

# ***MicroTCA™***

---

## **PICMG® MTCA.4 PCI Express Hot Plug Design Guide**

**MTCA\_DG.0 R1.0**

**January 9, 2017**



**Open Modular  
Computing Specifications**



## DISCLAIMER:

This design guide is not a specification. It contains additional detail information but does not replace any applicable PICMG MTCA.4 specifications.

For complete guidelines on the design of MTCA.4 compliant boards and systems, refer also to the full specification – do not use this design guide as the only reference for any design decisions.

© Copyright 2017, PCI Industrial Computer Manufacturers Group. The attention of adopters is directed to the possibility that compliance with or adoption of PICMG® specifications may require use of an invention covered by patent rights. PICMG® **shall** not be responsible for identifying patents for which a license may be required by any PICMG® specification or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. PICMG® specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

## NOTICE:

The information contained in this document is subject to change without notice. The material in this document details a PICMG® specification in accordance with the license and notices set forth on this page. This document does not represent a commitment to implement any portion of this specification in any company's products.

WHILE THE INFORMATION IN THIS PUBLICATION IS BELIEVED TO BE ACCURATE, PICMG® MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL INCLUDING, BUT NOT LIMITED TO, ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR PARTICULAR PURPOSE OR USE.

In no event **shall** PICMG® be liable for errors contained herein or for indirect, incidental, special, consequential, reliance or cover damages, including loss of profits, revenue, data or use, incurred by any user or any third party. Compliance with this specification does not absolve manufacturers of equipment from the requirements of safety and regulatory agencies (UL, CSA, FCC, IEC, etc.).

## IMPORTANT NOTICE:

This document includes references to specifications, standards or other material not created by PICMG. Such referenced materials will typically have been created by organizations that operate under IPR policies with terms that vary widely, and under process controls with varying degrees of strictness and efficacy. PICMG has not made any enquiry into the nature or effectiveness of any such policies, processes or controls, and therefore ANY USE OF REFERENCED MATERIALS IS ENTIRELY AT THE RISK OF THE USER. Users should therefore make such investigations regarding referenced materials, and the organizations that have created them, as they deem appropriate.

PICMG®, CompactPCI®, AdvancedTCA®, AdvancedTCA® 300, ATCA®, ATCA® 300, CompactPCI® Express, COM Express®, SHB Express®, and the PICMG, CompactPCI, AdvancedTCA,  $\mu$ TCA and ATCA logos are registered trademarks, and MicroTCA™, xTCA™, AdvancedMC™, IRTM™ and the IRTM logo are trademarks of the PCI Industrial Computer Manufacturers Group. All other brand or product names **may** be trademarks or registered trademarks of their respective holders.

---

# Contents

Contents .....	iii
<b>1 Introduction.....</b>	<b>1</b>
1.1 Objectives.....	1
1.2 Reference Documents.....	1
1.2.1 Reference Specifications.....	1
1.2.2 Environmental and Regulatory Documents.....	2
1.3 Contributors.....	2
1.4 Special word usage .....	2
1.5 Name and logo usage.....	2
1.6 Signal naming conventions.....	3
1.7 Intellectual property.....	4
1.7.1 Necessary Claims (referring to mandatory or recommended features):.....	4
1.7.2 Unnecessary Claims (referring to optional features or non-normative elements):.....	4
1.8 Acronyms & Definitions.....	5
<b>2 Operating Module Hardware-Software Components .....</b>	<b>6</b>
2.1 Introduction .....	6
2.2 PCI Express Hot Plug.....	6
2.3 MTCA Specific Hot Plug Environment and Hardware Components.....	9
2.4 Software Elements .....	10
2.5 Operating Models.....	11
2.6 Device Removal Procedure.....	11
2.7 Device Insertion Procedure.....	14
2.8 Soft Hot Plug.....	16
<b>3 Usage Models.....</b>	<b>17</b>
3.1 Introduction .....	17
3.2 Hot Plug Interaction of the Device Driver and User Application.....	18
<b>4 RTM Hot Plug.....</b>	<b>20</b>
<b>5 Using PCI Express Switches on the AMC .....</b>	<b>22</b>
<b>6 Revision History .....</b>	<b>24</b>
<b>7 Table of Requirements.....</b>	<b>25</b>

## Figures

Figure 2-1: PCI Express Hot Plug Hardware Elements.....	7
Figure 2-2: MTCA Specific Hot Plug Event & Hardware Components .....	9
Figure 2-3: Device Removal Procedure.....	12
Figure 2-4: Device Insertion Procedure .....	14
Figure 3-1: Linux Device Model and Generation of the /sys and /dev files.....	17
Figure 3-2: User Application Response to Hot Plug .....	19
Figure 4-1: Example Usage of RTM Status Register .....	20
Figure 5-1: PCI Express Bus numbering for case of AMC module with PCIe Switch .....	22
Figure 5-2: PCI Express Bus numbering after removing AMC module with PCIe Switch (c.f. Fig. 5-1).....	23

## Tables

Table 1-1: Acronyms .....	5
Table 2-1: PCI Express Configuration Registers Used for Hot Plug.....	8
Table 2-2: Major Hardware Elements for PCI Express Hot Plug in MTCA .....	8
Table 2-3: Major software elements required to support Hot Plug capability.....	10
Table 2-4: Device Removal Procedure .....	13
Table 2-5: Device Insertion Procedure .....	15

# Introduction

# 1 Introduction

## 1.1 Objectives

This guideline defines standard operating and usage models and the Application Programming Interface (API) of the MTCA.4 PCI Express hot plug implementation. The guideline defines a standard usage model for all devices and software components that support hot plug capabilities for both AMCs and RTMs (Rear Transition Modules).

## 1.2 Reference Documents

¶1 This guideline defines standard operating and usage models and the Application Programming Interface (API) of the MTCA for Physics PCI Express hot plug implementation. The guideline defines a standard usage model for all devices and software components that support Hot Plug. The publications cited in this section are relevant to this specification. Most of the specifications referred to are subject to periodic and independent updates, and are the responsibility of their respective organizations. The reader is advised to check carefully the version or revision of the referenced specification that is to be used in conjunction with this document

### 1.2.1 Reference Specifications

¶2 All documents may be obtained from their respective organizations.

<b>Document</b>	<b>Organization</b>	<b>Contact information</b>
PICMG 3.0 R3.0	PICMG	
PICMG AMC.0 R2.0	PICMG	
PICMG IRTM.0 R1.0	PICMG	
PICMG IRTM.1 R1.0	PICMG	
PICMG MTCA.0 R1.0	PICMG	
PICMG MTCA.1 R1.0	PICMG	
PICMG MTCA.4 R.1.0	PICMG	
PICMG MTCA.4.1 R1.0	PICMG	
IPMI v2.0 R1.0	Intel, HP, NEC, Dell	

## 1.2.2 Environmental and Regulatory Documents

None.

### 1.3 Contributors

¶ 3 The following PICMG Member companies and institutions contributed to the creation of this document:

DESY	Germany
Institute of High Energy Physics (HEP)	PR China
IPFN Lisbon	Portugal
ITER	France
N.A.T. GmbH	Germany
Pentair Schroff	Germany/USA
PowerBridge GmbH	Germany
SLAC National Laboratory	USA
TEWS Technologies	Germany
VadaTech	USA

### 1.4 Special word usage

¶ 4 In this specification, the following key words (in bold text) will be used:

<b>may:</b>	Indicates flexibility of choice with no implied preference.
<b>should:</b>	Indicates flexibility of choice with a strongly preferred implementation. The use of <b>should not</b> (in bold text) indicates a flexibility of choice with a strong preference that the choice or implementation should be prohibited.
<b>shall:</b>	Indicates a mandatory requirement. Designers shall implement such mandatory requirements to ensure interchangeability and to claim conformance with this specification. The use of shall not (in bold text) indicates an action or implementation that is prohibited.

*Note:* When not in bold text, the words “may,” “should,” and “shall” are being used in the traditional sense; that is, they do not adhere to the strict meanings described above.

### 1.5 Name and logo usage

¶ 5 The PCI Industrial Computer Manufacturers Group's policies regarding the use of its logos and trademarks are as follows:

¶ 6 Permission to use the PICMG organization logo is automatically granted to designated members only as stipulated on the most recent *Membership Privileges* document (available on the Web at [www.picmg.org](http://www.picmg.org)), during the period of time for which



their membership dues are paid. Non-members of PICMG may not use the PICMG organization logo.

¶ 7 The PICMG organization logo must be printed in black or in color as shown in the files available for download from the members' side of the Web site. The center bar of the logo containing the phrase "PICMG" is set horizontally and the aspect ratio of the entire logo must be maintained, but the size may be varied. Nothing may be added to or deleted from the PICMG logo.

¶ 8 Manufacturers' distributors and sales representatives may use the AdvancedTCA logos (but not the PICMG organization logo) in promoting products sold under the name of the manufacturer.

¶ 9 The use of the AdvancedTCA logos is a privilege granted by the PICMG organization to companies who have purchased the relevant AdvancedTCA specifications (or acquired them as a member benefit), and who believe their products comply with these specifications. Use of the AdvancedTCA logo by either members or non-members implies such compliance. Misuse of the AdvancedTCA logos may result in PICMG's revoking permission to use them.

¶ 10 The AdvancedTCA logos must be used exactly as shown in the files available for download from the PICMG Web site. The aspect ratios of the logos must be maintained, but the sizes may be varied. Nothing may be added to or deleted from the AdvancedTCA logos

¶ 11 Color information for the red parts of the AdvancedTCA logos is as follows:

- Pantone 032 CVC is the closest match.
- CMYK colors are C:0, M:100, Y:100, K:0.
- RGB is R:237, G:28, B:36.

¶ 12 The PICMG name and logo are registered trademarks of PICMG. Registered trademarks must be followed by the □ symbol, and the following statement must appear in all published literature and advertising material in which the logo appears:

¶ 13 PICMG and the PICMG logo are registered trademarks of the PCI Industrial Computer Manufacturers Group.

¶ 14 The AdvancedTCA□ name and logos and the ATCA□ name and logo are trademarks of PICMG in process of registration. These trademarks must be followed by the □ symbol, and the following statement must appear in all published literature and advertising material in which the logo appears:

¶ 15 AdvancedTCA, ATCA and the AdvancedTCA and ATCA logos are trademarks of the PCI Industrial Computer Manufacturers Group.

## **1.6 Signal naming conventions**

¶ 16 All signals are active high unless denoted by a trailing # symbol. Differential signals are denoted by a trailing + (positive) or – (negative) symbol.

## **1.7 Intellectual property**

¶ 17 The Consortium draws attention to the fact that compliance with this specification may involve the use of a patent claim(s) (“IPR”). The Consortium takes no position concerning the evidence, validity, or scope of any such IPR.

¶ 18 Attention is also drawn to the possibility that some of the elements of this specification may be the subject of IPR other than those identified below. The Consortium shall not be responsible for identifying any or all such IPR.

THIS SPECIFICATION IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NON-INFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY USE OF THIS SPECIFICATION **SHALL** BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE CONSORTIUM, NOR ANY OF ITS MEMBERS OR SUBMITTERS, **SHALL** HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER, DIRECTLY OR INDIRECTLY, ARISING FROM THE USE OF THIS SPECIFICATION.

No representation is made as to the availability of any license rights for use of any IPR inherent in this specification for any purpose other than to implement this specification.

### **1.7.1 Necessary Claims (referring to mandatory or recommended features):**

¶ 19 None.

### **1.7.2 Unnecessary Claims (referring to optional features or non-normative elements):**

¶ 20 None.

¶ 21 PICMG makes no judgment as to the validity of any claims or the licensing terms offered by the claimants.

## 1.8 Acronyms & Definitions

¶ 22 **Table 1-1** shows the new terms and acronyms used in specific ways throughout this document.

**Table 1-1: Acronyms**

AMC	Advanced Mezzanine Card
API	Application Programming Interface
HPC	Hot Plug Controller
MCH	MicroTCA Carrier Hub
MMC	Module Management Controller
MCMC	MicroTCA Carrier Managment Controller
MRL	Manually Operated Retention Latch and Sensor
OS	Oppering System
RTM	Rear TransitionModule

## 2 Operating Module Hardware-Software Components

### 2.1 Introduction

¶ 23 This section describes the hardware and software elements required to support the Hot Plug environment and differences of implementation of Hot Plug in MTCA.0 systems, specifically relating to both AMC and RTM procedures.

### 2.2 PCI Express Hot Plug

¶ 24 Key goals of the Hot Plug procedure are high availability and uninterrupted operation of the overall system during the procedure. To support these goals Hot Plug solutions provide three important capabilities:

- A method of replacing failed cards without turning the system off
- Keeping the OS and other services running during the repair
- Shutting down, restarting and reporting to software associated with the failed device

The main role in performance of the above-stated tasks lies with the Hot Plug Controller.

¶ 25 The PCI Express Hot Plug specification defines the Standard Hot Plug Controller (HPC) with associated port interface integrated within individual root or switch ports, and the PCI Express configuration registers set associated with the current PCI Express slot.

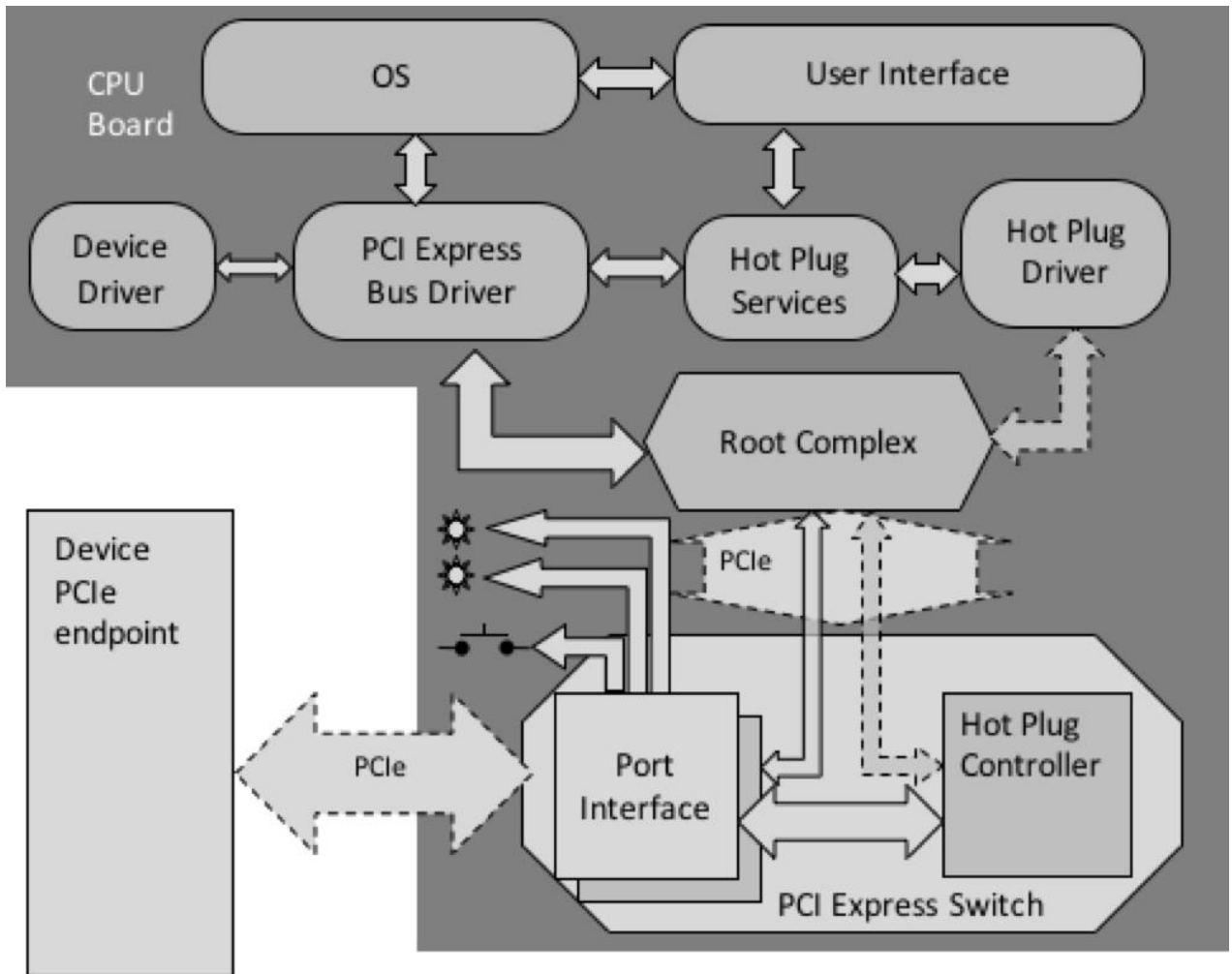
¶ 26 PCI Express is a point-to-point connection which permits the HPC to be distributed to each port interface to which a connector is attached.

¶ 27 Under Hot Plug software control, these HPCs and the associated port interface must control the card interface signals to ensure orderly power-down and power-up, and provide indications to the user as cards are removed and replaced. Hot Plug Controller must:

- Control power to the card connector
- Selectively turn on or off the Power and Attention Indicators associated with a specific card connector
- Monitor slot events and report to software via interrupts

The hardware elements needed to support Hot Plug are shown in **Figure 2-1**.

¶ 28 The standard programming interface to the HPC is provided via the PCI Express Capability register block. These registers have to be correctly configured. **Table 2-1** illustrates these registers. Hot Plug features are primarily provided via Slot Registers that are defined for each switch port.



**Figure 2-1: PCI Express Hot Plug Hardware Elements**

**Table 2-1** illustrates the PCI Express configuration registers needed for Hot Plug.

**Table 2-2** illustrates the major hardware elements necessary to support PCI Express Hot Plug operation in the MTCA environment.

**Table 2-1: PCI Express Configuration Registers Used for Hot Plug**

<b>PCI Express Capabilities Register</b>	<b>Next Cap. Register</b>	<b>PCI Cap. ID</b>
<b>Device Capabilities Register</b>		
Device Status Register	Device Control Register	
<b>Link Capabilities Register</b>		
Link Status Register	Link Control Register	
<b>Slot Capabilities Register</b>		
Slot Status Register	Slot Control Register	

**Table 2-2: Major Hardware Elements for PCI Express Hot Plug in MTCA**

<b>Hardware Element</b>	<b>Description</b>
Hot Plug Controller	Receives and processes commands issued by the Hot Plug System Driver. One Controller is associated with each switch port that supports hot plug operation. Resides on the PCI Express Switch integrated on the MCH. Port interface done via IPMI messages of the MMC on the AMC and MCMC on the MCH
Card Slot Power Switching Logic	Slot Power Switching controlled by the MCH under the direction of the Hot Plug Controller
Power Indicator	One per Slot. Indicates whether power is currently applied to the card slot or not. Controlled by the MCH under the direction of the Hot Plug Controller
Attention Indicator	One per Slot. The Attention Indicator is used to draw the attention of the operator to indicate a Hot Plug problem or failure. Controlled by the MCH under the direction of the Hot Plug Controller
Attention Button (AMC MRL)	One per Slot. Used by the operator to notify the Hot Plug software of a Hot Plug request. The MCH informs PCI Express switch if the MRL is pulled out or pushed in.

### 2.3 MTCA Specific Hot Plug Environment and Hardware Components

¶ 29 In MTCA systems the major role in Hot Plug is played by the PCI Express switch integrated within the MCH.

- REQ 2.1 To enable Hot Plug for every MTCA carrier slot the Hot Plug controllers of the PCI Express switch **shall** be enabled for every existing slot.
- REQ 2.2 Numbering of slots: Physical slot numbers in the Slot Capabilities register **shall** correspond to physical slot number of the MTCA shelf.
- REQ 2.3 The Hot Plug Controller **should not** be enabled on the Upstream Slot. The MCH **should not** wait for a Hot Plug event in the case of removing a CPU from an upstream slot.
- REQ 2.4 If the CPU in an upstream slot is switched off the MCH **should not** wait for Hot Plug events to shut down the slot.

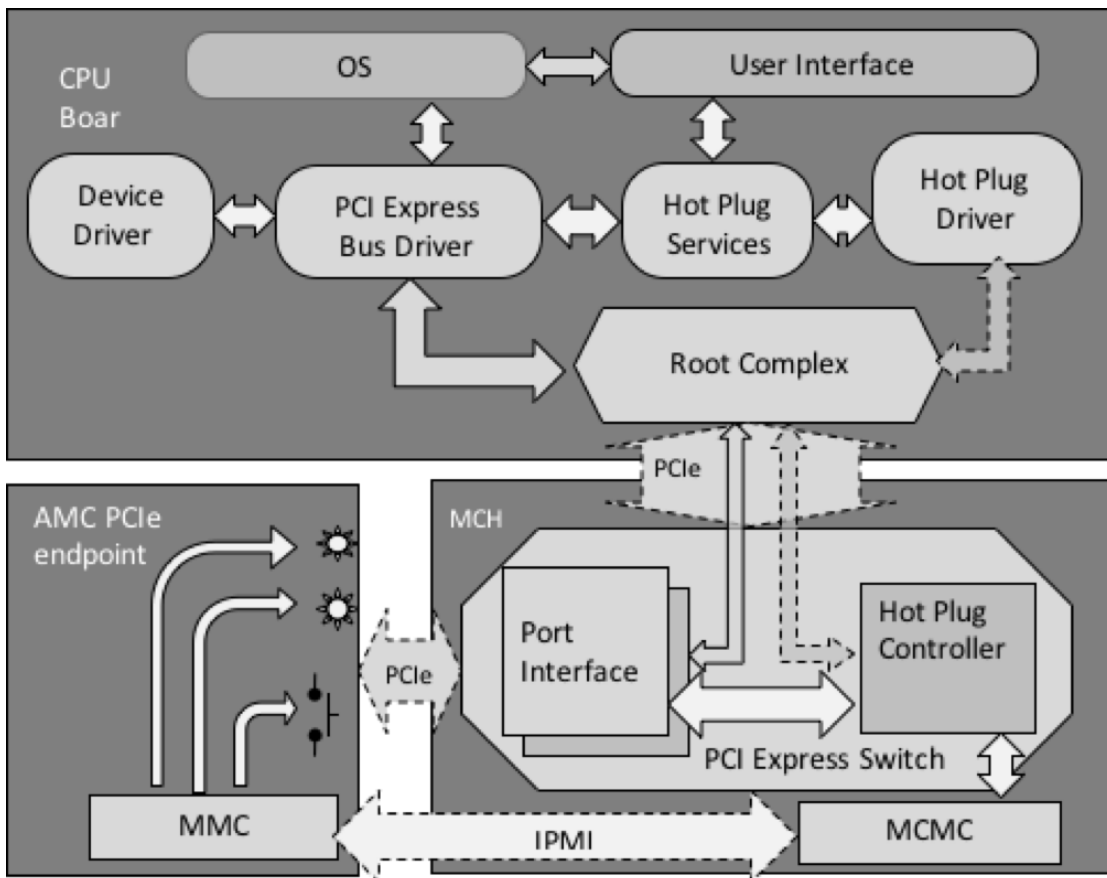


Figure 2-2: MTCA Specific Hot Plug Event & Hardware Components

¶ 30 It is possible to avoid the above specified restrictions by implementing a Hot Plug event expectation timeout in the MCH.

¶ 31 In MTCA the PCI Express port interface of the Hot Plug controller and the port itself may be on different devices. Thus there is no direct connection between them. In case of such a connection, IPMI messages between the AMC and MCH have to be used. On the MTCA systems some tasks of the Hot Plug Controller are fulfilled by the MMC and MCMC via IPMI messages See **Figure 2.2** c.f. Figure 2.1.

## 2.4 Software Elements

¶ 32 **Table 2-3** and **Figure 2-1** describe the major software elements required to support Hot Plug capability. The software elements are implemented by the OS and consequently beyond the scope of this guideline.

¶ 33 The device driver and user application Hot Plug protocol will be described in **Section 3**, Usage Models.

**Table 2-3: Major software elements required to support Hot Plug capability**

Software Element	Supplied	Description
User Interface	OS vendor, MCH vendor	An OS-supplied utility that permits the end-user to request that card connector be turned off/on. User Interface has to be implemented both at the level of an OS and at the level of Carrier management ( for example: using IPMI tool)
Hot Plug Services	OS vendor	A service that processes requests issued by the OS. This includes requests to: <ul style="list-style-type: none"> <li>• provide slot identifiers</li> <li>• turn card On or OFF</li> <li>• turn Attention Indicator ON or OFF</li> <li>• return current state of slot</li> </ul> The Hot Plug Service interacts with the Hot Plug System Driver to satisfy the requests.
Standardized Hot Plug System Driver	OS vendor	Receives requests from the Hot Plug Service within the OS. Interacts with the hardware Hot Plug Controllers to accomplish requests.
Device Driver	Card vendor	Some special Hot Plug specific capabilities must be incorporated in a Hot Plug capable device driver. This includes: <ul style="list-style-type: none"> <li>• support for the Quiesce command</li> <li>• optional implementation of the Pause command</li> <li>• support for Start or Optional Resume command</li> </ul>



## 2.5 Operating Models

¶ 34 For MTCA systems three main models of Hot Plug are used:

1. Hot Plug triggered by Attention Button (User run Hot Plug).
  - Operator pulls out the MRL to remove the AMC Card
2. Hot Plug triggered by the Management Controller (Management Hot Plug)
  - Operator sends IPMI message (shutdown FRU) to MCH
3. Hot Plug triggered by the Software (Soft Hot Plug, Development Hot Plug)
  - Operator request Hot Plug via User Interface of the OS Hot Plug Service

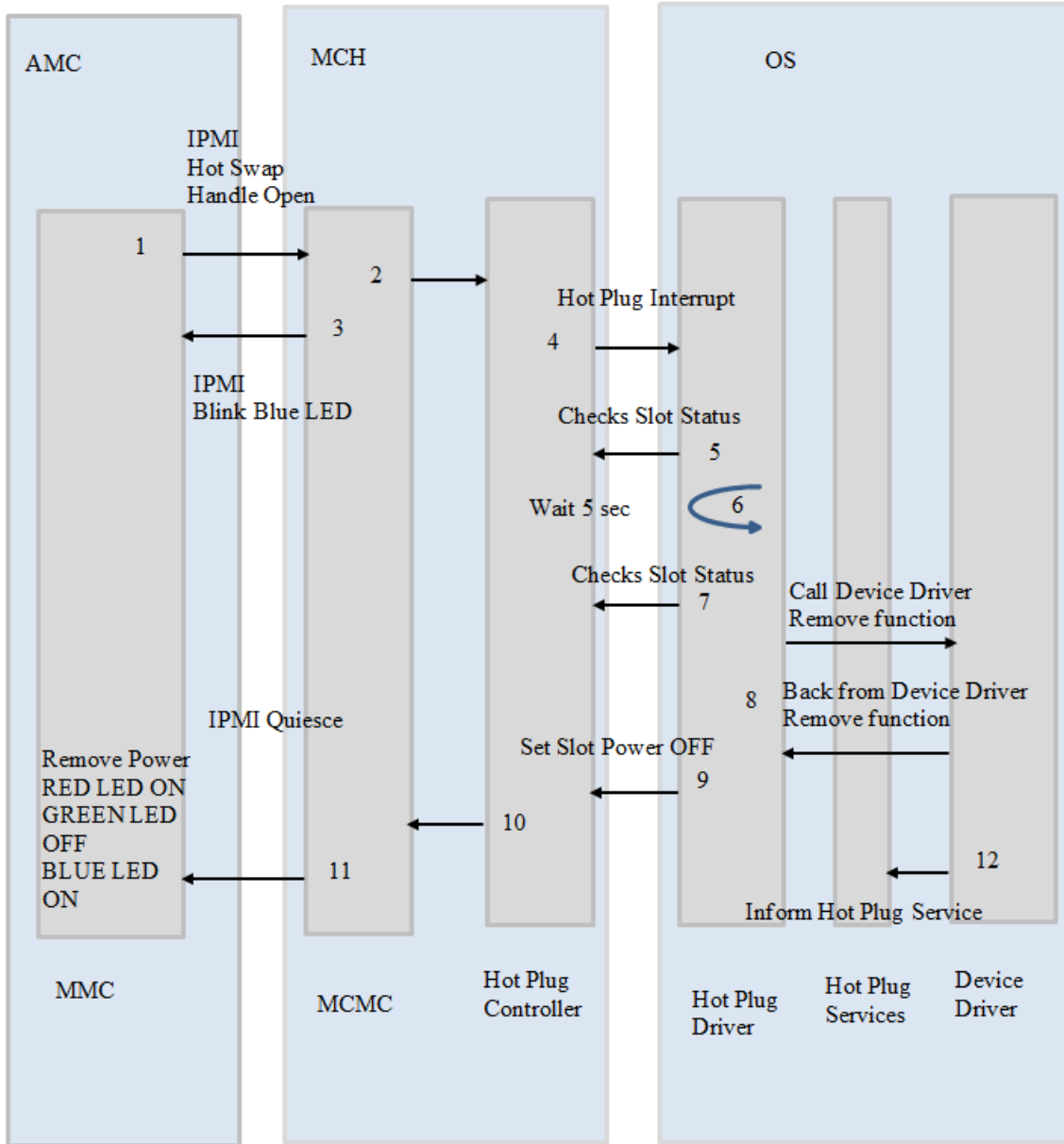
¶ 35 In the case where the Soft Hot Plug PCI Express device is removed from the PCI Express bus but not from the crate and remains powered On, the Developer can load new Firmware and then via the Hot Plug User Interface requests a PCI Express bus rescan to add the device to the Bus.

¶ 36 For this purpose, the Software triggered Hot Plug is used. Hot Plug takes place at the level of the Hot Plug Controller - Operating System, without intervention of the Management System.

## 2.6 Device Removal Procedure

¶ 37 When a device must be removed, a number of steps must occur to not only prepare the software and hardware for safe removal of the device, but also to control indicators that provide visual evidence of the request to remove the device being processed.

¶ 38 The exact sequences of events that occur when performing a Hot Plug device **removal** vary slightly depending on the Hot Plug model. Each sequence is described in **Figure 2-3** and **Table 2-4**.



**Figure 2-3: Device Removal Procedure**

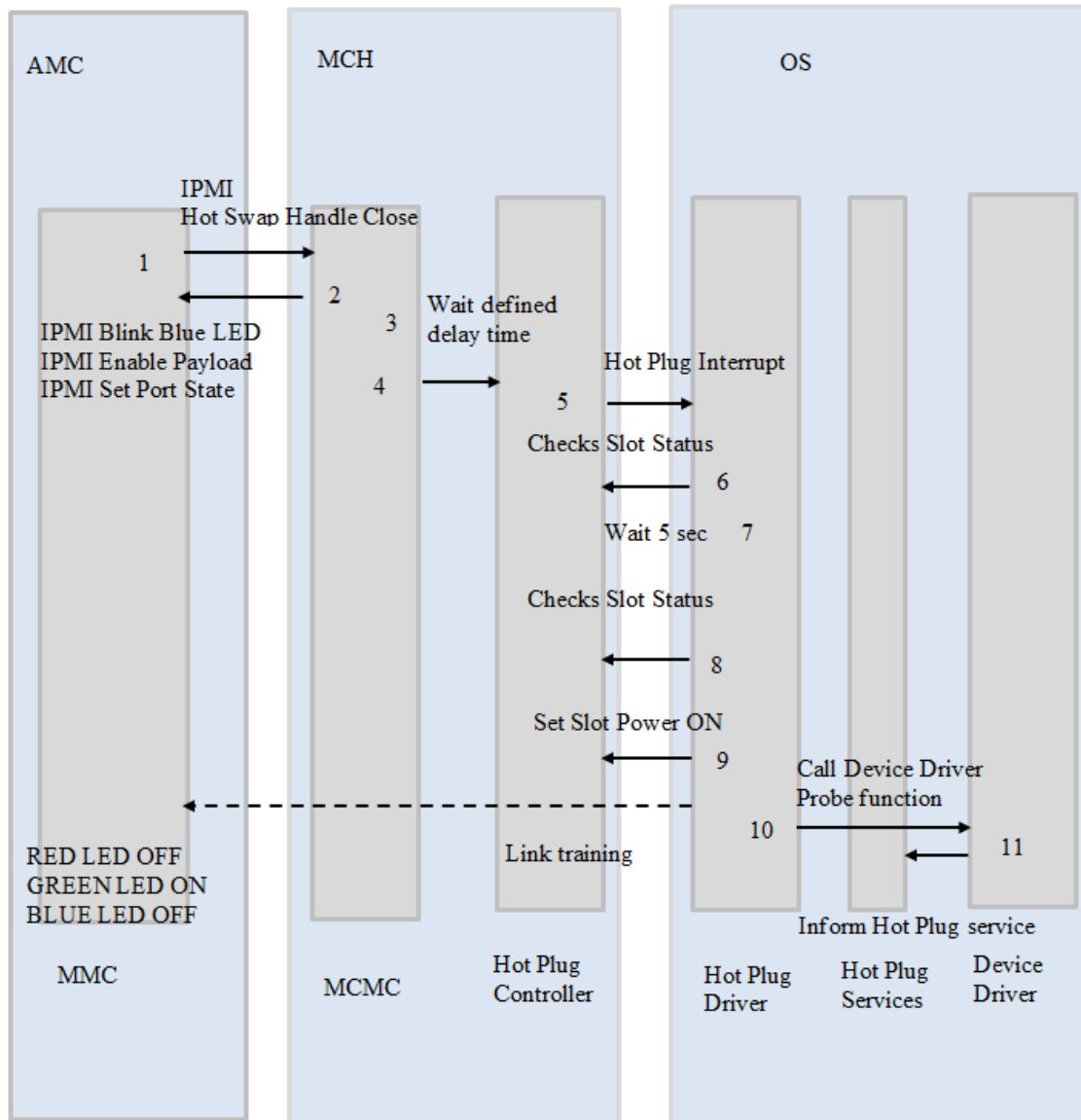
**Table 2-4: Device Removal Procedure**

	<b>Hot Plug triggered by Attention Button</b>	<b>Hot Plug triggered by Management Controller</b>
1	Operator initiates device removal by pulling out the MRL of the AMC	Operator initiates the device removal by sending IPMI command to MCH
	MMC of the AMC sends IPMI Hot Swap Open message to MCMC of the MCH	
2	MCH activates “Attention Button Pressed” line of the PCI Express Switch Hot Plug Controller of the current Slot	
3	MCH sends Blue LED blink command to the AMC	
4	Hot Plug Controller detects event, delivers an interrupt to the Root Complex.	
5	Hot Plug System Driver reads slot the status information (checks Slot Status), and detects Attention Button Request	
6	Operator is granted a 5 second abort interval, from the time that indicator starts to blink; during this time operator may press MRL back in to abort the request; in this case AMC informs MCH that „Attention Button pressed“ was removed. Without Abort the procedure follows as below.	
7	Hot Plug Software validates the request (checks Slot Status)	
8	If request is validated, the Hot Plug Service utility commands device driver to remove the node. Device driver must stop issuing requests to the device and complete or terminate all outstanding requests as well as disable its ability to generate transactions and interrupts	
9	Hot Plug Software commands Hot Plug Controller to turn the slot OFF	
10	Hot Plug Controller informs MCMC the Device can be removed	
11	MCH starts normal Quiesce sequence to power off the Slot	
12	Optional: Device Driver could inform Hot Plug Services about node removal (udev in Linux)	

REQ 2.5 If a timeout of the Quiesce command is implemented in the MMC of the AMC a 5 seconds delay **shall** be considered; otherwise during validation of the “Attention Button” Request the Hot Plug Controller will receive an error, as the device will be switched-off.

## 2.7 Device Insertion Procedure

¶ 39 The procedure for installing a new device reverses the steps listed for device removal. If the device is removed from the shelf the insertion steps for all Hot Plug models are the same. If the device is not removed from the shelf but only powered OFF the exact sequence of events that occur when performing a Hot Plug device insertion vary slightly depending on the Hot Plug model. Each sequence is described in **Figure 2-4** and **Table 2-5**.



**Figure 2-4: Device Insertion Procedure**

**Table 2-5: Device Insertion Procedure**

	<b>Hot Plug triggered by Attention Button</b>	<b>Hot Plug triggered by the Management Controller</b>
1	Operator installs the AMC and presses in the MRL	Operator sends IPMI command to MCH to start the current slot
	MMC of the AMC sends IPMI Hot Swap close message to MCMC of the MCH	
2	MCH starts powering ON the Slot	
3	MCH waits for delay time defined by operator	
4	MCH activates “Attention Button Pressed” line of the PCI Express Switch Hot Plug Controller of the current Slot	
5	Hot Plug Controller detects this event and delivers an interrupt to the Root Complex.	
6	Hot Plug System Driver reads slot status information and detects the Attention Button Request	
7	Operator is granted a 5 second abort interval, from the time the indicators starts to blink, during which the operator may pull the MRL back out to abort the request, in this case AMC has to inform MCH about the state change.	
8	Hot Plug Software validates the request (checks Slot Status)	
9	If request is validated, Hot Plug Service utility commands the Hot Plug Controller to turn slot on and starts Link training . Once Link training is complete, OS commands Platform Configuration Routine to configure the device by assigning the necessary resources	
10	OS locates the appropriate driver (using Vendor and Device IDs) and loads it into memory	
11	Optional: Device Driver informs Hot Plug Services about Device Node creation (udev, Linux)	

REQ 2.6 At the moment the Hot Plug Controller checks the status of the device the PCI links of the device **shall** be already initialized.

REQ 2.7 In cases when initialization of the device is slow, e.g. due to slow FPGA loading or FPGA PCI Express core initialization, the operator

**shall** have the option to enter the delay time of sending interrupt to the Root Complex.

## **2.8 Soft Hot Plug**

REQ 2.8 Soft Hot Plug is completely executed from the OS and Hot Plug Software, and does not require intervention of the MCH. The only condition is that the MCH **shall not** power off the slot

### 3 Usage Models

#### 3.1 Introduction

¶ 40 In the usual PCI Express environment, user applications communicate with the Hardware using a Device Driver bound to the current Device.

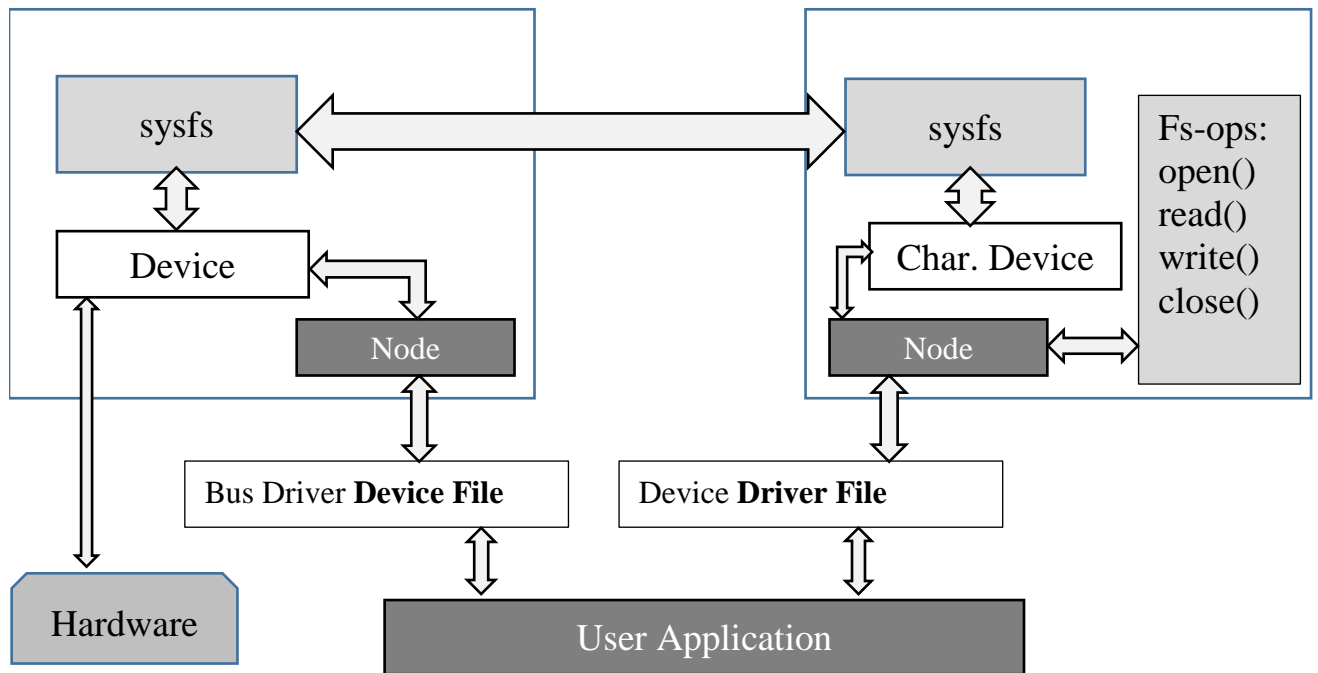
REQ 3.1 The Device driver in such an environment is responsible for the Device Hot Plugging and **shall** be written in a way that handles asynchronous device removal.

REQ 3.2 The user application **shall** react to Hot Plug events in a standardized way.

¶ 41 This section describes the common Hot Plug protocol between Device Driver and user applications.

¶ 42 The user application can be notified about Hot Plug Events of the current device by the Operating System Hot Plug Services or by the Device Driver.

¶ 43 **Figure 3-1** illustrates the Linux Device Model and generation of the /sys and /dev files.



**Figure 3-1: Linux Device Model and Generation of the /sys and /dev files**

¶ 44 During hardware insertion/removal the Bus Driver creates/deletes a device and node for the current device reflected in sysfs (/sys/bus/pci/devices/xxx). The user may interact with a device via sysfs attributes.

¶ 45 The OS Hot Plug Service generates events which could be used by the user application to be notified about Hot Plug events of the current device. The Bus Driver Device File name is based on the bus:subordinatebus:function structure. Device driver is attached to a parent bus driver and bound to the current device via PCI Vendor/Device IDs. In the insertion/removal of the current device, the Device Driver creates a Device Node and Device Driver File in /dev based on the Device name described in the MTCA.4 Standard Hardware API (SHAPI) Guideline.<sup>1</sup>

### ***3.2 Hot Plug Interaction of the Device Driver and User Application***

REQ 3.3 To support a method of safely replacing the device the Device Driver **shall** adhere to the following procedures:

1. Stop all current transactions in case of device removal
2. Guarantee that all future requests for transactions on removed device will be rejected with the appropriate error code
3. Create a Device Driver File in case of Device insertion and delete this file in case of Device removal
4. Optional; The Device Driver can inform the user application of the Hot Plug via OS Signals

The sequence of the Device Driver Hot Plug steps is as follow:

Device insertion:

1. Device Driver initializes the device and all device resources
2. Device Driver creates the Device Driver File
3. Set device status to 1 (indicating device is OK)

Device removal:

1. Set device status to 0 (indicating device is OFF)
2. Stop all outgoing transactions
3. While device status is OFF all future requests for transactions for this device are denied with the appropriate error code
4. Delete Device Driver File

---

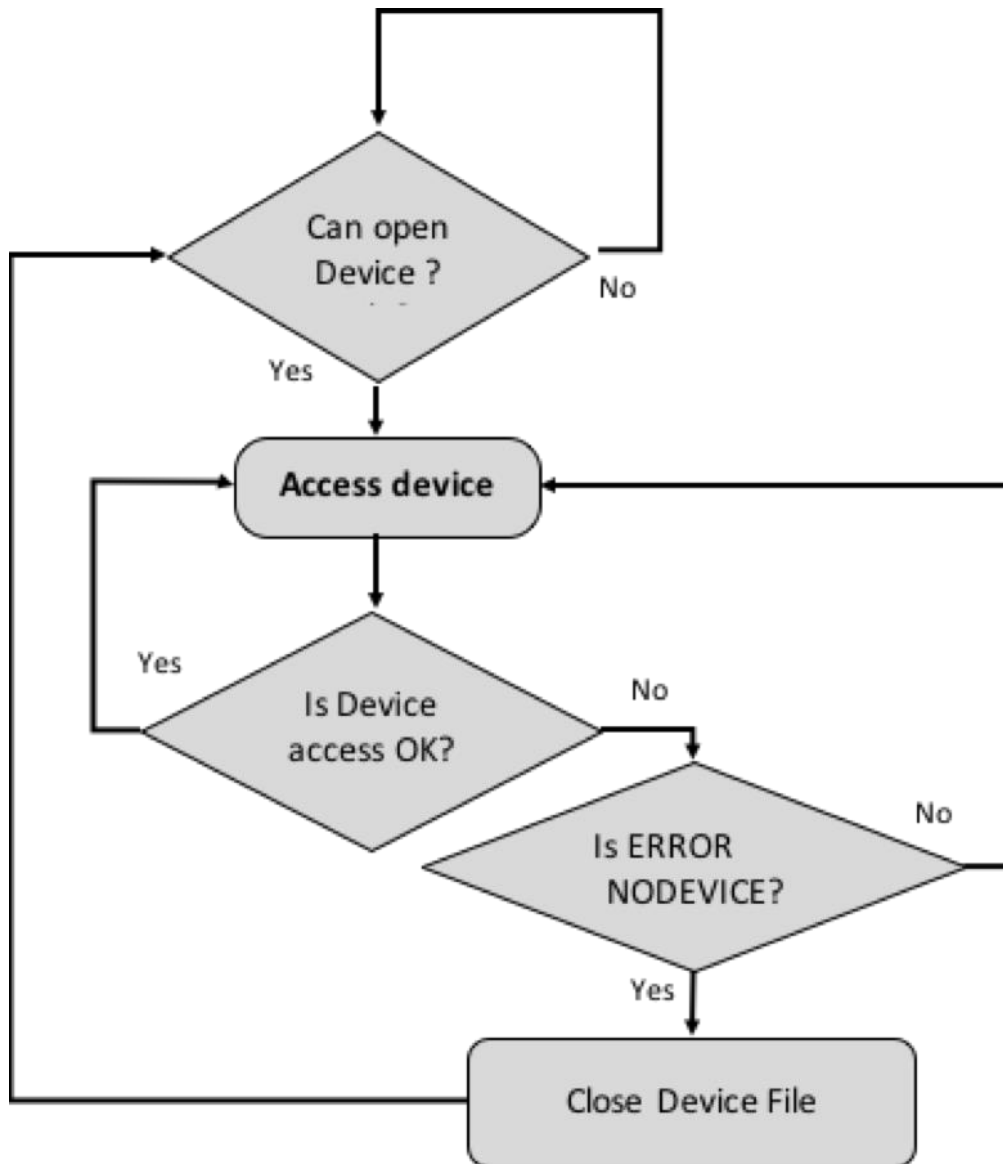
<sup>1</sup> SHAPI currently under development.



From the user application point of view, the Hot Plug protocol is:

1. User application tries to open device file
2. If device is ON the application uses the device
3. If user application receives an error in device access it **shall** check the error code, if it is NODEVICE it **shall** close the Device Driver file
4. User Application retries the open the Device Driver File.

The operation is summarized in **Figure 3-2**.



**Figure 3-2: User Application Response to Hot Plug**

## 4 RTM Hot Plug

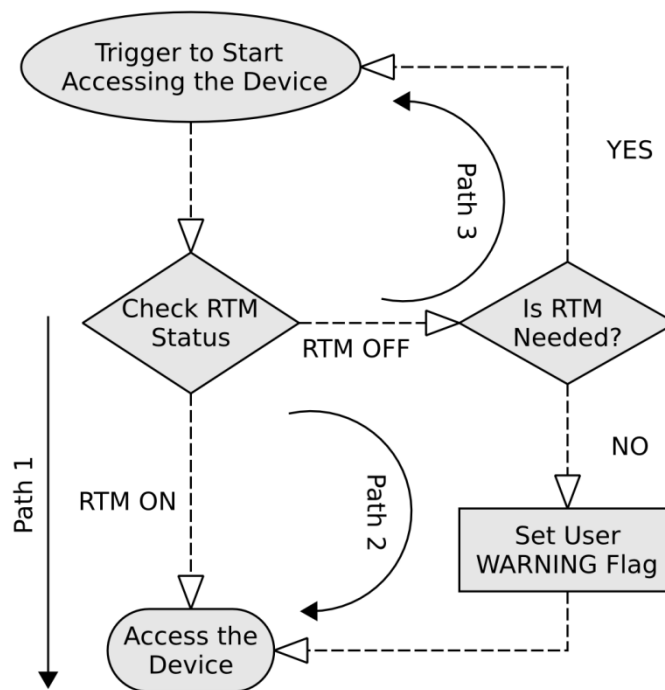
¶ 46 Hot Plug replacement of an RTM is completely controlled by an MMC and MCMC; however for stable and faultless operation of user applications the user application must have information on hot replacement of the RTM.

REQ 4.1 For these purposes the AMC **shall** provide an RTM status register which is readable from the user application, such as when fully controlled by the FPGA on the AMC or MMC.

¶ 47 Reading from this register gives information of whether the RTM module is IN or OUT. This Register should provide additional information of whether the RTM is needed for the normal operation of the AMC. Based on this information the user application can determine the procedure for RTM removal and insertion.

¶ 48 For more detailed information about RTM Status Register refer to the SHAPI guidelines of PICMG MTCA.4.

¶ 49 **Figure 4-1** shows three conditions of using the RTM status register. In all cases the AMC is an ADC board. The conditions are indicated as path 1 to 3.



**Figure 4-1: Example Usage of RTM Status Register**

¶ 50 **Path 1:** The RTM is on; in this case the device can be used without limitations and directly reach the termination point.

¶ 51 **Path 2:** The RTM is off, but not required for the firmware to be operational. In this case the application will raise a warning to the user because there may be limitations if some RTM features are missing.

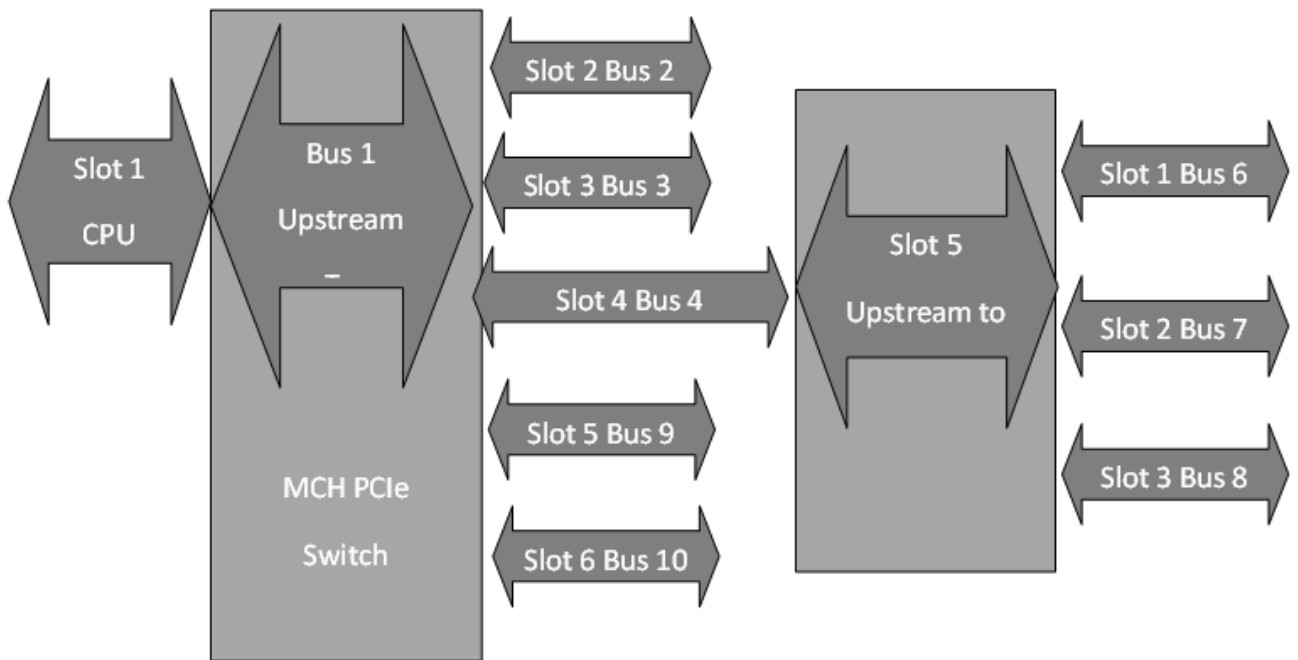
¶ 52 **Path 3:** The RTM is off but mandatory for the operation. The application cannot run and does not end on the termination point but closes the device and goes back to the beginning.

## 5 Using PCI Express Switches on the AMC

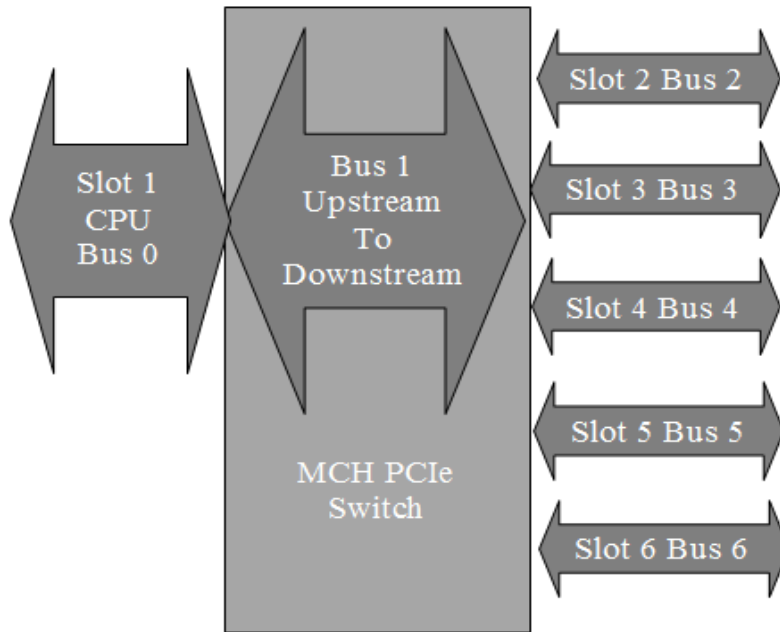
¶ 53 Using a PCI Express Switches on an AMC module brings additional Hot Plug issues.

1. Removing or inserting PCI Express switches changes the bus numbering for all busses.
2. Mismatching occurs in shelf Slot numbering

¶ 54 **Figures 5-1 and 5-2** indicate these problems. Fig 5-1 shows an AMC module with the PCI Express Switch on it and Fig.5-2 shows the bus and slot numbering after removing the AMC module.



**Figure 5-1: PCI Express Bus numbering for case of AMC module with PCIe Switch**



**Figure 5-2: PCI Express Bus numbering after removing AMC module with PCIe Switch (c.f. Fig. 5-1)**

REQ 5.1 After removing an AMC that includes a PCIe Switch, the OS **shall** be rebooted to update the PCI Express bus configuration. Hot Plugging AMC modules with a PCI Express Switch **should** be avoided.

¶55 Probably in future operating systems will have a feature of a preliminary reservation of bus numbers for each PCI Express slot, so AMC boards with PCI Express Switches will be involved in the Hot Plug procedure.

## 6 Revision History

<b>Date</b>	<b>Revision</b>	<b>Changes</b>
01/09/2016	R1.0	Initial Release

## 7 Table of Requirements

REQ 2.1	To enable Hot Plug for every MTCA carrier slot the Hot Plug controllers of the PCI Express switch <b>shall</b> be enabled for every existing slot. ....	9
REQ 2.2	Numbering of slots: Physical slot numbers in the Slot Capabilities register <b>shall</b> correspond to physical slot number of the MTCA shelf. ....	9
REQ 2.3	The Hot Plug Controller <b>should not</b> be enabled on the Upstream Slot. The MCH <b>should not</b> wait for a Hot Plug event in the case of removing a CPU from an upstream slot. ....	9
REQ 2.4	If the CPU in an upstream slot is switched off the MCH <b>should not</b> wait for Hot Plug events to shut down the slot. ....	9
REQ 2.5	If a timeout of the Quiesce command implemented in the MMC of the AMC the 5 seconds <b>shall</b> be considered; otherwise during validation of the “Attention Button” Request the Hot Plug Controller will receive an error as the device will be switched-off. ....	13
REQ 2.6	At the moment the Hot Plug Controller checks the status of the device the PCI links of the device <b>shall</b> be already initialized. ....	15
REQ 2.7	In cases when initialization of the device is slow (due to slow FPGA loading, FPGA PCI Express core initialization) the operator <b>shall</b> have the option to enter the delay time of sending interrupt to the Root Complex. ....	15
REQ 2.8	Soft Hot Plug is completely executed from the OS and Hot Plug Software, and does not require intervention of the MCH. The only condition is that the MCH <b>shall not</b> power off the slot. ....	16
REQ 3.1	The Device driver in such an environment responsible for the Device Hot Plugging and <b>shall</b> be written in a way that handles asynchronous device removal. ....	17
REQ 3.2	The user application <b>shall</b> to react to Hot Plug events in standardized way. ....	17
REQ 3.3	To support a method of safely replacing the field device the Device Driver <b>shall</b> adhere to the following procedures: ....	18
REQ 4.1	For these purposes the AMC <b>shall</b> provide an RTM status register which is fully controlled by the FPGA on the AMC or MMC and readable from the user application. ....	20
REQ 5.1	These examples show that after removing the PCIe Switch the OS <b>shall</b> be rebooted for correct PCI Express bus reconfiguration. Clearly, Hot Plugging AMC modules with a PCI Express Switch <b>should</b> be avoided. ....	23