LatticeECP3 and Broadcom 1 GbE (1000BASE-X) Physical/MAC Layer Interoperability

July 2010 Technical Note TN1217

Introduction

This technical note describes a 1000BASE-X physical/MAC layer Gigabit Ethernet interoperability test between a LatticeECP3™ device and the Broadcom BCM56800 network switch.

Specifically, the document discusses the following topics:

- Overview of LatticeECP3 devices and Broadcom BCM56800 network switch
- 1000BASE-X physical/MAC layer interoperability setup and results

A significant aspect of the interoperability test needs to be highlighted:

The BCM56800 uses a CX-4 HiGig™ port, whereas the LatticeECP3 Serial Protocol Board provides an SMA connector. A CX-4 to SMA conversion board was used as a physical medium interface to create a physical link between both boards. The SMA side of the CX-4 to SMA conversion board has four differential TX/RX channels (10 Gbps bandwidth total), but only one SMA channel (channel 2) was connected to the LatticeECP3 side.

LatticeECP3 Overview

The LatticeECP3 (EConomy Plus Third generation) family of FPGA devices is optimized to deliver high-performance features such as an enhanced DSP architecture, high-speed SERDES and high-speed source synchronous interfaces in an economical FPGA fabric. This combination is achieved through advances in device architecture and the use of 65 nm technology making the devices suitable for high-volume, high-speed, low-cost applications.

The LatticeECP3 device family also features high-speed SERDES with dedicated PCS functions. High jitter tolerance and low transmit jitter allow the SERDES and PCS blocks to be configured to support an array of popular data protocols including PCI Express, SMPTE, Ethernet (XAUI, GbE, and SGMII) and CPRI. Transmit Pre-emphasis and Receive Equalization settings make the SERDES suitable for transmission and reception over various forms of media.

LatticeECP3 PCS

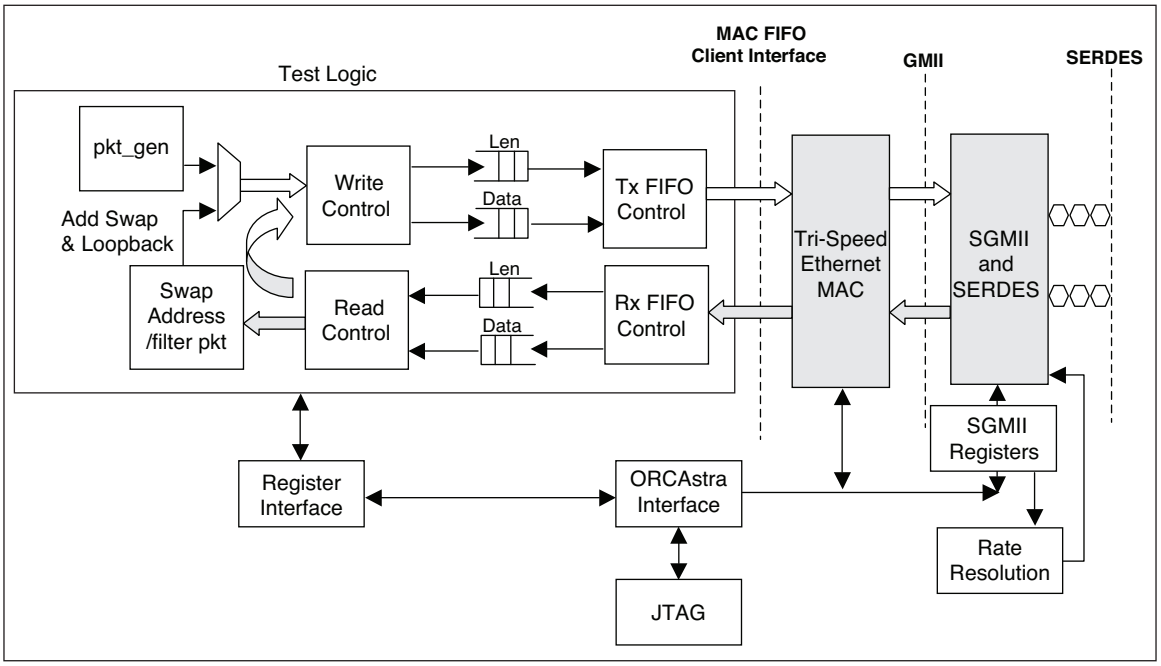
Each channel of PCS logic contains dedicated transmit and receive SERDES for high-speed, full-duplex serial data transfer at data rates up to 3.2 Gbps. The PCS logic in each channel can be configured to support an array of popular data protocols including GbE, XAUI, PCI Express, Serial RapidIO, CPRI, OBSAI, SD-SDI, HD-SDI and 3G-SDI. In addition, the protocol-based logic can be fully or partially bypassed in a number of configurations to allow users flexibility in designing their own high-speed data interfaces.

The PCS also provides bypass modes that allow a direct 8-bit or 10-bit interface from the SERDES to the FPGA logic. Each SERDES pin can be independently DC coupled and can allow for both high-speed and low-speed operation on the same SERDES pin for applications such as Serial Digital Video.

LatticeECP3 1000BASE-X PCS/Tri-Speed Ethernet MAC Reference Design

Figure 1 describes the Tri-Speed Ethernet MAC/SGMII reference design targeting the LatticeECP3 device. The reference design includes the SGMII and Gb Ethernet PCS IP core, the Tri-Speed Ethernet MAC IP core as well as test logic. ORCAstra logic controls and monitors the test logic, Tri-Speed Ethernet MAC and SGMII/Gb Ethernet PCS IP core registers. Register mapping information for all registers is described in Appendix A of this document. Please refer to IPUG51, Tri-Speed Ethernet MAC Core User's Guide and IPUG60, SGMII and Gb Ethernet PCS IP Core User's Guide for more information on the specific IP registers.

Figure 1. LatticeECP3 Tri-Speed Ethernet MAC + SGMII/GbE PCS IP Reference Design



SGMII and Gb Ethernet PCS IP Core

The SGMII and Gb Ethernet PCS IP core implements the PCS functions of both the Cisco SGMII and the IEEE 802.3z (1000BASE-X) specifications. The PCS mode is pin selectable. This IP core may be used in bridging applications and/or PHY implementations.

Features:

- Implements PCS functions of the Cisco SGMII Specification, revision 1.7
- Implements PCS functions for IEEE 802.3z (1000BASE-X)
- Dynamically selects SGMII/1000BASE-X PCS operation
- Supports MAC or PHY mode for SGMII auto-negotiation
- Supports (G)MII data rates of 1 Gbps, 100 Mbps, 10 Mbps
- Provides Management Interface Port for control and maintenance
- Includes Easy Connect option for seamless integration with the Tri-Speed Ethernet MAC IP core

Tri-Speed Ethernet MAC IP Core

The Tri-Speed Ethernet MAC IP core transmits and receives data between a host processor and an Ethernet network. The main function of the core is to ensure that the Media Access rules specified in the 802.3 IEEE standard are met while transmitting a frame of data over Ethernet.

Features:

- Compliant to IEEE 802.3z standard
- Generic 8-bit host interface
- 8-bit wide internal data path
- Generic transmit and receive FIFO interface
- Full-duplex operation in 1G mode

- Full- and half-duplex operation in 10/100 mode
- Transmit and receive statistics vector
- Programmable Inter-Packet Gap (IPG)
- Multicast address filtering
- Selectable MAC operating options:
  - Classic Tri-Speed MAC with G/MII
  - Gigabit MAC with GMII
  - SGMII Easy Connect MAC with GMII, configurable option available on LatticeECP3, LatticeECP2/M, and LatticeSC/M devices
- Supports:
  - Full-duplex control using PAUSE frames
  - VLAN tagged frames
  - Automatic re-transmission on collision
  - Automatic padding of short frames
  - Multicast and broadcast frames
  - Optional FCS transmission and reception
  - Optional MII management interface module
  - Jumbo frames of any length

Broadcom BCM56800 Overview

BCM56800 Features

The BCM56800 network switch is a high-density, 10-Gigabit Ethernet switching chip solution with 20 ports. Each of these flexible ports supports 10-Gigabit Ethernet or 1-Gigabit Ethernet. Additionally, the BCM56800 integrates all the SERDES required to interface to applicable copper and fiber physical interfaces. The integrated SERDES functionality includes 10-Gbps XAUI interfaces and 1-Gbps SGMII PHY interfaces. The integrated SERDES complies with the CX-4 and PICMG3.1 standards, ensuring interoperability with Ethernet line cards in an Advanced TCA chassis.

BCM56800 10 GbE/HiGig™ Ports

The Broadcom BCM56800 has twenty 10 GbE/1 GbE ports. The BCM56800 is based on the StrataXGS® field-proven, robust architecture. It has integrated high-performance SERDES: integrated XAUI SERDES for all twenty 10 GbE ports, and it uses a single SERDES lane per port at GbE speeds. The device supports 200 Gbps switching capacity at line rate.

Test Equipment

Below is the equipment used in the interoperability test.

LatticeECP3 Serial Protocol Board (Revision E)

Figure 2 shows the LatticeECP3 Tri-Speed Ethernet MAC + SGMII/GbE PCS IP Reference Design and other components on the LatticeECP3 Serial Protocol Board. All board components are described in detail in EB44, LatticeECP3 Serial Protocol Evaluation Board - Revision D User's Guide. 1000BASE-X Ethernet data is received at the LatticeECP3 PCS (PCSC) channel 2 SERDES inputs via SMA connectors J21 and J22. The data is transferred to the SGMII and Gb Ethernet PCS IP core and the Tri-Speed Ethernet MAC IP core. Test logic at the Tri-Speed Ethernet MAC IP core client interface then loops data back in the other direction back to SERDES ouptut SMA connectors J25 and J26. The Tri-Speed Ethernet MAC IP core keeps statistical information about Ethernet frames received and re-transmitted. This information can be accessed via ORCAstra.

Table 1 provides a description of the LatticeECP3 Tri-Speed Ethernet MAC + SGMII/GbE PCS IP Reference Design signals accessible on the LatticeECP3 Serial Protocol Board.

Figure 2. LatticeECP3 Serial Protocol Board (Version E)

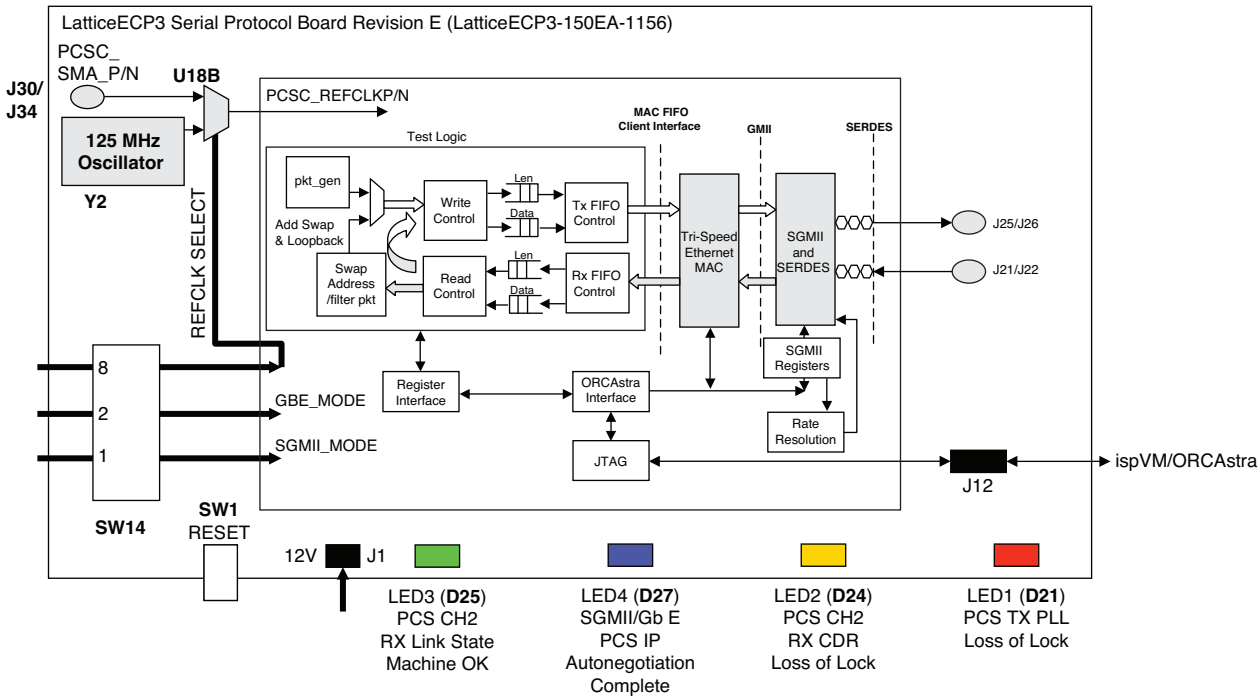


Table 1. LatticeECP3 Reference Design Signals

LatticeECP3 Signal Name	Signal Type	LatticeECP3 Serial Protocol Board Revision C or Newer Connection	Description
General Signals			
reset_n	I	SW1 push-button	FPGA global active low reset
Reference Design Signals			
GBE_MODE	I	Output of SW14, switch 2	SGMII and Gb Ethernet PCS IP core GBE_MODE input. This signal is controlled by SW14, switch 2. Press switch 2 down (0V) to set GBE_MODE low. The SGMII and Gb Ethernet PCS IP core is running in SGMII mode. Pull switch 2 up (3.3V) To set GBE_MODE high. The SGMII and Gb Ethernet PCS IP core is running in 1000BASE-X mode.
SGMII_MODE	I	Output of SW14, switch 1	SGMII and Gb Ethernet PCS IP core SGMII_MODE input. This input is valid when GBE_MODE=0 (SGMII mode). This signal is controlled by SW14, switch 1. Press switch 1 down (0V) to set SGMII_MODE low. The SGMII and Gb Ethernet PCS IP core is in MAC mode. Pull switch 1 up (3.3V) to set SGMII_MODE high. The SGMII and Gb Ethernet PCS IP core is in PHY mode.

Table 1. LatticeECP3 Reference Design Signals (Continued)

LatticeECP3 Signal Name	Signal Type	LatticeECP3 Serial Protocol Board Revision C or Newer Connection	Description
PCS TX PLL LOSS OF LOCK	O	LED1 (D21)	Red LED. Lattice ECP3 TX PLL loss of lock indication. The LED will not glow when a valid 125 MHz reference clock is provided to the LatticeECP3 PCS.
PCS CH2 RX CDR LOSS OF LOCK	O	LED2 (D24)	Yellow LED. Lattice ECP3 channel 2 RX CDR loss of lock indication. The LED will not glow when valid 1.25 Gbps data is provided to the LatticeECP3 Channel 2 SERDES inputs.
SGMII/GBE PCS IP AUTONEG COMPLETE	O	LED4 (D27)	Blue LED. This LED will glow upon successful SGMII auto-negotiation. Traffic will not flow through the system unless ANEG completes.
PCS CH2 RX LINK STATE MACHINE OK	O	LED3 (D25)	Green LED. Indicates that the LatticeECP3 channel RX GbE link state machine successfully synchronized to incoming Ethernet traffic. The LED will glow when valid 1.25 Gbps Ethernet data is provided to the LatticeECP3 Channel 2 SERDES inputs.
JTAG Signals			
tck	I	To J12 JTAG header	Connect the ispVM USB download cable to this header.
tdi	I		
tdo	O		
Tms	I		
PCS QUAD			
PCSC_REFCLKP/N	I	Output of U18B on-board clock MUX	LatticeECP3 PCS Quad C reference clock. The source of this clock is controlled by SW14, switch 8. Press switch 8 down (0V) to select the Y2 125 MHz on-board oscillator. Pull switch 8 up (3.3V) to select the J30/J34 SMA clock inputs.

Broadcom BCM56800 Network Switch

Figure 3 shows the BCM56800 network switch.

Figure 3. Broadcom BCM56800 Network Switch

The Broadcom ports can be configured by connecting its serial port to a PC and starting a HyperTerminal session. Figure 3 shows a serial cable connected to the serial port at the back of the BCM56800.

Figure 3 also shows a CX-4 connector inserted into one 10 GbE/HiGig port available on the front right side of the BCM56800. This port, referred to as xe0/hg0, was selected for the interoperation with the LatticeECP3 device. It was configured in XAUI mode.

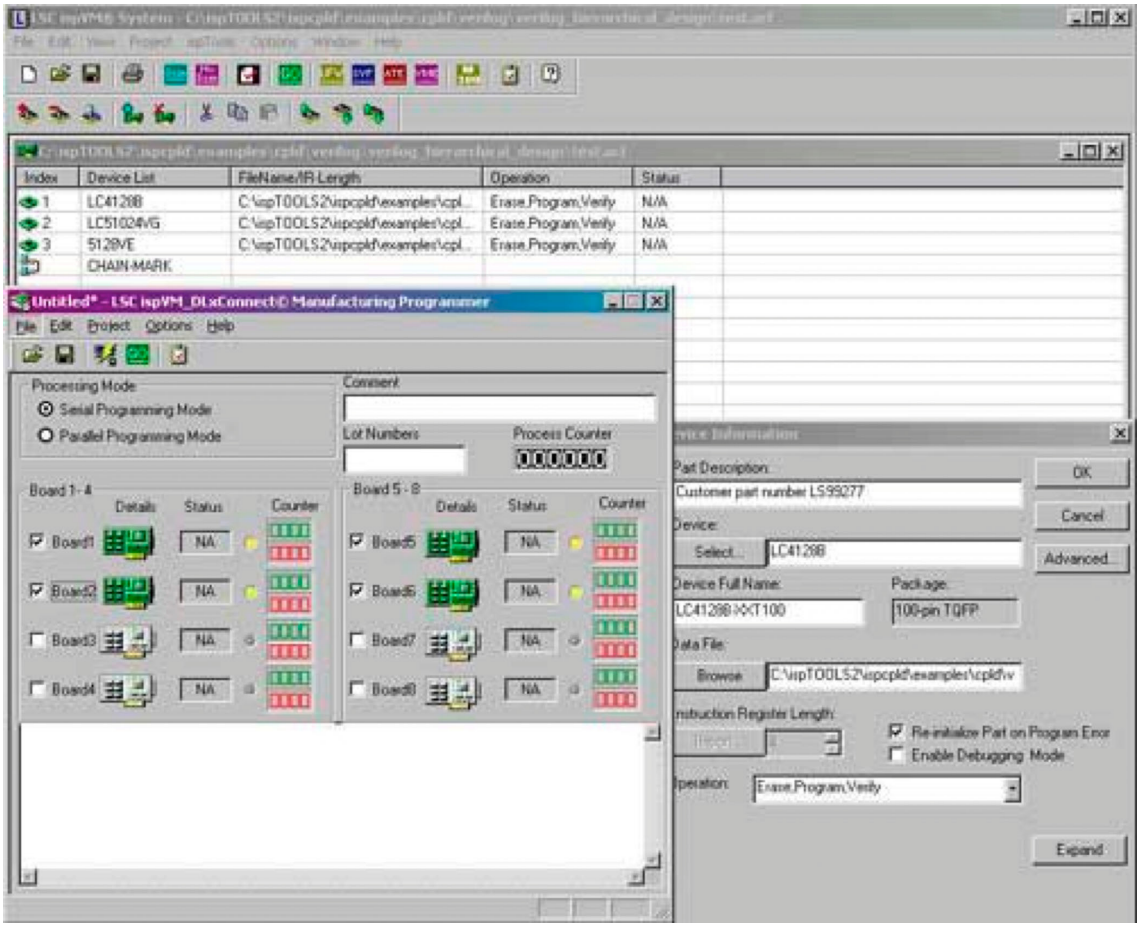
ispVM System

ispVM System software is included with the ispLEVER design tool suite, and is also available as a stand-alone device programming manager. The ispVM System is a comprehensive design download package that provides an efficient method of programming in-system programmable devices using JEDEC and bitstream files generated by Lattice, and other, design tools. This complete device programming tool allows the user to quickly and easily download designs through an ispSTREAM to devices and includes features that facilitate ispATE™, ispTEST and ispSVF programming as well as gang-programming with DLxConnect.

The ispVM System is used in this interoperability test to download the LatticeECP3 bitstream, which configures the flexiPCS™ in Gigabit Ethernet mode.

Figure 4 shows a screen shot of the ispVM system.

Figure 4. ispVM System



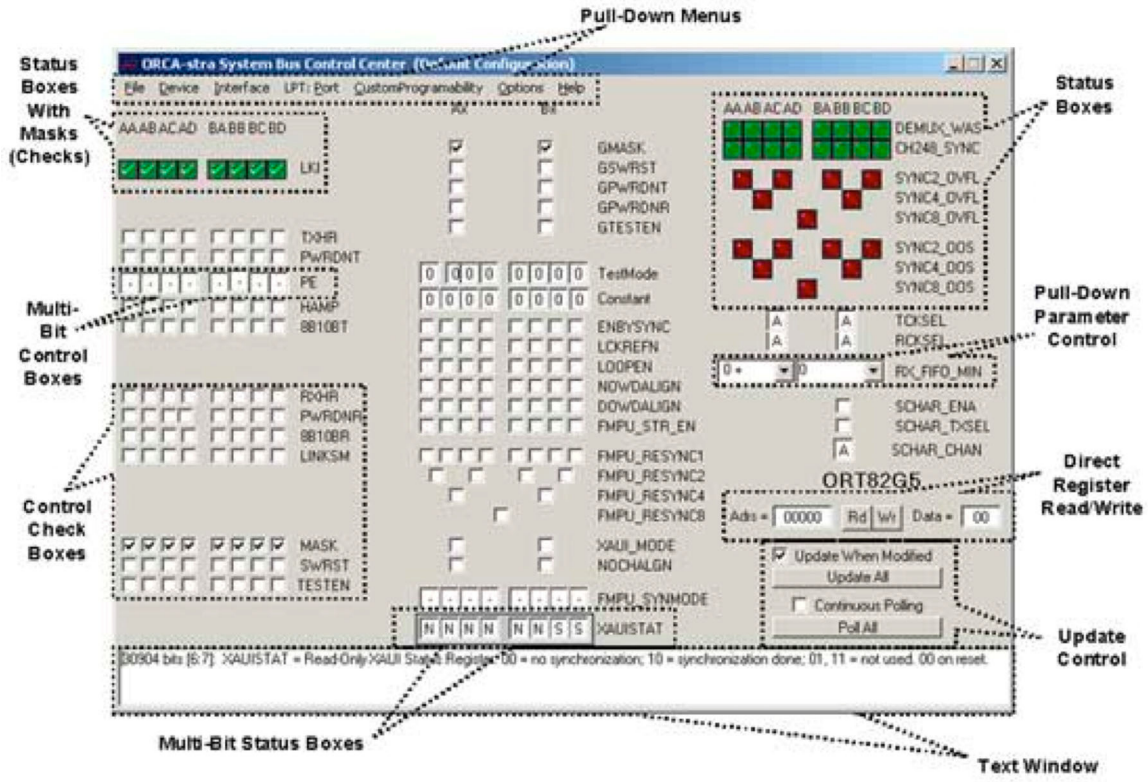
ORCAstra System

The ORCAstra software is a PC-based graphical user interface that allows the user to configure the operational mode of an FPGA by programming control bits in the on-chip registers. This helps users quickly explore configuration options without going through a lengthy re-compile process or making changes to their board.

Configurations created in the GUI can be saved to memory and re-loaded for later use. A macro capability is also available to support script-based configuration and testing. The GUI can also be used to display system status information in real time. Use of the ORCAstra software does not interfere with the programming of the FPGA portion of the LatticeECP3.

Figure 5 is a screen shot of the ORCAstra software.

Figure 5. ORCAstra System



Interoperability Testing

This section provides details the 1 GbE (1000BASE-X) Physical/MAC layer interoperability between a LatticeECP3 device and the Broadcom BCM56800 network switch. This interoperability tests the correct processing of Gigabit Ethernet data from the 88E1111 PHY to the Tri-Speed Ethernet MAC IP core, and then back in the other direction. Particularly, the test verifies the ability to transfer packets across the system in an asynchronous manner.

The test has the following characteristics:

- Independent (asynchronous) +/- 100 ppm clock sources clock the LatticeECP3 and BCM56800 devices
- The Broadcom switch transmits continuous Ethernet frames to the LatticeECP3 device
- The LatticeECP3 device loops the data at its MAC client interface back to the Broadcom switch

By the end of the test:

- The BCM56800 device visual window RX ERR counter should remain at zero
- The BCM56800 device visual window counters should report as many TX packets generated as RX packets received
- The amount of test time should be longer than 30 minutes to ensure the error rate is less than 10-12 with 99.999999% accuracy

Hardware Setup

Figure 6 shows the Broadcom BCM56800 board, the LatticeECP3 Serial Protocol Board, and the CX-4 to SMA conversion board connections.

Figure 7 is a block diagram of the test setup.

LatticeECP3 Serial Protocol Board Setup

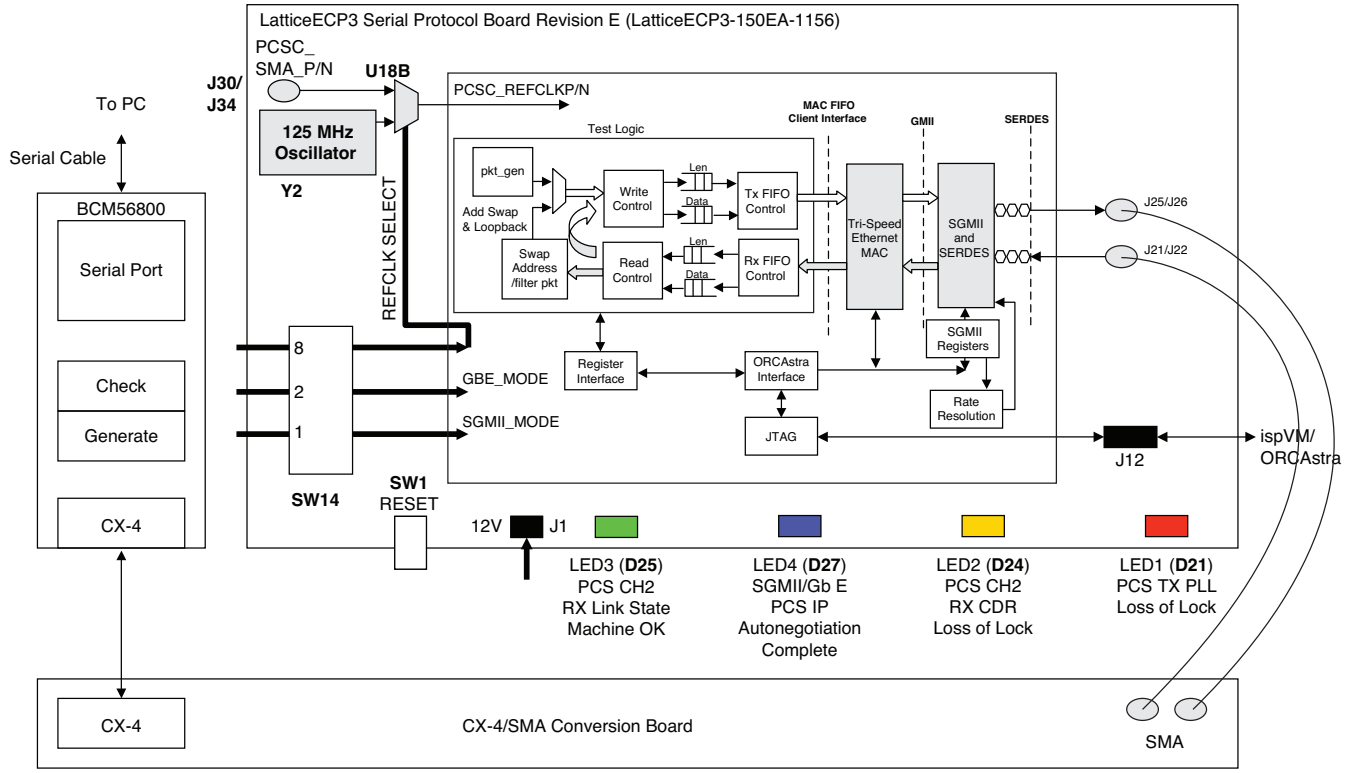
- Ensure the ispVM JTAG cable is connected to the J12 header
- A CX-4 to SMA conversion board was used as a physical medium interface to create a physical link between both boards. The SMA side of the CX-4 to SMA conversion board has four differential TX/RX channels, or 16 SMA connectors for a total bandwidth of 10 Gbps (12.5 Gbps aggregated rate). A total of four SMA cables (two for TX and two for RX channels are connected to the LatticeECP3 SMAs J21/J22 and J25/J26 on one end, and to channel 0 SMAs on the CX-4 to SMA conversion board (as shown in Figure 6).
- SW14, switch 2 should be in the upper position for GbE 1000BASE-X mode, and pressed down for SGMII mode. For this application, the switch is in the upper position.
- SW14, switch 1 should be in the upper position for SGMII PHY mode, and pressed down for SGMII MAC MODE (ONLY VALID IN SGMII MODE). For this application, the switch position is irrelevant.
- SW14, switch 8 must be pressed down (selects internal Y2 125 MHz oscillator as reference clock for the LatticeECP3 Tri-Speed Ethernet MAC + SGMII/GbE PCS IP Reference Design).

Broadcom BCM56800 Setup

- A CX-4 cable connected to BCM56800 HiGig port xe0/hg0 on one end and to the CX-4 to SMA conversion board on the other end.
- Serial cable for BCM56800 HyperTerminal access.

Figure 6. Board Connections

Figure 7. Test Setup Block Diagram



Test Description

This section describes how each interoperability partner is set up for 1 Gigabit Ethernet Physical/MAC Layer interoperability.

Broadcom BCM56800 Setup

The BCM56800 switch generates and checks full protocol-compliant Ethernet packets. The BCM56800 is configured in 1 Gigabit Ethernet.

Figure 8 illustrates the sequence of commands performed in a HyperTerminal from startup to configure HiGig port 0 (xe0) of BCM56800 in 1 Gigabit Ethernet, while disabling auto-negotiation. To prevent port xe0 statistics counters from overflowing, a few lines are added to create Ethernet traffic on port xe1, and then redirect it to xe0.

A shown in Figure 8, xe1 generates continuous Ethernet packets defined by the content of the bc_DA_1022 file and redirects the content to port xe0. Port xe0 then continuously outputs this data on channel 0 of its HiGig port. bc_DA_1022 is a VLAN tagged packet with 1022 bytes of data. The destination address in bc_DA_1022 is set to 00.00.00.00.00.02.

Figure 8. BCM56800 1 Gigabit Ethernet Configuration Commands

```
'SETUP BC BOARD
'TRANSMIT PACKETS FOR 3600 seconds (1hr)
rc
'ENABLE VLAN PACKETS ON ALL PORTS
vlan remove 1 pbm=0x00000000001fffff
vlan add 1 pbm=0x00000000001fffff ubm=0x000000
vlan show
'SETUP XE0 AND XE1 IN 1 GBE MODE
'REDIRECT XE1 TRAFFIC TO XE0
fp qset clear
fp qset add InPorts
fp group create 1 1
fp entry create 1 1
fp qual 1 InPorts 0x1 0xffffff
fp action add 1 RedirectPbmp 0x2
fp entry install 1
fp entry create 1 2
fp qual 2 InPorts 0x2 0xffffff
fp action add 2 RedirectPbmp 0x1
fp entry install 2
port xe0 speed=1000 AN=OFF
port xe1 speed=1000 AN=OFF
port xe1 lb=phy
ps
clear counters
'RUN TRAFFIC FOR 1 HR, THEN SHOW COUNTERS
tx 4 pbm=xe1 file=bc_DA_1022
sleep 3600
port xe1 lb=none
sleep 5
show counters
```

LatticeECP3 PCS Auto Configuration (.txt) file

Appendix B lists the auto-configuration file settings for the LatticeECP3 PCS used in the LatticeECP3 Tri-Speed Ethernet MAC + SGMII/GbE PCS IP Reference Design.

ORCAstra Setup

After the LatticeECP3 Serial Protocol Board is powered up and the LatticeECP3 bitstream is downloaded, the following steps explain the procedure for configuring the test logic and Tri-Speed Ethernet MAC/SGMII and Gb Ethernet PCS IP registers via ORCAstra.

1. Ensure that a valid Ethernet link exists at the J37 RJ-45 connector (SmartBits box up and running).
2. Start ORCAstra from the ispLEVER installation directory.
3. Select **Interface -> 3. ispVM JTAG USB Interface**. If the **Select Target JTAG Device** window appears, select the first device, and click **OK**.
4. From the ORCAstra main window, click **ORCAstra -> CustomProgrammability -> Scripts -> VBScripts**.
5. From the new window, select **File -> Open -> < PROJECT_PATH>\init_mac_1000BASE-X_Broadcom.vbs**. This file is shown in Appendix C. This file configures LatticeECP3 Tri-Speed Ethernet MAC/SGMII and Gb Ethernet PCS AUTONEG registers to run at 1000 Mbps Full duplex and advertises full pause capabilities. It also configures the Media Access Controller in UNICAST mode and the local address is set to 00 00 00 00 00 02. This is the same value as the destination address in the frames that the BCM56800 is transmitting. Click **Run**. The scripts will perform loops as it constantly checks whether autonegotiation completes or restarts and for link partner advertised PAUSE, ASYM pause capabilities to reconfigure its own MAC PAUSE capabilities. To exit the loop, hit the **Shift** key while the mouse is inside the VBScript window.

After running this script as well as the Broadcom script, the Broadcom box and LatticeECP3 LEDs will indicate a proper link.

Upon the termination of the test, the following steps can be used to record the LatticeECP3 MAC statistics counters:

1. From the ORCAstra main window, select **ORCAstra -> CustomProgrammability -> Scripts -> VBScripts**.
2. From the new window, select **File -> Open -> < PROJECT_PATH>\ reg_stats_1000BASE-X_Broadcom.vbs**. This file is shown in Appendix D. Click **Run**. This results in an output window similar to the one shown in Figure 9.

Note that the LatticeECP3 statistics counters are not wide enough to record enough frame captures. As a result, the counters will roll over and will not match the BCM56800 TX and RX packets counters. The purpose of reporting the events in Figure 9 is to show that no error events were recorded.

Figure 9. Output Example of Running reg_stats_1000BASE-X_Broadcom.vbs Script

```
AUTONEG Statistics Reading

SGMII AUTONEG STATUS (BIT 5=AN complete)  = 0x 20
AUTONEG LP Abiliy byte 1:
BIT7=Next Page,,  BIT6=ACK, BIT5:4=Remote Fault,
BIT0=PAUSE = 0xC0
AUTONEG LP Abiliy byte 0:
BIT7=ASYM_PAUSE, BIT6=HALF DUPLEX, BIT5=FULL
DUPLEX = 0xA0
TEST LOGIC

TX/RX status

RX FIFO STATUS (BIT2=RX_FIFO_FULL
BIT1=RX_FIFO_ERR BIT0=RX_ERRROR)  = 0x 0
TX FIFO STATUS ( BIT1=TX_FIFO_FULL
BIT0=TX_DISC_FRM)  = 0x 0
```

```
TX/RX Statistics Counters

RX Packet Ignored  = 0
RX Length Check Error  = 0
RX Long Frame  = 0
RX Short Frame  = 0
RX IPG Violation  = (not functional at time of interop)
RX CRC Error  = 0
RX OK PACKET  = 29763
RX Control Frame  = 0
RX PAUSE Frame  = 0
RX Multicast Frame  = 0
RX Broadcast Frame  = 0
RX VLAN tagged Frame  = 29763
TX UNICAST FRAME  = 29763
TX PAUSE FRAME  = 0
TX MULTICAST FRAME  = 0
TX BROADCAST FRAME  = 0
TX VLAN FRAME  = 29763
TX CRC Error  = 0
TX JUMBO FRAME  = 0
```

Results

Figure 10 illustrates the section of the HyperTerminal output that resulted from running the last few commands in the BCM56800 1 Gigabit Ethernet Configuration sequence of Figure 8.

As shown in Figure 10, HiGig ports 0 (xe0) and 1 (xe1) of BCM56800 were configured for 1 Gigabit Ethernet.

The last “show counter” command reports the status of BCM56800 port xe0 TX and RX packet (GTPKT.xe0 and GRPKT.xe0) and byte (GTBYT.xe0 and GRBYT.xe0) counters. Figures 9 and 10 do not show any errors.

This is an indication that the test between the two ports ran error-free. Additionally, the BCM56800 TX and RX packet counters in Figure 10 show an identical number of frames recorded.

Figure 10. Output of BCM56800 Configuration Script

```
BCM.0> ps
      ena/  speed/  link  auto    STP               lrn  inter   max  loop
port  link  duplex  scan  neg?   state   pause  discrd  ops   face frame  back
 xe0  up     1G FD    SW   No    Forward  TX RX   None    FA  XGMII 16360
 xe1  up     1G FD    SW   No    Forward  TX RX   None    FA  XGMII 16360   PHY
 xe2  down    -       SW   Yes   Forward  TX RX   None    FA  XGMII 16360
 xe3  down    -       SW   Yes   Forward  TX RX   None    FA  XGMII 16360
 xe4  down    -       SW   Yes   Forward  TX RX   None    FA  XGMII 16360
 xe5  down    -       SW   Yes   Forward  TX RX   None    FA  XGMII 16360
 xe6  down    -       SW   Yes   Forward  TX RX   None    FA  XGMII 16360
 xe7  down    -       SW   Yes   Forward  TX RX   None    FA  XGMII 16360
 xe8  down    -       SW   Yes   Forward  TX RX   None    FA  XGMII 16360
 xe9  down    -       SW   Yes   Forward  TX RX   None    FA  XGMII 16360
xe10  down    -       SW   Yes   Forward  TX RX   None    FA  XGMII 16360
xe11  down    -       SW   Yes   Forward  TX RX   None    FA  XGMII 16360
xe12  down    -       SW   Yes   Forward  TX RX   None    FA  XGMII 16360
xe13  down    -       SW   Yes   Forward  TX RX   None    FA  XGMII 16360
xe14  down    -       SW   Yes   Forward  TX RX   None    FA  XGMII 16360
xe15  down    -       SW   Yes   Forward  TX RX   None    FA  XGMII 16360
xe16  down    -       SW   Yes   Forward  TX RX   None    FA  XGMII 16360
xe17  down    -       SW   Yes   Forward  TX RX   None    FA  XGMII 16360
xe18  down    -       SW   Yes   Forward  TX RX   None    FA  XGMII 16360
xe19  down    -       SW   Yes   Forward  TX RX   None    FA  XGMII 16360
BCM.0> clear counters
BCM.0> tx 4 pbm=xe1 file=bc_DA_1022
BCM.0> sleep 3600
Sleeping for 3600 seconds
BCM.0> port xe1 lb=none
BCM.0> sleep 5
Sleeping for 5 seconds
BCM.0> show counters
RUC.xe0         :          422,933,571        +422,933,571
GR1518.xe0      :          422,933,571        +422,933,571
GRPKT.xe0       :          422,933,571        +422,933,571
GRBYT.xe0       :      441,542,648,124    +441,542,648,124
GT1518.xe0      :          422,933,571        +422,933,571
GTPKT.xe0       :          422,933,571        +422,933,571
GTBYT.xe0       :      441,542,648,124    +441,542,648,124
```

Summary

In conclusion, the LatticeECP3 family offers a 1000BASE-X 1 Gigabit Ethernet Physical/MAC layer solution that is fully inter-operable with the Broadcom BCM56800 network switch.

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
July 2010	01.0	Initial release.

Appendix A. Lattice Tri-Speed Ethernet MAC/SGMII Reference Design Test Logic Map and Bit Descriptions

There are three address spaces in the Test Application design. The first address space (Table 2) starts at offset 0x00 and ends at 0x35. This space contains the SGMII and Tri-Speed Ethernet MAC IP core user registers as described in the Tri-Speed Ethernet MAC IP Core's User's Guide. The second address space (Table 3) starts at offset 0x40 and ends at 0x4D. This space contains SGMII/GbE PCS IP registers as described in the SGMII and Gb Ethernet PCS IP Core User's Guide. The third address space (Table 4) starts at offset 0x880 and ends at 0x8C1. This space contains ID, control, status and statistics registers used by the Test Logic Application.

Table 2. Tri-Speed Ethernet MAC IP Core Internal Registers

Register Description	Mnemonic	I/O Address	POR Value
Mode register	MODE	00H - 01H	0000H
Transmit and Receive Control register	TX_RX_CTL	02H - 03H	0000H
Maximum Packet Size register	MAX_PKT_SIZE	04H - 05H	05EEH
Inter Packet Gap register	IPG_VAL	08H - 09H	0048H
Tri-Speed MAC Address register 0	MAC_ADDR_0	0AH - 0BH	0000H
Tri-Speed MAC Address register 1	MAC_ADDR_1	0CH - 0DH	0000H
Tri-Speed MAC Address register 2	MAC_ADDR_2	0EH - 0FH	0000H
Transmit and Receive Status	TX_RX_STS	12H - 13H	0000H
GMII Management Interface Control register	GMII_MNG_CTL	14H - 15H	0000H
GMII Management Data register	GMII_MNG_DAT	16H - 17H	0000H
VLAN Tag Length/type register	VLAN_TAG	32H - 33H	0000H
Multicast_table_0	MLT_TAB_0	22H - 23H	0000H
Multicast_table_1	MLT_TAB_1	24H - 25H	0000H
Multicast_table_2	MLT_TAB_2	26H - 27H	0000H
Multicast_table_3	MLT_TAB_3	28H - 29H	0000H
Multicast_table_4	MLT_TAB_4	2AH - 2BH	0000H
Multicast_table_5	MLT_TAB_5	2CH - 2DH	0000H
Multicast_table_6	MLT_TAB_6	2EH - 2FH	0000H
Multicast_table_7	MLT_TAB_7	30H - 31H	0000H
Pause_opcode	PAUS_OP	34H - 35H	0080H

Table 3. SGMII/GbE PCS IP Core Registers

Register Description	Mnemonic	I/O Address	POR Value
Control Register LO [7:0]	Control Register	40H	00H
Control Register HI [15:8]	Control Register	41H	10H
Status Register LO [7:0]	Status Register	42H	—
Status Register HI [15:8]	Status Register	43H	—
Advertised Ability LO[7:0]	mr_adv_ability	48H	01H ** read-only in
Advertised Ability HI[15:8]	mr_adv_ability	49H	40H MAC Mode
Link Partner Ability LO[7:0]	mr_lp_adv_ability	4AH	—
Link Partner Ability HI[15:8]	mr_lp_adv_ability	4BH	—
Auto Neg Expansion LO [7:0]	Auto Neg Expansion	4CH	—
Auto Neg Expansion HI [15:8]	Auto Neg Expansion	4DH	—

The REGINTF logic block in the Test Application provides the address decoding, for the RO and R/W Registers used by the test logic and statistics counters. All registers are 8 bits wide and are byte addressable. Note that the statistics counter registers are composed of two 8-bit registers, a low and a high byte register. Therefore, in order to access these registers, two byte accesses must be made. For example, to access to all 16 bits of the RXOKCNT requires an access to both 0x899 (high byte) and 0x898 (low byte). Note that since the statistics counter registers are Clear On Read (COR), the high byte should be read first before reading the low byte, since a read of the low byte clears all the combined 16 bits of the low and high registers. The address map for the Test Application Related Resisters is as follows:

Table 4. Test Logic Application Related Registers

Address	Register Description	Mnemonic	Type
0x0880	VERsion/IDentification Register	VERID	RO
0x0881	TeST CoNTroL Register	TSTCNTL	RW
0x0882	TeST CoNTroL Register 2	TSTCNTL_2	RW
0x0883	MAC CoNTroL Register	MACCNTL	RW
0x0884	PAUSe TiMeR Register - Low byte	PAUSTMRL	RW
0x0885	PAUSe TiMeR Register - High byte	PAUSTMRH	RW
0x0886	FIFO Almost Full Threshold Register - Low	FIFOAFTL	RW
0x0887	FIFO Almost Full Threshold Register - High	FIFOAFTH	RW
0x0888	FIFO Almost Empty Threshold Register - Low	FIFOAETL	RW
0x0889	FIFO Almost Empty Threshold Register - High	FIFOAETH	RW
0x088a	RX Status Register	RXSTATUS	RO/COR
0x088b	TX Status Register	TXSTATUS	RO/COR
0x088c, 0x088d	RX Packet Ignored Counter Register (L,H)	RXPICNT	RO/COR
0x088e, 0x088f	RX Length Check Error CouNTer (L,H)	RXLCECNT	RO/COR
0x0890, 0x0891	RX Long Frames CouNTer Register (L,H)	RXLFCNT	RO/COR
0x0892, 0x0893	RX Short Frames CouNTer Register (L,H)	RXSFCNT	RO/COR
0x0894, 0x0895	RX IPG violations CouNTer Register (L,H)	RXIPGCNT	RO/COR
0x0896, 0x0897	RX CRC errors CouNTer Register (L,H)	RXCRCCNT	RO/COR
0x0898, 0x0899	RX OK packets CouNTer Register (L,H)	RXOKCNT	RO/COR
0x089a, 0x089b	RX Control Frame CouNTer Register (L,H)	RXCFCNT	RO/COR
0x089c, 0x089d	RX Pause Frame CouNTer Register (L,H)	RXPFCNT	RO/COR
0x089e, 0x089f	RX Multicast Frame CouNTer Register (L,H)	RXMFCNT	RO/COR
0x08a0, 0x08a1	RX Broadcast Frame CouNTer Register (L,H)	RXBFCNT	RO/COR
0x08a2, 0x08a3	RX VLAN tagged Frame CouNTer Register (L,H)	RXVFCNT	RO/COR
0x08a4, 0x08a5	TX Unicast Frame CouNTer Register (L,H)	TXUFCNT	RO/COR
0x08a6, 0x08a7	TX Pause Frame CouNTer Register (L,H)	TXPFCNT	RO/COR
0x08a8, 0x08a9	TX Multicast Frame CouNTer Register (L,H)	TXMFCNT	RO/COR
0x08aa, 0x08ab	TX Broadcast Frame CouNTer Register (L,H)	TXBFCNT	RO/COR
0x08ac, 0x08ad	TX VLAN tagged Frame CouNTer Register (L,H)	TXVFCNT	RO/COR
0x08ae, 0x08af	TX BAD FCS Frame CouNTer Register (L,H)	TXBFCCNT	RO/COR
0x08b0, 0x08b1	TX Jumbo Frame CouNTer Register (L,H)	TXJFCNT	RO/COR
0x08b2, 0x08b3	UNUSED		
0x08b4, 0x08b5	TX Packet Generator DEST ADD (B1,B2)	PG_DA_W1	RW
0x08b6, 0x08b7	TX Packet Generator DEST ADD (B3,B4)	PG_DA_W2	RW
0x08b8, 0x08b9	TX Packet Generator DEST ADD (B5,B6)	PG_DA_W3	RW

Table 4. Test Logic Application Related Registers (Continued)

Address	Register Description	Mnemonic	Type
0x08ba, 0x08bb	TX Packet Generator SRC ADD (B1,B2)	PG_SA_W1	RW
0x08bc, 0x08bd	TX Packet Generator SRC ADD (B3,B4)	PG_SA_W2	RW
0x08be, 0x08bf	TX Packet Generator SRC ADD (B5,B6)	PG_SA_W3	RW
0x08c0, 0x08c1	TX Packet Generator PYLD LEN (L,H)	PG_PL	RW
0x08c2 - 0x0FFF	UNUSED		

Register Bit Descriptions

0x0880 **VERsion/IDentification Register** VERID Read Only
default value = 0xA3

0x0881 **TeST CoNTroL Register** TSTCNTL Read/Write
default value = 0x00

bit_0 =Destination/source address swap – 1 = Swap, 0 = No swap

bit_1 =Loopback enable – 1 = Loopback, 0 = No loopback

bit_2 =reset_phy_n – 1 = No reset, 0 = Reset PHY device

bit_3 =pkt_loop_clksel – 1 = sys_clk tied to rx_clk, 0 = sys_clk from external pin

bit_4 =flag_large_pkt_en – 1 = Enables Rx packets larger than those set in Max_PKT_SIZE reg to be flagged

bit_5 =flag_errored_pkt_en – 1 = Enables Rx errored packets (rx_error set) to be flagged

bit_6 = flag_pause_pkt_en – 1 = Enables pause packets be flagged

bit_7 =drop_flagged_en – 1 = Enables the discarding of flagged packets

0x0882 **TeST CoNTroL Register 2** TSTCNTL 2 Read/Write
default value = 0x00

bit_0 =pkt_gen_en – 1 = Enable Tx packet generator

bit_1 =pkt_mode[0] – Mode 00 = Generate single packet, 01 = Continous packets, 10 = A burst of packets

bit_2 =pkt_mode[1] – Mode 00 = Generate single packet, 01 = Continous packets, 10 = A burst of packets

bit_3 –pkt_gen_burst_size [0] = 0000 – 1 packet, 0001 – 2 packets , …, 1111 – 15 packets

bit_4 –pkt_gen_burst_size [1] = 0000 – 1 packet, 0001 – 2 packets , …, 1111 – 15 packets

bit_5 –pkt_gen_burst_size [2] = 0000 – 1 packet, 0001 – 2 packets , …, 1111 – 15 packets

bit_6 –pkt_gen_burst_size [3] = 0000 – 1 packet, 0001 – 2 packets , …, 1111 – 15 packets

bit_7 –Not used

0x0883 **MAC CoNTroL Register** MACCNTL Read/Write
default value = 0x00

bit_0 =Send pause request – 1 = Send request, 0 = Don't send request. This register bit is ORed with the tx_fifo almost full signal. The ORed output sets the tx_sndpausreq pin on the Tri-Speed Ethernet MAC IP core. See the Tri-Speed Ethernet MAC IP Core User's Guide for more information.

bit_1 =FIFO control frame – This register bit sets the tx_fifoctrl pin on the Tri-Speed Ethernet MAC IP Core. See the Tri-Speed Ethernet MAC IP Core User's Guide for more information.

bit_2 =Not used.

bit_3 =tx_fifo empty – This register bit is ORed with the tx_fifo empty signal. The ORed output sets the tx_fifo_empty pin on the Tri-Speed Ethernet MAC IP core. See the device data sheet for more information. Note that by setting this bit you can mimic the Tx FIFO being empty.

bit_4 =Ignore next packet – This register bit sets the ignore_next_pkt pin on the Tri-Speed Ethernet MAC IP core. See the Tri-Speed Ethernet MAC IP Core User's Guide for more information.

bit_5 =Rx_fifo_flush –1 = Flush Rx FIFO

bit_6 =Tx_fifo_flush. 1 = Flush Tx FIFO

bit_7 = Not used

0x0884 **PAUSe TiMeR Register - Low byte** PAUSTMRL Read/Write
0x0885 **PAUSe TiMeR Register - High byte** PAUSTMRH Read/Write
default value = 0x0000

Low Reg
bit[7:0] = Pause timer low bits. These register bits set the tx_sndpaustim[7:0] pins on the Tri-Speed Ethernet MAC IP core. See the Tri-Speed Ethernet MAC IP Core User's Guide for more information.

High Reg
bit[7:0] = Pause timer high bits. These reg bits set the tx_sndpaustim[15:8] pins on the Tri-Speed Ethernet MAC IP core. See the Tri-Speed Ethernet MAC IP Core User's Guide for more information.

0x0886 **FIFO Almost Full Threshold Register - Low** FIFOAFTL Read/Write
0x0887 **FIFO Almost Full Threshold Register - High** FIFOAFTH Read/Write
default value = 0x0000

Low Reg
bit[7:0] = FIFO Almost Full Threshold - Low
These register bits set the loopback FIFO threshold [7:0] bits

High Reg
bit[0] = FIFO Almost Full Threshold - High
This register bit set the loopback FIFO [8] bit

bit[7:1] = Not used

0x0888 **FIFO Almost Empty Threshold Register - Low** FIFOAETL Read/Write
0x0889 **FIFO Almost Empty Threshold Register - High** FIFOAETH Read/Write
default value = 0x0000

Low Reg
bit[7:0] = FIFO Almost Empty Threshold - Low
These register bits set the loopback FIFO threshold [7:0] bits

High Reg
bit[0] = FIFO Almost Empty Threshold - High
This register bit sets the loopback FIFO [8] bit
bit[7:1] = Not used

0x088a **RX Status Register** RXSTATUS Read Only/clear on read
default value = 0x00
0x088b **TX Status Register** TXSTATUS Read Only/clear on read
default value = 0x00

RX status
bit_0 = Latches rx_error value from the Tri-Speed Ethernet MAC IP core pin
bit_1 = Latches rx_fifo_error value from the Tri-Speed Ethernet MAC IP core pin
bit_2 = Latches rx_fifo_full value from test logic loopback FIFO
bit_3 =Not used
bit_4 =Not used
bit_5 =Not used
bit_6 =Not used
bit_7 =Not used

TX status
bit_0 = Latches tx_discfrm value from the Tri-Speed Ethernet MAC IP core pin
bit_1 = Latches tx_fifo_full value from test logic loopback FIFO
bit_2 =Not used
bit_3 =Not used
bit_4 =Not used
bit_5 =Not used
bit_6 =Not used
bit_7 =Not used

The following counter registers are all 16 bits with 8-bit low and 8-bit high address locations. The counters count different Rx and Tx statistics as defined by the statistics vectors in the Tri-Speed Ethernet MAC IP Core User's Guide. All counters have a power-on default value of 0x0000.

0x088c **RX Packet Ignored Counter Register**RXPICNTRO/COR

0x088e **RX Length Check Error CouNTer** RXLCECNTRO/COR

0x0890 **RX Long Frames CouNTer Register**RXLFCNTRO/COR

0x0892 **RX Short Frames CouNTer Register**RXSFCNTRO/COR

0x0894 **RX IPG violations CouNTer Register**RXIPGCNTRO/COR

0x0896 **RX CRC errors CouNTer Register**RXCRCCNTRO/COR

0x0898 **RX OK packets CouNTer Register**RXOKCNT RO/COR

0x089a **RX Control Frame CouNTer Register**RXCFCNTRO/COR

0x089c **RX Pause Frame CouNTer Register**RXPFCNTRO/COR

0x089e **RX Multicast Frame CouNTer Register**RXMFCNTRO/COR

0x08a0 **RX Broadcast Frame CouNTer Register**RXBFCNTRO/COR

0x08a2 **RX VLAN tagged Frame CouNTer Register**RXVFCNTRO/COR

0x08a4 **TX Unicast Frame CouNTer Register**TXUFCNTRO/COR

0x08a6 **TX Pause Frame CouNTer Register**TXPFCNTRO/COR

0x08a8 **TX Multicast Frame CouNTer Register**TXMFCNTRO/COR

0x08aa **TX Broadcast Frame CouNTer Register**TXBFCNTRO/COR

0x08ac **TX VLAN tagged Frame CouNTer Register**TXVFCNTRO/COR

0x08ae **TX BAD FCS Frame CouNTer Register**TXBFCCNTRO/COR

0x08b0 **TX Jumbo Frame CouNTer Register**TXJFCNTRO/COR

Appendix B. LatticeECP3 PCS Auto-Configuration File

```
# This file is used by the simulation model as well as the ispLEVER bitstream
# generation process to automatically initialize the PCSD quad to the mode
# selected in the IPexpress. This file is expected to be modified by the
# end user to adjust the PCSD quad to the final design requirements.

DEVICE_NAME "LFE3-95E"
CH2_PROTOCOL            "GIGE"
CH0_MODE                "DISABLED"
CH1_MODE                "DISABLED"
CH2_MODE                "RXTX"
CH3_MODE                "DISABLED"
CH2_CDR_SRC       "REFCLK_EXT"
PLL_SRC                 "REFCLK_EXT"
TX_DATARATE_RANGE       "MED"
CH2_RX_DATARATE_RANGE   "MED"
REFCK_MULT              "10X"
#REFCLK_RATE            125.0
CH2_RX_DATA_RATE        "FULL"
CH2_TX_DATA_RATE        "FULL"
CH2_TX_DATA_WIDTH       "8"
CH2_RX_DATA_WIDTH        "8"
CH2_TX_FIFO       "ENABLED"
CH2_RX_FIFO        "ENABLED"
CH2_TDRV      "0"
#CH2_TX_FICLK_RATE      125.0
#CH2_RXREFCLK_RATE        "125.0"
#CH2_RX_FICLK_RATE      125.0
CH2_TX_PRE              "DISABLED"
CH2_RTERM_TX            "50"
CH2_RX_EQ               "DISABLED"
CH2_RTERM_RX            "50"
CH2_RX_DCC              "AC"
CH2_LOS_THRESHOLD       "2"
PLL_TERM                "50"
PLL_DCC                 "AC"
PLL_LOL_SET             "0"
CH2_TX_SB               "DISABLED"
CH2_RX_SB               "DISABLED"
CH2_TX_8B10B            "ENABLED"
CH2_RX_8B10B            "ENABLED"
CH2_COMMA_A             "1100000101"
CH2_COMMA_B             "0011111010"
CH2_COMMA_M             "1111111111"
CH2_RXWA                "ENABLED"
CH2_ILSM                "ENABLED"
CH2_CTC                 "DISABLED"
CH2_CC_MATCH4           "0001010000"
CH2_CC_MATCH_MODE       "1"
CH2_CC_MIN_IPG          "3"
CCHMARK                 "9"
CCLMARK                 "7"
CH2_SSLB                "DISABLED"
```

```
CH2_SPLBPORTS            "DISABLED"
CH2_PCSLBPORTS           "DISABLED"
INT_ALL                  "ENABLED"
QD_REFCK2CORE            "ENABLED"
```

Appendix C init_mac_1000BASE-X_Broadcom.vbs ORCAstra Visual Basic Script

```
Sub Main()

'NOTE; 1000 Mbps Half Duplex is not supported

        Call V.SPut(&H00800,&h00) ' to host bus - mode reg 00 DISABLE TX/RX

        Call V.SPut(&H00884,&h0F) ' TX PAUTE TIME LOW BYTE-0F

Call V.SPut(&H00886,&hC1) 'fifo AFL C1
Call V.SPut(&H00887,&h01) 'fifo AFH 01
Call V.SPut(&H00888,&h05) 'fifo AEL 05
Call  V.SPut(&H00881,&hFE)  'tstcntl  (b0=swap,b1=loop_en,b2=reset_phy_n,b3(0=sys_clk
from external pin),b4=flaglargepackets,b5=flarerrorpkt,b6=flagpause,b7-dropflagged)-
FE
       ' 1518 bytes = 0x5ee
'Call V.SPut(&H008b2,&h77) 'max_pkt_sz LEE
'Call V.SPut(&H008b3,&hA0) 'max_pkt_sz H05

' Pkt generator registers
        Call V.SPut(&H008b4,&h0A) '  pkt_gen_destadd_B1 0A
        Call V.SPut(&H008b5,&h0B) '  pkt_gen_destadd_B2 0B
        Call V.SPut(&H008b6,&h0C) '  pkt_gen_destadd_B3 0C
        Call V.SPut(&H008b7,&h0D) '  pkt_gen_destadd_B4 0D
        Call V.SPut(&H008b8,&h0E) '  pkt_gen_destadd_B5 0E
        Call V.SPut(&H008b9,&h0F) '  pkt_gen_destadd_B6 0F
        Call V.SPut(&H008ba,&h01) '  pkt_gen_srcadd_B1 01
        Call V.SPut(&H008bb,&h02) '  pkt_gen_srcadd_B2 02
        Call V.SPut(&H008bc,&h03) '  pkt_gen_srcadd_B3 03
        Call V.SPut(&H008bd,&h04) '  pkt_gen_srcadd_B4 04
        Call V.SPut(&H008be,&h05) '  pkt_gen_srcadd_B5 05
        Call V.SPut(&H008bf,&h06) '  pkt_gen_srcadd_B606

       ' 46 bytes = 0x2e
        Call V.SPut(&H008c0,&h2E) '  pkt_gen_pyld_len L 2e
        Call V.SPut(&H008c1,&h00) '  pkt_gen_pyld_len H

' MDIO registers
'       `ifdef MIIM_MODULE
'        Call V.SPut(&H00816,&h00) ' to  - MDIO DATA reg
'        Call V.SPut(&H00817,&h80) ' to  - MDIO DATA reg
'        Call V.SPut(&H00814,&h00) ' to  - MDIO ACCESS CTL reg
'        Call V.SPut(&H00815,&h21) ' to  - MDIO ACCESS CTL reg
'#20000
'        jtag_drv.JTAG_read_byte(&H00814, read_data) ' to  - MDIO ACCESS CTL reg
'        jtag_drv.JTAG_read_byte(&H00815, read_data) ' to  - MDIO ACCESS CTL reg
'       `endif


' MAC registers  00 00 00 00 00 02
        Call V.SPut(&H0080a,&h00)           ' MAC Addr reg 0 CD
        Call V.SPut(&H0080b,&h00)           ' MAC Addr reg 0 AA
```

```
        Call V.SPut(&H0080c,&h00)           ' MAC Addr reg 1 12
        Call V.SPut(&H0080d,&h00)           ' MAC Addr reg 1 EF
        Call V.SPut(&H0080e,&h02)           ' MAC Addr reg 2 56
        Call V.SPut(&H0080f,&h00)           ' MAC Addr reg 2 34

        'Don't Drop Control (will be dropped by test logic), UNICAST :9A
        Call V.SPut(&H00802,&h9A) ' to host bus - TX_RX_CTL H reg 9A
        Call V.SPut(&H00803,&h00) ' to host bus - TX_RX_CTL L -NO short pkts

'12 Bytes IPG
        Call V.SPut(&H00808,&h0C) ' to host bus - IPG reg 0C
'Gbit enable
        Call V.SPut(&H00800,&h0F) ' to host bus - mode reg 0F


'PROGRAM Lattice SGMII  AN register
'USE 1000BASE-X ADV FORMAT
'DEVICE ALSO DOES NOT ADVERTISE PAUSE
' mr_adv_ability[7:0], bit7=PAUSE=1,bit6=HALFDUP=0,bit5=FULLDUP=1,bit4:0=RES=00000
        Call V.SPut(&H00848,&hA0)           ' mr_adv_ability[7:0] A0
'           mr_adv_ability[15:8],           bit15=NP=0,bit14=RES=0,bit13:12=REmore-
Fault=00,bit11:9=RES=000,bit8=ASYM_PAUSE=1
        Call V.SPut(&H00849,&h01)           ' mr_adv_ability[15:8] 01
' mr_autoneg expansion [7:0], bit1=mr_page_rx=0
        Call V.SPut(&H0084C,&h00)           ' mr_autoneg expansion [7:0] 00
' mr_control[15:8], bit 15=ANRESET=0, bit12=ANENABLE=1, bit9=ARRESTART=1
        Call V.SPut(&H00841,&h12)           ' mr_control[15:8] 12
V.Wait(100)



'AUTOMATIC PAUSE CONGIFURATION SECTION


'V.Show_Display()
'V.Clear_Display()
ANStatusReg=V.SGet(&H842&)
ANcomplete= ANStatusReg And &H20

Do
Do until ( ANcomplete <> &H00)   'wait until ANcomplete bit is 1
ANStatusReg=V.SGet(&H842&)
ANcomplete= ANStatusReg And &H20
'V.echo ("ANcomplete=" & Hex(ANcomplete) & "")
V.DO_EVENTS 'allow other events to happen while in loop
If V.shiftkey_pressed Then Exit Sub 'If shift Key pressed, then exit

Loop


LPAdvReg0=V.SGet(&H84A&)' bit7=PAUSE
LPPause= LPAdvReg0 And &H80
'V.echo ("LPPause=" & Hex(LPPause) & "")'
```

```
LPAdvReg1=V.SGet(&H84B&)' bit0=ASYM_PAUSE
LPAsymPause= LPAdvReg1 And &H01

'V.echo ("LPAsymPause=" & Hex(LPAsymPause) & "")'

   If  (LPPause = &H00) Then
   If  (LPAsymPause = &H00) Then
   'PAUSE=0, ASYM_PAUSE=0
   'DISABLE PAUSE TRANSMIT AND RECEIVE
   Call V.SPut(&H00800,&H00)' DISABLE MAC FIRST
   NEWMACCONTROLREG= V.SGet(&H802&) And &HF7 ' DISABLE RX PAUSE, BIT3
   Call V.SPut(&H00802,NEWMACCONTROLREG)
   Call V.SPut(&H00800,&H0D) ' DISABLE TX  MAC PAUSE , bit 1

   Else
   'PAUSE=0, ASYM_PAUSE=1
   'DISABLE PAUSE TRANSMIT , Enable PAUSE RECEIVE
   Call V.SPut(&H00800,&H00)' DISABLE MAC FIRST
   NEWMACCONTROLREG= V.SGet(&H802&) Or &H08 ' Enable RX PAUSE,BIT3
   Call V.SPut(&H00802,NEWMACCONTROLREG)
   Call V.SPut(&H00800,&H0D) ' DISABLE TX  MAC PAUSE , bit 1

   End If

Else
'PAUSE=1, ASYM_PAUSE=DON'T CARE
'Enable PAUSE TRANSMIT , Enable PAUSE RECEIVE
Call V.SPut(&H00800,&H00)' DISABLE MAC FIRST
NEWMACCONTROLREG= V.SGet(&H802&) Or &H08 ' Enable RX PAUSE,BIT3
Call V.SPut(&H00802,NEWMACCONTROLREG)
Call V.SPut(&H00800,&H0F) ' Enable TX  MAC PAUSE , bit 1
'V.echo ("NEWMACCONTROLREG=" & Hex(NEWMACCONTROLREG) & "")'
End If

Do until ( ANcomplete = &H00)   'wait until ANcomplete bit is 0
ANStatusReg=V.SGet(&H842&)
ANcomplete= ANStatusReg And &H20
'V.echo ("ANcomplete=" & Hex(ANcomplete) & "")
V.DO_EVENTS
If V.shiftkey_pressed Then Exit Sub 'If shift Key pressed, then exit
Loop

V.DO_EVENTS

Loop


end sub

Function HexX(Num, Lngth)
    TempStr = Hex(Num)
    HexX = Mid((String(8 - Len(TempStr), "0") & TempStr), (9 - Lngth))
End Function
```

Appendix D. reg_stats_1000BASE-X_Broadcom.vbs ORCAstra Script

```
Sub Main()

V.Show_Display()
V.Clear_Display()

V.Echo("AUTONEG Statistics Reading")
V.Echo("")

' Autnegotiation  STATUSV.Echo("Lattice SGMII AUTONEG:")
temp0 = V.SGet(&h842&)
temp = hex(temp0)
V.Echo("SGMII AUTONEG STATUS (BIT 5=AN complete)  = 0x "& temp &"")

' Autnegotiation  LP Ability register
temp0 = V.SGet(&h84B&)
temp = hex(temp0)
V.Echo("AUTONEG LP Abiliy byte 1:")
V.Echo("BIT7=Next Page,,  BIT6=ACK, BIT5:4=Remote Fault, BIT0=PAUSE = 0x"& temp &"")
temp0 = V.SGet(&h84A&)
temp = hex(temp0)
V.Echo("AUTONEG LP Abiliy byte 0:")
V.Echo("BIT7=ASYM_PAUSE, BIT6=HALF DUPLEX, BIT5=FULL DUPLEX = 0x"& temp &"")




V.Echo("TEST LOGIC")
V.Echo("")

V.Echo("TX/RX status")
V.Echo("")

' RX Status
temp0 = V.SGet(&h88a&)
temp = hex(temp0)
V.Echo("RX FIFO STATUS (BIT2=RX_FIFO_FULL BIT1=RX_FIFO_ERR BIT0=RX_ERRROR)  = 0x "&
temp &"")

' TX Status
temp0 = V.SGet(&h88b&)
temp = hex(temp0)
V.Echo("TX FIFO STATUS ( BIT1=TX_FIFO_FULL BIT0=TX_DISC_FRM)  = 0x "& temp &"")
V.Echo("")

V.Echo("TX/RX statistics counters")
V.Echo("")

' RX PKT Ignored
temp0 = V.SGet(&h88d)
temp1 = V.SGet(&h88c)
temp = temp1 + temp0*2^8
V.Echo("RX Packet Ignored  = " & temp &"")
```

```
' RX Length Check Error
temp0 = V.SGet(&h88f)
temp1 = V.SGet(&h88e)
temp = temp1 + temp0*2^8
V.Echo("RX Length Check Error  = " & temp &"")

' RX Long Frame
temp0 = V.SGet(&h891)
temp1 = V.SGet(&h890)
temp = temp1 + temp0*2^8
V.Echo("RX Long Frame  = " & temp &"")

' RX Short Frame
temp0 = V.SGet(&h893)
temp1 = V.SGet(&h892)
temp = temp1 + temp0*2^8
V.Echo("RX Short Frame  = " & temp &"")

' RX IPG Violation
temp0 = V.SGet(&h895)
temp1 = V.SGet(&h894)
temp = temp1 + temp0*2^8
V.Echo("RX IPG Violation  = " & temp &"")

' RX CRC Error
temp0 = V.SGet(&h897)
temp1 = V.SGet(&h896)
temp = temp1 + temp0*2^8
V.Echo("RX CRC Error  = " & temp &"")

' RX OK PACKET
temp0 = V.SGet(&h899)
temp1 = V.SGet(&h898)
temp = temp1 + temp0*2^8
V.Echo("RX OK PACKET  = " & temp &"")

' RX Control Frame
temp0 = V.SGet(&h89b)
temp1 = V.SGet(&h89a)
temp = temp1 + temp0*2^8
V.Echo("RX Control Frame  = " & temp &"")

' RX PAUSE Frame
temp0 = V.SGet(&h89d)
temp1 = V.SGet(&h89c)
temp = temp1 + temp0*2^8
V.Echo("RX PAUSE Frame  = " & temp &"")

' RX Multicast Frame
temp0 = V.SGet(&h89f)
temp1 = V.SGet(&h89e)
temp = temp1 + temp0*2^8
V.Echo("RX Multicast Frame  = " & temp &"")
```

```
' RX Broadcast Frame
temp0 = V.SGet(&h8a1)
temp1 = V.SGet(&h8a0)
temp = temp1 + temp0*2^8
V.Echo("RX Broadcast Frame  = " & temp &"")

' RX VLAN tagged Frame
temp0 = V.SGet(&h8a3)
temp1 = V.SGet(&h8a2)
temp = temp1 + temp0*2^8
V.Echo("RX VLAN tagged Frame  = " & temp &"")

' TX UNICAST FRAME
temp0 = V.SGet(&h8a5)
temp1 = V.SGet(&h8a4)
temp = temp1 + temp0*2^8
V.Echo("TX UNICAST FRAME  = " & temp &"")

' TX PAUSE FRAME
temp0 = V.SGet(&h8a7)
temp1 = V.SGet(&h8a6)
temp = temp1 + temp0*2^8
V.Echo("TX PAUSE FRAME  = " & temp &"")

' TX MULTICAST FRAME
temp0 = V.SGet(&h8a9)
temp1 = V.SGet(&h8a8)
temp = temp1 + temp0*2^8
V.Echo("TX MULTICAST FRAME  = " & temp &"")

' TX BROADCAST FRAME
temp0 = V.SGet(&h8ab)
temp1 = V.SGet(&h8aa)
temp = temp1 + temp0*2^8
V.Echo("TX BROADCAST FRAME  = " & temp &"")

' TX VLAN FRAME
temp0 = V.SGet(&h8ad)
temp1 = V.SGet(&h8ac)
temp = temp1 + temp0*2^8
V.Echo("TX VLAN FRAME  = " & temp &"")

' TX CRC Error
temp0 = V.SGet(&h8af)
temp1 = V.SGet(&h8ae)
temp = temp1 + temp0*2^8
V.Echo("TX CRC Error  = " & temp &"")

' TX JUMBO FRAME
temp0 = V.SGet(&h8b1)
temp1 = V.SGet(&h8b0)
temp = temp1 + temp0*2^8
```

```
V.Echo("TX JUMBO FRAME  = " & temp &"")




End Sub

Function HexX(Num, Lngth)
    TempStr = Hex(Num)
    HexX = Mid((String(8 - Len(TempStr), "0") & TempStr), (9 - Lngth))
End Function
```