APB I2C Master/Slave Controller

SILVACO

Overview

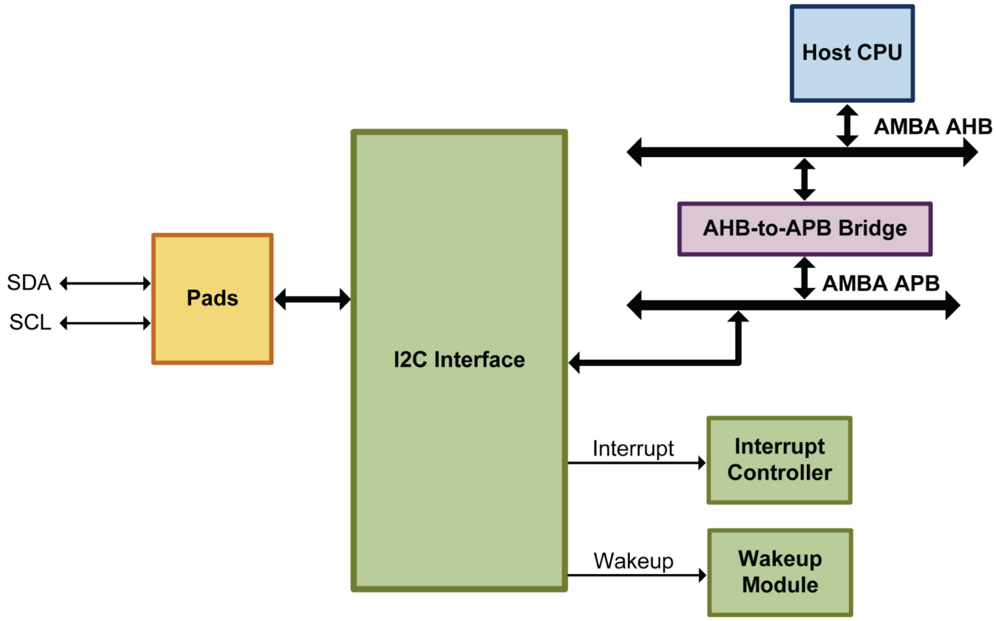
The I2C Interface provides full support for the two-wire I2C synchronous serial interface, compatible with the ACCESS. Bus physical layer, with additional support for the SMBus protocol, including Packet Error Checking (PEC). Through its I2C compatibility, it provides a simple interface to a wide range of low-cost memories and I/O devices, including: EEPROMs, SRAMs, timers, A/D converters, D/A converters, clock chips, and peripheral drivers. The I2C is the same I2C Interface IP is proven in high-volume devices.

The host interface of the I2C Interface complies with the AMBA 2 APB protocol. Control registers provide CPU control of Serial Clock Line (SCL) frequency, start and stop condition generation, PEC byte generation, I2C address assignment, 7-bit or 10-bit addressing, and enabling/disabling interrupts. Status registers indicate current operating mode, packet error, and interrupt status. A serial data register shifts the serial I2C data into and out of the I2C Interface during receive and transmit operations.

The I2C serial interface consists of the standard bidirectional I2C signals: Serial Clock Line (SCL) and Serial Data Line (SDA). At the I2C IP level, there are separate unidirectional signals (scl_in_pin/scl_out and sda_in_pin/sda_out). To reduce chip-level pin count, the I2C bus interface signals can be shared with other on-chip functions through a General Purpose I/O (GPIO) Controller.

Features

- Compliant to SMBus (versions 1.1 and 2.0), ACCESS. Bus, and I2C (version 2.1)
- Supports Standard Mode, Fast Mode (F/S), and High Speed (Hs) Mode
- Programmable master or slave operation
- Multi-master capable
- One software-defined slave address
- 7-bit or 10-bit slave addressing
- Global Call (broadcast) address support
- Specific SMBus features:
- SCL Timeout detection
- Packet Error Checking (PEC)
- Alert Response Address
- Supports polling and interrupt-controlled operation
- Wakeup signal generation upon detection of a start condition while in power-down mode
- Local clock gating for minimal power consumption



I2C Protocol

The I2C protocol uses a two-wire interface for bidirectional communications between the devices connected to the bus. The two interface lines, SCL and SDA, are connected to a positive supply through pull-up resistors. Any device connected to the bus uses open-collector or open-drain drivers and can only pull the respective bus line low. Therefore, both the SCL and SDA are wired-AND type signals.

The device that initiates the transaction becomes the master of the bus. The bus master generates SCL and terminates the transaction once finished. One data bit is transferred on SDA during each pulse on SCL. Data is sampled during the high state of SCL and can change while SCL is low.

Each data transaction is composed of a start condition, a number of byte transfers, and a stop condition to end the transaction. Each byte is transferred with the Most Significant Bit (MSB) first. After each byte (8 bits), an acknowledge signal must follow.

A slave device can stall the master by extending the low period of the SCL clock while it handles the previous data or prepares new data. This process can occur after each bit is transferred or on a byte boundary. The slave stalls the bus by pulling SCL low to extend the clock-low period. Typically, slaves extend the first clock cycle of a transfer if a byte read has not yet been stored or if the next byte to be transmitted is not yet ready.

SMBUS Features

The SMBus protocol (supported by the I2C Interface) is a superset of the I2C protocol and adds the following features:

- SCL Timeout detection – If the SCL low period exceeds the SCL timeout value, the device aborts the current operation
- Packet Error Checking (PEC) through the use of a Cyclic Redundancy Check (CRC) byte
- Alert Response Address, which provides slave-only devices an interrupt capability

Interfaces

- AMBA 2 APB host interface
- 8-bit read/write data buses
- 10-bit address bus
- I2C bus interface pins (SCL and SDA) through chip I/O pads (optionally through a GPIO Controller)
- System (APB) clock
- One interrupt signal
- One asynchronous reset input
- One wakeup signal
- Signals to control idle mode and halt mode
- DFT signals

Deliverables

- Synthesizable Verilog source code
- Integration testbench and tests
- Documentation
- Scripts for simulation and synthesis with support for common EDA tools