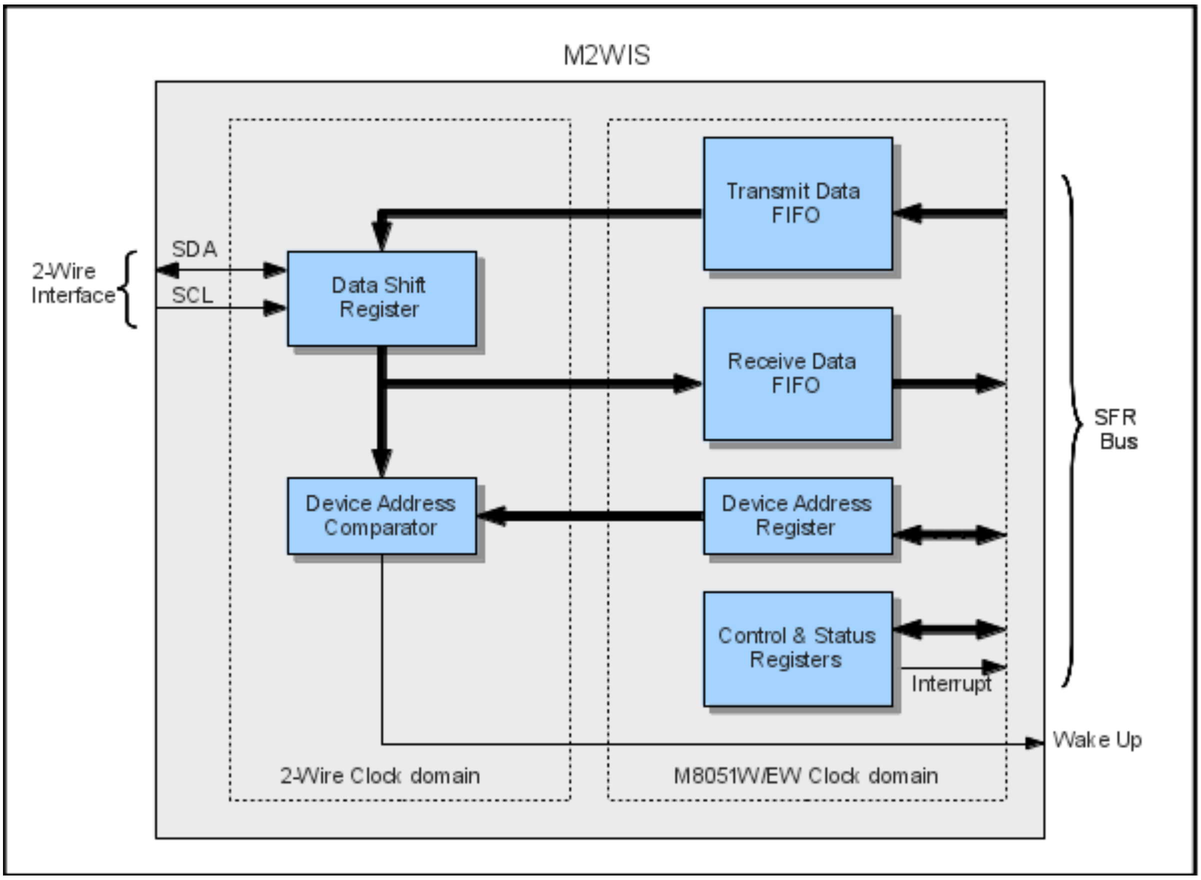
M2WIS
Two-Wire Slave Interface for M8051W and M8051EW Microcontrollers

Overview

The M2WIS adds two-wire slave capability to M8051W and M8051EW designs. Use of standard synchronous design methodology makes this core simple to integrate into both ASIC and FPGA SoC designs.

M2WIS Two-Wire Slave Structure



- Implementation of a two-wire slave interface, compatible with I^{2}C bus standard
- Compatible with I^{2}C Fast and Fast Mode plus signalling
- Supports seven and ten-bit addressing modes, with optional response to general call address
- Can implement flow control with stretching
- Supports any two-wire clock rate, independent of microcontroller clock rate
- Integral data FIFOs minimise processor overhead required to service the link
- Supports both interrupt-driven and polled transfers
- Optional clock prescaler minimises power consumption whilst active
- Power saving mode when the interface is not enabled
- Slave select wake up feature can be used to cold start a M8051W and M8051EW microcontroller, in addition to interrupt driven wake ups
- Binds tightly to M8051W and M8051EW external interrupt and SFR buses with no glue logic required

M2WIS
Two-Wire Slave Interface for M8051W and M8051EW Microcontrollers

Supported Protocols

The M2WIS is compatible with the I^2C Fast protocol. It includes hardware support for seven-bit and ten-bit device addressing modes and a general call address and address ranges. Clock stretching can be achieved under register control in order to implement flow control.

Interfacing

The M2WIS is design to bind tightly to an M8051W or M8051EW SFR bus. Data, control and status information are exchanged with a host microcontroller using memory-mapped 8-bit words. A single interrupt signal is used to indicate that a matching address has been received, that data buffers require service or that an exception has been detected.

Structure

The M2WIS comprises two independent clock domains: The two-wire clock domain and the host peripheral clock domain.

A nine-bit shift register, clocked by the external two-wire clock, transfers data octets in and out of the core. An associated acknowledge bit is generated by the receiving two-wire agent for each octet transferred. Start and stop framing events that are generated by the corresponding two-wire master device are detected in the M2WIS.

Data for transmission over the two-wire interface is first written by the M8051W and M8051EW into the M2WIS transmit FIFO, which is located in the SFR address space. When the external two-wire master executes a read transfer, data from the transmit FIFO is shifted out onto the two-wire interface. When the external two-wire master executes a write transfer, data received by the M2WIS shift register is written to a receive FIFO. Data in the receive FIFO is read by the host M8051W and M8051EW via the SFR address space.

Power Management

The M2WIS is compatible with the standard M8051W and M8051EW three power saving states. In the M8051W and M8051EW idle state, the M2WIS continues to function, transferring data in or out of data FIFOs and generating an interrupt when a transfer is complete or a FIFO requires attention.

A clock prescaler enables designs to select the minimum clock frequency for correct operation of the M2WIS logic. This may be many times slower than the host microcontroller clock frequency.

When the M2WIS is not enabled, and there is no activity on the two-wire interface, power consumption drops to leakage levels only.

The M2WIS can be configured to generate a host wake-up signal whenever a device address match is detected. This enables the host logic to be left in a deep power saving state between two-wire messages.

Configuration Options

The core RTL is highly configurable at compile time allowing users to implement only the features required by their application.

Major configuration options include:

- Data FIFO depths from 1 to 255 bytes
- Single clock design, or independent two-wire and microcontroller clock domains
- Optional clock stretching control
- Seven-bit or ten-bit device addressing
- Device address range matching
- General call address recognition
- User defined SFR base address
- User defined interrupt channel
- Microcontroller clock domain prescaler
- Optional microcontroller wake-up signal

Deliverables

- VHDL '93 and Verilog 2001 RTL source code
- VHDL and Verilog functional demonstration testbench
- Demonstration assembler code
- Simulation scripts for Modelsim and Cadence
- Synopsys synthesis compile scripts and SDC timing constraint files
- Example Mentor DFT and ATPG scripts
- Example netlist implementation with SDF files
- Detailed product specification and a user guide containing implementation notes