

InfiniBand Trade Association

Revision 0.34
07/23/2018

IBTA CIWG MOI for QSFP Module Select Testing

This material is provided for reference only. The InfiniBand Trade Association does not endorse the vendors equipment outlined in this document.

This document is provided "AS IS" and without any warranty of any kind, including, without limitation, any express or implied warranty of non-infringement, merchantability or fitness for a particular purpose. In no event shall IBTA or any member of IBTA be liable for any direct, indirect, special, exemplary, punitive, or consequential damages, including, without limitation, lost profits, even if advised of the possibility of such damages

TABLE OF CONTENTS

TABLE OF TABLES.....	3
TABLE OF FIGURES	3
TABLE 1: MODIFICATION RECORD	4
ACKNOWLEDGMENTS	5
INTRODUCTION.....	6
COMMON PROCEDURES FOR CXP AND QSFP+ CABLE TESTING	6
CONNECTING THE DIOLAN U2C-I2C ADAPTER	9
MEMORY MAP APPLICATION SETUP	10
I2C EXPLORER APPLICATION	10
UPPER PAGE 03	10
AARDVARK I2C/SPI ADAPTER.....	11
MMPARSER APPLICATION	12
TEST PROCEDURES.....	14
MEMORY MAP VALIDATION SPREADSHEET PROCEDURE	14
TEST 7.4.01 – QSFP MEMORY MAP VERIFICATION TEST PROCEDURE.....	15
TEST 7.4.02 – CXP MEMORY MAP VERIFICATION TEST PROCEDURE	16
TEST 7.4.03 – QSFP MODULE SELECT TEST PROCEDURES	17
TEST 7.4.04 – MECHANICAL SPEC VALIDATION.....	18

Table of Tables

Table 1: Modification Record.....	4
-----------------------------------	---

Table of Figures

Figure 1 - QSFP+ Mechanical Specification for the connector – Page 25 of the SFF-8436 Rev 3.6	18
Figure 2 - CXP Mechanical Specification for the connector – Page 32 of the CXP Spec Rev 1.1.....	19
Figure 3 - Shielded Free (Plug) Integrated Cable Connector.....	19

Table 1: Modification Record

Revision	Release Date	Author	Comments
0.10	2/21/2011	Cliff Cole	<ul style="list-style-type: none"> Created the ModSel Test
0.11	2/22/2011	Llolsten Kaonga	<ul style="list-style-type: none"> Added CXP Memory Map Test Added QSFP Memory Map Test
0.12	3/15/2011	Llolsten Kaonga	<ul style="list-style-type: none"> Incorporated the MemoryParserNotes-2011-02-15 document into the MOI
0.13	3/16/2011	Rupert Dance	<ul style="list-style-type: none"> Review and update the content and structure of the document Added Test 7.4.04 – Mechanical Spec Validation
0.14	3/18/2011	Rupert Dance	<ul style="list-style-type: none"> Corrected the conversion of Newton to foot lbs of force
0.15	3/21/2011	Rupert Dance	<ul style="list-style-type: none"> Editorial Updates to the ModSel Test Procedure as per Cliff Cole Fix typo on QSFP width in 7.4.04 – Catalin Muntean
0.16	3/23/2011	Rupert Dance	<ul style="list-style-type: none"> Added additional discussion in procedure 7.4.01 after updates were made to the QSFP Memory Map Validation spreadsheet
0.17	09/28/2011	Llolsten Kaonga	<ul style="list-style-type: none"> Added device driver information for USB-I2C Adapter
0.18	10/11/2011	Llolsten Kaonga	<ul style="list-style-type: none"> Added instructions on how to update device driver for the USB-I2C Adapter
0.19	12/13//2011	Llolsten Kaonga	<ul style="list-style-type: none"> Added instructions for CXP cables in the I2C Explorer common setup procedure
0.20	03/20/2012	Llolsten Kaonga	<ul style="list-style-type: none"> Updated the section on the Diolan USB-I2C Adapter to include firmware installation
0.21	03/21/2012	Llolsten Kaonga	<ul style="list-style-type: none"> Reviewed with Rupert and brought document into synch with PF21
0.22	08/28/2012	Llolsten Kaonga	<ul style="list-style-type: none"> Added instructions on how to read Upper Page 03 data
0.23	09/26/2012	Llolsten Kaonga	<ul style="list-style-type: none"> Updated instructions on DLN-2 and I2CExplorer installer
0.24	02/12/2013	Llolsten Kaonga	<ul style="list-style-type: none"> Updated the Common Procedures section and fixed some broken references
0.25	01/24/2014	Llolsten Kaonga	<ul style="list-style-type: none"> Updated I2CExplorer installation procedure and some obsolete test steps
0.26	07/28/2014	Llolsten Kaonga	<ul style="list-style-type: none"> Fixed some minor typos and updated document for PF26
0.27	04/17/2015	Llolsten Kaonga	<ul style="list-style-type: none"> Updated the Diolan USB-I2C adapter driver update procedure
0.28	08/17/2015	Llolsten Kaonga	<ul style="list-style-type: none"> Updating the MOI for PF28
0.29	01/15/2016	Llolsten Kaonga	<ul style="list-style-type: none"> Updating the MOI for PF29
0.30	08/09/2016	Llolsten Kaonga	<ul style="list-style-type: none"> Update the MOI for PF30
0.31	02/15/2017	Llolsten Kaonga	<ul style="list-style-type: none"> Updated the MOI for PF31
0.32	07/14/2017	Llolsten Kaonga	<ul style="list-style-type: none"> Updated the MOI for PF32
0.33	02/12/2018	Llolsten Kaonga	<ul style="list-style-type: none"> Updated the MOI for PF33
0.33.05	03/27/2018	Llolsten Kaonga	<ul style="list-style-type: none"> Updated the Aardvark API steps for C and C# builds
0.34	07/23/2018	Llolsten Kaonga	<ul style="list-style-type: none"> Updated the MOI for PF34

ACKNOWLEDGMENTS

The IBTA would like to acknowledge the efforts of the following individuals in the development of this test suite.

QLogic/Intel

- Cliff Cole
- Brent Rothermel

Intersil Corporation

- Peter Baughman

Software Forge

- Llolsten Kaonga
- Rupert Dance

Introduction

This MOI describes the standard CXP and QSFP memory map validation procedures. The spreadsheets titled [QSFP-Memory Map Validation.xlsx](#) and [CXP-Tx-Rx-Upper-Page-00h-Memory Map Validation.xlsx](#) are used as references in these procedures. The content of the memory map is obtained by using an I2C-USB reader. The output of the memory map is then copied and pasted into the spreadsheets where specific memory map addresses are verified for accuracy. *These spreadsheets and/or the formulas therein may not work with older versions of Microsoft Excel.*

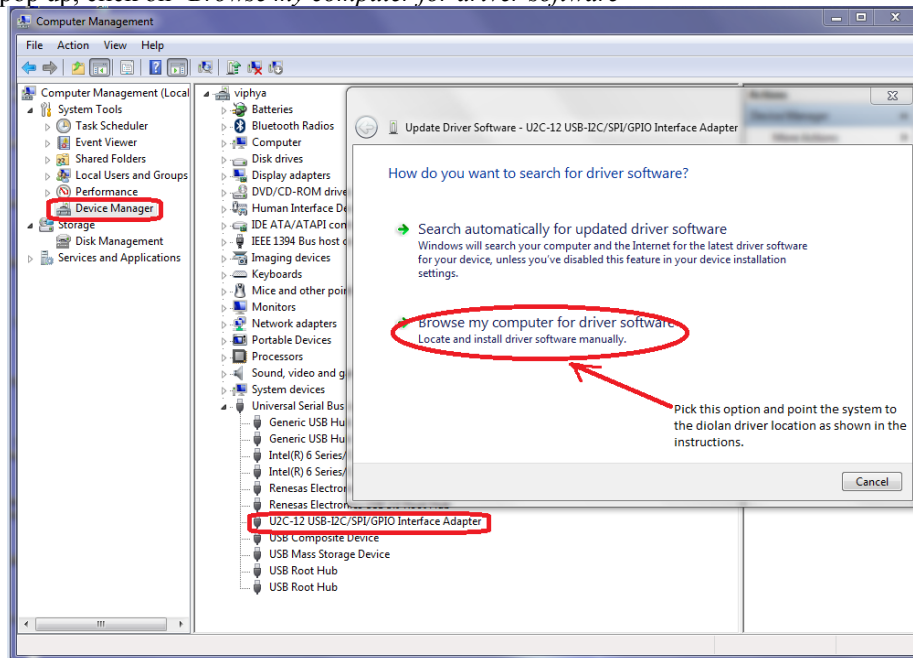
Common Procedures for CXP and QSFP+ Cable Testing

If you plan to use the I2CExplorer application to read the Memory Map data, you need to either use the FTDI adapter or the Diolan USB-I2C adapter. You will need to install drivers for either of these adapters. The latest FTDI CDM drivers may be found here <http://www.ftdichip.com/FTDrivers>.

You may also need to build or make sure you have the mmparser.exe if you need an application to read binary data.

If you are going to use the Diolan USB-I2C device, then please make sure you have installed the appropriate device drivers. Here is the procedure for updating the Diolan USB-I2C adapter driver

1. Connect the Diolan device to your laptop via a USB cable
2. Go to <http://dlnware.com/Downloads>
3. Download DLN.3.0.2.exe(zip) or the latest version of the DLN executable and install it. This executable has the drivers and other apps
4. Go to My Computer→Device Manager→USB Controller and locate U2C-12 USB-I2C/SPI/GPIO Interface Adapter
5. Right click on the device and select *Update Driver*
6. In the next pop up, click on "*Browse my computer for driver software*"



7. Point it to *C:\Program Files\Diolan\I2CBridge\Driver\x64 (x86)* (or a similar path to the driver location on your system) then click Next.
8. A dialog box may pop up asking you to confirm that you are OK installing the driver from Diolan.
9. The update should start and when it completes, your computer should be able to discover the Diolan adapter.
10. Once the update is completed, the status light at D2 (pointed to by arrow in **Figure 1**) should be flashing. This means that your computer is able to discover the Diolan adapter.

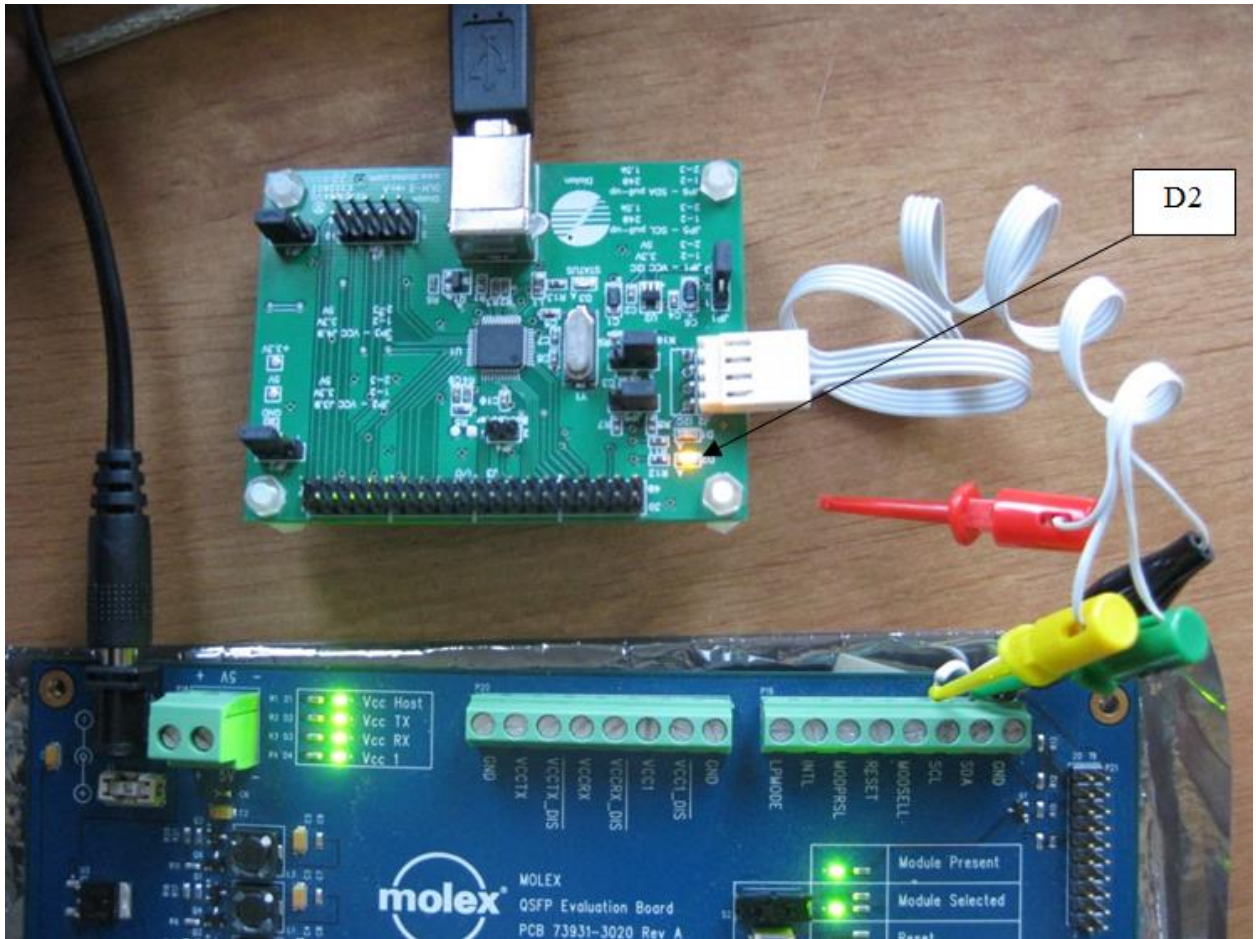






Figure 1

It is our experience that some USB ports do not work well. If you are unable to see the device on one USB port, it does not necessarily mean you cannot see it on all USB ports. You may wish to try a different USB port. Also, you may have the D2 LED flashing even when the driver is not installed properly. Please make sure that the folder *C:\Program Files\Diolan\I2CBridge\Driver* (or similar location) has the following files in it:

-  i2cbrdg.dll
-  I2cBrdg.inf
-  i2cbrdg.sys
-  I2cBrdgf.SYS

If you have old installations of the pre DLN-2 device drivers, that is OK too. You can skip step (1) and just do an update. We have confirmed that this works as well.

Note: If the *Status* light (D3) flashes but the light at D2 is OFF, then your adapter has the DLN-2 firmware and you will need to convert it to U2C-I2. To make the conversion, you need to burn new firmware on the adapter:

1. Download the firmware (you may have to contact Diolan <<http://www.diolan.com/home/contact.html>> for a link to download or have them send you the firmware)
2. Make sure you have the files *dlm.dll* and *i2cbrdg.dll* in the same location as the *dlm_2_u2c.exe* firmware executable, or whatever file Diolan gives you, or the conversion will not work. *dlm_2_u2c.exe* will complain that these files are missing and it will not run if you place them in a different location. You can get the two DLLs from

wherever location you installed the DLN-2 Adapter. The dln.dll will be located in the Diolan\DLN\bin directory while the i2cbrdg.dll will be located in the Diolan\I2CBridge.x** folder.

3. Close the JP4 jumper ← you can use the switch from the JP5 jumper to do this.
4. Plug in the board to a USB port on your computer
5. Open the JP4 jumper ← you can return the switch back to the JP5 jumper at this point
6. Run the dln_2_u2c.exe ← this takes just a few seconds

You do not need to close the JP4 jumper if all you need to do is update the firmware. You just need to plug in the board to a USB port and run dln_2_u2c.exe and update the firmware.

Connecting the Diolan U2C-I2C adapter

You must connect the mini-hooks and the mini-crocodile connectors to the correct terminal on the board. Connect the SDA mini-hook to the SDA terminal, the SCL mini-hook to the SCL terminal. The black mini-crocodile connector is the ground and must be connected to the ground terminal on the board. Please see the picture in **Figure 1**.

You are now ready to install the I2CExplorer.

1. Unpack the setup files
2. Click on the I2CExplorer-Setup executable.
3. If your system does not already have the **Microsoft.NET Framework Client Profile** installed, setup will ask you if you would like to install it. You *must* install this or already have it on your system for I2CExplorer to work. You can either go on the web and get .NET or close the setup, click on the MSI and install the .NET. You may get an error but just close this then click setup again to complete the I2CExplorer installation.

Memory Map Application Setup

I2C Explorer Application

Purpose: Common setup procedure to read cable memory data for CXP and QSFP cable assemblies.

Test Procedure

1. Unpack the I2C Explorer Tool zip file and install it on your system
2. The default installation directory is C:\Program Files (x86)\CIWG\I2CExplorer
3. Start the I2C Explorer GUI (it is in the bin directory)
4. In the GUI under the **Device** menu, please choose the correct device (either *Devantech USB-I2C* or *Diolan U2C-I2*)
5. Select the appropriate COM port the adapter connects to (Serial number if using the Diolan U2C-I2 adapter)
6. Click on Connect/Disconnect button to connect or disconnect
7. Select Address **160** if it is not the default address. For **CXP** cables, you will need to also read from address 168. This may not apply to passive cables but may apply to active cables. If you are using the Molex boards for active **CXP** cables, you must set the current to some non-zero value (about 0.5 amps) or you won't be able to read anything.
8. Select the **Single-byte Read** radio button
9. Make sure you have valid data (you can click on a byte address to check that you do have valid data. For example byte 127 should have **00h**). If the cable or program is incompatible, you may see just **FFh**
10. If the program returns valid data from the cable, click **File**→**Save as** and type in the file name to save the binary or text file
11. Use the mmparser application or equivalent program to read the binary data. You do not need this if you saved your data as text in step 10.

Upper Page 03

Purpose: To verify that cable assemblies properly implement Upper Page 03 (**Figure 2**)

Test Procedure

1. Start the I2C Explorer GUI (See preceding procedure)
2. Connect to the cable
3. Check that **Byte 2, bit 2** is low (0). If this bit is 0, then Upper Page 03 *must* be implemented
4. Click the **Page Select Byte (Byte 127)**.
5. Set the Page Select Byte to **03**. This is under the Hex Value column in the top right hand side of the GUI as shown in the screen shot below.

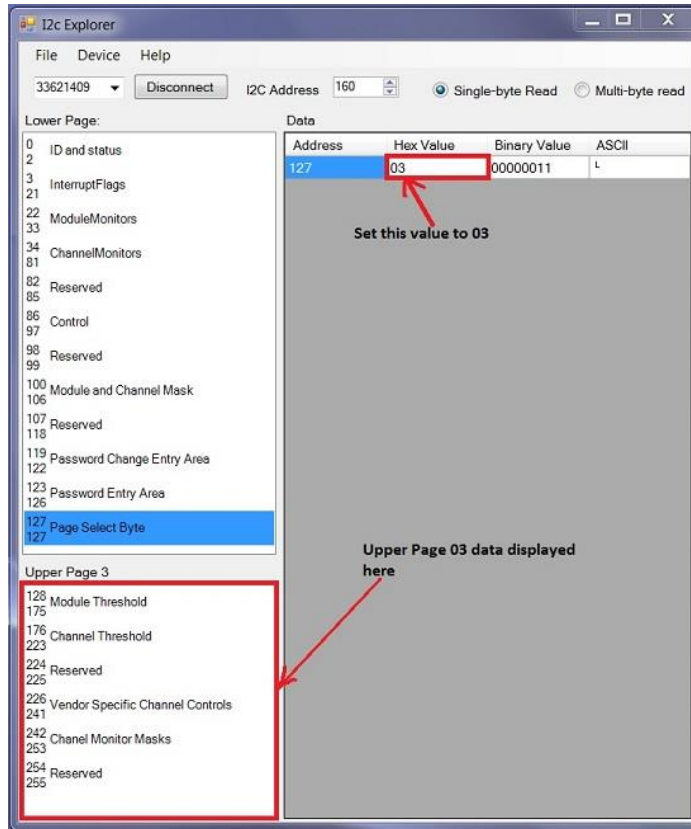


Figure 2

- The Upper Page 03 data is displayed in the bottom left textbox. If you click on any of the fields in this textbox, the details will be displayed in the main (grey) box on the right hand side.

Aardvark I2C/SPI Adapter

Purpose: Common setup procedure to read cable memory data for CXP and QSFP cable assemblies using the Total Phase Aardvark I2C/SPI Adapter.

You may use the Total Phase Aardvark I2C/SPI Adapter to read the memory map data. The data may not be presented to you in the same form as that from the I2C Explorer Application. So you may need to do additional processing to get it into the correct form to allow you to use the Memory Map Validation Spreadsheet.

You need to install the [Windows driver](#) if you use a Windows system or the [Linux driver](#). Please review the appropriate user's manual ([Windows](#) or [Linux](#)) for more information about the drivers and their installation. Download the [Aardvark Software API](#) (current version 5.30) appropriate for your system.

Building the API Executables

You need to build the API executables for your system.

To build the C executables,

- cd to the directory where you unpacked the API and open the Visual Studio solution *Makefile.sln* located in the directory "c:\MSVC2008"
- aadetector** should be the *Startup Project* if it is not, set it as the *Startup Project*
- Right click on the projects and select "Properties" to set up your projects, including the location where you want to save your executables

- 4) Run the “Build Solution” command
- 5) Copy the aardvark.DLL to the location where you saved your executables in step 3 above.

To build C# executables,

- 1) cd to the directory where you unpacked the API zip file and open the solution “*csharp\VisualStudio\Examples.sln*”
- 2) **aadetct** should be set as the *Startup Project*
- 3) Set the project properties as appropriate or use the default API settings
- 4) Run the “Build Solution” command
- 5) Copy the aardvark.DLL to the directory where your executables are located.

Connect the Aardvark I2C/SPI adapter via a USB cable to your computer. Connect the Aardvark crocodile connector to the GND terminal on the Molex PCB board; connect the SDA mini-hook to the SDA on the PCB and the SCL mini-hook to the SCL on the board.

- 1) Start a DOS command window
- 2) Right click the top of the DOS window and select “*Properties*”. Make sure that “*QuickEdit Mode*” is checked. This enables you to highlight and copy the memory map data from your DOS window
- 3) cd to the location of your API executables
- 4) invoke “*aadetct*” ← this detects the device and provides a port for you to use. You should see output similar to the lines below

```
D:\aardvark-api-windows-x86_64-v5.30\c\_output>aadetct
1 device(s) found:
  port=0 (avail) (2238-301884)
```

- 5) Invoke the command “*aaai2c_eeeprom 0 400 read 80 0 256*”. This says read from the device at port **0**, the first 256 bytes starting at offset **0** from slave address **80**. The 400 is the bitrate. You should now see output similar to this:

```
Bitrate set to 400 kHz
Bus lock timeout set to 150 ms

Data read from device:
0000: 0d 06 06 00 00 00 00 00 00 00 00 00 00 00 00 00
0010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0060: 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 04
0070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0080: 0d 00 23 88 00 00 00 00 00 00 00 00 00 ff 00 00 00
0090: 00 00 01 a0 4d 65 6c 6c 61 6e 6f 78 20 20 20 20
00a0: 20 20 20 20 1f 00 02 c9 4d 43 50 31 36 30 30 2d
00b0: 45 30 30 41 20 20 20 20 41 32 02 03 04 06 00 3e
00c0: 0b 00 00 00 4d 54 31 35 33 31 56 53 30 33 31 32
00d0: 33 20 20 20 31 35 30 39 30 31 20 20 00 00 67 4f
00e0: 31 36 33 30 35 32 35 37 33 31 53 43 00 00 00 00
00f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 30 00 00
```

- 6) You can now copy the memory map data to the Memory Map Validation spreadsheet (See [Test Procedures](#) below). Just highlight it and press “Enter” to copy it since we enabled copying in the DOS window in Step 2 above.

MMParser Application

Purpose: This application converts the binary memory map data so it can be copied to the validation spreadsheet. If you saved your memory map data as text, then you do not need this.

Data Conversion Procedure

- 7) Start a DOS command window
- 8) CD to the location of the mmparser.exe script
- 9) Type the command **mmparser -qsfp+ -d binFileName textFileName**
 - a) **TextFileName** is the name of the text file you want the data to be saved to

- b) ***binFileName*** is the name of the binary file saved previously with the I2c Explorer application
 - c) ***“-qsfp+”*** is the option you use if you want to analyze a QSFP cable.
 - d) ***“-cxp”*** is the option you use if you want to analyze a CXP cable.
 - e) ***–d*** if you would like to see dynamic data
- 10) You do not need Step 3 if you saved your data as text files. You will just need to open the files and copy and paste the data to the validation spreadsheet. In fact, you do not need the mmparser if you saved your data as text files.
- 11) Typing ***“mmparser –h”*** will give you the usage message and the options you have available to use.

Test procedures

Memory Map Validation Spreadsheet Procedure

Purpose: This spreadsheet is used to analyze and validate the memory map data

Test Procedure

1. Open the QSFP or CXP Memory Map Validation spreadsheet
2. Copy the text data only at the bottom of the DOS window as discussed in the [MMParser Application](#) (or directly from the folder where you saved the memory map data if you saved it in text format) and paste it in the Memory Map Validation spreadsheet “Conversions” tab. It should go in the section which says “Binary data to be pasted in”, beginning with the cell next to byte 0. The data you need to copy should look similar to this:

```
0D 00 02 1F 00 00 00 00 FF 00 00 00 00 00 00 00
00 00 00 FF FF FF 11 92 00 00 7E EE 00 00 FF FF
FF FF 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 FF FF 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0D 00 23 04 01 30 00 40 40 06 E0 00 64 00 00 00
00 00 0A 00 53 61 6D 74 65 63 20 49 6E 63 20 20
20 20 20 20 07 04 C8 80 51 53 46 50 4F 20 20 20
20 20 20 20 20 20 20 20 30 31 42 68 0F A0 46 CC
00 00 04 98 30 30 30 30 30 30 30 30 30 30 30 30
30 30 30 32 31 31 30 31 31 31 30 30 00 00 00 23
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF 00 01 0E
```

3. After step 2 is completed, go to the DUTs tab
 - a. Select the appropriate Company Name
 - b. Select the Cable Type
 - c. If you are testing a CXP cable, make the appropriate choice about whether the cable is a hybrid or not.
 - d. Enter the DUT number in the Cable Values column
4. Copy the information from the Temp column and paste it under the column where you just entered the DUT number in step 3. You must select **Paste** → **Paste Values** and not **Paste** → **Formulas**. This must be done as a Paste Special and not as a standard Paste.
5. Once you have pasted the values as indicated in step 4, please check for failures and record them as appropriate in the *Issues* tab.
6. **Note:** Fractional values ending in .50 or greater should be rounded up whereas values ending in less than .50 should be rounded down. For example a 3.5 meter cable would be listed as a 4 meter cable.
7. Please make sure you enter the **Module Selected** test result at the bottom of the spreadsheet.
8. You may see errors if you paste your data for just one end of the cable. Please paste data for the other end of the cable to confirm the errors. Some fields, such as cable length, have been programmed to check that both ends of the cable are programmed with the same information. This is why you need to paste data for both ends.

Possible Problems:

1. Make sure the Vendor Name in the Memory Map is 16 characters long – the name must be padded if it is shorter. So if a DUT fails in the Vendor Name field, it may be that the name is not padded if it otherwise looks correct.
2. We are now allowing more than one vendor name for vendors who have submitted requests for multiple names. The validation spreadsheet takes this into account for the most part so you (mostly) do not need to worry except if a failure occurs.

Test 7.4.01 – QSFP Memory Map Verification Test Procedure

Purpose: To verify that QSFP cable assemblies are compliant with the [SFF-8436](#) and [InfiniBand QSFP specifications](#)

References:

- [IBTA Volume 2 Specification](#)
- [QSFP-Memory Map Validation.xlsx](#)

Resource Requirements:

- MOLEX QSFP Evaluation Board PCB 73931-3020

Discussion:

- Please refer to the [Common Procedures](#) section for system requirements and setup procedure details.
- Byte 164: On 9/22/10 the CIWG passed the following **Motion**:
Address 164 indicates the InfiniBand data rates that the cable assembly is intended for. This does not imply that the particular cable assembly will pass the InfiniBand Compliance tests for that particular data rate. It indicates which data rates are intended to be supported for interoperability, especially in homogeneous environments where the cables are used with one device vendor's equipment. All bits are to be set for the data rates that are appropriate. The SDR bit must always be set to a 1 because SDR is used to establish the link. For example, a QDR cable could be set to 101 indicating QDR and SDR, but not DDR compatibility or it could be set to 111 indicating QDR, DDR and SDR compatibility. This bit is checked at IBTA Plugfests for consistency with the data rates for which the vendor submitted the cable. It is not being used by the device vendors to determine the appropriate link speeds of the cable.
- Bytes 186 & 187: On 8/5/2010 the CIWG decided to leave the difference between measured IL and the reported Attenuation to +/- 2.0 dB for Plugfest 18
- Bytes 184-185: On 10/6/10 the CIWG passed the following **Motion**:
IBM submits a motion that a cable with the Revision number field being an Alpha-numeric field, of ASCII characters (including ASCII code for '0') , would be considered a compliant Rev. number. Software Forge makes a motion that the CIWG does not have to check for deltas in Rev field between Plugfests (but recommends that vendors increment this field when making changes to the connector/cable).

Test Setup:

Set up the cable assembly to be tested according to the [Common Procedures](#) section. This requires the use of the MOLEX QSFP Evaluation Board PCB 73931-3020. Make sure the Molex board is connected to its power source.

Test Procedure:

1. Plug in the cable under test to the QSFP Evaluation Board
2. Perform a QSFP Read of the assembly under test as described in the [I2C Explorer Application](#) section.
3. It is assumed that the QSFP read was successful – if not the problem should be debugged.
4. If the QSFP read was successful, save the data to a file for analysis with the validation spreadsheet
5. Connect the other end of the cable to the Evaluation Board and repeat steps 2-4
6. Repeat steps 1-5 for each cable
7. Copy the hex data into the validation spreadsheet as described in the [Memory Map Validation Spreadsheet Procedure](#) for each cable end and verify that the data is valid. If there are failures, note them.

Observable Results:

1. Both ends of each cable's memory map data should be valid and compliant with respect to the SFF-8436 QSFP Specification as updated by the IBTA and recorded in the tab SFF8436-IBTA of the [QSFP-Memory Map Validation](#) spreadsheet.

Possible Problems:

1. The Molex evaluation board needs to be powered up.
2. In the past, we have found that some active fiber cables might fail in some fields if the read is done too quickly after the cable is connected to the evaluation board.
3. If there is a failure with the dynamic data, please try the read again after waiting for a few seconds.

Test 7.4.02 – CXP Memory Map Verification Test Procedure

Purpose: To verify that CXP cable assemblies are compliant with respect to the CXP Specification

References:

- [CXP Specification \(IBTA Volume 2 Spec\)](#)
- [CXP-Tx-Rx-Upper-Page-00h-Memory Map Validation.xlsx](#)

Resource Requirements:

- MOLEX CXP Evaluation Board

Discussion:

Please refer to the [Common Procedures](#) section for system requirements and setup procedure details.

Test Setup:

Set up the assembly to be tested according to the [Common Procedures](#) section. This requires the use of the MOLEX CXP Evaluation Board. Make sure the Molex board is connected to its power source. The CXP Evaluation Board requires a power supply with 3.3V.

Test Procedure:

1. Plug in the cable under test to the CXP Evaluation Board
2. Perform a CXP Read of the assembly under test as described in the [I2C Explorer Application](#) section.
 - a. Depending on the bits in Byte 127, you may need to read from both address 160 and address 168
3. It is assumed that the CXP read was successful – if not the problem should be debugged.
4. If the CXP read was successful, save the data to a file for analysis with the validation spreadsheet
5. Connect the other end of the cable to the Evaluation Board and repeat steps 2-4
6. Repeat steps 1-5 for each cable
7. Copy the hex data into the CXP validation spreadsheet as described in the [Memory Map Validation Spreadsheet Procedure](#) for each cable end and verify that the data is valid. If there are failures, note them.

Observable Results:

1. Both ends of each cable's memory map data should be valid and compliant with respect to the [IBTA Volume 2 Specification](#)

Possible Problems:

1. If the read to address 168 fails, it may be that that address is not used by the cable. Check the bits in byte 127 before failing the cable for this read failure.

Test 7.4.03 – QSFP Module Select Test Procedures

Purpose: To verify that QSFP cable assemblies are compliant with the section “4.1.1.1 ModSel” of the SFF-8436 QSFP specification

References:

- [IBTA Volume 2 Specification](#)
- [QSFP-Memory Map Validation.xlsx](#)

Resource Requirements:

- MOLEX QSFP Evaluation Board PCB 73931-3020

Discussion:

The following procedure is an extension to the QSFP memory map procedures. The “Module Selected” switch (S2) on the Molex QSFP Evaluation Board (PCB 73931-3020) is disabled and the QSFP memory map is then read. When the module select is implemented on the QSFP assembly as described in the [IBTA Volume 2 Specification](#), the QSFP memory map will no longer read, as the module will now be “disabled”. If the memory map still can be read, then module select feature is not implemented correctly on the assembly or there is a hardware or firmware error causing a problem.

Test Setup:

Set up the cable assembly to be tested according to the Common Procedures section for verifying the QSFP memory map. This requires the use of the MOLEX QSFP Evaluation Board PCB 73931-3020. Make sure the Molex board is connected to its power source.

Test Procedure:

1. Plug in the cable under test to the QSFP Evaluation Board
2. Locate the “Module Select Switch” (S2) and LED (D6) on the MOLEX QSFP evaluation board
3. Make sure that S2 is in the enable position which is indicated by the LED D6 being lit
4. Perform a QSFP Read of the assembly under test as described in the Memory Parser Notes
5. It is assumed that the QSFP read was successful – if not the problem should be debugged
6. After a successful QSFP memory map read has been accomplished proceed to next step
7. Set the module select switch (S2) to the disable position. Verify that LED D6 is no longer illuminated
8. Use the same procedure as above to attempt a QSFP Memory Map read.
9. The expected outcome is for the memory map read to fail. The QSFP Memory EEPROM has been disabled so no output should be displayed.
10. Depending on the I2C-USB reader, this may be indicated by an error message, all bytes read as 0xFF or all bytes read as 0x00.
11. If the QSFP Memory Map is successfully read with the switch S2 disabled, then the test FAILED
12. Mark this test as a failure or a pass depending on the outcome from steps 8-10.
13. If the Memory map does not read as expected:
14. Set S2 to enabled
15. Re-try the QSFP Memory map read to verify the assembly is still good
16. If there is doubt repeat the procedure with the S2 enabled, then disabled then enabled again
17. When the test is complete put Switch S2 Module Selected back to the enabled position LED D6 illuminated
18. It would be a good idea to verify a QSFP Memory map read with a known good assembly before the next cable is tested.

Observable Results:

1. When S2 Module Selected switch is enabled the QSFP memory map should read out successfully.
2. When S2 Module Switch is disabled (LED D6 off) then the QSFP memory map should NOT read.

Possible Problems:

1. The S2 switch could be left in the disabled position before or after the test.
2. The Molex evaluation board needs to be powered up

Test 7.4.04 – Mechanical Spec Validation

Purpose: To verify that CXP and QSFP cable assemblies are compliant with the module mechanical specifications

References:

- [IBTA Volume 2 Specification](#)
- [Force Conversion](#)

Resource Requirements:

- Digital Calipers and/or micrometers
- Mechanical Force Gauges - Ametek [Chatillon® LG Series](#)

Discussion:

This test procedure insures that cables meet the mechanical dimensions in the appropriate Specification. This includes but is not limited to measuring the width and the height of the connector and the force required to insert and remove the connector.

Test Procedure:

1. For QSFP+ cables measure each end of the cable - Figure 3 - QSFP+ Mechanical Specification for the connector – Page 25 of the SFF-8436 Rev 3.6
 - a. The height of the connector must be 8.5 mm +/- .10 mm
 - b. The width of the connector must be 18.35 mm +/- .10 mm

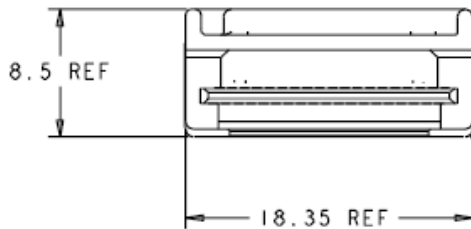


Figure 3 - QSFP+ Mechanical Specification for the connector – Page 25 of the SFF-8436 Rev 3.6

2. For CXP cables measure each end of the cable.
 - a. The height of the connector (A02) must be 9.81 mm +/- .13 mm
 - b. The width of the connector (A01) must be 21.20 mm +/- .13 mm

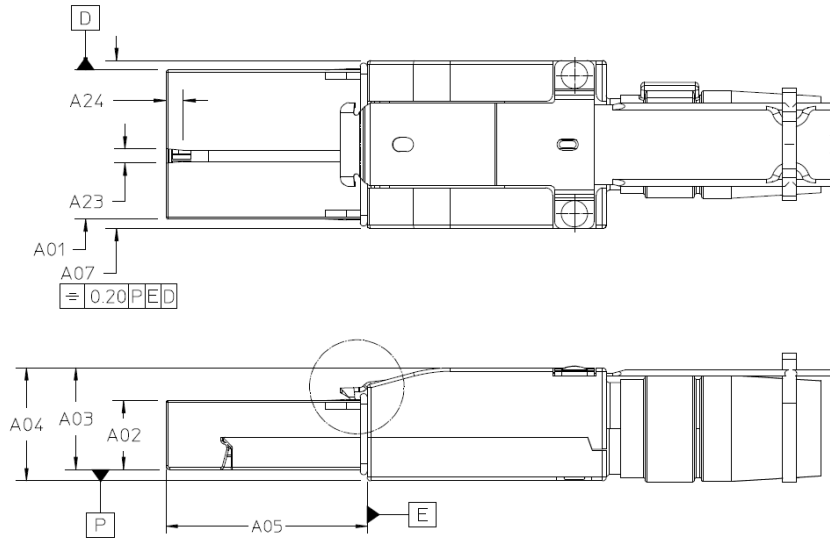


Figure 4 - CXP Mechanical Specification for the connector – Page 32 of the CXP Spec Rev 1.1

	Description	Dim.	Tol.		Description	Dim.	Tol.
A01	Snout Width	21.20	0.13	A13	Base of snout to top of 1 st paddle card	2.99	0.20
A02	Snout Thickness	9.81	0.13	A14	Top of 1st to top of 2 nd card	4.50	0.10
A03	Snout Bottom to Plug Top	14.60	Max	A15	Plug Body (Datum E) to short pad (Datum C)	26.67	0.20
A04	Plug Body Thickness	16.21	Max	A16	Tongue Width - Base	16.95	0.10
A05	Snout Length	28.45	0.13	A17	Tongue Width - Tip	16.10	0.20
A06	Plug Body (Datum E) to Latch	1.74	0.13	A18	Length of Tongue - Straight Section	1.80	0.10
A07	Body Width	24.05	Max	A19	Length of Tongue	6.00	Min.
A08	Barb Lead-in Height	1.14	0.10	A20	Inside Width of Snout	20.00	0.05
A09	Barb Lead-in Angle	45°	1°	A21	Inside Height of Snout	8.60	0.25
A10	Barb Length	2.05	0.10	A22	Tongue Thickness	0.60	Ref.
A11	Plug Overall Body Length	62.00	Ref	A23	Orientation Key Lead-In Width	2.00	0.25
A12	Latch Width	11.90	0.10	A24	Orientation Key Lead-in Length	2.40	0.25

Figure 5 - Shielded Free (Plug) Integrated Cable Connector

3. For QSFP+ cables
 - a. Measure insertion force and verify that it does not exceed 40 N (9 lbf.)
 - b. Measure extraction force and verify that it does not exceed 30 N (6.75 lbf.)
4. For CXP cables
 - a. Measure insertion force and verify that it does not exceed 150 N (33.72 lbf.)
 - b. Measure extraction force and verify that it does not exceed 50 N (11.24 lbf.)