I^2C Controller for Serial EEPROMs

February 2002 | Reference Design RD1006

Introduction

The I^2C bus provides a simple two-wire means of communication. This protocol is used in many applications. SDRAM modules implement a serial EEPROM that supports the I^2C protocol. This is used so that a microprocessor can read the EEPROM for configuration purposes.

This reference design documents an I^2C Controller designed to interface with serial EEPROM devices. It is intended to be a simple controller providing random reads cycles only. Typically, serial EEPROMs are programmed at board assembly time and store configuration information, which is read by a microprocessor during power-up.

This design was implemented in Verilog, synthesized and fitted using Lattice design tools into an ispMACH™ 4A or ispMACH 4000 device. The design requires 46 macrocells and 24 I/O pins. Using an M4A-64/32-55 or an LC4256B-5T100C yields approximately 153MHz performance. Results may vary according to the synthesis tool.

This design assumes the user has experience with I^2C controllers. Information available in documents listed in the Applicable Documents section, below, is not repeated in this document.

Applicable Documents

- National NM24C16 16,384-Bit Serial EEPROM
- Philips I^2C Specification
- Lattice Semiconductor Data Book CD-ROM

Theory of Operation

Overview

This I^2C controller provides an interface between standard microprocessors and I^2C serial EEPROM devices. It supports random read cycles only. The design consists of the following modules:

- I2c_sep Top level module
- I2c_clk Clock generation module
- I2c_rreg Read register module
- I2c_st State machine module
- I2c_tbuf Tri-state buffer module
- I2c_wreg Write register module

Top Level Signal Description

Table 1 provides the input/output signals of the I^2C Controller. Signals ending with "_L" indicate an active low signal. This convention is used throughout the design. The chip select signal is assumed synchronous to the clock. Address, data and the read/write signals are assumed valid at the assertion of the chip select.

Register Description

Table 2 lists the I^2C Controller registers.

Table 1. I²C Signals

Signal	Type	Description
CS_L	Input	I²C Chip Select from processor
RD_WR_L	Input	Write pulse from processor
ADDR[1:0]	Input	Address bus from microprocessor
DATA[7:0]	I/O	Microprocessor data bus
CLK	Input	Input clock
RST_L	Input	Asynchronous Reset
REG_CLK_IN	Input	Clock used to latch Word Address
ACK_L	Output	Microprocessor Cycle Acknowledge
REG_CLK_OUT	Output	Generated clock to latch Word Address
SCL_PIN	Output	I²C clock pin
SDA_PIN	I/O	I²C data pin

Table 2. I²C Registers

Register	Address	Type
Word Address	00	Read/Write
Data	01	Read
Status	10	Read

Design Module Description

I2C_CLK Module

The I2C_CLK module provides a one-microprocessor clock tick wide pulse every 5µsec. This is used to control the state machine. The 8-bit counter in this module can be adjusted to meet the needs of any design. If a faster microprocessor clock is used, a bigger counter will be needed. If it is desired to run the I²C bus at a faster speed, the counter can be smaller. This design assumes a microprocessor clock frequency of 50MHz.

I2C_RREG Module

The I2C_RREG module provides a multiplexor for reading data back to the CPU. The word address, the data read from the I²C device and status information can be read back. The status register bits are defined in Table 3.

Table 3. Status Register Bits

Bit	Signal	Description
7	ready	The I²C device has responded back with read data.
6	ack_err	The I²C device did not acknowledge the current read cycle.
0	Active	An I²C cycle is in progress.

I2C_ST Module

The I2C_state module provides:

- A state machine to control the I²C cycles
- A bit counter to count the bit phases
- I²C clock generation
- I²C data generation
- Ready generation
- Error generation

I2C_TBUF Module

The I2C_TBUF module provides the tri-state buffer for the microprocessor data. It also provides the open drain outputs for the I^{2}C clock and data signals.

I2C_WREG Module

The I2C_WREG module provides the word address register. When this register is written to, an i2c_go signal is generated. This causes the state machine to start a random read cycle to the address stored in this register.

A clock, reg_clk_out, is generated in this module to allow the use of input registers for the word address register. In order to use the input registers, this clock signal must be routed on the board to a dedicated clock input pin on the ispMACH device.

A microprocessor acknowledge signal is provided in this module. This is the chip select signal delayed by one clock tick. This signal could be omitted from the design if this function is not needed or performed elsewhere.

Test Bench Description

The test bench for this design includes the following modules:

- CLK_RST
- I2C_SLAVE
- I2C_TB
- MICRO

CLK_RST Module

The CLK_RST module provides the clock and reset signals to the test bench. Editing the CLK_PERIOD parameter changes the clock frequency. Editing the RESET_TIME parameter changes the duration of reset. The reset_recovery parameter holds off the clock until a fixed time allowing all the ispMACH registers to recovery from reset.

I2C_SLAVE Module

The I2C_SLAVE module provides a model of a I^{2}C memory device which features 256 memory locations. The slave module is able to detect START and STOP commands from the I^{2}C controller. It generates ACK after the word address cycle and leaves the I^{2}C bus tristated after data read. The data stored in the memory is the same as its address. This provides a convenient way to check the data by comparing “Slave Data Receive on Write” and “Slave Data Transmitted on Read” messages during simulation.

I2C_TB Module

The I2C_TB module is the top-level test bench for the design. It instantiates the I^{2}C controller as well as the modules in the test bench.

MICRO Module

The MICRO module provides the microprocessor stimulus to the I^{2}C Controller. This module provides tasks to simulate write and read cycles to the I^{2}C Controller. It also performs error checking on the data read back from the I^{2}C Controller.

Design Flow

Simulation

The design was simulated using the Model Technology ModelSim simulator. Scripts are provided for RTL and gate level simulation.

Synthesis

This design was synthesized using Exemplar Spectrum and Synplify from Synplicity. The output file from either tool will be an EDIF file.

Fitting

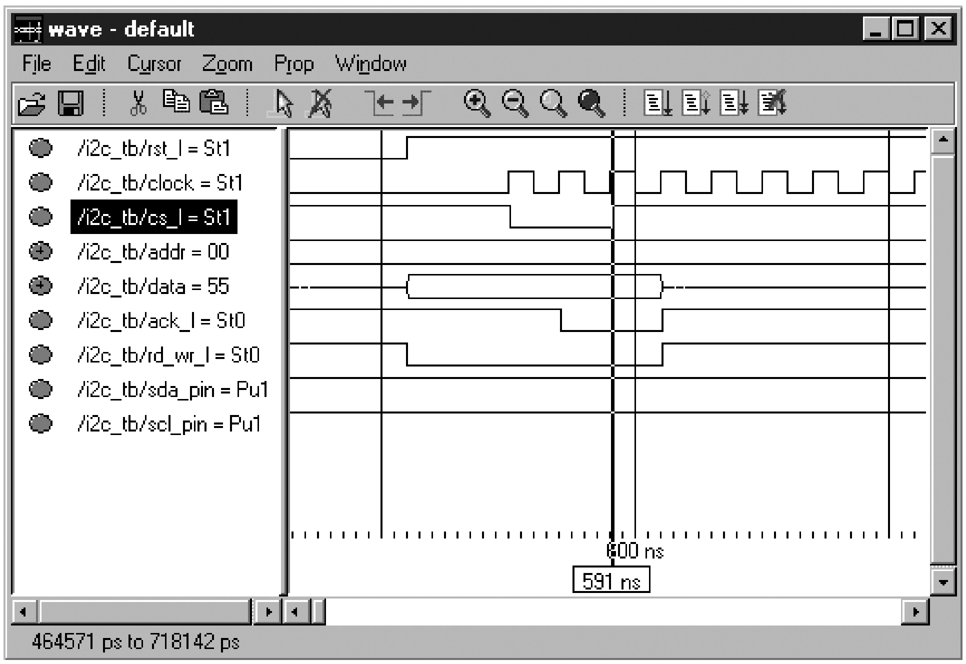
The EDIF file is imported into the Lattice design tool and targeted to an ispMACH 4A or ispMACH 4000 device. Running the Timing Analyzer shows approximately 153MHz performance when a 5ns or 5.5ns device is targeted.

Timing Diagrams

The following timing diagrams were created from Model Technology's ModelSim version 5.2 Simulator.

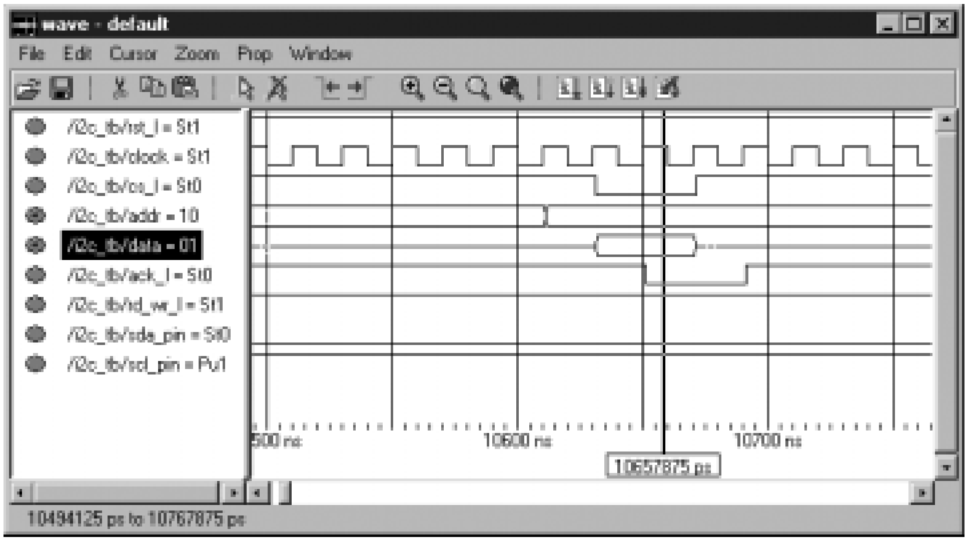
Microprocessor Write Cycle

The microprocessor is writing a "55" to the word address register. This will cause the I^{2}C Controller state machine to start a random read cycle to the I^{2}C device.



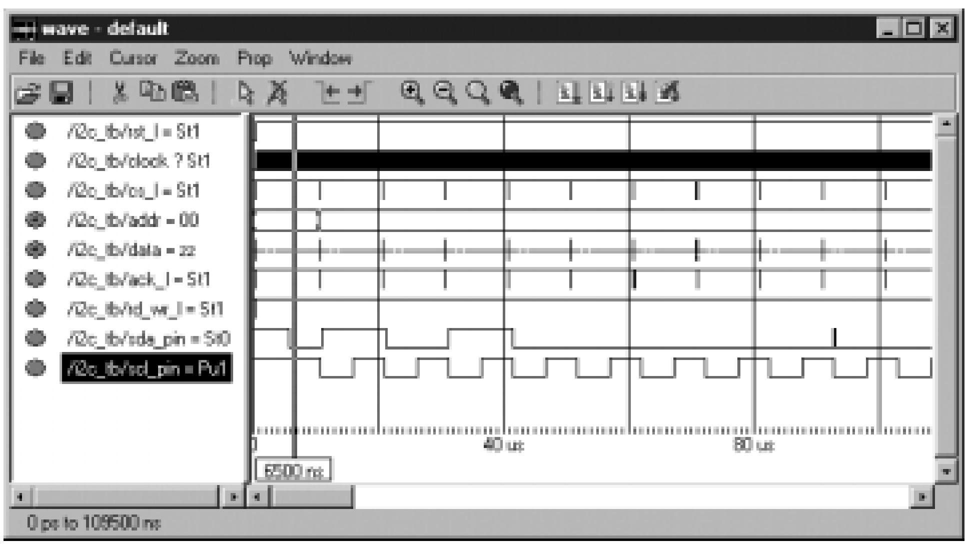
Microprocessor Read Cycle

The microprocessor is reading the status register. Bit 1 is set indicating an I^{2}C cycle is in progress. Bits 6 and 7 are cleared. This indicates the cycle has not completed and that an error has not been detected.



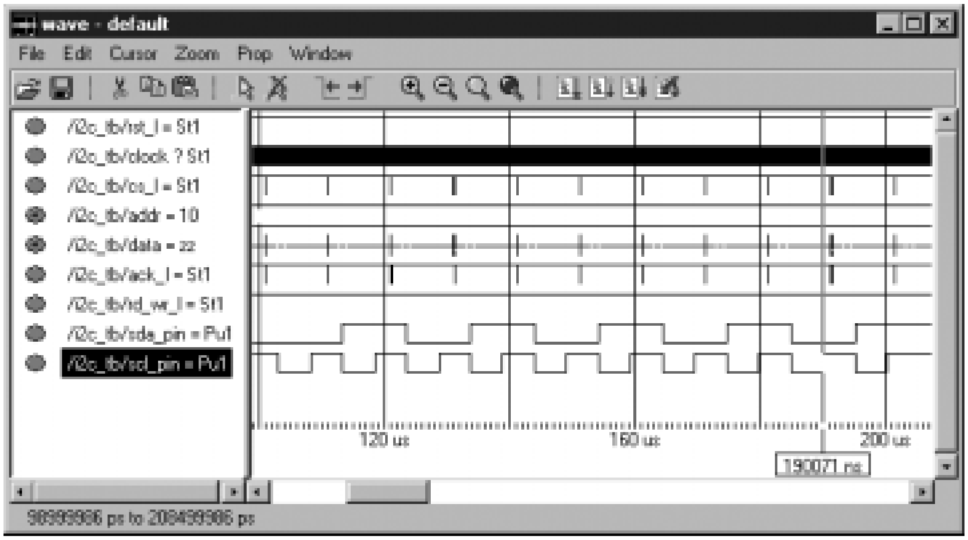
I^{2}C Device Address Write Cycle

An I^{2}C cycle starting followed by the device address and an acknowledge from the I^{2}C device.



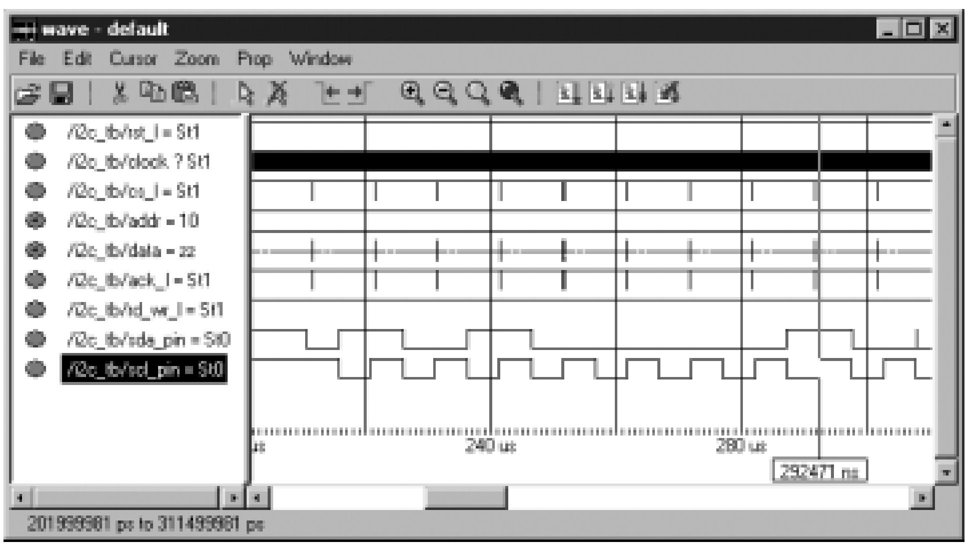
I^2C Word Address Cycle

The word address cycle followed by an acknowledge from the I^2C device.



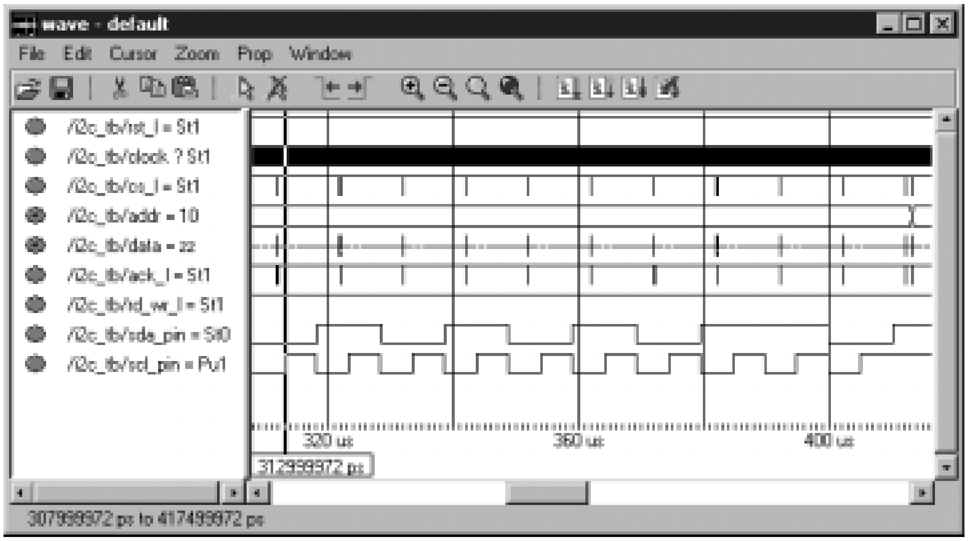
I^2C Device Address Read Cycle

The second device address cycle, this time requesting a read.



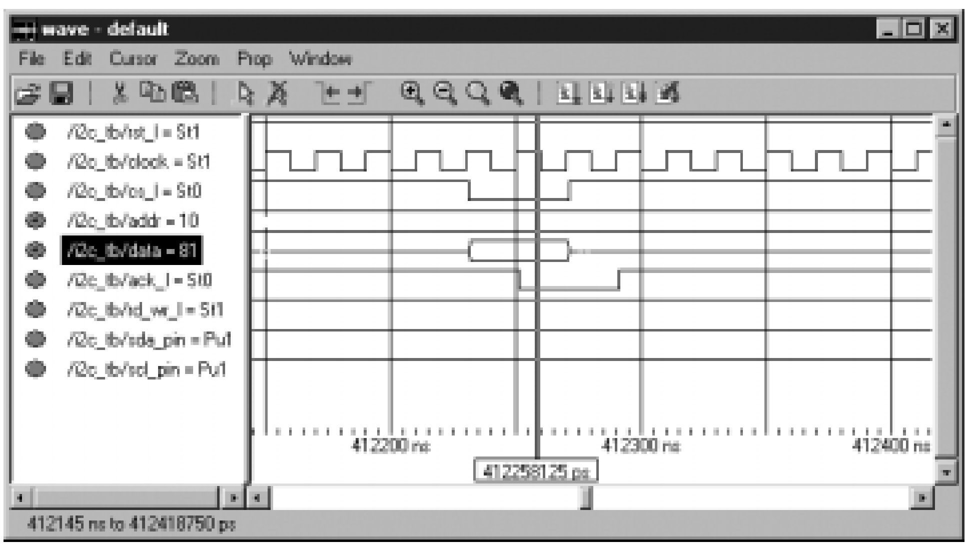
I^2C Read Data Cycle

The data is read from the I^2C device. This cycle is done without an acknowledge. A stop is asserted by the controller.



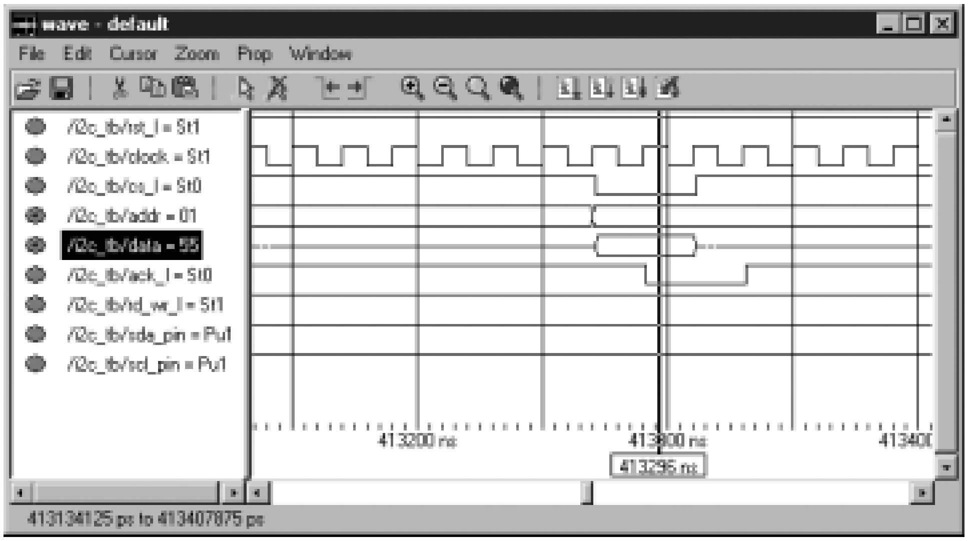
Microprocessor Read Status Cycle

The status register is read indicating the I^2C cycle has completed.



Microprocessor Read Data Cycle

The I²C data is read by the microprocessor.



The I²C bus is patented by Philips Electronics Corporation. An I²C license is required from Philips before obtaining and using any source code mentioned in this reference design document. Contact Lattice Technical Support (below) for instructions regarding this process.

Technical Support Assistance

Hotline: 1-800-LATTICE (Domestic)

1-408-826-6002 (International)

e-mail: techsupport@latticesemi.com