Width** | **Description** |
|---|---|---|
| clk_i | 1 | Input clock |
| rst_ni | 1 | Input reset(active low) |
| tx_en_i | 1 | Transfer enable |
| tx_data_i | 8 | Input the data to be transferred |
| clks_per_bit | 16 | Input the baud rate |
| tx_o | 1 | Output the byte(bit by bit) |

| | | |
|---|---|---|
| tx_done_o | 1 | |

Uart_fifo_tx:

Uart_fifo_tx is the fifo for the transmitter, the data to be transferred is first stored in the fifo and when transfer is enabled the data is transferred. Fifo stores the data byte by byte when the transfer is enabled the data is output from the fifo.

Uart_fifo_rx:

Uart_fifo_rx is the fifo for the receiver when the data is received from the receiver, the data starts to store in the fifo. When 1 byte is received the write signal is set high and fifo stores the byte in it.And when we have to read the data the read signal is provided to the fifo to output the data from the fifo.

Timer_rx:

Timer_rx is the module which is used to check whether the bytes receiving from the rx_i pin is completed or not.It detect the start bit of the next byte and if the start bit doesn't comes unit the clk_per_bit which is set as according to the baud rate.

Uart_core:

This is the module where configurable registers are present and status registers are there. We configure the register through the configurable register,we select the address of the register and writes the desired data to the register. There are status registers too, which gives the status