Accessing Control Registers Through the MDIO Bus

Reference Design

FPGA-RD-02130-1.2

January 2020

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS and with all faults, and all risk associated with such information is entirely with Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

© 2011-2020 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Contents

Figures

Tables

© 2011-2020 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

1. Introduction

Management Data Input/Output Interfaces, or MDIO, are specified in the IEEE 802.3 standard. Their primary application is to provide a Serial Management Interface (SMI) to transfer management data between an Ethernet Media Access Controller (MAC) and a physical layer device (PHY). The MDIO interface consists of two pins, a bidirectional MDIO pin and a Management Data Clock (MDC) pin. All data is transferred synchronously to the MDC which is usually provided by the MAC core or a master controller and sourced to all slave devices. The MDIO is a relatively slow interface running up to 2.5 MHz. However, its ability to access and modify various registers in PHY devices by the MAC often extends the application beyond the Ethernet system. The two-wire interface also provides a solution for systems where a limited pin count is desired.

The device that controls the MDIO bus is called a Station Management Entity (STA), while the device being managed is called the MDIO Manageable Device (MMD). The STA device is often embedded in the MAC core to handle parallel-to-serial conversion. It is responsible for all read and write transactions to and from slave devices. The MMD is often embedded in the PHY device. It updates the registers and outputs the status to the STA device. This design is based on, and is a subset of, the OpenCores Ethernet IP Core design. It implements a MDIO slave interface for a MMD device. It accomplishes the writing and reading of registers with an MDIO frame structure as defined in the IEEE 802.3 Standard, Clause 22. Figure 1.1 shows a typical application environment of the MDIO bus. This design also has a WISHBONE interface. The WISHBONE interface updates the read-only registers defined in the IEEE 802.3 Standard, Clause 22.

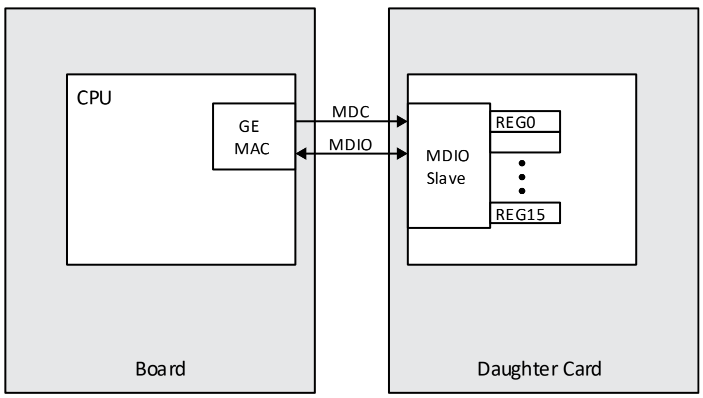

Figure 1.1. MDIO Interface for a Generic Application

2. Features

- Implements the IEEE 802.3 Standard, Clause 22 interface
- Support 16 registers, as defined in the IEEE 802.3 Standard, Clause 22
- All registers can be read through the MDIO bus
- All registers can be read through the WISHBONE bus
- All R/Wa registers can be written through the MDIO bus
- All ROb registers can be written through the WISHBONE bus
- The slave PHY address can be set with the WISHBONE bus

Note: R/W[a] = Read/Write, RO[b] = Read Only

© 2011-2020 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

3. Functional Description

The "clause 22" MDIO interface can access up to 32 registers in 32 different PHY devices. The STA initiates a command using an MDIO frame and provides the target port address and register address. The STA provides data during the write command while the MMD takes over the bus and supplies the STA with data during the read command.

The MDIO slave interface serially receives the configuration information from the STA device to control the user registers and transmits the status information to the STA device. The bus is initially in idle state (tristated). To initialize a transaction, the STA device supplies 32 '1's to the MDIO pin. This is the preamble of the MDIO frame. The MDIO frame format can be found in Table 3.1

Table 3.1. Frame Definition

	Management Frame Fields							IDLE
	PRE	ST	OP	PHYAD	REGAD	TA	DATA	
Read	1...1	01	10	AAAAA	RRRR	Z0	DDDDDDDDDDDDDDDD	Z
Write	1...1	01	01	AAAAA	RRRR	10	DDDDDDDDDDDDDDDD	Z

The timing diagrams for read/write operations on the MDIO bus are shown in Figure 3.2 and Figure 3.2.

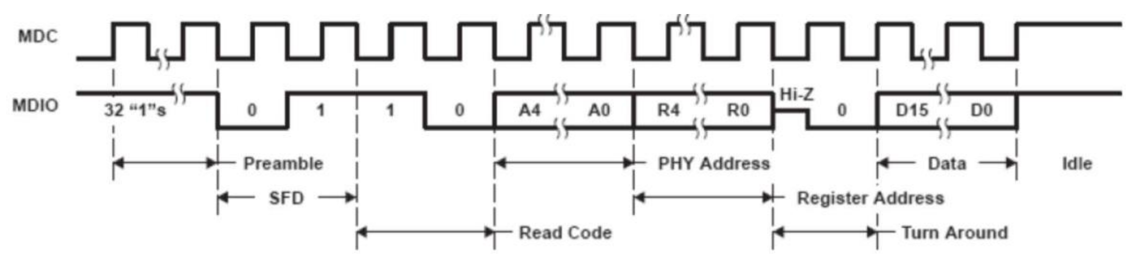

Figure 3.1. Read Timing

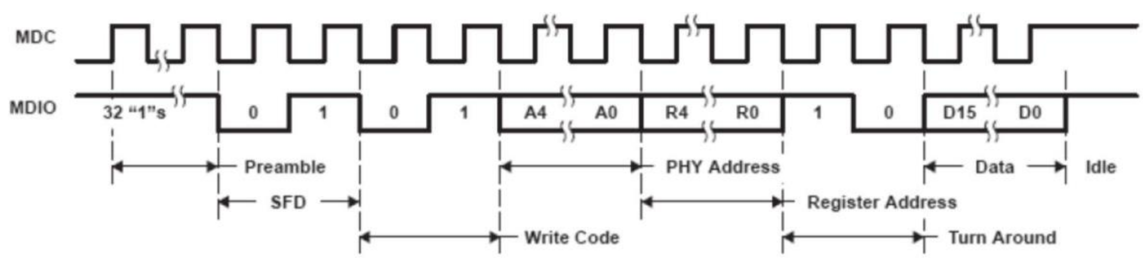

Figure 3.2. Write Timing

According to the frame structure and the read/write timing, the MDIO data transmission can be divided into several stages, as shown below.

- The MDIO bus keeps hi-Z in the idle state
- STA sends 32 bits of '1's to initialize the transaction
- Start of frame bits "01"
- The MMD determines which operation is the next step according to the 2-bit op-code
- The STA sends the 5-bit PHY address
- The STA sends the 5-bit register address
- The MDIO bus is in the 2-bit turn-around state according to the command
- The MMD receives or transmits data serially
- The MDIO bus keeps hi-Z and enters into the idle state

© 2011-2020 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Table 3.2 lists the management registers defined in the IEEE 802.3 Standard, Clause 22.

Table 3.2. Management Register Set

Register Address	Description	Read/Write
0	Control	Read and Write
1	Status	Read Only
2	PHY Identifier	Read Only
3	PHY Identifier	Read Only
4	Auto-Negotiation Advertisement	Read and Write
5	Auto-Negotiation Link Partner Base Page Ability	Read Only
6	Auto-Negotiation Expansion	Read Only
7	Auto-Negotiation Next Page Transmit	Read and Write
8	Auto-Negotiation Link Partner Received Next Page	Read Only
9	Master-Slave Control Register	Read and Write
10	Master-Slave Status Register	Read Only
11	PSE Control Register	Read and Write
12	PSE Status Register	Read Only
13	MMD Access Control Register	Read and Write
14	MMD Access Address Data Register	Read and Write
15	Extended Status	Read Only

4. Design Module

A block diagram of the design is shown in Figure 4.1 Pin descriptions are listed in Table 4.1

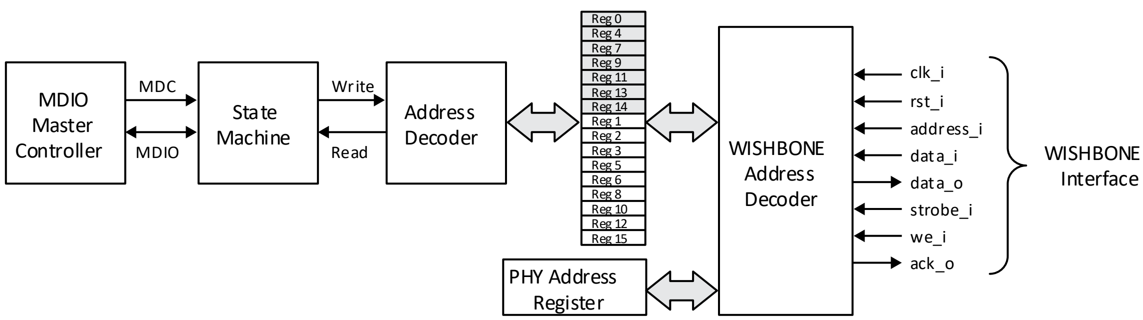

Figure 4.1. Block Diagram

© 2011-2020 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Table 4.1. Signal Descriptions

Port Name	Direction	Description
MDIO Interface		
mdio	Input	Serial data line of the MDIO bus
mdc	Input	MDIO bus clock
rst_n	Input	Reset signal, active low
WISHBONE Interface		
clk_i	Input	CPU input clock
rst_i	Input	WISHBONE interface synchronous reset
address_i	Input[7:0]	WISHBONE interface address
data_i	Input[7:0]	WISHBONE interface input data
data_o	Output[7:0]	WISHBONE interface output data
strobe_i	Input	WISHBONE interface strobe signal
we_i	Input	WISHBONE interface write enable
ack_o	Output	WISHBONE interface acknowledge

This design implements a MDIO slave interface. According to the IEEE 802.3 Standard, Clause 22 registers 0, 4, 7, 9, 11, 13 and 14 are read-write registers. The host can read and write these registers through the MDIO bus. Readonly registers can be read by the host through the MDIO bus. The WISHBONE interface provides a back-end generic interface to the MMD device. All the registers can be read through this interface. Read-only registers can be modified and written to through the WISHBONE interface.

The design also includes a phy_address register. This register is used to configure the physical address of the slave MDIO device. This register is controlled by the WISHBONE interface.

When the MDIO bus is idle, the state machine waits for the preamble and the start signal. The "01" on the MDIO bus indicates a valid start signal. When the start signal is received, the state machine moves to the next state to check the op-code. The op-code "01" indicates a write request and the "10" indicates a read request by the STA. After the op-code is sent, the STA transmits the 10-bit address data on the MDIO bus. The first five bits are the phy_address and the last five bits are the reg_address. After these bits are received, the state machine enters the turn-around state.

During the read operation, the state machine verifies the phy_address in the turn-around state. If verification of the phy_address fails, the state machine returns to the idle state to wait for the next start signal. If verification is successful, the state machine begins to control the MDIO bus and moves to the transmit data state until the selected register data is completely transmitted.

During the write operation, the state machine verifies the phy_address in the last state. If the verification of the phy_address fails, the state machine does not write the data and enters the idle state. If verification is successful, the state machine generates a flag signal to write data into the selected register. Figure 4.2 shows the state machine of the design. Table 4.2 lists the WISHBONE register addresses and their default values. The WISHBONE register addresses are decoded to match to the MDIO management addresses in the design.

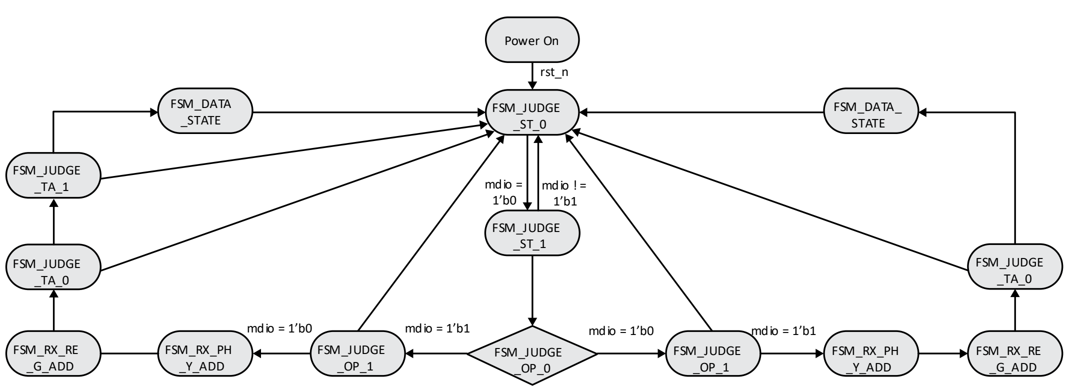

Figure 4.2. Read/Write State Machine

© 2011-2020 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Table 4.2. WISHBONE Register Addresses Mapping to Management Register Addresses

Register Address	Register Name	WISHBONE Interface	Default Value	Description
8'h0	Control register high byte	Read-only	8'h00	Control register
8'h1	Control register low byte	Read-only	8'h00	Control register
8'h2	Status register high byte	Read/Write	8'h00	Status register
8h'3	Status register low byte	Read/Write	8'h00	Status register
8'h4	PHY id1 high byte	Read/Write	8'h00	PHY Identifier register
8'h5	PHY id1 low byte	Read/Write	8'h00	PHY Identifier register
8'h6	PHY id2 high byte	Read/Write	8'h00	PHY Identifier register
8'h7	PHY id2 low byte	Read/Write	8'h00	PHY Identifier register
8'h8	Register4 high byte	Read-only	8'h00	Auto-negotiation advertisement register
8'h9	Register4 low byte	Read-only	8'h00	Auto-negotiation advertisement register
8'ha	Register5 high byte	Read/Write	8'h00	Auto-negotiation link partner ability register
8'hb	Register5 low byte	Read/Write	8'h00	Auto-negotiation link partner ability register
8'hc	Register6 high byte	Read/Write	8'h00	Auto-negotiation expansion register
8'hd	Register6 low byte	Read/Write	8'h00	Auto-negotiation expansion register
8'he	Register7 high byte	Read-only	8'h00	Auto-negotiation next page transmit register
8'hf	Register7 low byte	Read-only	8'h00	Auto-negotiation next page transmit register
8'h10	Register8 high byte	Read/Write	8'h00	Auto-negotiation link partner received next page register
8'h11	Register8 low byte	Read/Write	8'h00	Auto-negotiation link partner received next page register
8'h12	Register9 high byte	Read-only	8'h00	100BASE-T2 control register/1000BASE-T2 control register (master slave)
8'h13	Register9 low byte	Read-only	8'h00	100BASE-T2 control register/1000BASE-T2 control register (master slave)
8'h14	Register10 high byte	Read/Write	8'h00	100BASE-T2 control register/1000BASE-T2 status register (master slave)
8'h15	Register10 low byte	Read/Write	8'h00	100BASE-T2 control register/1000BASE-T2 status register (master slave)
8'h16	Register11 high byte	Read-only	8'h00	PSE control register
8'h17	Register11 low byte	Read-only	8'h00	PSE control register
8'h18	Register12 high byte	Read/Write	8'h00	PSE status register
8'h19	Register12 low byte	Read/Write	8'h00	PSE status register
8'h1a	Register13 high byte	Read-only	8'h00	MMD access control register
8'h1b	Register13 low byte	Read-only	8'h00	MMD access control register
8'h1c	Register14 high byte	Read-only	8'h00	MMD access address data register
8'h1d	Register14 low byte	Read-only	8'h00	MMD access address data register
8'h1e	Register15 high byte	Read/Write	8'h00	Reserved register
8'h1f	Register15 low byte	Read/Write	8'h00	Reserved register
8'h40	phy_address	Read/Write	8'h01	PHY address register

© 2011-2020 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

5. Test Bench Description

The test bench includes three files:

- **tb_mdio_slave.v** – Top-level simulation file that emulates simple behaviors of the MDIO bus.
- **wb_master_model.v** – WISHBONE master model.
- **mdio_mdc_master.v** – Emulates the MDIO STA device according to Clause 22 of the IEEE802.3 Standard.

The WISHBONE interface includes six user registers in the test bench that can be configured to begin the transaction between the STA and MMD. The test bench registers are described in Table 5.1 Signal definitions can be found in the test bench source files.

Table 5.1. Test Bench Registers

Register Address	Register Name	WISHBONE Interface	Default Value	Description
8'h42	Execute register	Write Only	8'h1	The value of this register, '1' or '0', indicates one operation.
8'h43	PHY_address_master register	Write Only	8'h0	Users can set this register to select the device.
8'h44	reg_address register	Write Only	8'h0	Users can set this register to select a slave register.
8'h45	Data high byte register	Write Only	8'h0	Users can set this register to send data to a write slave register.
8'h46	Data high byte register	Write Only	8'h0	Users can set this register to send data to a write slave register.
8'h47	Request register	Write Only	8'h0	Request signal. '0' indicates one write request; '1' indicates one read request.

Read/write simulation is shown in Figure 5.1 and Figure 5.2

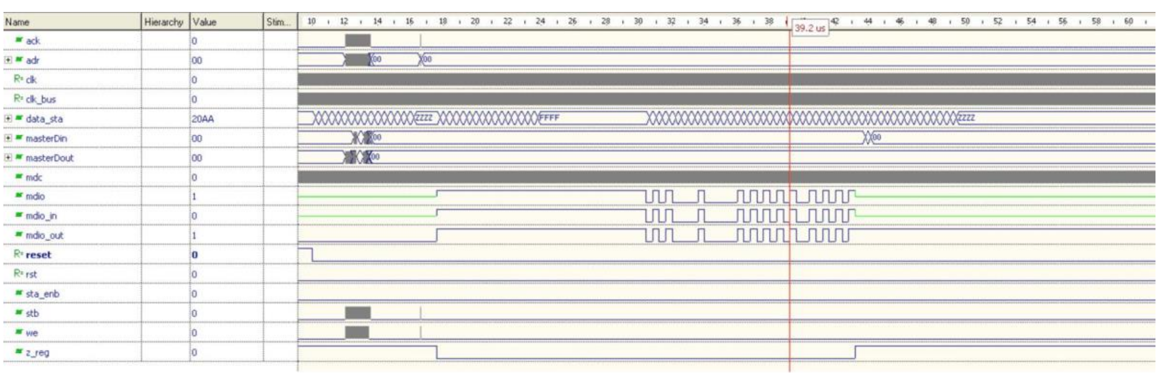

Figure 5.1. Write Operation (phy_address=5'd1, reg_address =5'd0 and data =16'hAA55)

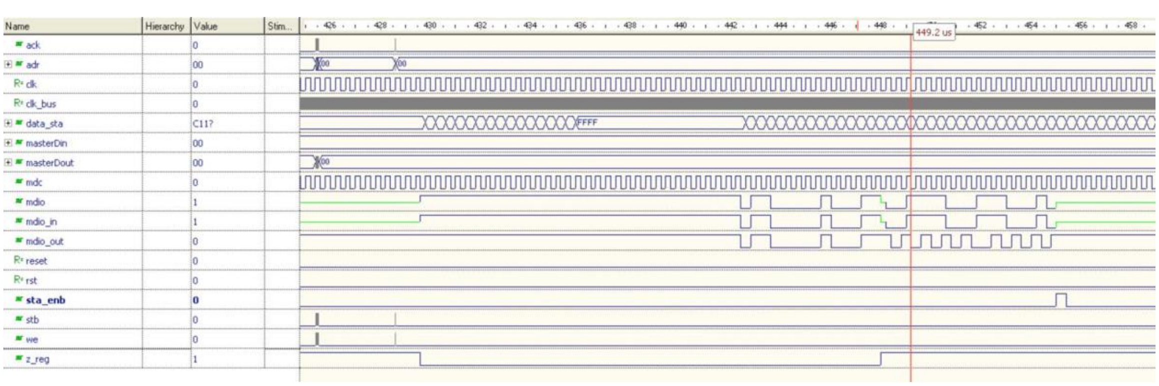

Figure 5.2. Read Operation (phy_address = 5'd1, reg_address=5'd3)

© 2011-2020 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

6. Implementation

This design is implemented in Verilog and VHDL. When using this design in a different device, density, speed, or grade, performance and utilization may vary. Default settings are used during the fitting of the design.

Table 6.1. Performance and Resource Utilization

Device Family	Language	Speed Grade	Utilization (LUTs)	f_{MAX} (MHz)	I/Os	Architecture Resources
MachXO™ [1]	Verilog	-4	460	>50	32	N/A
	VHDL	-4	440	>50	32	N/A
LatticeECP3™ [2]	Verilog	-7	460	>150	32	N/A
	VHDL	-7	440	>150	32	N/A

Notes:

1. Performance and resource utilization characteristics are generated using LCMXO640C-4T100C with Lattice Diamond™ 1.2 design software.
2. Performance and resource utilization characteristics are generated using LFE3-95EA-7FN1156C with Lattice Diamond 1.2 design software.

© 2011-2020 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

References

- Ethernet IP Core design from OpenCores, authors Tadej Markovic and Igor Mohor, www.opencores.org/projects/ethmac/
- IEEE 802.3 Standard

© 2011-2020 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

© 2011-2020 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Revision History

Revision 1.2, January 2020

Section	Change Summary
All	• Changed document number from RD1074 to FPGA-RD-02130. • Updated document template.
Disclaimers	Added this section.

Revision 1.1, April 2011

Section	Change Summary
All	• Updated for LatticeECP3 FPGA support. • Updated for Lattice Diamond design tool support.

Revision 1.0, February 2010

Section	Change Summary
All	Initial release.

© 2011-2020 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

www.latticesemi.com