Mellanox IB DDR Auto-negotiation Protocol

Rev 1.0
Contents

Contents 3
List of Tables 5
List of Figures 7
Chapter 1: Introduction 9
Chapter 2: Mellanox IB DDR Auto-negotiation 10
  1.1 Auto-negotiation Protocol 10
    1.1.1 Outline 10
    1.1.2 Flow Diagrams 10
    1.1.3 Auto-negotiation Vendor Specific MAD 14
    1.1.4 Port Info for DDR Support 16
  2.2 SDR to DDR Auto-Negotiation State Machine 17
Chapter 3: Revision History 21
List of Tables

Table 1:  Vendor Specific MAD Layout 14
Table 2:  Vendor Specific MAD Field Descriptions 15
Table 3:  Affected PortInfo Attribute Fields 16
Table 4:  Example of Configuration Parameters Used in Sweep Mode 20
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>SDR to DDR Auto-negotiation (Successful Case 1)</td>
<td>11</td>
</tr>
<tr>
<td>Figure 2</td>
<td>SDR to DDR Auto-negotiation (Successful Case 2)</td>
<td>12</td>
</tr>
<tr>
<td>Figure 3</td>
<td>SDR to DDR Auto-negotiation (Failure Case)</td>
<td>13</td>
</tr>
<tr>
<td>Figure 4</td>
<td>SDR to DDR Auto-negotiation State Machine</td>
<td>18</td>
</tr>
<tr>
<td>Figure 5</td>
<td>DDRSweep (Sweep Mode) State</td>
<td>19</td>
</tr>
</tbody>
</table>
1 Introduction

Mellanox provides HCA and switch devices having InfiniBand ports which are capable of transmitting/receiving data at SDR (Single Data Rate) or DDR (Double Data Rate). That is, their InfiniBand SERDESes (links) can operate at 2.5Gb/s (SDR) or 5Gb/s (DDR) when communicating with a DDR-capable device.

To set SDR or DDR SERDES operation for a device’s InfiniBand (IB) port, both hardware and firmware configuration is required. A SERDES can be configured to perform one of the following:

1. Operate at SDR only
2. Operate at DDR only
3. Auto-negotiate with the SERDES at the other end of the link to operate at DDR

This document describes the specification of Mellanox Technologies’ IB DDR Auto-negotiation for devices compliant with the *InfiniBand Architecture Specification, Release 1.1*.

Note that a ‘link’ in this document refers to a pair of SERDES lanes – one TX and one RX.
2 Mellanox IB DDR Auto-negotiation

2.1 Auto-negotiation Protocol

2.1.1 Outline
The following is an outline of the Mellanox IB DDR auto-negotiation protocol for the speed of the link connecting between two IB ports on two different devices. For reference, the two ends of the link are termed Node 0 and Node 1.

1. **Link @ SDR:** Node 0 and Node 1, both configured to auto-negotiate for DDR operation, rise initially operating at SDR.
2. **Capabilities Advertisement:** Each of the nodes sends on the link a Vendor Specific MAD (see “Auto-negotiation Vendor Specific MAD” on page 14) to figure out whether the other node is DDR capable. If a positive acknowledgement is received, then a parameter check procedure is initiated.
   Note: The above is true for both nodes even if only one of the two gets acknowledged.
3. **Capabilities Advertisement + ACK:** The nodes transmit to each other how many SERDES TX and RX configurations each node wishes to test. The nodes transmit to each other the duration for testing each configuration.
4. **Sweepmode @ DDR:** Both nodes transfer to Sweep Mode state where each node transmits on its TX one configuration after the other, while the RX verifies whether the received configuration matches one of its options (set in the firmware configuration file), and that the received data is free of symbol errors.
   Note: Transmission and testing in Sweep Mode are performed at DDR rate. This means that the physical link (at SDR) becomes down and is represented as failing.
   Note: This note applies only to a connection between a no-sweep HCA device and an InfiniScale III based Switch. A node may choose to skip Sweep Mode entirely. This is requested in the **Capabilities Advertisement** and **Capabilities Advertisement + ACK** stages above, and the **Sweepmode @ DDR** stage will not be reached. See the **NoSweep** field in Table 1, “Vendor Specific MAD Layout,” on page 14.
5. **Return to SDR:** Both nodes return to SDR state after Sweep Mode.
6. **Result Advertisement:** Both nodes transmit to each other the desired configuration (“my desired RX is your TX configuration # X”).
7. **Result Advertisement + ACK:** Each node acknowledges the configuration request of the other and configures its SERDES TX configuration according to what the requested, and its SERDES RX configuration according to what it found to suit itself.
   Note: The above is true for both nodes even if only one of the two gets acknowledged.
8. **GO DDR/SDR:** If a suitable DDR configuration is agreed upon, both nodes will go to DDR operation with the chosen configuration parameters. If no suitable configuration was found, then both nodes will operate at SDR.

2.1.2 Flow Diagrams
The following three figures show flow diagrams of auto-negotiation between Node 0 and Node 1. The first two cases (Figure 1 on page 11 and Figure 2 on page 12) yield a successful transition to DDR operation upon finding appropriate configurations for the two nodes. The third case (Figure 3 on page 13) yields operation at SDR only due to failure in finding configurations suitable for the two nodes to operate together at DDR.
Figure 1: SDR to DDR Auto-negotiation (Successful Case 1)

Node 0

- Link @ SDR
- Capabilities Advertisement
- Capabilities Advertisement + ACK
- Sweepmode at DDR
- Back to SDR
- Result Advertisement
- Result Advertisement + ACK
- Go DDR with the chosen parameters

Node 1

- Link @ SDR
- Capabilities Advertisement
- Capabilities Advertisement + ACK
- Sweepmode at DDR
- Back to SDR
- Result Advertisement
- Result Advertisement + ACK
- Go DDR with the chosen parameters
Note that in Figure 2, Node 1 does not receive an ACK for its Capabilities Advertisement. Nevertheless, it carries along with the protocol since it re-issued the capabilities with its Capabilities Advertisement + ACK to Node 0. Similarly, Node 0 does not receive an ACK for its Result Advertisement. Nevertheless, it carries along with the protocol since it re-issued the result with its Result Advertisement + ACK to Node 1.

Figure 2: SDR to DDR Auto-negotiation (Successful Case 2)
Figure 3: SDR to DDR Auto-negotiation (Failure Case)

Node 0

- Link @ SDR
- Capabilities Advertisement
- Capabilities Advertisement + ACK
- Sweepmode at DDR
- Back to SDR
- Result Advertisement
- Result Advertisement + ACK
- SDR Only (No suitable config. for DDR)

Node 1

- Link @ SDR
- Capabilities Advertisement
- Capabilities Advertisement + ACK
- Sweepmode at DDR
- Back to SDR
- Result Advertisement
- Result Advertisement + ACK
- SDR Only (No suitable config. for DDR)
2.1.3 Auto-negotiation Vendor Specific MAD

The Request and Ack packets exchanged between two link nodes auto-negotiating for DDR operation use the Vendor Specific MAD format described in Table 1 and Table 2.

The MAD should use the following attributes:
- MgmtClass = 0x81 (Direct Route)
- Permissive LID
- VL15
- QP0
- Method: Send (0x3)
- Attribute ID: 0x02C9
- AttributeModifier: 0x000002C9

Table 1 - Vendor Specific MAD Layout

<table>
<thead>
<tr>
<th>Offset</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>SerDesConf</td>
<td>NoSweep</td>
</tr>
<tr>
<td>29</td>
<td>Reserved</td>
<td>Sustained DDR Receive BW</td>
</tr>
<tr>
<td>22</td>
<td>Iteration Duration</td>
<td>00004h</td>
</tr>
<tr>
<td>18</td>
<td>Number of Iterations / Chosen Iteration Number</td>
<td>00004h</td>
</tr>
<tr>
<td>15</td>
<td>Reserved</td>
<td>00000h</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>S7:</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>00000h</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>00000h</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>00000h</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>00000h</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>00000h</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>00000h</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>00000h</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>00000h</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>00000h</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>00000h</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>00000h</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>00000h</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>00000h</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>00000h</td>
</tr>
<tr>
<td>Offset</td>
<td>Bits</td>
<td>Name</td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>00000h</td>
<td>31</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>SerDesConf</td>
</tr>
<tr>
<td></td>
<td>29</td>
<td>NoSweep</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note:</td>
</tr>
<tr>
<td></td>
<td>28:1</td>
<td>Reserved</td>
</tr>
<tr>
<td>00004h</td>
<td>31:0</td>
<td>Sustained DDR Receive BW</td>
</tr>
<tr>
<td>00008h</td>
<td>31:0</td>
<td>Iteration Duration (IT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Otherwise</td>
</tr>
<tr>
<td>0000Ch</td>
<td>31:0</td>
<td>Number of Iterations / Chosen Iteration Number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Otherwise</td>
</tr>
<tr>
<td>00010h</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>000FCh</td>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>
2.1.4 Port Info for DDR Support

If an IB port is configured (via hardware and firmware) to auto-negotiate for DDR speed, the following link-speed fields of the PortInfo attribute should be updated as described in Table 3.

Table 3 - Affected PortInfo Attribute Fields

<table>
<thead>
<tr>
<th>PortInfo Field</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LinkSpeedSupported</td>
<td>3</td>
<td>2.5 or 5.0 Gbps</td>
</tr>
<tr>
<td>LinkSpeedActive</td>
<td>1 or 2</td>
<td>Reflects the actual port speed:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 1 for SDR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 2 for DDR</td>
</tr>
<tr>
<td>LinkSpeedEnabled</td>
<td>3</td>
<td>2.5 or 5.0 Gbps</td>
</tr>
</tbody>
</table>
2.2 SDR to DDR Auto-Negotiation State Machine

Figure 4 on page 18 shows the Mellanox SDR to DDR Auto-Negotiation (i.e., Link Training) state machine diagram. The state machine uses the following two variables:

- RetryCnt0 - number of transmissions of request packets
- RetryCntAck - number of transmissions of Ack packets

Also, the following parameters are used to configure the state machine:

- BurstLen - number of Request/Ack packets to be sent in each transmission. Default value = 4.
- MaxRetry - Maximum number of retries for Request/Ack transmission. Default value = 4.
- XmitTimeOut - timeout for receiving Ack. Default value = 1ms.
- DDRTimeOut - Timeout before falling back to SDR. Default value = 400ms.

Once the IB ports are up, the auto-negotiation protocol is executed as described in the state machine diagram. Subnet Manager queries to the ports should be responded to with LinkPhyState=Polling and LinkState=Down until the protocol is completed (i.e., either SDR state or DDR state is reached).

The DDRSweep (Sweep Mode) sub-state-machine is shown in Figure 5 on page 19. See also the example following the figure.
Figure 4: SDR to DDR Auto-negotiation State Machine

- **INIT**
  - Train Link in SDR
  - SerDesConf = 0

- **SENDREQ**
  - Send BurstLen packets
  - RetryCnt0++

- **RECV0**
  - Wait for received packets
  - If (Packet.SerDesConf != SerDesConf) Drop packet

- **SENDACK**
  - Send BurstLen packets with AckBit set
  - RetryAckCnt++

- **RECV1**
  - Wait for received packets
  - Received Request
  - Received Ack & SerDesConf == 1

- **DDRSWEEP**
  - SerDesConf = 1
  - See Figure 5 for sub states

- **TRYDDR**
  - Set DDR parameters
  - Set SerDes DDR params
  - Train Link in DDR
  - (DDRTTimeOut) or (LinkWidth DDR != LinkWidth SDR)

- **TRYSDR1**
  - Set SDR parameters
  - Set SerDes SDR params
  - Train Link in SDR
  - LinkUp

- **SDR**
  - Link will remain in SDR

**Transition Rules**:
- If PhyLinkDown and state is: INIT, SENDREQ, RECV0, SENDACK, DDR, SDR, Move to INIT state
- If (Packet.SerDesConf != SerDesConf) Drop packet
- (SerDesConf | NoSweep) & (Received Ack or PhyLinkDown)
- (XmitTimeOut) & (RetryCnt = MaxRetry)

**Notes**:
- SEE Figure 5 for sub states
- SerDesConf = 1
- LinkUp
- 2* DDRTimeOut
- 2* DDRTTimeOut
**Legend of variables:**
- X – Index of TX config.
- Y – Index of RX config.
- NOI – # of TX iterations
- NRI – # of RX iterations
- soak time – 150ms (Recommended)
- delta – 50ms (Recommended)

**Example:** Consider a case of auto-negotiation on a link between an InfiniScale III switch device (Node 0) and an InfiniHost III Lx HCA device (Node 1). Table 4 on page 20 provides a summary of Sweep Mode parameters assuming that: delta=50ms and soak time=150ms.
With the above settings, 3 SERDES TX configurations are transmitted by Node 0, and will be checked against 4 SERDES RX configurations by Node 1, and 5 SERDES TX configurations transmitted by Node 1 will be checked against 9 SERDES RX configurations by Node 0. Each iteration lasts for 750ms.

### Table 4 - Example of Configuration Parameters Used in Sweep Mode

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Node 0</th>
<th>Node 1</th>
<th>Value Used in Sweep Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOI</td>
<td>3</td>
<td>5</td>
<td>5 (by both)¹</td>
</tr>
<tr>
<td>NRI</td>
<td>9</td>
<td>4</td>
<td>9 by Node 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 by Node 1</td>
</tr>
<tr>
<td>Iteration Duration (IT)</td>
<td>soak + NRI<em>delta + soak = 150 + 9</em>50 +150 = 750ms</td>
<td>soak + NRI<em>delta + soak = 150 + 4</em>50 +150 = 500ms</td>
<td>750ms (by both)</td>
</tr>
</tbody>
</table>

1. Since Node 0 has less transmit configurations to test, it can re-transmit two out of the three configurations it has.
3 Revision History

Rev 1.0 – May 7, 2009:

• Initial release