

UART

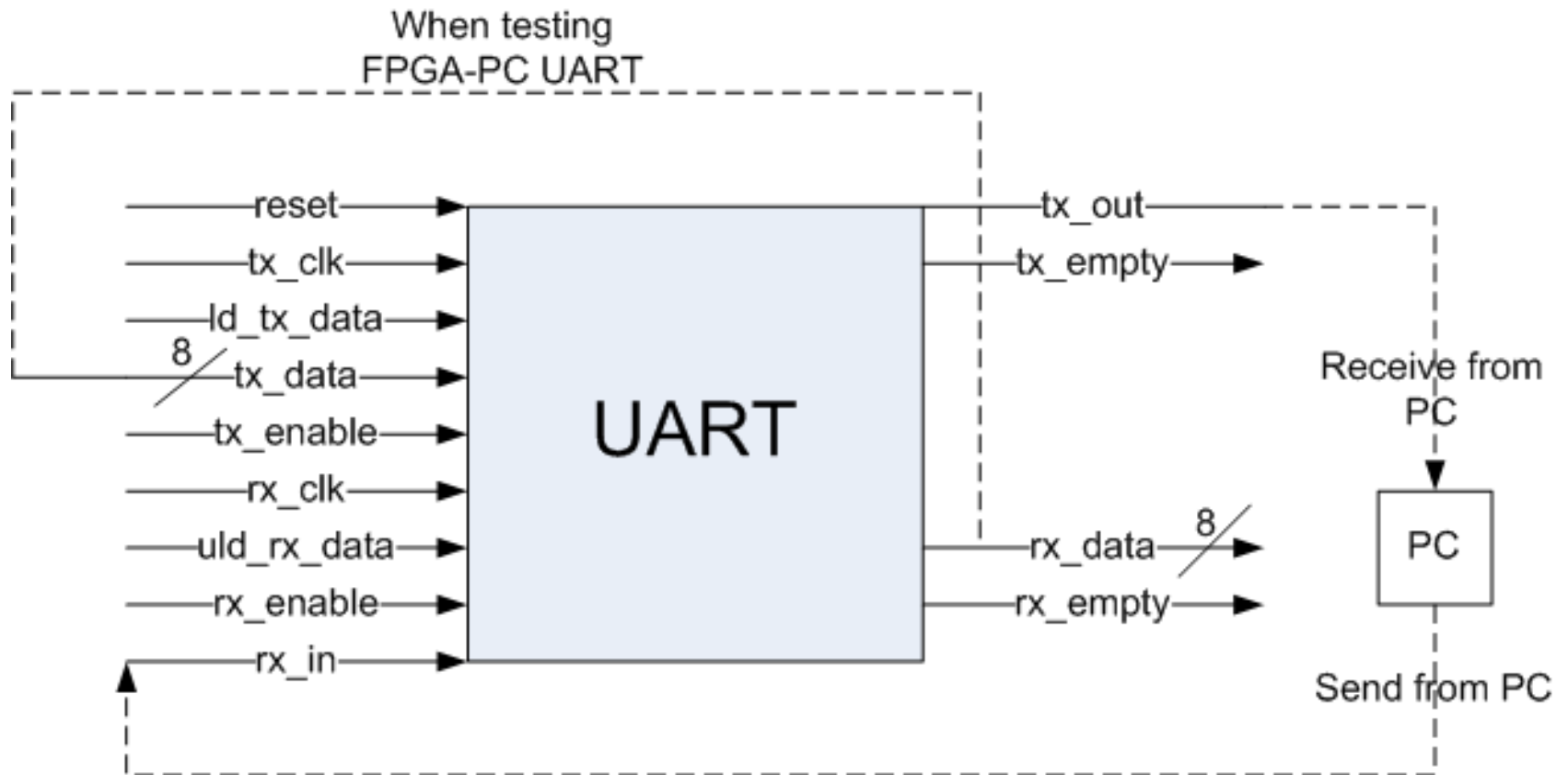
Universal asynchronous receiver and transmitter

UART

- A protocol that translates data between parallel and serial form.
- Interface with PC is serial. i.e Data is sent serially, bit by bit.
- There are only two pins between UART & PC interface: rx_in and tx_out.
-

UART IO pins for the UART verilog example code

- <http://www.asic-world.com/examples/verilog/uart.html>



Specifics for the UART verilog example code

- For this module: rx_clk set to be 16x faster than tx_clk which is desired baudrate.
- Baudrate values 110, 150, 300, 1200, 2400, 9600, 19200, 38400, 115200, ...
- If tx_empty = 1 then ld_tx_data = 1 => tx_data is loaded

For receiving , check if rx_empty = 0 then make uld_rx_data = 1

- For this code rx_in = tx_out (In test bench)

- You need to generate rx-clk and tx_clk from a reference clk that is generated on board according to your UCF file. Tx_clk is set as baudrate and rx-clk is 16 times faster. You need to find the appropriate counter numbers to divide the reference clk to generate both tx_clk and rx_clk.
- For example if reference clk is 50MHz and the baud rate is 115200 bps. You need to divide the reference clk by $50M/115200 = 434$ clock cycles to generate tx_clk. The rx_clk is 16 times faster, so we need to divide the reference clk by $50M/115200 / 16 = 27$ cycles to generate rx_clk.
- Verilog Code example:

```
always@(posedge clk)
begin
    tx_counter <= tx_counter + 1;
    rx_counter <= rx_counter + 1;

    if(tx_counter == 434)
    begin
        tx_counter <= 0;
        txclk <= ~ txclk;
    end
    if(rx_counter == 27)
    begin
        rx_counter <= 0;
        rxclk <= ~ rxclk;
    end
end
```

end