H16550S

UART with FIFOs and Synchronous CPU Interface

The H16550S is a standard UART providing 100% software compatibility with the popular Texas Instruments 16550 device. It performs serial-to-parallel conversion on data originating from modems or other serial devices, and performs parallel-to-serial conversion on data from a CPU to these devices.

The H16550S can be run in either 16450-compatible character mode or in 16550-compatible FIFO mode, where an internal FIFO relieves the CPU of excessive software overhead.

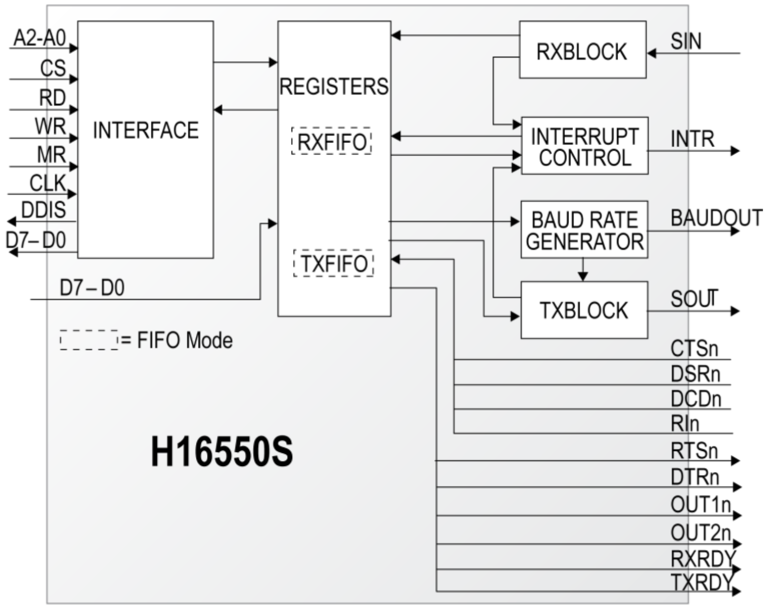
Developed for easy reuse in FPGA or ASIC applications, the H16550S is available optimized for several technologies with competitive utilization and performance characteristics.

Applications

The H16550S can be utilized for a variety of serial communication applications including:

- Serial or modem computer interface
- Serial interface within modems and other devices

Block Diagram



FEATURES

- Capable of running all existing 16450 and 16550a software
- Fully Synchronous design. All inputs and outputs are based on rising edge of clock
- In FIFO mode, the transmitter and receiver are each buffered with 16 byte FIFOs to reduce the number of interrupts presented to the CPU
- Adds or deletes standard asynchronous communication bits (start, stop and parity) to or from the serial data
- Independently controlled transmit, receive, line status and data set interrupts
- Programmable baud generator divides any input clock by 1 to (2^{16} - 1) and generates the 16 x clock
- Modem control functions (CTSn, RTSn, DSRn, DTRn, RIn, and DCDn)
- Fully programmable serial interface characteristics:
  - 5, 6, 7, or 8 bit characters
  - Even, odd, or no-parity bit generation and detection
  - 1, 1½, or 2 stop bit generation
  - Baud generation
- False start bit detection
- Complete status register
- Internal diagnostic capabilities: loopback controls for communications link fault isolation
- Full prioritized interrupt system controls
- Optional AMBA^{TM} APB Bus Interface

Functional Description

The H16550S includes the following six major blocks. All the core’s inputs and outputs are fully synchronous to the rising edge of the CLK input.

Interface (RWCONTROL)

Handles communication with the processor (or parallel) side of the system. Manages all writing and reading of internal registers.

Registers (UART_REG)

Holds all of the device’s internal registers. Some information comes from the other blocks, but it is all gathered in the Registers block and made available to all blocks.

Functional Description (continued)

RXBlock

Handles the receiving of the incoming serial word. It is programmable to recognize data widths such as 5, 6, 7 or 8 bits, various parity settings, and different stop bits of 1, 1½ and 2 bits. It checks for errors in the input data stream, and if the incoming word has no problems it is placed either in the Receiver Holding register or in the Receiver FIFO depending on the mode programmed.

Interrupt Control

Sends an interrupt signal back to the processor depending on the state of the FIFO and its received and transmitted data. Various levels of interrupt can be read from the Interrupt Identification register. Interrupts are sent in the condition of empty transmission or receiving buffers (or FIFOs), an error in the receiving of a character, or other conditions requiring the attention of the processor.

Baud Rate Generator

Takes the input clock, CLK, and divides it by a programmed value (from 1 to $2^{16} – 1$). This divided clock is then divided by 16 to create the transmission clock called the Baudout clock.

TXBlock

Handles the transmission of data written to the Transmission Holding register (or transmit FIFO). It adds required start, parity and stop bits to the data being transmitted so that the receiving device can do the proper error handling and receiving.

Component Substitution

The H16550S core is modeled after the Texas Instruments 16550, with the following differences:

- No provision is made for a crystal. The CLK input is designed to accept a standard digital input.
- The RCLK input in the Asynchronous version is replaced by CLK with rclk_enb which has the same frequency as BAUDOUTn.
- The bi-directional Data Bus has been split into an input and an output component. In order to use the core with a bi-directional Data Bus, the DDIS signal can be used as the controlling signal for the tri-state drivers.
- RD2, WR2, CS1 and CS2 have been eliminated. A single signal takes their place. These are RD, WR and CS.
- The ADSN signal has been removed. The H16550S functions as if the ADSN signal is held low. The included wrapper can be used to add the ADSN functionality latching the address and data buses.
- The main clock input CLK must be active from power-up.
- The Baudrate Generator is reset to the 0001h value upon activation of the MR signal. Programming the BRG to 0000h is an illegal value. The minimum value for the BRG is 0001h. Until the BRG is programmed, no output is generated.
- The Output Data Bus always shows the value of the last register read.

Implementation Results

H16550S reference designs have been evaluated in a variety of technologies. The following are sample AMD results with IOBs assuming all core I/Os are routed off-chip. Different results can be obtained by optimizing slices for area or speed.

Supported Family	Slices	BRAM	IOBs	Fmax (MHz)
Spartan-6 6SLX25-3	152	2	39	149
Virtex-6 6VLX130T-3	135	2	39	270
Virtex-7X 7VX330T-3	154	0	39	387
Artix-7 7A350T-3	160	0	39	266
Kintex-7 7K325T-3	177	0	39	392

Support

The core as delivered is warranted against defects for ninety days from purchase. Thirty days of phone and email technical support are included, starting with the first interaction. Additional maintenance and support options are available.

Verification

The core has been verified through extensive synthesis, place and route, and simulation runs. It has been embedded in several shipping customer products, and is proven in both ASIC and FPGA technologies.

Deliverables

The core is available in synthesizable RTL and FPGA netlist forms, and includes everything required for successful implementation, including a sophisticated self-checking testbench, simulation scripts, test vectors, and expected results, synthesis scripts and comprehensive user documentation.