AU-G0600: Serial Peripheral Interface (SPI) AMBA APB Core

The AU-G0600 Serial Peripheral Interface (SPI) AMBA APB Core provides an industry standard Serial Peripheral Interface (SPI) for AMBA based SOCs. It supports master (including multi master) and slave SPI modes with SPI word lengths up to 32 bits. Either the most significant bit or least significant bit of each SPI word may be transferred first. All four combinations of SPI clock idle polarity and SPI clock data transfer phase are implemented. The SPI AMBA APB Core connects seamlessly to the AMBA APB Bus as an AMBA Bus slave. The Serial Peripheral Interface (SPI) AMBA APB Core is available as a synthesizable Verilog model from Aurora VLSI, Inc. Contact CustomerService@auroravlsi.com.

Software configures the SPI AMBA APB Core to operate in either master or slave mode. In master mode, the SPI Core generates the SPI clock from the AMBA Bus clock with a SPI clock period resolution of twice the AMBA Bus clock period, as programmed by software. Sixteen dedicated slave select lines support sixteen SPI slaves. The slave select signals can be hardware generated or driven directly by software. When hardware generated, the slave select signals can be configured to remain asserted between SPI words or to be de-asserted for a software programmable time. The delay from slave select assertion or the end of the previous SPI word, to the first clock of the next SPI word is software configurable to support slaves that can not respond immediately. Optionally, the SPI clock can also be delayed by a wait input signal. To support multiple master systems, the master mode outputs can be configured as open drain or high impedance when de-asserted. Optionally in master mode, when the SPI Core is not idle, a master collision error is detected when another master asserts a de-asserted slave select. This causes the SPI Core to switch to slave mode and become disabled.

In slave mode, the SPI Core transmits and receives data when selected and clocked by an external SPI master. Typically, the external SPI master first asserts an interrupt that causes transmit data to be written to the SPI core. The external SPI master then selects the SPI Core by asserting its slave select input, and drives the SPI clock to it to transfer data to and from the SPI Core. If there is no transmit data in the SPI Core, it can be configured to send zeros or the last transmit data. When receiving data, if there is no space available, the SPI Core can be configured to drop the incoming data or overwrite existing receive data. A software driven wait signal is provided for use when the external SPI master supports it. Master broadcast mode is supported by optionally disabling the slave data output.

Each 32 bit AMBA Bus transfer is packed with four byte size data, two halfword data, or one word datum as configured by software. Each of these AMBA Bus data holds one right justified SPI word. Receive data is optionally zero extended or sign extended to fill each AMBA Bus datum. To facilitate DMA and CPU data transfers to/from the SPI Core, there is a transmit data FIFO and a receive data FIFO at the AMBA Bus interface. The sizes of these data FIFOs are user configurable from 2 to 32 entries where each entry is 32 bits wide. A transmit DMA request or interrupt is optionally asserted when the transmit data FIFO empties below a software programmable threshold. When the receive data FIFO is filled above a software programmable threshold, a receive DMA request or interrupt is optionally signaled.

The SPI Core detects several error conditions- transmit underrun, receive overrun, master collisions, transmit data FIFO writes to a full transmit data FIFO, receive data FIFO reads from an empty receive data FIFO, and slave selection of the SPI Core when it is disabled. Each of these error conditions can signal an independently maskable interrupt.

AMBA is a trademark of ARM Limited

The Serial Peripheral Interface (SPI) AMBA APB Core features are summarized:

- Master mode including multi master capability, and slave mode
- Transmit and receive SPI word lengths up to 32 bits- software configurable
- MSB or LSB transferred first- software configurable
- Implements all SPI clock polarity and clock phase modes
- Master mode:
  - supports 16 SPI slaves
  - configurable SPI clock period- resolution of 2 x AMBA clock period
  - software configurable delay to first clock of each SPI word
  - optional wait signal to delay first clock of each SPI word
  - optional de-assertion of slave selects between each SPI word
  - hardware generated or software driven slave selects- configurable
  - SPI clock, slave select, and data outputs support push pull (both driven and hiZ when de-asserted), and open drain pad drivers
  - optionally detects master collision error with multiple masters
- Slave mode:
  - sign or zero extend receive data to fill AMBA bus datum
  - last data or zero sent on transmit underrun
  - drop or overwrite on receive overrun
  - optional wait/enable output
  - master broadcast supported
  - data and wait outputs support push pull (both driven and hiZ when de-asserted), and open drain pad drivers
- AMBA interface
  - byte, halfword, or word AMBA bus datum size- configurable
  - one SPI word per AMBA bus datum of the same or greater size
  - Each 32 bit AMBA bus transfer is packed with four AMBA data bytes, two AMBA data halfwords, or one AMBA datum word
  - transmit and receive FIFOs- 8, 16, 32, 64, or 128 bytes each (configurable)
- DMA support
  - programmable count up to 64K SPI words
  - optional DMA request signals based of empty/full FIFO status
- Independently maskable interrupts
  - SPI word transfer complete
  - transmit FIFO empty below a configurable threshold
  - receive FIFO full above a configurable threshold
  - slave select de-assertion (for master collision error handling)
  - SPI core in slave mode selected while disabled- error
  - slave mode transmit underrun- error
  - slave mode receive overrun- error
  - transmit FIFO write when full- error
  - receive FIFO read when empty- error
  - master collision with multiple masters- error

The core is delivered as a synthesizeable RTL Verilog model. Deliverables include:

- RTL Verilog source code model of the core
- Verilog testbench and test cases
- Synthesis scripts examples
- Complete detailed documentation and training class notes

AMBA is a trademark of ARM Limited