
**Release Notes**

# InfiniScale III MT47396 Firmware

Rev 0.5.0

InfiniScale III 47396 Firmware Release Notes

**Document Number:**

Mellanox Technologies, Inc.
2900 Stender Way
Santa Clara, CA 95054
U.S.A.
www.Mellanox.com

Tel: (408) 970-3400
Fax: (408) 970-3403

Mellanox Technologies Ltd
PO Box 586 Hermon Building
Yokneam 20692
Israel

Tel: +972-4-909-7200
Fax: +972-4-959-3245

# 1 Introduction

The firmware in this package complements the InfiniScale III MT47396 silicon architecture with a set of advanced features, allowing easy and remote management of the switch.

Note: This firmware revision supports IB port operation at 2.5 Gb/s transmission rates (SDR) only.

This document includes the following sections:



# 2 Notes on this Release

- This firmware release has passed all link and management tests in the CIWG Compliance Pack v26.
- This firmware release complies with the *InfiniScale III MT47396 Program Reference Manual, Doc. #2235PM*, except for features listed below in Figure 4,"UnSupported Features," on page 3.
- This firmware release supports the MT47396A1-FCC and MT47396A1-FDC InfiniScale III devices. (The two devices differ in the substrate, but not in functionality.) In both devices, the IB ports operate in SDR (single data rate) transmission frequency (2.5 Gb/s) only.

# 3 Supported New Features

This firmware release supports the following new features:

- The new MT47396A1-FDC InfiniScale III version
- LEDs now blink in case that LinkWidthAct < LinkWidthEn
- Automatic input sensitivity calibration during boot time
- Failsafe boot
- 64-byte data counters

# 4 UnSupported Features

The following features are not supported in this release:

- Baseboard Management Agent (BMA)

# 5 Requirements

- One of the following burning tools:
  - IBADM ver. 1.6.0 or later - ibfwmgr application
  - is3burn 3.1.8 or later
- Supported I2C cards:
  - MTUSB-1 (USB to I2C adapter)
  - ISA Calibre
- This firmware supports the InfiniScale III switch device only

# 6 Supported Mellanox Products

The following table lists the switch system that were tested with this firmware release and were approved by the MQAS (Mellanox QA System).

Table 1 - Mellanox Supported Switch Systems

| Board Name | Description |
|---|---|
| MTS2400 (Reindeer) | A 24-port switch |

# 7 SMA/GSA Attributes

The following tables summarize the attributes supported by the management agents provided in this release.

Table 2 - SMA Supported Attributes

| Attribute | Support |
|---|---|
| Notice | X |
| NodeDescription | X |
| NodeInfo | X |
| SwitchInfo | X |
| GUIDInfo | X |
| PortInfo | X |
| Partition Key Table | X |
| SLtoVLMappingTable | X |
| VLArbitration | X |
| LinearForwardingTable | X |
| RandomForwardingTable | |

Table 2 - SMA Supported Attributes (Continued)

| Attribute | Support |
|---|---|
| MulticastForwardingTable | X |
| SMInfo | |
| VendorDiag | |
| LedInfo | |

Table 3 - Performance Management Supported Attributes

| Attribute | Support |
|---|---|
| ClassPortInfo | X |
| PortSamplesControl | X |
| PortSamplesResult | X |
| PortCounters | X |
| PortCountersExtended (for data counters only) | X |

# 8 Fixed Bugs

## 1 : ENP0 - PKey check in the receive flow does not treat Partnership bit (#15) properly

Keywords: Enhanced Switch Port 0, PKey

Description:

PKey of in-bound ENP0 packet is compared to the appropriate PKey table entry of InfiniScale III, but the Partnership bit (#15) is disregarded.

Impact:

Enhanced Switch Port 0.

Fix:
Fixed in firmware.

Internal Reference: 28583

## 2 : Sub-ports 1 and 2 in an AutoConfig cluster may not rise if the master port is not connected

Keywords: Master port, Sub-port, AutoConfig

Description:

If two auto-configured cluster nodes are connected through their sub-ports and their master ports are not connected, then these sub-ports cannot rise.

Impact:

No activity on some auto-configured IB sub-ports.

Fix:
Fixed in firmware.

Internal Reference: None

## 3 : The initialization sequencing for the InfiniBand ports may cause a system bring-up failure on very rare occasions

Keywords: Initialization, Bring-up

Description:

The initialization sequencing for the IB ports may prevent, on very rare occasions, the switch system to come up stable thus preventing its proper operation.

Impact:

System operation failure.

Fix:
Fixed in firmware

Internal Reference: None

# 9 Known Issues

## 1 : Cannot set PKey Table of the disabled sub-ports of a 3-port cluster configured as a single port

Keywords: PKey

Description:

If a 3-port cluster is configured as a single 4X/12X port, then the PKey Table of two disabled sub-ports of this cluster cannot be changed by Set(PKeyTable) MAD.

Impact:

IB compliance.

Internal Reference: 16268, 20969

# 10 InfiniScale III Firmware Initialization And Configuration (.INI) File

The Mellanox firmware burning tools enable setting initialization and configuration attributes to suit a user's specific system by the use of a special (.INI) file. This section describes this user-supplied initialization and configuration file.

To begin with, the .INI file is a text file composed of several initialization and configuration *sections*. The user may choose to include all sections and all attribute settings in the final .INI file, and modify some of the attributes as required. Alternatively, the user may choose to keep only the sections with changes to the existing settings, with only those attributes that are to be modified.

This .INI file is described in the following sub-sections:

• ".INI File Format" on page 9
• "Description and Usage of .INI File Sections" on page 10

## 10.1 .INI File Format

The .INI file is actually a concatenation of (a part or all) section specific initialization and configuration settings. Each section in the .INI file starts with its name between square brackets, e.g. [EEPROM], [General], etc. The section name is followed by one or more lines of configuration settings and comments, as in the partial .INI file shown below. Note that comment lines start with a semicolon.

**Example:**

```
[EEPROM]
; This is a comment line
amount = 0x4
eeprom1_address = 0x56
eeprom1_size = 64

[General]
PortsBufferingMode = SAF
;This is another comment line
DEVID = 0xb924
VID = 0x2c9
AutoPowerSave = Enable

; End of (partial) .INI file
```

## 10.2 Description and Usage of .INI File Sections

The .INI file sections are:

- EEPROM
- PS_INFO
- PSID
- General
- Special
- ENP0
- MISC
- IB_TO_HW_MAP
- IB_TO_LED_MAP
- PortDisable
- LinkWidthSupp
- PLL
- REV_LANE
- Polarity
- SERDES_Equa_CFG
- SERDES_OutPut_Voltage
- SERDES_Pre_Amp_OutPut
- SERDES_Pre_Emp_Out
- SERDES_Pre_Emp_Pre_Amp
- Credits_Time

Each of these sections is described next.

### 10.2.1 [EEPROM]

The EEPROM section is used to indicate to the burning tool, and later to the firmware itself, the amount, size, slave address, and order of the EEPROMs on the board.

NOTE: The first boot EEPROM must be consistent with the number indicated through strapping pins, as described in InfiniScale III Hardware Reference Manual. (HRM).

**Example.** The example below describes a board with 4 EEPROMs. The size of each EEPROM is 64KB. The I2C slave address of each EEPROM is defined as well.

```
[EEPROM]
;number of EEPROMs
amount = 0x4
;I2C slave address of EEPROM #1
eeprom1_address = 0x56
; size of EEPROM in KB
eeprom1_size = 64
eeprom2_address = 0x57
```

eeprom2_size = 64

eeprom3_address = 0x52

eeprom3_size = 64

eeprom4_address = 0x53

eeprom4_size = 64

## 10.2.2 [PS_INFO]

This section provides firmware configuration (ini) information used by the IB Administration tools. It has no impact on the generated firmware image.

**Example:**

Name = Reindeer4X

Version = 0.5.0

Description = Mellanox Reindeer switch with 4X ports

## 10.2.3 [PSID]

Parameter Set ID (PSID) describes a pre-defined set of parameters. It is used for quick identification of different board configurations. This attribute can be assigned a string built from up to 16 characters of the set: [A-Z,0-9].

### Example:

PSID = MT_0060000001

## 10.2.4 [General]

The General section includes several general configuration attributes of the InfiniScale III device. These attributes are described in the following sub-sections.

### 10.2.4.1 INI file number

This attribute serves as a .INI file revision number. Must be a number of 32 bits.

### Example:

INIFileNumber = 0

### 10.2.4.2 Buffering Mode

Set the attribute PortsBufferingMode to force *all IB ports of the device* to operate in one of two possible buffering modes: either Store and Forward (SAF) or Cut-Through (CT).

In case CT is chosen for buffering mode, then each IB port of the device will get dynamically configured so as to use the minimum buffer size possible. This size depends on the port link width.

In the [General] section write either

PortsBufferingMode = CT

or

PortsBufferingMode = SAF

### 10.2.4.3 Device and Vendor IDs

It is possible to configure the Device and Vendor IDs that will be used by the device as part of NodeInfo.

For InfiniScale III (MT47396) set Device ID to 0xb924 (= 4739610). In the [General] section write:

DEVID = 0xb924

For Mellanox Technologies set the Vendor ID to 0x2c9. In the [General] section write:

VID = 0x2c9

#### 10.2.4.4 Power-Save

The SerDes devices can operate with Auto-Power-Save mode *enabled* or *disabled*. If enabled, the SerDes will be automatically powered-down if the physical link of the port is *disabled*. If this attribute is disabled, the SerDes will consume power whether the physical link of the port is enabled or not.

In the [General] section write either

AutoPowerSave = Enable

or

AutoPowerSave = Disable

#### 10.2.4.5 GPIO-Led Enable

The GPIOLinkStatusLedEn attribute is used to enable or disable encoding of the link status of the ports on LEDs via GPIO pins of the device. For details, please refer to the InfiniScale III (MT47396) Hardware Reference Manual.

In the [General] section write either

GPIOLinkStatusLedEn = Enable

or

GPIOLinkStatusLedEn = Disable

#### 10.2.4.6 SDO-Led Enable

The GMONLinkStatusGmonEn attribute is used to enable or disable encoding of the link status of the ports on LEDs via the serial System Monitoring pin (SDO) of the device. For details, please refer to the InfiniScale III (MT47396) Hardware Reference Manual.

In the [General] section write either

GMONLinkStatusGmonEn = Enable

or

GMONLinkStatusGmonEn = Disable

#### 10.2.4.7 Led Blinking Enable

The LEDBlinkEn attribute is used to enable or disable encoding of the data transfer rate by *blinking* the LEDs. When disabled, the LEDs will only indicate whether the logical link is up or down.

In the [General] section write either

LEDBlinkEn = Enable

or

LEDBlinkEn = Disable

### 10.2.4.8 Single Led Indication Enable

The SingleLedIndication attribute is used to control the physical link status indication of the port cluster in Single mode. If enabled, a single led is used to indicate the physical link status (up or down). If disabled, in addition to the physical link status, two sub-port leds are used to indicate link width (1x, 4x, 12x).

In the [General] section write either

SingleLedIndication = Enable

or

SingleLedIndication = Disable

### 10.2.4.9 Trap Enable

The TrapFlagEn attribute is used to enable or disable the generation of all traps by the InfiniScale III device.

In the [General] section write either

TrapFlagEn = Enable

or

TrapFlagEn = Disable

### 10.2.4.10 GPIO Direction register

The InfiniScale III device has 64 GPIO pins. Each pin may be configured to act as input or output via the GPIO Direction register. If bit x of this GPIO Direction register is cleared (‘0), then GPIO pin number x will act as an input pin to InfiniScale III; if bit x is set (‘1), then GPIO pin number x will act as an output pin of InfiniScale III.

**Example.** To configure GPIO pins [31:16] as outputs, GPIO pins [63:32] and [15:0] as inputs, write in the [General] section:

GPIODir = 0xffff0000

**Possible Conflict.** If the attribute GPIOLinkStatusLedEn is set to *Enable*, the GPIO pins of InfiniScale III that are connected to LEDs should be set to act as outputs in the GPIO Direction register. Otherwise, there will be a conflict between the two configuration settings.

### 10.2.4.11 GPIO Polarity register

The GPIO Polarity register is used to set the input polarity or output open-drain mode of the GPIO pins. Table 4 summarizes the possible configurations.

Table 4 - GPIO pin behavior based on Direction and Polarity register values

| GPIO bit x Direction | GPIO bit x Polarity | GPIO pin x acts as |
|---|---|---|
| 0 | 0 | Non-Inverting Input |
| 0 | 1 | Inverting Input |
| 1 | 0 | Open Drain Non-Inverting Output |
| 1 | 1 | Open Drain Inverting Output |

This feature is especially useful for applications where GPIO pins are used as interrupt inputs. Write in the [General] section the desired 64-bit value to GPIOPol.

**Example.** To set all GPIO pins as non-inverting inputs/outputs, write

GPIOPol = 0x0

#### 10.2.4.12 Parity Error Flow

Describes the desired device behavior in case a Parity Error occurs in its internal memory. This attribute has two possible settings:

1. ParityErrorFlow = SwReset
   - This setting instructs the device to execute a software reset upon Parity Error.
2. ParityErrorFlow = HaltFW
   - Uupon a Parity Error, this setting causes all IB links to go down, the GPIO pin defined by ParityErrorGpioBit (see below) to be asserted, and the firmware to halt.

#### 10.2.4.13 Parity Error GPIO pin

Specifies the GPIO pin to be asserted in case of a firmware halt due to a Parity error. Possible values:

[0..15] if CpuMode=CPU

[0..15, 32..63] if CpuMode=GPIO

See explanation about CpuMode in “CPU Pins Mode” on page 16.

**Example:**

ParityErrorGpioBit = 0x0

### 10.2.5 [Special]

The special section is used for configuring items which have special implications on firmware.

#### 10.2.5.1 System Image GUID

System image GUID is a 64-bit number. Use sysimage_GUID attribute to set it.

Example: To set the system image GUID to 0, write in the [Special] section

sysimage_GUID = 0x0

#### 10.2.5.2 Node Description

Node description is a string written to the node_descr attribute of the Subnet Management class.

Example: Describe the InfiniScale III device as follows in the [Special] section:

node_descr = MT47396 InfiniScale-III Mellanox Technologies

#### 10.2.5.3 Node GUID

Node GUID is a 64-bit number used for NodeInfo. Set it using the NodeGUID attribute.

Example:

Node_GUID = 0x2c90000000000

#### 10.2.5.4 Board Serial Number

The Board Serial Number (BSN) attribute can be used only if the board is supplied by Mellanox Technologies. (Please consult your FAE to know which Mellanox boards are currently supported.)

For example, if the board is an MTS2400, then this attribute should indicate the unique serial number of the board about to burnt. In the case the attribute is not applicable to your board, the following line should be left unchanged in the .INI file:

BSN = NO_BSN

#### 10.2.5.5 Vendor Specific Enable

The attribute VendorSpecEn is used to enable the InfiniScale III Firmware to respond to Vendor Specific Class attributes, or disable it from responding. It determines the value of the bit IsVendorClassSupported in the Capability Mask register.

Write in the [Special] section either

VendorSpecEn = Enable

or

VendorSpecEn = Enable

### 10.2.6 [ENP0]

This section is dedicated for Enhanced Port 0 related attributes.

#### 10.2.6.1 Enhanced Switch Port 0 Enable

If ENP0En attribute is set to Enable, the management port of the InfiniScale III device will function as Enhanced Port 0. If set to Disable, the management port will function as Base Port 0. This attribute also determines the value of the SwitchInfo.ENP0 bit.

Write in the [ENP0] section either

ENP0En = Enabled

or

ENP0En = Disabled

#### 10.2.6.2 CPU Master Target Port Size

The attribute CPUMPortSize is used to indicate to the InfiniScale III CPU bus Master what is the port size of the target on the CPU bus.

For a 32-bit port size write

CPUMPortSize = T32

For a 16-bit port size write

CPUMPortSize = T16

#### 10.2.6.3 CPU Interrupt Request

The attribute CPUIrq is used to indicate to the host which of its interrupt requests is connected to the InfiniScale III interrupt output.

Possible values are 0x0 - 0x5.

Example: To map the InfiniScale III interrupt output to host interrupt request no. 2 write:

CPUIrq = 0x2

### 10.2.7 [MISC]

This section combines miscellaneous attributes.

#### 10.2.7.1 I2C Primary and Secondary slave addresses and enables

Use the I2CSlavePrimaryAddr attribute to set the slave address of the InfiniScale III device on the primary I2C bus; and use I2CSlavePrimaryEnable to enable this slave.

Use the I2CSlaveSecondaryAddr attribute to set the slave address of the InfiniScale III device on the secondary I2C bus; and use I2CSlaveSecondaryEnable to enable this slave.

Note: Setting the primary slave address overrides the information from the strapping pins.

Possible values for the I2C slave enables are: 0x0 or 0x1.

Examples:

I2CSlavePrimaryAddr = 0x6c

I2CSlavePrimaryEnable = 0x1

I2CSlaveSecondaryAddr = 0x2a

I2CSlaveSecondaryEnable = 0x1

#### 10.2.7.2 CPU Pins Mode

The InfiniScale III pins that are used for PPC bus interface can also be used as GPIO pins. The CpuMode attribute allows the user to configure these pins as CPU or GPIO pins in advance.

Write in the [MISC] section either

CpuMode = Cpu

or

CpuMode = Gpio

#### 10.2.7.3 System Monitoring Chain Disable

The SystemMonitoringDis attribute is used to disable the System Monitoring serial chain (via SDO pin). If disabled, the SDO output pin will remain undriven by the device, i.e. in the High-Z state.

Write in the [MISC] section either

SystemMonitoringDis = false

or

SystemMonitoringDis = false

### 10.2.7.4 System Monitoring clock cycle time

The SystemMonitoringClCyT attribute is used to determine the System Monitoring chain clock cycle time. The Core clock cycle time is divided by the value specified by this attribute +1.

Range of values: 10 through 255

Example: If the following line is written in the [MISC] section, then the cycle time of the System Monitoring chain will be the core clock cycle time divided by 0x13 + 1 = 2010.

SystemMonitoringClCyT = 0x13

That is, System Monitoring Cyc Time = (Core clock cycle time) / 20.

### 10.2.7.5 I2C bus master frequency (Primary and Secondary)

The PriI2CBusFreq and SecI2CBusFreq attributes are used to determine the bus frequency of the InfiniScale III device as a master on the primary and secondary I2C busses respectively. The number specified is in KHz units.

Range of values: 50 through 300.

**Examples**:

PriI2CBusFreq = 300

SecI2CBusFreq = 0x41

### 10.2.7.6 CPU Bus Request mode

This attribute sets the ***CBR#*** pin output mode: regular active

For regular active output mode (1 for inactive, 0 for active) write:

CPUBusReqMode = 0x0

For Open Drain output mode write:

CPUBusReqMode = 0x1

Default output mode: Open Drain

This attribute is applicable only for A1 silicon devices. In A0 devices the ***CBR#*** pin is an Open Drain output, regardless of the value of this attribute.

### 10.2.7.7 Target Acknowledge Timer

Specifies the period of time, measured in cycles of CPU bus clock, the CPU bus master waits for a Target Acknowledge signal (CTA#, CTEA#, CRETRY#) after it asserted the Transaction Start signal (CTS#). When this timer expires the transaction is terminated with error status 4. If this attribute is set to 0 the timer is disabled.

Possible values: [0..0xFFF].

**Example**:

TATimer = 0x0

## 10.2.8 [IB_TO_HW_MAP]

This section describes the board connectivity between the Logical InfiniBand ports (as labeled on the chassis) and the actual InfiniScale III device internal ports.

To map a Logical IB port on the chassis to an actual internal port of the InfiniScale III device, a simple mathematical transformation needs to be applied. It requires the InfiniScale III port pin number (the index number attached to the corresponding RX/TX port pins) to which the Logical IB port is connected (See **Figure 1** below).

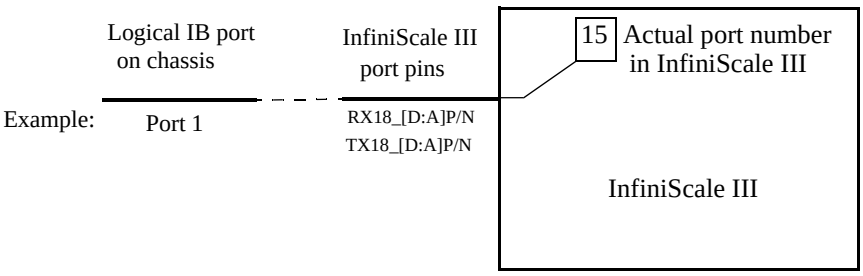


**Figure 1: mapping a chassis Logical IB port number to actual port number inside InfiniScale III**

The transformation formula provided below is necessary since the numbering of the InfiniBand port pins of the InfiniScale III device does not represent its internal (actual) port numbering.

INTERNAL_PORT_NUM = (PIN_PORT_NUM + 21) mod 24

The figure above includes an example. Logical IB port number 1 is connected to port pins RX18_[D:A]P/N and TX18_[D:A]P/N of InfiniScale III; i.e. to port pins number 18. Therefore, *(18+21) mod 24* will give 15 as the internal IB port number. In the [IB_TO_HW_MAP] section of the .INI file, the following line should appear:

PORT1 = 15

where PORT1 indicates the Logical IB port number on the chassis, and 15 is the corresponding port number inside InfiniScale III. There should be 24 such mappings in the [IB_TO_HW_MAP] section, PORT1 through PORT24. For convenience, the mapping of InfiniScale III port pins to actual port numbers inside the InfiniScale III are provided in Table 5, “Mapping InfiniScale III port pins to Actual (internal) port numbers”.

Table 5 - Mapping InfiniScale III port pins to Actual (internal) port numbers

| **InfiniScale III port pins index** RX?_[D:A]P/N and TX?_[D:A]P/N | **Actual port number (in InfiniScale III)** |
|---|---|
| 1 | 22 |
| 2 | 23 |
| 3 | 24 |
| 4 | 1 |
| 5 | 2 |
| 6 | 3 |
| 7 | 4 |

Table 5 - Mapping InfiniScale III port pins to Actual (internal) port numbers (Continued)

| **InfiniScale III port pins index** RX?_[D:A]P/N and TX?_[D:A]P/N | **Actual port number (in InfiniScale III)** |
|---|---|
| 8 | 5 |
| 9 | 6 |
| 10 | 7 |
| 11 | 8 |
| 12 | 9 |
| 13 | 10 |
| 14 | 11 |
| 15 | 12 |
| 16 | 13 |
| 17 | 14 |
| 18 | 15 |
| 19 | 16 |
| 20 | 17 |
| 21 | 18 |
| 22 | 19 |
| 23 | 20 |
| 24 | 21 |



## 10.2.9 [IB_TO_LED_MAP]

For each Logical IB port on the chassis corresponds a (green or yellow) LED. The InfiniScale III employs 6 GPIO pairs (GPIO 16-27) and four GPIO qualifier signals (GPIO 28-31) to light up to 24 chassis IB port LEDs. This section defines the mapping between the Logical (chassis) IB port number and its GPIO control pins.

The mapping scheme to be presented here allows different board designs to have various connectivity and control schemes in a very simple and flexible manner.

**Table 5** below provides code numbers by which it is possible to map logical port to their corresponding LED GPIO controls. Each code in a table cell defines a pair of GPIO pins and a qualifying GPIO pin.

Table 6 - Code numbers for mapping Logical (chassis) IB ports to their corresponding LED GPIO control pins

| **GPIO Pair \ Qualifier** | **GPIO28** | **GPIO29** | **GPIO30** | **GPIO31** |
|---|---|---|---|---|
| **GPIO 16,17** | 1 | 7 | 13 | 19 |
| **GPIO 18,19** | 2 | 8 | 14 | 20 |
| **GPIO 20,21** | 3 | 9 | 15 | 21 |
| **GPIO 22,23** | 4 | 10 | 16 | 22 |

Table 6 - Code numbers for mapping Logical (chassis) IB ports to their corresponding LED GPIO control pins

| GPIO Pair \ Qualifier | GPIO28 | GPIO29 | GPIO30 | GPIO31 |
|---|---|---|---|---|
| **GPIO 24,25** | 5 | 11 | 17 | 23 |
| **GPIO 26,27** | 6 | 12 | 18 | 24 |

Example 1: If Logical IB port 1 is controlled by the pair [GPIO26, GPIO27] and the qualifier GPIO30, then the [IB_TO_LED_MAP] section should include the following line:

PORT1 = 18

Example 2: If Logical IB port 24 is controlled by the pair [GPIO22, GPIO23] and the qualifier GPIO28, then the [IB_TO_LED_MAP] section should include the following line:

PORT24 = 4

### 10.2.10 [PortDisable]

This section is used to enable or disable the IB ports of the InfiniScale III. If a port is disabled here, then its physical link will remain disabled after boot, and its SerDes TX will be powered down.

Examples:

PORT1 = Enabled

PORT2 = Enabled

PORT3 = Disabled

PORT4 = Disabled

...

...

### 10.2.11 [LinkWidthSupp]

The 24 IB ports of InfiniScale III are viewed as eight 3-port clusters. Each such cluster gets a single command line in the [LinkWidthSupp] section to configure the link widths of its 3 ports. The configuration settings in this section determine both the LinkWidthSupported and the LinkWidthEnabled attributes of all InfiniScale III ports during boot.

The clusters are numbered CL1 through CL8, and the ports are assigned to these clusters in an ascending order. Thus, cluster CL1 groups ports 1, 2, and 3; cluster CL2 groups ports 4, 5, and 6;..., cluster CL8 groups ports 22, 23, and 24.

To configure a cluster with 3 ports, write in the [LinkWidthSupp] section lines in the following format:

CLi = config_value

where i represents a cluster number in the range 1,2,...,8, and config_value represents one of the values described in the table below.

Table 7 - Configuration values for cluster port link widths

| config_value | Resulting Cluster Configuration |
|---|---|
| TRIO_1X | the cluster is configured as 3 separate ports, each limited to 1X link width only |
| TRIO_1X4X | the cluster is configured as 3 separate ports, each supporting 1X or 4X link width |
| SINGLE_1X | the cluster is configured as a single port, limited to 1X link width only |

Table 7 - Configuration values for cluster port link widths

| config_value | Resulting Cluster Configuration |
|---|---|
| SINGLE_1X4X | the cluster is configured as a single port supporting 1X and 4X link width |
| SINGLE_1X4X12X | the cluster is configured as a single port supporting 1X, 4X and 12X link widths |
| AUTOCONF_1X4X12X | the cluster will auto-configure itself to match the configuration of the(peer) cluster connected to it through the IB cable. |

Note that in case a cluster is set as AUTOCONF_1X4X12X, then it may rise as a single 12X port or as 3 4X ports, depending on the peer cluster capabilities at the time of link bring-up. This may occur in every instance of bring-up. Thus the cluster may start out as 3 4X ports prior to link (re)establishment, and turn into a single 12X port after (re)establishment, or vice versa.

Examples:

CL1 = TRIO_1X4X

CL2 = TRIO_1X4X

CL3 = SINGLE_1X4X

CL4 = SINGLE_1X

CL5 = TRIO_1X4X

CL6 = SINGLE_1X4X12X

CL7 = TRIO_1X4X

CL8 = AUTOCONF_1X4X12X

## 10.2.12 [PLL]

This section is used to set the PLL boot record values.

### 10.2.12.1 Serial EEPROM Clock Rate Divider

When a serial EEPROM is present, the attribute ClockRateDivider is used to set the desired serial clock rate for reading the remaining part of the PLL Configuration Record via the I2C bus. It is an 8-bit number that the InfiniScale III multiplies by 16 to obtain the number by which to divide the external clock frequency. Thus, the I2C serial clock frequency will be given by the following equation:

*Serial_Clock_Frequency = External_Clock_Frequency / (16 x ClockRateDivider)*

In the [PLL] section, write a line in the following format:

ClockRateDivider = number

where number is an integer between 1 and 255.

### 10.2.12.2 Core PLL Divider

The core clock PLL in the InfiniScale III includes a divider for the external clock input. This divider requires two parameters, M and N, to produce required output clock frequency. The parameter M is supplied by the ClockDividerNumerator attribute, and N is supplied by the ClockDividerDenominator attribute in the [PLL] section of the INI file.

The core PLL output clock frequency is calculated by the following equation:

*Fout = [M/(2 x N)] x Fin*

where,

*Fout* is the core PLL output clock frequency

*Fin* is the external input clock frequency

*N* is a 3-bit wide Clock Divider Denominator

*M* is a 4-bit wide Clock Divider Numerator

For example, if the external clock input to InfiniScale III runs at 66MHz, i.e. Fin = 66, and the divider parameters are N=1 and M=5, then the PLL output core clock frequency will be:

*Fout = [5/(2 x 1)] x 66 = 165MHz.*

In the [PLL] section the attribute settings will appear as follows (for the example above):

ClockDividerDenominator = 0x1

ClockDividerNumerator = 0x5

### 10.2.12.3 VCO Range

The VCO_RANGE attribute should be to 1 to indicate that the VCO frequency is greater than or equal to 133MHz. Otherwise, it should be set to 0.

Example:

VCO_RANGE = 0x1

### 10.2.12.4 SerDes PLL Ratio – LDIV

InfiniScale III has two PLLs which provide the high speed differential clocks to the SerDes devices. PLLr0 provides the clock to ports 1 through 12, while PLLr1 provides the clock to ports 13 through 24.

The PLLr0Ldiv and PLLr1Ldiv attributes are used to set the ratio between the frequency of the external refclk pins V0RCLKP/N and V1RCLKP/N and the SerDes differential clock outputs. The SerDes differential clock nominal frequency is 2.5GHz.

In order to obtain the nominal 2.5 GHz clock, the frequency of the external refclk is summarized in the following table:

Table 8 - PLLr0/1Ldiv settings according to input reference clock frequency

| PLLr0/1Ldiv | Required External Reference Clock Frequency |
|---|---|
| 00 | 250MHz |
| 01 | 125MHz |
| 10 | 62.5MHz |
| 11 | 31.25MHz |

**Example**: With a 250MHz reference input clock, set the attributes in the [PLL] section to 00 to obtain 2.5 GHz link speed.

PLLr0Ldiv = 0x0

PLLr1Ldiv = 0x0

### 10.2.12.5 Monitoring of the SerDes PLL Reference Clock

SerDes PLL Reference Clock can be monitored on the special device pin VPTEST. There are two parameters defining which of the PLL's clocks is issued on this pin: Er0Out and Er1Out, as summarized in the following table:

Table 9 - Selection of RefClk for Monitoring

| Er0Out | Er1Out | Reference clock driven out |
|---|---|---|
| 0 | 0 | VPTEST pin is Hi-Z |
| 0 | 1 | RefClk1 |
| 1 | 0 | RefClk0 |
| 1 | 1 | VPTEST pin is Hi-Z |

**Example**:

Er0Out = 0x0

Er1Out = 0x0

### 10.2.12.6 Core PLL Stabilization Time

The PLLStabilizationTime attribute is used to define the core PLL stabilization time. The value supplied indicates the number of external clock cycles required for PLL stabilization. During this stabilization period, the internal reset signal of InfiniScale III will remain asserted.

In the [PLL] section write a line in the following format:

PLLStabilizationTime = number

where number is an integer *larger* than 0.

## 10.2.13 [REV_LANE]

Each port of the InfiniScale III has 4 lanes for data transfer. This section is used to configure the Lane Reversal feature of these IB ports, taking into account the 3-port clustering mentioned in Section 10.2.11, "[LinkWidthSupp]," on page 20. Due to differences between receive (RX) and transmit (TX) attributes, they are discussed here separately.

### 10.2.13.1 RX attributes

The RX lines of each IB port have two attributes for setting in the [REV_LANE] section: Rx_Rev_lane_IB_port*X* and Rx_Sub_Rev_lane_IB_port*X*, where *X* is actually a cluster number [1..24].

**The Rx_Rev_lane_IB_port*X* attribute.** The usage of this attribute depends on the cluster configuration:

3. If a port is part of a cluster is in "Trio" mode, i.e. it consists of 3 separate ports, each supporting 1X or 4X link widths, then the Rx_Rev_lane_IB_port*X* attribute affects only the referenced port, and determines whether the 4 lanes of the port are to be reversed, not reversed, or automatically configured by firmware. To configure such a port in the cluster using this attribute, write in the [REV_LANE] section a line in the following format:

   Rx_Rev_lane_IB_port*X* = AutoConfig | Off | On

   where one of the 3 optional values is chosen. If AutoConfig is chosen, the firmware will configure the lanes; Off will keep the order 0..3; On will reverse it to 3..0.

   Note: It is highly recommended to set this RX parameter in this section to "AutoConfig".

4. If a port is part of a cluster in “Single” mode, i.e. it is a single IB port supporting up to 12X link width, then the Rx_Rev_lane_port*X* attribute determines whether *three ports* in the cluster are to be reversed, not reversed, or automatically configured by firmware. The term “reversed” in this context refers to port-ordering but not lane-ordering. This attribute has no effect on the lanes within each of these three ports. Thus, if the value “On” is set to Rx_Rev_lane_port*X* attribute, and the cluster ports are p1, p2, and p3, their order will be reversed to p3, p2, and p1. To configure the lanes of each port, see the next RX attribute.

**The Rx_Sub_Rev_lane_IB_port*X* attribute.** The second RX parameter in the [REV_LANE] section is used only when a cluster is in “Single” mode; it has no effect on a cluster in “Trio” mode. When in “Single” mode, this parameter determines whether the lanes of a specific port of the 3-port 12X cluster are to be reversed, not reversed, or automatically configured by firmware. If the Rx_Sub_Rev_lane_IB_port*X* attribute is “On”, then the lanes 0..3 of the port will be reversed to 3..0.

Note: It is highly recommended to set this RX parameter in this section to “AutoConfig”.

Example: Suppose we want to configure the RX attributes of a cluster comprised of ports 7, 8, and 9, and set it in SINGLE_1X4X12X mode. To have the cluster ports reversed in order and the port-lanes reversed in order, we would set the RX attributes as follows:

Rx_Rev_lane_IB_port*7* = On
Rx_Sub_Rev_lane_IB_port7 = On
Rx_Sub_Rev_lane_IB_port8 = On
Rx_Sub_Rev_lane_IB_port9 = On

Note that when setting the Rx_Rev_lane_IB_port*X* attribute for a single-mode cluster, *X* must be the Master Port of the cluster (i.e. the smallest port number in the cluster, 7 in this example). However, it is necessary to specify for each port separately whether its lanes should be reversed or not. Thus, to keep the lanes of port 8 not reversed, we should set the attributes as follows:

Rx_Rev_lane_IB_port*7* = On
Rx_Sub_Rev_lane_IB_port7 = On
Rx_Sub_Rev_lane_IB_port8 = Off
Rx_Sub_Rev_lane_IB_port9 = On

### 10.2.13.2 TX attribute

**Tx_Rev_lane_port*X***. This attribute is used in a similar manner to the Rx_Rev_lane_port*X* (See above). The reversal of port lanes of a cluster in “Single” mode is automatically handled by firmware in accordance with the configuration of Section 10.2.8, “[IB_TO_HW_MAP],” on page 18.

To set this attribute in the [REV_LANE] section write a line in the following format:

Tx_Rev_lane_IB_port*X* = On | Off | AutoConfig

**Tx_Trio_Rev_lane_IB_cluster*Y***. This attribute is applicable only in A1 silicon devices. It allows TX lane reversal in a cluster configured in TRIO mode in the manner described in the following table:

Table 10 - TX Trio Lane Reversal

| Original lanes | Reversed lanes |
|---|---|
| 0..3 | 11..8 |
| 4..7 | 4..7 (no change) |

| Original lanes | Reversed lanes |
|---|---|
| 8..11 | 3..0 |

Possible values: Off, On

**Example:**

Tx_Trio_Rev_lane_IB_cluster*Y* = On

## 10.2.14 [Polarity]

This section is used to configure the polarity of the RX and TX lines of the SerDes interfaces for each IB port of the InfiniScale III device. Due to differences between RX and TX attributes, they are discussed here separately.

### 10.2.14.1 RX Polarity

The polarity of RX port lines can either be forced in the .INI file or left to be auto-configured. There are two attributes in this regard: Rx_Force_Pol_IB_port*X* and Rx_Pol_IB_port*X*, where *X*=1, 2,...24. The attribute Rx_Force_Pol_IB_port*X* determines whether the RX polarity is be forced in the INI file or to be auto-configured: if it is to be forced, then it requires setting the second attribute Rx_Pol_IB_port*X* as well; otherwise, Rx_Pol_IB_port*X* will be ignored.

Rx_Force_Pol_IB_port*X* can assume one of two values: Enable or Disable. To force a polarity, set this attribute to Enable; to have it auto-configured, set the attribute to Disable.

Rx_Pol_IB_port*X* can assume one of 16 possible values: 0x0 through 0xf. Each of the four bits determines the polarity of one RX lane of the port. A ‘0 bit for a lane will have it operate in normal (active high) polarity, whereas a ‘1 bit will have it operate in reverse (active low) polarity.

Examples:

1. To have the polarity of RX lanes of port 5 *auto-configured*, write in the [Polarity] section the following line:

   Rx_Force_Pol_IB_port5 = Disable

2. To *force* the polarity of all four RX lanes of port 5 to act in reverse polarity, write in the [Polarity] section the following lines:

   Rx_Force_Pol_IB_port5 = Enable

   Rx_Pol_IB_port5 = 0xf

3. To *force* the polarity of RX lanes 0, 1, and 3 of port 5 to act in normal polarity, and lane 2 in reverse polarity, write in the [Polarity] section the following lines:

   Rx_Force_Pol_IB_port5 = Enable

   Rx_Pol_IB_port5 = 0x4

### 10.2.14.2 TX Polarity

The polarity of TX port lines is always forced in the INI file. For each port there 4 lanes, and each lane cab be set to operate in normal (active high) polarity, or in reverse (active low) polarity. In this case, one attribute is required per port, i.e. Tx_Pol_IB_Port*X*, where *X*=1, 2,...,24. This attribute takes the same range of values as in the case of its RX parallel.

Example: To *force* the polarity of TX lanes 0 and 1 of port 9 to act in normal polarity, and lanes 2 and 3 in reverse polarity, write in the [Polarity] section the following line:

Rx_Pol_IB_port9 = 0xc

### 10.2.15 [SERDES_Equa_CFG]

This section is optionally used to configure the SerDes Equalization level in Single Data Rate (SDR) mode. Each of the 96 SerDes devices (24 ports x 4 lanes) of InfiniScale III is configured separately. The attribute used for each lane is PORTi_SERj, where i represents a port number (1,2,...24), and j represents a lane number (0,1,2,3).

PORTi_SERj can assume an equalization level number between 0x0 and 0xf. By default, if this attribute is not specified for some lane(s) in the .INI file, it will hold the value 0x0.

Example: To set lane 2 of port 3 to level 7, write in the [SERDES_Equa_CFG] section the following line:

PORT3_SER2 = 0x7

### 10.2.16 [SERDES_OutPut_Voltage]

This section is optionally used to tune the SerDes output voltage level in SDR mode. Each of the 96 SerDes devices (24 ports x 4 lanes) of InfiniScale III is configured separately. The attribute used for each lane is PORTi_SERj, where i represents a port number (1,2,...24), and j represents a lane number (0,1,2,3).

PORTi_SERj can assume an output voltage level number between 0x0 and 0xf. By default, if this attribute is not specified for some lane(s) in the .INI file, it will hold the value 0x2.

Example: To set lane 1 of port 9 to output voltage level 0xb, write in the [SERDES_OutPut_Voltage] section the following line:

PORT9_SER1 = 0xb

### 10.2.17 [SERDES_Pre_Amp_OutPut]

This section is optionally used to tune the SerDes Pre-Amplifier output voltage level in SDR mode. Each of the 96 SerDes devices (24 ports x 4 lanes) of InfiniScale III is configured separately. The attribute used for each lane is PORTi_SERj, where i represents a port number (1,2,...24), and j represents a lane number (0,1,2,3).

PORTi_SERj can assume a pre-amplifier output voltage level number between 0x0 and 0xf. By default, if this attribute is not specified for some lane(s) in the .INI file, it will hold the value 0x2.

Example: To set lane 0 of port 23 to pre-amplifier output voltage level 0xe, write in the [SERDES_Pre_Amp_OutPut] section the following line:

PORT23_SER0 = 0xe

### 10.2.18 [SERDES_Pre_Emp_Out]

This section is optionally used to tune the SerDes Pre-Emphasis output level in SDR mode. Each of the 96 SerDes devices (24 ports x 4 lanes) of InfiniScale III is configured separately. The attribute used for each lane is PORTi_SERj, where i represents a port number (1,2,...24), and j represents a lane number (0,1,2,3).

PORTi_SERj can assume a pre-emphasis output level number between 0x0 and 0xf. By default, if this attribute is not specified for some lane(s) in the .INI file, it will hold the value 0x0.

Example: To set lane 3 of port 19 to pre-emphasis output level 0x6, write in the [SERDES_Pre_Emp_Out] section the following line:

PORT19_SER3 = 0x6

### 10.2.19 [SERDES_Pre_Emp_Pre_Amp]

This section is optionally used to tune the SerDes Pre-Emphasis level of the Pre-Amplifier in SDR mode. Each of the 96 SerDes devices (24 ports x 4 lanes) of InfiniScale III is configured separately. The attribute used for each lane is PORTi_SERj, where i represents a port number (1,2,...24), and j represents a lane number (0,1,2,3).

PORTi_SERj can assume a pre-emphasis level number between 0x0 and 0xf. By default, if this attribute is not specified for some lane(s) in the .INI file, it will hold the value 0x0.

Example: To set lane 2 of port 15 to pre-emphasis level 0x4 of pre-amplifier, write in the [SERDES_Pre_Emp_Pre_Amp] section the following line:

PORT15_SER2 = 0x4

### 10.2.20 [Credits_Time]

This section is optionally used to set, for each InfiniScale III IB port, the maximum and minimum elapsed time periods between two successive transmissions of credit packets on each Virtual Lane (VL). The two attributes used are MAX_TIME_PORT*i* and MIN_TIME_PORT*i*, where *i* is a port number (1,2,...24).

MIN_TIME_PORT*i* is specified in units of (64 x symbol-time), where symbol-time equals 256ns. After this minimum time has elapsed, the output port *i* schedules for transmission a credit-update packet for every VL. The transmission occurs once new credits become available.

MAX_TIME_PORT*i* is used to force the delivery of credit packets for every VL after a certain amount of time has elapsed (even if no new credits are available). MAX_TIME_PORT*i* is specified in units of “cycles of the minimum time counter”. This means that the actual maximum elapsed time between successive credit packets will be given my the following expression: PORT_MAX_TIME*i* x PORT_MIN_TIME*i* x (64 x symbol time).

Note that in the case where MAX_TIME_PORT*i* = 0, no VL credit packets will be transmitted as long as *no new* credits are available, regardless of the elapsed time since the last transmission.

The possible values for *both* attributes are 0x0 through 0xff. By default, i.e. if these attributes of a specific port *i* are not defined in the .INI file, then MAX_TIME_PORTi = 0x20 and MIN_TIME_PORTi = 0x8.

Example:

Suppose a user entered in the [Credits_Time] section the following two lines:

MIN_TIME_PORT5 = 0x8

MAX_TIME_PORT5 = 0x80

In this case, the shortest time period between two successive credit updates on a VL of port 5 will be 8 x 64 = 512 symbol-times; of course, this requires the availability of credits. The longest time period between two successive credit updates will be 128 x 512= 64K symbol-times. (0x80 = 12810)