Fusion Engineering and Design 96–97 (2015) 738–741

Contents lists available at ScienceDirect

Fusion Engineering and Design

journal homepage: www.elsevier.com/locate/fusengdes

PCI express hotplug implementation for ATCA based instrumentation

Paulo F. Carvalho[a], Bruno Santos[a], Miguel Correia[a], Álvaro M. Combo[a], António P. Rodrigues[a], Rita C. Pereira[a,*], Ana Fernandes[a], Nuno Cruz[a], Jorge Sousa[a], Bernardo B. Carvalho[a], António J.N. Batista[a], Carlos M.B.A. Correia[b], Bruno Gonçalves[a]

[a] Instituto Plasmas e Fusão Nuclear, Instituto Superior Técnico, Universidade Técnica de Lisboa, 1049-001 Lisboa, Portugal
[b] Centro de Instrumentação, Departamento de Física, Universidade de Coimbra, 3004-516 Coimbra, Portugal

HIGHLIGHTS

- Hotplug capabilities are designed as an expected or graceful methodology in which the user is not permitted to install or remove a PCIe endpoint device without first notifying the system software.
- Hotswap capabilities allow endpoints or PCIe switches with endpoints to be inserted or removed from a PCIe system gracefully or unexpectedly without special consideration.
- ATCA, advanced telecommunication computer architecture is a new specification with high availability and high reliability key features which improves data acquisition systems.
- Data acquisition systems are used almost everywhere and a demand in the nuclear fusion research field.
- Nuclear fusion is a future alternative for power and energy resources generation for world humanity consumption.

ARTICLE INFO

Article history:
Received 24 September 2014
Received in revised form 13 May 2015
Accepted 13 May 2015
Available online 20 June 2015

Keywords:
Hotplug
Hotswap
Control
Acquisition
ATCA systems
Nuclear fusion

ABSTRACT

This paper describes a Peripheral Component Interconnect Express (PCIe) hotplug and hotswap capability implementation for advanced telecommunication computer architecture (ATCA) based instrumentation. PCIe hotplug provides card insertion and removal capability from a running PCIe-based platform without causing system damages and not requiring an entire system shutdown. PCIe hotswap allows endpoints or PCIe switches with endpoint cards to be inserted or removed from a PCIe system gracefully or unexpectedly without special considerations. Control and data acquisition (C&DAQ) cards need to be replaced from a system for fault-condition repair, hardware malfunction, firmware updates or upgrades and hardware reconfiguration. ATCA specification key features such as high reliability and high availability for C&DAQ systems strongly benefits from these capabilities taking advantage from Redhat Enterprise Linux, installed operating system, and corresponding kernel with built-in mechanisms and embedded software modules for hotplug and hotswap support. PCIe hotplug and hotswap implemented solutions in the ATCA-based prototype provides described capabilities to the C&DAQ and PCIe switch cards providing a fast replacement strategy for damaged cards and system downtime reduction. In addition, provides capability to automatically remove PCIe device nodes and corresponding device files from ATCA system as expected by ITER.

1. Introduction

Peripheral Component Interconnect Express (PCIe) [1] hotplug is derived from revision 1.0 of the standard hotplug controller specification for PCI [2]. This specification describes the methodology by which PCIe endpoint devices may be added/removed from an operational system without compromising the operational state of the system. Moreover, defines a number of standard hardware registers and signals, such as attention button and attention indicator which allows development of a common hotplug device driver.

PCIe hotplug is designed as an expected or graceful methodology in which the user is not permitted to install or remove a PCIe endpoint device without first notifying the system software.

* Corresponding author at: Instituto de Plasmas e Fusão Nuclear, Instituto Superior Técnico, Av. Rovisco Pais, 1049-001 Lisboa, Portugal. Tel.: +351 239 410108; fax: +351 239 829158.
E-mail address: pricardofc@ipfn.ist.utl.pt (R.C. Pereira).

http://dx.doi.org/10.1016/j.fusengdes.2015.05.030
0920-3796/

PCIe hotswap has no standard and it is system dependent [3]. Usually it is implemented by the system vendor. The PCIe hotswap capability allows endpoints or PCIe switches with endpoints to be inserted or removed from a PCIe system gracefully or unexpectedly without special consideration.

PCIe endpoint devices are connected to PCIe switches or to Root Complex bridges in a hotswap environment. Moreover, an external PCIe cable connects the host computer to the advanced telecommunication computer architecture (ATCA) [4] system.

The aim of this work is to implement PCIe hotplug and hotswap software capabilities in an ATCA based system prototype.

These implementations are being tested in the ATCA-based ITER fast plant system controller (FPSC) prototype [5].

2. System overview

2.1. Hardware components

Fig. 1 depicts the ATCA system used to implement both solutions which includes:

- A computer hosting a PCIe x16 switch cable adapter using a PCIe x16 from One Stop Systems (OSS), OSS-PCIe-HIB38-x16, that is directly connected to the rear transition module (RTM) of a PCIe switch card through a standard PCIe cable. The RTM follows PICMG 3.8 “Advanced RTM Zone 3A”;
- The data acquisition cards (PCIe endpoints) connect to the PCIe switch card through the ATCA fabric backplane;
- The ATCA shelf AXP 1440 [6], the C&DAQ cards, ATCA-IO-PROCESSOR [7], and the PCIe Timing Switch card, ATCA-PTSW-AMC4 [8,9], which will be detailed in this section.

2.1.1. AXP 1440 SHELF

The AXP 1440 shelf supports 14 ATCA blade slots, 2 of which for system controller and switching blades.

The ATCA dual star backplane provides peer-to-peer links between hub and node slots on the fabric interface, which are used to set up the PCIe data network.

In the rear of the shelf there are 2 shelf management alarm modules (SAMs) slot locations. The shelf includes also 2 power entry modules (PEMs) and 2 fan tray modules (FTMs).

2.1.2. ATCA-IO-PROCESSOR

The ATCA-IO-PROCESSOR card was designed for fast control and data acquisition systems. It is compliant with PICMG 3.0/3.4 and AXIe specifications [10], comprising a passive RTM for rear IO connectivity to ease hotswap. It is a high channel number IO card with galvanic isolation and FPGA based processing.

The digitized data from the ADC modules can be filtered or decimated, decreasing data throughput, increasing ADC resolution. Data are sent through PCIe Fabric Interface to the switch located in the ATCA shelf hub slots. The card includes DAC modules which can be updated in real-time.

Full-duplex point-to-point communication links between FPGAs and PCIe switches of peer cards inside the shelf allows the implementation of distributed algorithms and MIMO systems. Moreover, support for several timing and synchronization architectures is provided [9].

2.1.3. ATCA-PTSW-AMC4

ATCA-PTSW-AMC4 card is a PCIe and timing switch with 4 advanced Mezzanine cards (AMCs) [11] slots. It is a cutaway quad-AMC module carrier where most mid/full-size AMC modules available from industry can be installed.

AMC slots support a broad range of these module types, including digitizers, waveform generators and processor cards.

The ATCA-PTSW-AMC4 is connected through RTM to the external computer via a PCIe host adapter card (x16, x8 or x4) installed on the host computer by a PCIe cable plugged-in from host to carrier.

2.2. Software modules

The ATCA system software modules used are:

- Operating system: Redhat Enterprise Linux v6.3. Kernel 3.0.9-rt26.46.el6rt.x86_64;
- Hotplug service: Linux Universal Device Manager module daemon;
- PCI Express Hotplug Controller Driver v0.4 kernel module;
- PCIe endpoint cards Linux device driver [12];
- PCIe switch cards Linux device driver;
- PLX service device driver which includes a required library to read from and write to the PEX 8733/8696 internal registers.

The ATCA system software modules architecture is organized according to Fig. 2 diagram.

3. Implementation methods

Fig. 3 details two possible methods that can be used to implement PCIe hotplug and hotswap solutions.

The first method is based on a register polling implementation and the second method consists of an interrupt driven mechanism.

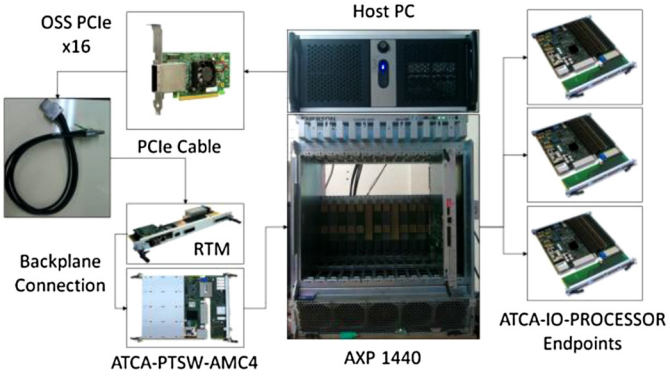

Fig. 1. ATCA system overview diagram.

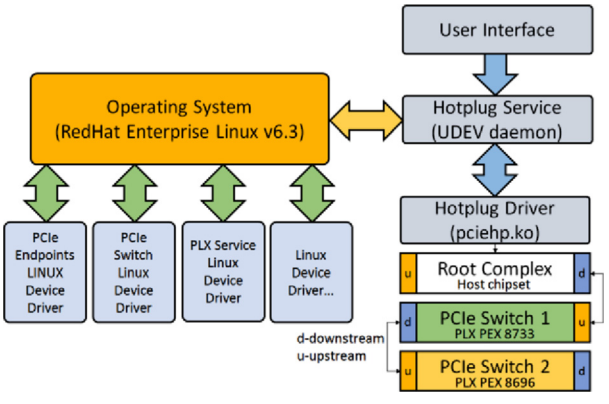

Fig. 2. Hotplug and hotswap software modules system architecture.

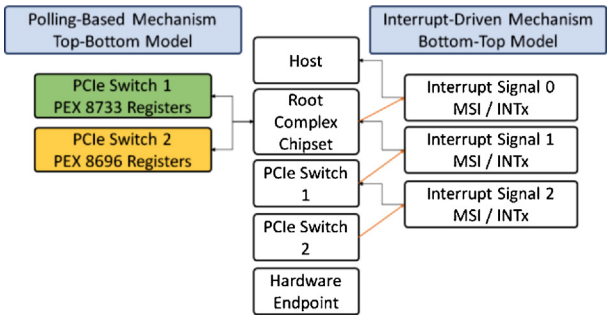

Fig. 3. Hotplug and hotswap methods: (i) polling-based (left side) and (ii) interrupt-driven (right side).

3.1. Polling-based method

Fig. 3 Hotplug and hotswap methods: (i) polling-based (left side) and (ii) interrupt-driven (right side).

Polling-based method can be carried out by a system service to monitor specific known PCIe switch device registers in the host computer side.

Changes in the polled PEX device register values means that some physical PCIe endpoint or PCIe switch card was inserted or removed from the ATCA system.

These registers changes compared to a reference value corresponding to the PCIe device present/absent states can be implemented by the system service to trigger an action based on two possible conditions:

- If the read value matches the device present status, an instruction is called to install device node into the PCIe bus tree and add device file to the system/*dev* directory;
- If the device is not present, an instruction is called to uninstall the device node from the PCIe bus tree and remove device file from system/*dev* directory.

3.2. Interrupt-driven mechanism

Interrupt-driven mechanism can be used whenever the PCIe hardware generates interrupt signals to the level-up system hierarchy component and it reaches the host CPU.

With this method, PEX device internal registers can be configured to handle two types of trigger interrupt signals for hotplug events: message signaled interrupts (MSI) or INTx.

These interrupt signals reach the PEX device ports and are propagated to the host root complex chipset which forwards them to the CPU.

An interrupt service routine is called by the CPU to acknowledge the received interrupt using the operating system hotplug kernel driver and hotplug running service.

The procedure installs/uninstalls the PCIe device nodes from the PCIe bus tree and adds/removes device file from system/*dev* directory in the host side.

The installed/uninstalled PCIe device nodes directly correspond to the physically inserted/removed PCIe cards from the ATCA system side.

4. Polling-based implementation

Polling-based method was the selected to develop and implement hotplug and hotswap solutions in the ATCA system because it is the less complex compared to the interrupt-driven mechanism, providing similar results with a fastest implementation. Two PCIe switch devices from PLX Technology exist on the present PCIe hierarchy, a PLX PEX 8696 [13] in the ATCA-PTSW-AMC4 card and a PLX PEX 8733 [14] located in the OSS-PCIe-HIB38-x16 PCIe adapter.

To read/write the PEX 8733/8696 internal registers the application programming interface (API) library provided by the PLX Technology manufacturer was used to implement both system services. The first service to hotplug the PCIe endpoints by monitoring PEX 8696 Lane Status Registers and second service to hotswap the PCIe switch cards by monitoring PEX 8733 Lane Status Registers.

Fig. 4 depicts the implementation that checks if PCIe switch cards, ATCA-PTSW-AMC4, are present in the ATCA prototype. Checking is done by reading PEX 8733 Lane Status Register.

- If PCIe switch card is present, the service checks the number of active lanes (half-full/full) sending an information message to the Linux default system logger (/*var*/*log*/*messages*).
- If a PCIe switch card is not detected, a warning message is sent to the system logger instead and the service checks if the PCIe device node is already installed. If a node is correctly detected, the insert script is called. This procedure is done to install the PCIe device node and corresponding device file in the/*dev* system directory performed by the PCIe endpoints and PCIe switch cards corresponding Linux device drivers.

Insertion procedure for both hotplug and hotswap implementations was organized in two blocks:

- For PCIe endpoints, the insert script is called with rescan instruction executed from a known point in the PCIe bus system and device node is installed in the system with corresponding device file created in/*dev* directory. An entire PCIe bus rescan is not required because it can be done from PEX 8733 ports until the end of the chain in that branch.
- For PCIe switch cards, a rescan to the PCI bus from the OSS card is performed followed by a check of any endpoint node dependency. If any endpoint presence is detected then corresponding insert script is called.

Removal procedure calls PCIe cards removing scripts with checking and validation implemented algorithms to ensure:

- Termination of any running user application using specific physically removed PCIe cards. The scripts try to stop any processes using the nodes to remove and only allow the system to remove the nodes without processes dependence, avoiding kernel crashes. Also, the other nodes keep operational without any interference on its operation;
- Removal of all dependent PCIe endpoint nodes. The remove script procedure for PCIe endpoints, before removing the PCIe switch cards nodes, removes the dependents nodes from them. This is

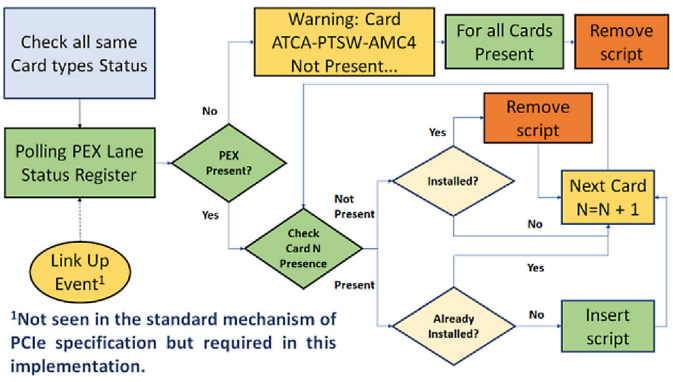

Fig. 4. ATCA system hotplug and hotswap polling services.

done in a complete transparent mode to the ATCA system and users;

Removing PCIe nodes for both PCIe endpoints and PCIe switch cards from the host computer, automatically removes the corresponding device file in/*dev* system directory.

5. Results

The mechanism was developed, implemented and tested in the ATCA system prototype successfully.

Removal procedure was initiated by the ATCA system operator to test the hotplug and hotswap services remove response. In result, the detected physically removed PCIe card node was cleared from host system with corresponding device file uninstalled from/*dev* directory as expected.

Insertion procedure was also initiated by the ATCA system operator to test the hotplug and hotswap services insertion response. In result, the detected physically inserted PCIe card node was installed to the host system with corresponding device file.

The polling periodicity, test with one second, did not cause visible system performance degradation as we could analyze from the Linux system manager.

From this module interface, the memory resource consumption was not significant as expected by a polling service with this time rate, as well as the system CPU load. The host computer Redhat Linux operating system running without any hanging, slowing down issues, halting occurrences or crashing events detected.

6. Conclusions and future work

Polling-based method demonstrated to be valuable with good results according to the initial specification demands. The mechanism worked for both hotplug and hotswap processes but new buses could not be added to the PCI tree. Only the buses enumerated during a cold boot could be hot-plugged or hot-swapped. Nevertheless new devices can be added on a cold-boot detected bus. The Linux kernel contains the concise procedures to enumerate all buses and devices correctly, however, the kernel is not able to add a device after a LINK_UP event, acting only on changes of the link status after a surprise removal. Therefore, the kernel sources can be patched to allow correct re-enumeration after a LINK_UP event, dropping the need for an external re-enumeration module. This method is currently under development. Interrupt-driven method is being studied to compare results with polling-based approach hopping to solve some previous described issues.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 633053. IST activities also received financial support from "Fundação para a Ciência e Tecnologia" through project Pest-OE/SADG/LA0010/2013. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

References

[1] PCI Express® Base Specification, Revision 3.0, November 10, 2010.
[2] PCI Hot-Plug Specification Revision 1.1. http://www.pcisig.com, June 20, 2001.
[3] Integrated Device Technology, Inc., Hotswap in PCIe Based Systems, Application Note AN-701 Preliminary, September 16, 2008.
[4] PICMG, PICMG® 3.0 Revision 3.0 AdvancedTCA® Base Specification, March 24, 2008.
[5] B. Gonçalves, et al., Engineering design of ITER prototype fast plant system, Fusion Eng. Des. 86 (6–8) (2011) 556–560.
[6] AXP1440 Installation and Use, P/N: 6806800H23J, July 2013.
[7] A.J.N. Batista, et al., Control and data acquisition ATCA/AXIe board designed for high system availability and reliability of nuclear fusion experiments, Fusion Eng. Des. 88 (6–8) (2013) 1332–1337.
[8] M. Correia, et al., Implementation of IEEE-1588 timing and synchronization for ATCA control and data acquisition systems, Fusion Eng. Des. 87 (December (12)) (2012) 2178–2181.
[9] M. Correia, et al., ATCA/xTCA-based hardware for control and data acquisition on nuclear fusion fast control plant systems, in: Real Time Conference (RT) 17th IEEE-NPSS, 2010, pp. 1–5.
[10] AXIe 1.0, Base Architecture Specification, June 30, 2010.
[11] PCIMG, PICMG® AMC.0 R2.0 – Advanced Mezzanine Card base specification, November 15, 2006.
[12] Linux Device Drivers Book, 3rd ed., Jonathan Corbet, Alessandro, Rubini, and Greg Kroah-Hartman, OíReilly. http://lwn.net/Kernel/LDD3/
[13] PEX_8696-AA Data Book Version 1.5, 25 January 2013.pdf. Available at PLX Technology website: http://www.plxtech.com/download/file/591
[14] PEX 8733-BA-CA Data Book Version 1.2, 30 September 2013.pdf. Available at PLX Technology website: http://www.plxtech.com/download/file/2239