Async/Clocked Serial Channel

Features

Async mode with optional parity and address bit

Clocked serial mode with internal or external clock, full- or half-duplex

Four bytes of buffering for both transmit and receive

IRDA mode for async

Internal baud-rate divider

DMA request for both transmit and receive

Standard byte-wide interface

Description

This dual-function serial port can be used in either asynchronous or clocked-serial mode. Four bytes of buffering are available for both the transmitter and the receiver.

The serial port can be clocked by an external baud rate generator or an internal dedicated 15-bit divider, except in the clocked serial mode with an external clock. In asynchronous mode this clock is sixteen or eight times the data rate, while in clocked serial mode the clock drives the serial port at the data rate.

In asynchronous mode the port can send and receive seven or eight data bits and has the option of appending and recognizing parity plus an additional address bit. The parity options are odd, even, Mark and Space and are the same for transmit and receive. The receiver reports parity errors in the status register. Asynchronous mode operates full-duplex.

The clocked serial mode allows full-duplex or half-duplex synchronous communication. The data direction is under program control, as is the selection of an internal or external clock. Clocked serial mode is strictly byte-oriented, with no provision for start, address, or stop bits in the serial bit stream.

Interface

```
module ser_top (dreq_rxx, dreq_txx, dreq2_txx, serx_drv, serx_clk, serx_int, serx_rbus,
                serx_txd, clkp, peri_addr, pwrite_bus, resetb, rt_sync, serx_eclki, serx_rd,
                serx_rxdi, serx_test, serx_wr, tmrb_out);

  input         clkp;          /* main peripheral clock                                    */
  input         resetb;        /* internal reset                                           */
  input         rt_sync;       /* receiver/transmitter clock enable                        */
  input         serx_rd;       /* serial port peripheral read strobe                       */
  input         serx_test;     /* serial port x test mode                                  */
  input         serx_wr;       /* serial port peripheral write strobe                      */
  input   [1:0] serx_eclki;    /* serial port external clock (sync mode)                   */
  input   [1:0] tmrb_out;      /* timer b output bus (for synced cks transfers)            */
  input   [2:0] serx_rxdi;     /* receiver data input                                      */
  input   [3:0] peri_addr;     /* internal peripheral address bus                          */
  input   [7:0] pwrite_bus;    /* internal peripheral write bus                            */
  output        dreq_rxx;      /* dma request for receiver                                 */
  output        dreq_txx;      /* dma request for transmitter                              */
  output        dreq2_txx;     /* dma request2 for transmitter                             */
  output        serx_drv;      /* serial port clock output enable                          */
  output        serx_clk;      /* serial port clock output                                 */
  output        serx_txd;      /* transmitter data output                                  */
  output  [3:1] serx_int;      /* serial port interrupt request                            */
  output  [7:0] serx_rbus;     /* serial port peripheral read bus                          */
```