Communication between Host machine and the Host Bus Adapter over PCIE bus using Diagnostic window

Naveen G[1], Sheetal Kulkarni[2], Naresh Madhusudhana[3], K N Raja Rao[4]

[1]M.Tech Student, Digital Communication Engg. R V College of Engg. Bangalore-59
[2]M.Tech Student, Communication System Engg. R V College of Engg. Bangalore-59
[3]System Application Engineer, LSI India R & D India Private Limited Bangalore-102
[4]Professor, Dept. of Telecommunication Engg. R V College of Engg. Bangalore-59

Abstract

In this paper a study of the PCIE interface and the Host Bus Adapter is carried and demonstrated the communication between the HBA and the Host through a Diagnostic Window Access of PCIE. The API are written in java to access the DDR3 and flash memory on the HBA The diagnostic method is a back door mechanism to access the register and memory on HBA through PCIE. The flash memory is used to store the firmware from which the HBA becomes operational. The Flash memory is accessed and the following operation is carried out. The Flash erase, Flash read, Flash program and Flash QRY program is successfully demonstrated. The DDR3 Serial Presence Detect-SEEPROM is accessed using I2C protocol which is contains vital information about the timing parameter and the manufacturer information of DDR3 memory. The SPD data is used to initiate the training of any DDR3 chip and to make the DDR3 operational.

Keywords: DDR3, Flash Memory, HBA, PCIE**.**

Introduction

The steady increase in the huge volume of data and usage of internet application has motivated several organizations to carry out research in the area of data storage. The storage industry is moving at the faster pace to incorporate high end and high speed technologies in their designs to compete and increase their market size.

The communication is a taking drastic change in its speed and becoming more and more reliable. Sectors like banks, industries, scientific research organization, and social networking sites need huge storage network for data storage for different other applications and even that data should be secured by variety of different methods one such method is RAID (Redundant Array of Independent Disks). In order to communicate with storage network, needs an understanding of how the host communicates with device for example Host Bus Adapter residing on the PCIE (Peripheral Component Interconnect-Express) slot.

PCI Express [1] is the third generation high performance I/O bus that is used for communication between the peripheral devices in the applications such as the storage, communication and the computing platform at a rate of 2.5 Giga bits/second. PCIE implements the dual simplex link which implies that the data can be transmitted to from the device attached to the PCIE simultaneously.

Memory map in the host system

One of the major improvements the PCI Local Bus had over other I/O architectures was its configuration mechanism. In addition to the normal memory-mapped and I/O port spaces, each device on the bus has a configuration space. This is 256 bytes that are addressable by knowing the 8-bit PCI bus, 5-bit device, and 3-bit function numbers for the device (commonly referred to as the BDF bus/device/function). This allows up to 256 buses, each with up to 32 devices, each supporting 8 functions. A single PCI expansion card can respond as a device and must implement at least function number zero. The first 64 bytes of configuration space are standardized; the remainders are available for vendor-defined purposes. A portion of the PCIE configuration register memory map is illustrated in the figure 1.

During the boot process the host machine performs the PCIE Configuration cycles. During this cycle the device attached to the PCIE slot will be uniquely identified by the Bus ID, Device ID and Function ID. Host will allocate the bar register address. This address varies with the system or may change during different boot. The Bar register contain the host memory address from where the device connected will be accessed. Mem0 and Mem1 in a PCIE memory map contain the Base Address Register (BAR) to use in the PCIE memory cycle. BAR will be mapped to Program Inbound Memory (PIM).The size of the Mem1 BAR is programmable, and occupies the specified amount of the possible 2^64 bytes of PCI Memory address space. The entire region maps either to SHELL address space or to PLB Address space of the controller in the HBA connected. The host accessing the device memory map is illustrated in the figure 2.

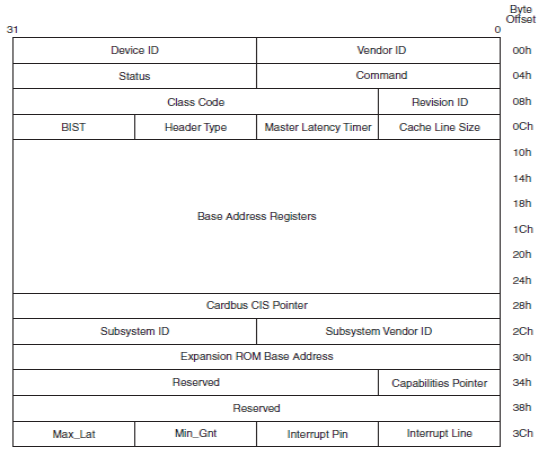

Fig.1 A Portion of the PCIE configuration register memory map for Physical function 0

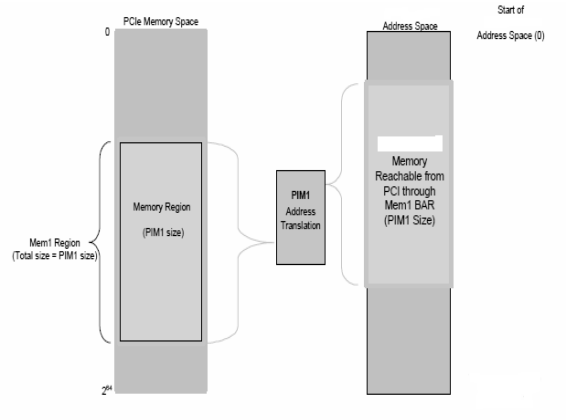

Fig.2 Host accessing the device memory map

PIM Register setup

The PIM registers (Programmable Inbound Map) define a region in the PLB memory map that can be accessed by a corresponding region in the PCIE memory map. PIM0 uses PCI Memory BAR 0 and PIM1 uses PCI Memory BAR 1. By convention, PIM0 maps to the Messaging Unit registers contained in PCIE core. PIM1 is available to provide arbitrary PLB access to the host. i.e. BAR0 for messaging unit registers and BAR1 for PLB access of the device. The PIM registers are defined as a set of Address and Attribute Registers. The Address registers define an arbitrary PLB address to which PCI Slave transactions will be steered. The Attribute Register defines the operation state of the PIM window. The Host interaction with the Flash /DDR3 memory on HBA is illustrated in the figure 3.

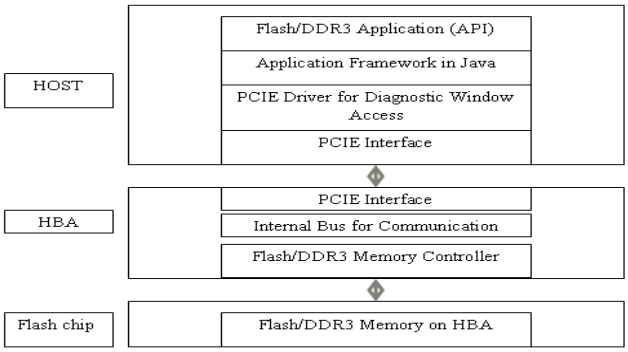

Fig 3: Host Interaction with the Flash /DDR3 Memory on HBA

Methodology

The methodology explains the mapping of the internal memory of HBA can be mapped to host memory

- Enable the Diagnostic Window.
- Use Mem0 to enable the access to diagnostic window. Here we are using PIM0 to configure PIM1.
- In diagnostic window we are writing to the PIM1 address low and PIM1 address high with the PLB base address low and PLB base address high of the region to be accessed.
- Use the Java API's to perform the register read/write, memory read/write operation.

Results and Discussion

COMPILATION ENVIRONMENT

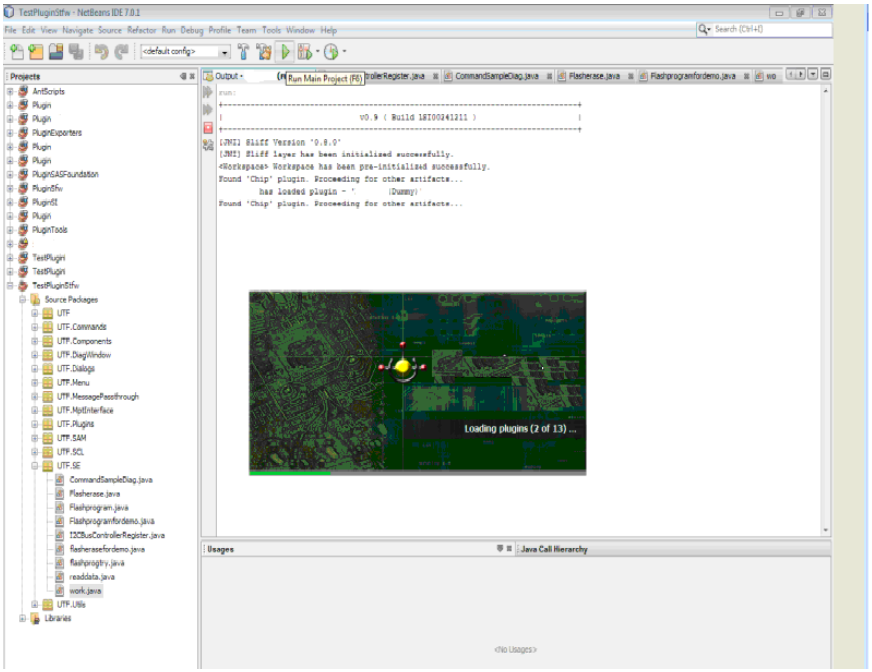

Figure 4 Launching of DDR3/Flash Application

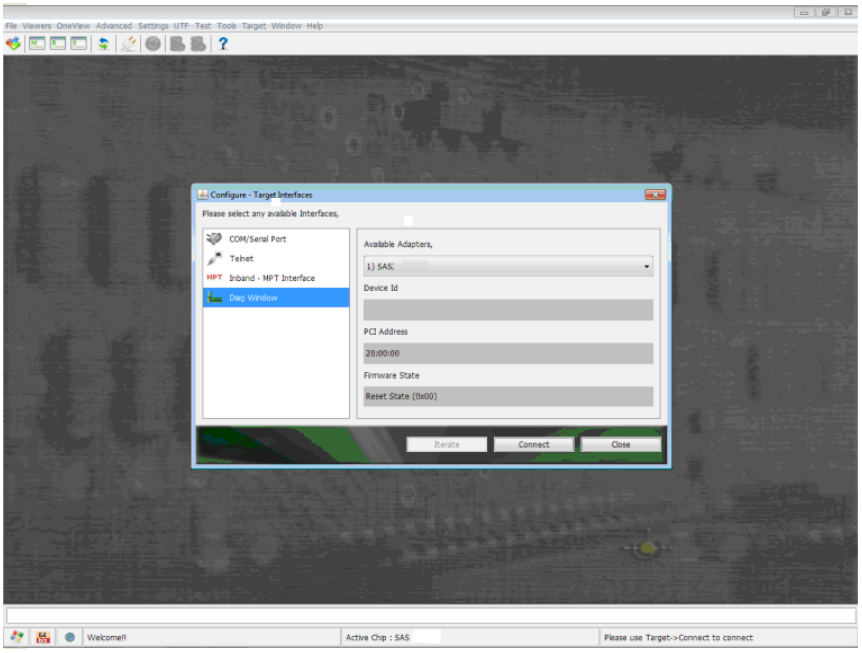

Figure 5 Listing of the device under Diag Window

Flash erase operation

Debug : The Given Command is utf se fler
(0xC20C2200) = 0x3370011F
(0xC20C2200) = 0x3371011F
Verbose : Chip erase successful
(0xFC000000) = 0xFF
All done....

Flash Read After Erase

Debug : File has been saved successfully to 'C:\Users\Administrator\Desktop\Flash Erase.html'
Debug : The Given Command is utf se flrd
Verbose : Readng data from flash memory
(0x0) = 0xFF
(0x1) = 0xFF
(0x2) = 0xFF
(0x3) = 0xFF
All done....

Flash Program

Debug : File has been saved successfully to 'C:\Users\Administrator\Desktop\Flash read after the erase.html'
Debug : The Given Command is utf se flprde
Verbose : Flash operation start
(0xC20C2200) = 0x3371011F
(0xC20C2200) = 0x3371011F
Verbose : Chip program in progress :0xFFFFFFAA
Verbose : Chip program in progress :0x55
Verbose : Chip program in progress :0xFFFFFFA0
Verbose : Chip program in progress :0x40
Read value is '0xBBCC
Verbose : 1st byte programmed
Read value is '0x4433
Verbose : 2nd byte programmed

Flash Read after Programming the region

Debug : File has been saved successfully to 'C:\Users\Administrator\Desktop\Program flash.html'
Debug : The Given Command is utf se flrd
Verbose : Readng data from flash memory
(0x0) = 0xCC
(0x1) = 0xBB
(0x2) = 0x33
(0x3) = 0x44
All done....

2. SPD read from DDR3

The first 12 location of SEEPROM is listed below

```
Verbose : Sequence transfer in progress....
The SEEPROM loc '0x0 0x92
The SEEPROM loc '0x1 0x10
The SEEPROM loc '0x2 0xB
The SEEPROM loc '0x3 0x2
Verbose : Recieve transfer in progress....
Verbose : Transmit transfer in progress....
Verbose : Sequence transfer in progress....
The SEEPROM loc '0x4 0x2
The SEEPROM loc '0x5 0x11
The SEEPROM loc '0x6 0x0
The SEEPROM loc '0x7 0x1
Verbose : Recieve transfer in progress....
Verbose : Transmit transfer in progress....
Verbose : Sequence transfer in progress....
The SEEPROM loc '0x8 0xB
The SEEPROM loc '0x9 0x52
The SEEPROM loc '0xA 0x1
The SEEPROM loc '0xB 0x8
```

The result described above shows the diag window method to access the controller memory and registers on the HBA.

Conclusion:

The host communication with the HBA using the diag window access (without firmware on HBA) is one of the methods widely helpful in the communication with HBA. This method establishes communication between the host and the HBA and is implemented in Java. The communication is demonstrated by performing operations on Flash/DDR3 memory in the HBA.

Acknowledgements:

We thank the Management of LSI India Research and Development India Private Limited, Bangalore and R.V.College of Engineering, Bangalore for providing us an opportunity, resource to carry out the experiment.

References

[1] Ravi Budruk et.al. A text book on PCI Express System Architecture

[2] Standard Development Group http//:www.pcisig.com

[3] PCI Express 3.0 x 8 8GT/s Fusion MPT 2.0 Core Architecture Specification.