User Guide

For

XVSEC Drivers

CONTENTS

LIST OF TABLES

1 Introduction

1.1 Document Overview

Xilinx Vendor Specific Capabilities(XVSEC) are additional features added to PCI Express configuration space. Some of these capabilities includes MCAP, XVC, etc.,

The Media Configuration Access Port (MCAP) is a new configuration interface available for UltraScale and UltraScale+ devices. This interface is integrated into the PCI Express hard block and provides access to the FPGA configuration logic through the PCI Express hard block when enabled. The MCAP can be enabled and used within the PCI Express solution IP for endpoint configurations that use either Tandem or PR over PCIe. The MCAP VSEC can be used to load additional bitstreams and configuration command sequences into the FPGA after the initial configuration has been loaded.

This User Guide provides drivers and software that can be run on a PCI Express root port host PC to interact with the XVSEC features in PCI Express.

The drivers and software referenced in this User Guide are designed for Linux operating systems and can be used for lab testing or as a reference for driver and software development.

1.2 Document References

Document References	Version
[1] https://www.xilinx.com/support/answers/64761.html	11/10/2015

Table 1-1: Document References

1.3 Glossary

Acronym / Term	Description
FPGA	Field Programmable Gate Array
IP	Intellectual Property
MCAP	Media Configuration Access Port
PCI	Peripheral Component Interconnect
PR	Partial Reconfiguration
RAM	Random Access Memory
VSEC	Vendor Specific Capability
XVC	Xilinx Virtual Cable
XVSEC	Xilinx Vendor Specific Capabilities

Table 1-2: Glossary

2 XVSEC Driver for Linux Operating Systems

This User Guide document describes the following for XVSEC Linux Driver that will be generally available for customers:

- Dependencies to be met for using the driver
- Environment to execute the driver
- Instructions for Compiling and loading the driver
- Sample commands to describe the usage of the driver

2.1 Dependencies

The release was tested with the following system configurations.

Directory		Description
Host System Configuration	Operating System	Ubuntu 18.04 LTS, Ubuntu 16.04 LTS
	Linux Kernel	4.15.0-23-generic, 4.4.0-93-generic
	RAM	32GB

Table 2-1: System Configuration

The following kernel functions shall be included in the OS kernel being used. Make sure that these functions are included in the kernel.

- Timer Functions
- PCIe Functions
- Kernel Memory functions
- Memory and GFP Functions

2.2 Environment

To execute the XVSEC driver on example design, following system requirements are to be met:

1. Host System with at least one Gen 3 x16 PCIe slot and minimum 32GB RAM
2. Any one of the Linux OS listed in Table 3-1
3. VCU1525 FPGA Board with JTAG cables
4. Xilinx 2018.3 Vivado tools for programming the FPGA.

2.3 XVSEC Driver Software directory structure

Pull the XVSEC 2018.3 version from GitHub and after uncompressing/pulling the drivers and software files, the provided directories contain the source code for a kernel mode driver, user space library and an example application that demonstrate the capabilities of the XVSEC driver and IP. The steps below describe how to install the driver and must be performed with root permissions.

From now on, the top directory of the software driver source files is referred as "**xvsec_lin_2018.3**". The following list describes the uncompressed directory structure of the software files:

- xvsec_lin_2018.3
  - COPYING
  - LICENSE
  - README
  - RELEASE
  - Makefile
  - docs/
  - drv/
  - libxvsec/
  - tools/

From now onwards all paths are relative to this base directory “xvsec_lin_2018.3”

2.4 Building the XVSEC Driver Software

To compile the Xilinx XVSEC software, a configured and compiled Linux kernel source tree is required. The source tree may be only header files, or a complete tree. The source tree needs to be configured and the header files need to be compiled. And, the Linux kernel must be configured to use modules.

Compile the driver:

- cd into “xvsec_lin_2018.3”

```
[xilinx@] # cd xvsec_lin_2018.3
```

- Build the driver, library and tools

```
[xilinx@] # make clean all
```

A sub-directory build/ will be created in “xvsec_lin_2018.3” after running "make". Once build is completed, executables can be found at following locations

XVSEC kernel driver(xvsec.ko)	build/modules/xvsec.ko
XVSEC user space library	build/lib/libxvsec.a
XVSEC application tool	build/xvsecctl

- To compile the kernel driver

```
[xilinx@] # make drv
```

- To compile user space library

```
[xilinx@] # make libxvsec
```

- To compile application tool

```
[xilinx@] # make tools
```

2.5 Installing the Compiled XVSEC SW binaries

To install the XVSEC driver software, the installer must have the root permission.

Install the driver:

- Enter “xvsec_lin_2018.3”

```
[xilinx@] # sudo make install
```

- The XVSEC module will be installed at below location
  `/lib/modules/<linux_kernel_version>/updates/kernel/drivers/xvsec`
- The “xvsecctl” tool will be installed in
  `/user/local/sbin`

2.6 Loading the QDMA Driver modules

Before loading the XVSEC driver, make sure that an intended board is connected to the Host System and required bitstream is flashed on to the board.

Load the XVSEC driver:

XVSEC driver does not have any module parameters and hence can be loaded directly.

- Load the driver as

```
[xilinx@] # sudo insmod xvsec.ko
```

Now the XVSEC software is ready for use.

2.7 XVSEC functionality usage

2.7.1 Control XVSEC through “xvsecctl” application

XVSEC SW includes a command-line utility called “xvsecctl” to manage the driver which gets built with driver and allows controlling of vendor specific functionality.

It supports the following functionalities:

- Generic VSEC functionality
  - ✓ Lists the supported VSECs by the device
  - ✓ Verbose information about the device

- MCAP VSEC functionality
  - ✓ Configuration Logic Reset
  - ✓ MCAP Module Reset
  - ✓ Full Reset
  - ✓ Dump the MCAP Read Data registers
  - ✓ Dump the MCAP register set
  - ✓ Dump FPGA configuration register set
  - ✓ Access MCAP register of given offset (Read and Write)
  - ✓ Access FPGA configuration register of given register number(Read and Write)
  - ✓ Program the clear bitstream
  - ✓ Program the bitstream
- For help run
  - ./xvsecctl –h
- Refer to Appendix 1 – User Application “xvsecctl” command options for all supported options by application tool

The preliminary version of the driver has the following limitation:

- *Only MCAP VSEC is supported*

2.7.1.1 Example: Get the list of VSECs supported by device

List the devices using lspci to cross check the devices are detected as PCIe devices

```
[xilinx@] # lspci | grep -i Xilinx
65:00.0 Serial controller: Xilinx Corporation Device 9034
```

Enter in to root mode and execute the below command to list the devices.

```
[root@] # ./xvsecctl -b<bus_no> -F <dev_no> -l
[root@] # ./xvsecctl -b 0x65 -F 0x0 -l

No of Supported Extended capabilities : 3
Capability ID : 0x0001, Capability Name : MCAP
Capability ID : 0x0000, Capability Name : UNKNOWN
Capability ID : 0x0008, Capability Name : XVC
```

2.7.1.2 Example: Get the MCAP register info

```
[root@] # ./xvsecctl -b 0x65 -F 0x0 -c 0x1 -d
BYTE OFFSET     Register Name           Data Value
-----------     ----------------        ----------
0x0000          Ext Capability          0x4801000B
0x0004          VSEC Header             0x02C10001
0x0008          FPGA JTAG ID            0x14B31093
0x000C          FPGA BitStream Ver      0x00000000
0x0010          Status                  0x00000004
   bit  0       MCAP Error                       0
   bit  1       MCAP EOS                         0
   bit  4       MCAP Read Complete               0
   bit 5:7      MCAP Read Count                  0
```

```
   bit  8       MCAP FIFO Overflow              0
   bit 12:15    MCAP FIFO Occupancy             0
   bit 24       Req for MCAP Release            0
0x0014          Control                 0x00001000
   bit  0       MCAP Enable                     0
   bit  1       MCAP Read Enable                0
   bit  4       MCAP Reset                      0
   bit  5       MCAP Module Reset               0
   bit  8       Req for MCAP by PCIe            0
   bit 12       MCAP Design Switch              1
   bit 16       Write Data Reg Enable           0
0x0018          FPGA Write Data         0x00000000
0x001C          FPGA Read Data[0]       0xDEADBEEF
0x0020          FPGA Read Data[1]       0x00000000
0x0024          FPGA Read Data[2]       0x00000000
0x0028          FPGA Read Data[3]       0x00000000
```

2.7.1.3 Example: Program the bitstream

```
[root@] # ./xvsecctl -b 0x65 -F 0x0 -c 0x1 -p tandem2.bit
[XVSEC] : xvsec_mcap_configure_fpga : Bitstream Program successful
FPGA configuration successful
```

2.8 Un-installing the XVSEC Driver modules

Standard Linux commands should be used to uninstall.

- Uninstall the XVSEC driver, application tool using the below command.

```
[xilinx@] # make uninstall
```

3 Appendix 1 – User Application "xvsecctl" command options

xvsecctl support common VSEC functionality and MCAP functionality. This section describes the details of each option provided for xvsecctl commands.

Format	Parameter	Description/Range
-h/H	--	Prints the usage of the tool
-b <bus number>	Bus number	PCI bus number on which device sit
-F <device number>	Device number	Device number in the PCI bus
-l	--	List the supported VSECs
-v	--	Prints the verbose information about the device
-c <capability ID>	Capability Identifier	This ID can be identified from
-r	--	MCAP Option to reset the configuration logic area
-m	--	MCAP option to reset the MCAP module
-f	--	MCAP option to reset both configuration logic and MCAP module
-D	--	MCAP option to dump Read Data Registers
-d	--	MCAP option to dump the MCAP registers
-o	--	MCAP option to dump FPGA configuration registers
-a <offset> <access> [<data>]	**offset:** MCAP VSEC offset **access:** b – for byte access h – for short access w – for word access **data:** Data to perform write operation	MCAP option to Read the data at given VSEC offset (or) Writes the given data at given VSEC offset

	NA to perform read operation	
-s <register number> [w] [data]	**register number:** Valid FPGA CFG register number **w** – command which indicates write operation **data:** data to write at given register	MCAP option to Read the data at given register number (or) Writes the data at given register number with given data
-C <partial clear file>	**Partial clear file:** Clear bitstream file to program Supported formats are .bit .bin .rbt	MCAP option to Program the partial clear bitstream
-p <bit stream>	**Bitstream:** Bitstream file to program Supported formats are .bit .bin .rbt	MCAP option to Programs the bitstream

Note:<> is a mandatory parameter to the requested operation

[] is an optional parameter to the requested operation

Except for -p and -C options, all other options are supported one at a time. i.e., It is not suggested to request two options in a single command, in such scenario only option will get executed

4 Appendix 2 – Release Directory Structure

4.1 SW Directory

The entire software source is under *"***xvsec_lin_2018.3"** folder

Directory	Description
xvsec_lin_2018.3	Top-level directory for XVSEC Linux SW driver, User space Library, application tool, and documents
docs/	Documentation for the XVSEC Linux SW
drv/	Kernel driver which manages VSEC functionality
libxvsec/	User Space Library which can be used to integrate XVSEC functionality into the customers application
tools/	User space application to configure and control the QDMA IP Tools to perform DMA operations
Makefile	Make file to compile the Linux XVSEC SW
README	README
RELEASE	Release Document
COPYING	GNU GPL License
LICENSE	BSD License

Table 4-1: SW Directory

5 Document Revision History

Version	Date	Description	State
3	05-Dec-2018	Reviewed and Approved for 2018.3 Release	Released

Table 5-1: Document Review History