

Data sheet for RS485 Transceiver IP

Functional Descriptions

RS485 transceiver IP is used for applications where many devices communicate with each other using RS485 physical layer. This IP has both transmitter and receiver parts. Transmitter part will convert the parallel data into serial bits and transmits the same. The Receiver part receives the serial data and converts them to parallel data. The block diagram of RS485 IP is shown in the Fig 1. Architectural block diagram is shown in Fig 2.

Features:

Configuration data is used for setting parameters for IP which includes the following.

- Mode of data transmission and reception namely RTU or ASCII.
- Even, odd, or no-parity bit.
- 1 or 2 stop bits.
- Different baud rates up to 4000K Baud.
- Half duplex communication.

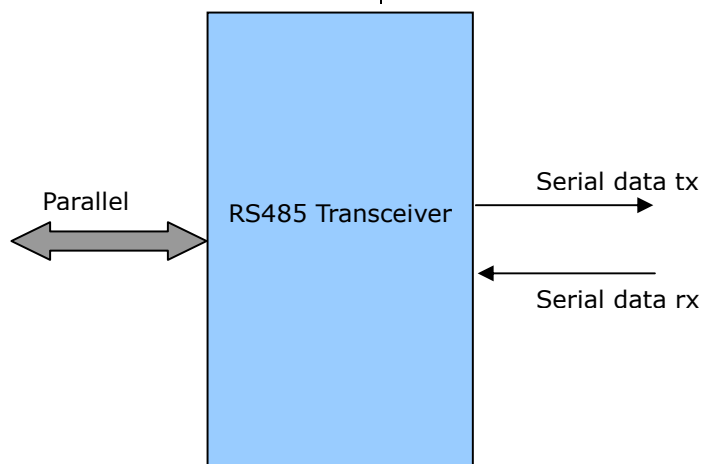


Fig 1: Block Diagram of RS485 Transceiver IP.

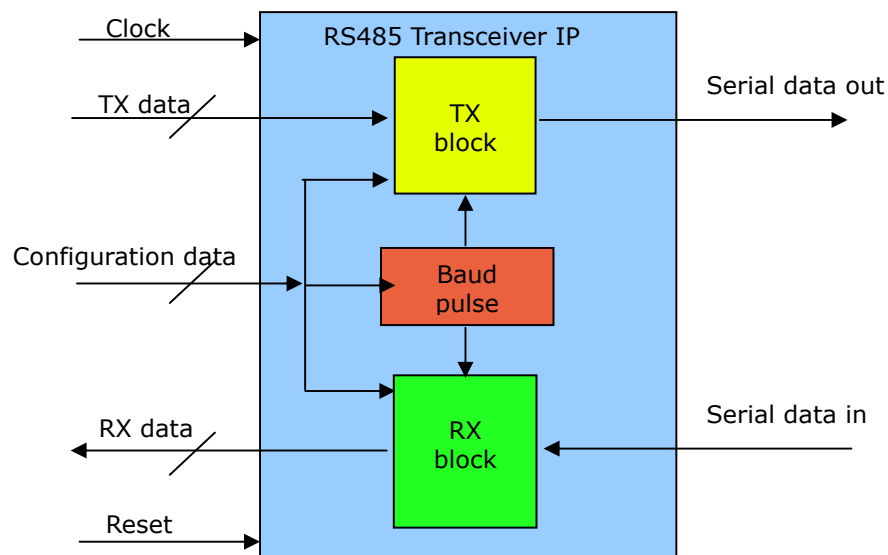


Fig 2: Architectural Diagram of RS485 Transceiver IP

Description:

RS485 transceiver IP supports two modes of data transmission and reception.

1. RTU Transmission mode.

In this mode, each 8-bit byte of message is transmitted as one RTU character. The Data format for each byte of data is 8-bit binary. Number of stop bits, parity and type of parity can be changed. Bits are transmitted from left to right as shown in Fig 3.

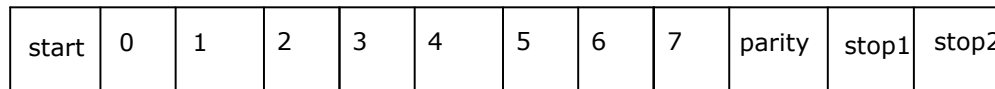


Fig 3: Transmission of bits for RTU mode (From left to right)

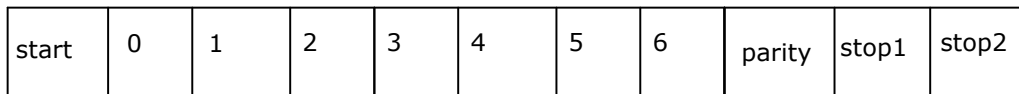


Fig 4: Transmission of bits for ASCII mode(From left to right)

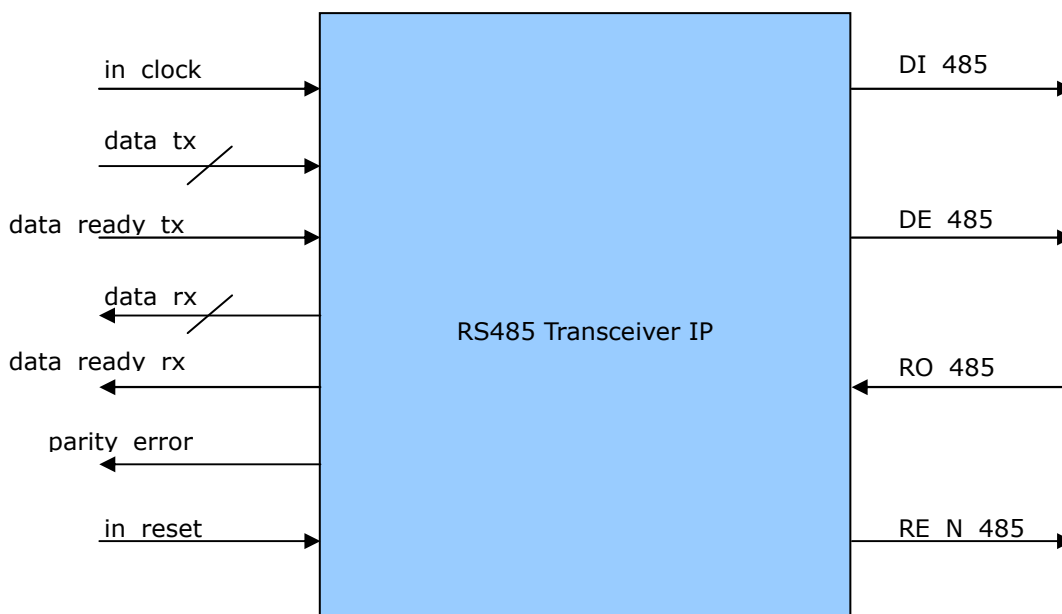


Fig 5: Schematic symbol RS485 transceiver IP

2. ASCII Transmission mode.

Here each 8 bit-byte of message is transmitted as 2 ASCII characters. Data format for each byte of data is two ASCII characters 0-9, A-F. Number of stop bits, parity and type of parity can be changed. Bits are transmitted from left to right as shown in Fig 4.

Schematic Symbol:

Schematic symbol for RS485 IP is shown in the Fig 5 and description for each signal is in Table 1.

Table 1: Signal descriptions of RS485 transceiver IP

Signal	Width	Input/Output	Interface	Description
in_clock	1	input	FPGA(user logic)	This is clock input pin.
in_reset	1	input	FPGA(user logic)	This is a reset input pin.
data_tx	7 / 8	input	FPGA(user logic)	This is the data to be transmitted. This is active high signal.
data_ready_tx	1	input	FPGA(user logic)	When this signal is high, data_tx should contain valid data to be transmitted. This is active high signal.
busy_tx	1	output	FPGA(user logic)	When this signal is high data_ready_tx should not go high meaning transmitter is busy sending the data. This is active high signal.
data_rx	7 / 8	output	FPGA(user logic)	This is the data received. This is active high signal.
data_ready_rx	1	output	FPGA(user logic)	When signal goes high a valid data recedes on data_rx. This is active high signal.
parity_error	1	output	FPGA(user logic)	This is asserted to 1 when there is a parity error in the received data. This is active high signal.
DI_485	1	output	RS485 IC	This is the serial interface pin used to transmit data bits. This is active high signal.
DE_485	1	output	RS485 IC	This is tristate control signal. When this signal is 1 DI_485 becomes valid. This is active high signal.
RO_485	1	output	RS485 IC	This is the serial interface pin used to receive data bits. This is active high signal.
RE_N_485	1	input	RS485 IC	This is tristate control signal. When this signal is 0, data on RO_485 is read. This is active low signal.

Example:

An Example demonstrating the RS485 Transceiver IP usage is shown in Fig 6.

There are two designs which communicate using RS485. Design 'S' supplies the data and design 'R' receives the data. Here MAX485 chip is being used as physical layer for rs485 transmission and reception of data. MAX485 converts TTL signals into differential signals A and B, and vice versa as

in Fig 6. MAX485 supports half duplex operation, where a device can either receive or transmit but cannot do both simultaneously. This gives rise to Master and slave concept where master first initiates the data transfer by addressing a slave device in network. The addressing of slave and other functionalities are implemented in data link layer (in this case, serial communication).

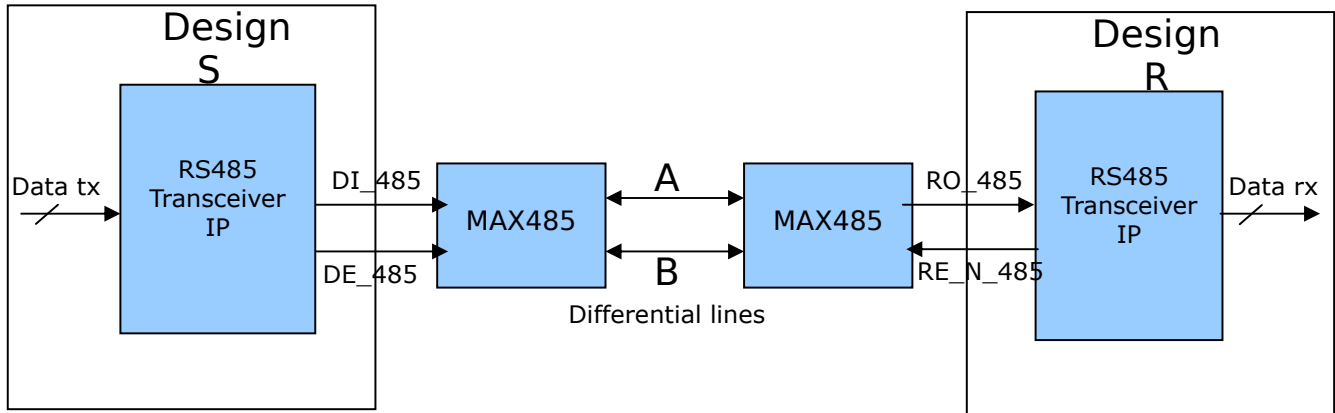


Fig 6: Device S sending data to Device R using Maxim MAX485

Performance:

Device	Slice Count	Frequency
Spartan-3A (xc3s700a-4fg484)	74	100 MHz

Verification:

RS485 transceiver IP is tested and verified on spartan 3A board for large of data of transmission and reception, with various baud rate.

Deliverables:

- Verilog RTL source code
- Test benches
- Synthesis and Simulation scripts
- Detailed user documentation, including RTL source code documentation