Linux XVSEC Driver

2000-0142

Generated by Doxygen 1.8.11

Contents

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

Data Structure Documentation

3.1 args Struct Reference

Data Fields

- uint16_t **bus_no**
- uint16_t **dev_no**
- uint16_t **cap_id**
- bool **parse_err**
- struct help_args **help**
- struct verbose_args **verbose**
- struct list_caps_args **list_caps**
- struct mcap_reset_args **reset**
- struct mcap_module_reset_args **module_reset**
- struct mcap_full_reset_args **full_reset**
- struct mcap_data_dump_args **data_dump**
- struct mcap_reg_dump_args **reg_dump**
- struct fpga_cfg_reg_dump_args **fpga_reg_dump**
- struct mcap_access_reg **access_reg**
- struct fpga_cfg_access_reg **fpga_access_reg**
- struct mcap_program_bitstream **program**

The documentation for this struct was generated from the following file:

- main.h

3.2 bitstream_file Struct Reference

```
#include <xvsec_cdev.h>
```

Data Fields

- char * partial_clr_file
- char * bitstream_file
- enum bitstream_program_status status

3.2.1 Detailed Description

MCAP bitstream parameters for programming.

3.2.2 Field Documentation

3.2.2.1 char* partial_clr_file

Partial clear bitstream file to program ultrascale devices

3.2.2.2 char* bitstream_file

bitstream file to program

3.2.2.3 enum bitstream_program_status status

Status of the bitstream programming

The documentation for this struct was generated from the following file:

- xvsec_cdev.h

3.3 cfg_data Struct Reference

```
#include <xvsec_cdev.h>
```

Data Fields

- char access
- uint16_t offset
- uint32_t data

3.3.1 Detailed Description

MCAP configuration parameters to perform read and writes.

3.3.2 Field Documentation

3.3.2.1 char access

access field. 'b' for byte access, 'h'for half word access, 'w' for word access

3.3.2.2 uint16_t offset

VSEC address offset

3.3.2.3 uint32_t data

data field holds the information to write into the provided offset for Write operation. Holds the information at the provided offset for read operation

The documentation for this struct was generated from the following file:

- xvsec_cdev.h

3.4 context Struct Reference

Data Fields

- dev_t **dev_no**
- int **major_no**
- char **name** [20]
- struct class * **class**
- struct cdev **cdev**
- struct device * **sys_device**
- struct pci_dev * **pdev**
- int **mcap_cap_offset**
- int **fopen_cnt**
- struct xvsec_capabilities **capabilities**

The documentation for this struct was generated from the following file:

- xvsec_cdev_int.h

3.5 device_info Struct Reference

```
#include <xvsec_cdev.h>
```

Data Fields

- uint16_t vendor_id
- uint16_t device_id
- uint16_t device_no
- uint16_t device_fn
- uint16_t subsystem_vendor
- uint16_t subsystem_device
- uint16_t class_id
- uint32_t is_msi_enabled
- uint32_t is_msix_enabled
- int cfg_size

3.5.1 Detailed Description

PCIe device information for verbose option.

3.5.2 Field Documentation

3.5.2.1 uint16_t vendor_id

PCIe Vendor Identifier

3.5.2.2 uint16_t device_id

PCIe Device Identifier

3.5.2.3 uint16_t device_no

PCIe Device number

3.5.2.4 uint16_t device_fn

PCIe Device function

3.5.2.5 uint16_t subsystem_vendor

PCIe Subsystem Vendor Identifier

3.5.2.6 uint16_t subsystem_device

PCIe Subsystem Device Identifier

3.5.2.7 uint16_t class_id

PCIe Class Identifier

3.5.2.8 uint32_t is_msi_enabled

Flag which indicates MSI enabled status

3.5.2.9 uint32_t is_msix_enabled

Flag which indicates MSIx enabled status

3.5.2.10 int cfg_size

Size of the PCIe Device configuration space

The documentation for this struct was generated from the following file:

- xvsec_cdev.h

3.6 file_priv Struct Reference

Data Fields

- void * **ctx**

The documentation for this struct was generated from the following file:

- xvsec_cdev_int.h

3.7 fpga_cfg_access_reg Struct Reference

Data Fields

- bool **flag**
- bool **write**
- char **cmd**
- uint16_t **offset**
- uint32_t **data**

The documentation for this struct was generated from the following file:

- main.h

3.8 fpga_cfg_reg Struct Reference

```
#include <xvsec_cdev.h>
```

Data Fields

- uint16_t offset
- uint32_t data

3.8.1 Detailed Description

FPGA configuration parameters to perform read and writes.

3.8.2 Field Documentation

3.8.2.1 uint16_t offset

FPGA configuration register number

3.8.2.2 uint32_t data

data field holds the information to write into the provided offset for Write operation. Holds the information at the provided offset for read operation

The documentation for this struct was generated from the following file:

- xvsec_cdev.h

3.9 fpga_cfg_reg_dump_args Struct Reference

Data Fields

- bool **flag**
- xvsec_fpga_cfg_regs_t **fpga_cfg_regs**

The documentation for this struct was generated from the following file:

- main.h

3.10 fpga_cfg_regs Struct Reference

```
#include <xvsec_cdev.h>
```

Data Fields

- uint32_t valid
- uint32_t crc
- uint32_t far
- uint32_t fdri
- uint32_t fdro
- uint32_t cmd
- uint32_t ctl0
- uint32_t mask
- uint32_t stat
- uint32_t lout
- uint32_t cor0
- uint32_t mfwr
- uint32_t cbc
- uint32_t idcode
- uint32_t axss
- uint32_t cor1
- uint32_t wbstar
- uint32_t timer
- uint32_t scratchpad
- uint32_t bootsts
- uint32_t ctl1
- uint32_t bspi

3.10.1 Detailed Description

FPGA configuration register set(See UG570 for more information)

3.10.2 Field Documentation

3.10.2.1 uint32_t valid

Valid flag to indicate registers validity

3.10.2.2 uint32_t crc

CRC Register

3.10.2.3 uint32_t far

Frame Address Register

3.10.2.4 uint32_t fdri

Frame Data Register, Input Register (write configuration data)

3.10.2.5 uint32_t fdro

Frame Data Register, Output Register (read configuration data)

3.10.2.6 uint32_t cmd

Command Register

3.10.2.7 uint32_t ctl0

Control Register 0

3.10.2.8 uint32_t mask

Mask Register for CTL0 and CTL1 Registers

3.10.2.9 uint32_t stat

Status Register

3.10.2.10 uint32_t lout

Legacy Output Register for Daisy Chain

3.10.2.11 uint32_t cor0

Configuration Option Register 0

3.10.2.12 uint32_t mfwr

Multi Frame Write Register

3.10.2.13 uint32_t cbc

Initial CBC Value Register

3.10.2.14 uint32_t idcode

Device ID Register

3.10.2.15 uint32_t axss

User Access Register

3.10.2.16 uint32_t cor1

Configuration Option Register 1

3.10.2.17 uint32_t wbstar

Warm Boot Start Address Register

3.10.2.18 uint32_t timer

Watchdog Timer Register

3.10.2.19 uint32_t scratchpad

Scratch Pad Register for Dummy Read and Writes

3.10.2.20 uint32_t bootsts

Boot History Status Register

3.10.2.21 uint32_t ctl1

Control Register 1

3.10.2.22 uint32_t bspi

BPI/SPI Configuration Options Register

The documentation for this struct was generated from the following file:

- xvsec_cdev.h

3.11 handle_t Struct Reference

Data Fields

- uint16_t **xvsec_magic_no**
- uint8_t **bus_no**
- uint8_t **dev_no**
- uint16_t **index**
- bool **valid**

The documentation for this struct was generated from the following file:

- xvsec_int.h

3.12 help_args Struct Reference

Data Fields

- bool **flag**

The documentation for this struct was generated from the following file:

- main.h

3.13 ioctl_ops Struct Reference

Data Fields

- uint32_t **cmd**
- long(∗ **fpfunction**)(struct file ∗filep, uint32_t cmd, unsigned long arg)

The documentation for this struct was generated from the following file:

- xvsec_cdev_int.h

3.14 list_caps_args Struct Reference

Data Fields

- bool **flag**
- xvsec_cap_list_t **cap_list**

The documentation for this struct was generated from the following file:

- main.h

3.15 mcap_access_reg Struct Reference

Data Fields

- bool **flag**
- bool **write**
- char **access_type**
- uint16_t **offset**
- uint32_t **data**

The documentation for this struct was generated from the following file:

- main.h

3.16 mcap_data_dump_args Struct Reference

Data Fields

- bool **flag**
- uint32_t **mcap_data_reg** [4]

The documentation for this struct was generated from the following file:

- main.h

3.17 mcap_full_reset_args Struct Reference

Data Fields

- bool **flag**

The documentation for this struct was generated from the following file:

- main.h

3.18 mcap_module_reset_args Struct Reference

Data Fields

- bool **flag**

The documentation for this struct was generated from the following file:

- main.h

3.19 mcap_ops Struct Reference

Data Fields

- enum mcap_operation **op**
- int(∗ **execute**)(xvsec_handle_t ∗xvsec_handle, struct args ∗args)

The documentation for this struct was generated from the following file:

- mcap_ops.h

3.20 mcap_program_bitstream Struct Reference

Data Fields

- bool **flag**
- char ∗ **abs_clr_file**
- char ∗ **abs_bit_file**

The documentation for this struct was generated from the following file:

- main.h

3.21 mcap_reg_dump_args Struct Reference

Data Fields

- bool **flag**
- xvsec_mcap_regs_t **mcap_regs**

The documentation for this struct was generated from the following file:

- main.h

3.22 mcap_regs Struct Reference

```
#include <xvsec_cdev.h>
```

Data Fields

- uint32_t valid
- uint32_t ext_cap_header
- uint32_t vendor_header
- uint32_t fpga_jtag_id
- uint32_t fpga_bit_ver
- uint32_t status_reg
- uint32_t control_reg
- uint32_t wr_data_reg
- uint32_t rd_data_reg [4]

3.22.1 Detailed Description

MCAP register set.

3.22.2 Field Documentation

3.22.2.1 uint32_t valid

Valid flag to indicate registers validity

3.22.2.2 uint32_t ext_cap_header

Extended capability header register

3.22.2.3 uint32_t vendor_header

Vendor Specific header register

3.22.2.4 uint32_t fpga_jtag_id

FPGA JTAG ID register

3.22.2.5 uint32_t fpga_bit_ver

FPGA bit-stream version register

3.22.2.6 uint32_t status_reg

Status Register

3.22.2.7 uint32_t control_reg

Control Register

3.22.2.8 uint32_t wr_data_reg

Write Data Register

3.22.2.9 uint32_t rd_data_reg[4]

Read Data Register: 4 data words

The documentation for this struct was generated from the following file:

- xvsec_cdev.h

3.23 mcap_reset_args Struct Reference

Data Fields

- bool **flag**

The documentation for this struct was generated from the following file:

- main.h

3.24 type1_header Union Reference

Data Fields

- struct {
    uint32_t **word_count**: 11
    uint32_t **reserved02**: 2
    uint32_t **address**: 5
    uint32_t **reserved01**: 9
    uint32_t **opcode**: 2
    uint32_t **header_type**: 3
  };
- uint32_t **data**

The documentation for this union was generated from the following file:

- xvsec_cdev_int.h

3.25 verbose_args Struct Reference

Data Fields

- bool **flag**

The documentation for this struct was generated from the following file:

- main.h

3.26 xvsec_cap_list_t Struct Reference

```
#include <xvsec.h>
```

Data Fields

- uint16_t no_of_caps
- xvsec_cap_t cap_info [MAX_CAPS_SUPPORTED]

3.26.1 Detailed Description

Capability List.

3.26.2 Field Documentation

3.26.2.1 uint16_t no_of_caps

No of capabilities supported

3.26.2.2 xvsec_cap_t cap_info[MAX_CAPS_SUPPORTED]

Capabilities Information

The documentation for this struct was generated from the following file:

- xvsec.h

3.27 xvsec_cap_t Struct Reference

```
#include <xvsec.h>
```

Data Fields

- uint16_t cap_id
- char cap_name [10]

3.27.1 Detailed Description

Capability Information.

3.27.2 Field Documentation

3.27.2.1 uint16_t cap_id

Capability Identifier

3.27.2.2 char cap_name[10]

Capability Name

The documentation for this struct was generated from the following file:

- xvsec.h

3.28 xvsec_capabilities Struct Reference

```
#include <xvsec_cdev.h>
```

Data Fields

- uint16_t no_of_caps
- uint16_t capability_id [MAX_CAPABILITIES_SUPPORTED]
- uint16_t capability_offset [MAX_CAPABILITIES_SUPPORTED]

3.28.1 Detailed Description

Xilinx Vendor Specific Capabilities.

3.28.2 Field Documentation

3.28.2.1 uint16_t no_of_caps

Number of VSEC capabilities supported by the device

3.28.2.2 uint16_t capability_id[MAX_CAPABILITIES_SUPPORTED]

Capability ID Info

3.28.2.3 uint16_t capability_offset[MAX_CAPABILITIES_SUPPORTED]

Capability Offset Info in PCIe configuration space

The documentation for this struct was generated from the following file:

- xvsec_cdev.h

3.29 xvsec_dev Struct Reference

Data Fields

- uint32_t **dev_cnt**
- struct context * **ctx**

The documentation for this struct was generated from the following file:

- xvsec_cdev_int.h

3.30 xvsec_fpga_cfg_regs_t Struct Reference

```
#include <xvsec.h>
```

Data Fields

- uint32_t crc
- uint32_t far
- uint32_t fdri
- uint32_t fdro
- uint32_t cmd
- uint32_t ctl0
- uint32_t mask
- uint32_t stat
- uint32_t lout
- uint32_t cor0
- uint32_t mfwr
- uint32_t cbc
- uint32_t idcode
- uint32_t axss
- uint32_t cor1
- uint32_t wbstar
- uint32_t timer
- uint32_t scratchpad
- uint32_t bootsts
- uint32_t ctl1
- uint32_t bspi

3.30.1 Detailed Description

FPGA Configuration Register set.

3.30.2 Field Documentation

3.30.2.1 uint32_t crc

CRC Register

3.30.2.2 uint32_t far

Frame Address Register

3.30.2.3 uint32_t fdri

Frame Data Register, Input Register (write configuration data)

3.30.2.4 uint32_t fdro

Frame Data Register, Output Register (read configuration data)

3.30.2.5 uint32_t cmd

Command Register

3.30.2.6 uint32_t ctl0

Control Register 0

3.30.2.7 uint32_t mask

Mask Register for CTL0 and CTL1 Registers

3.30.2.8 uint32_t stat

Status Register

3.30.2.9 uint32_t lout

Legacy Output Register for Daisy Chain

3.30.2.10 uint32_t cor0

Configuration Option Register 0

3.30.2.11 uint32_t mfwr

Multi Frame Write Register

3.30.2.12 uint32_t cbc

Initial CBC Value Register

3.30.2.13 uint32_t idcode

Device ID Register

3.30.2.14 uint32_t axss

User Access Register

3.30.2.15 uint32_t cor1

Configuration Option Register 1

3.30.2.16 uint32_t wbstar

Warm Boot Start Address Register

3.30.2.17 uint32_t timer

Watchdog Timer Register

3.30.2.18 uint32_t scratchpad

Scratch Pad Register for Dummy Read and Writes

3.30.2.19 uint32_t bootsts

Boot History Status Register

3.30.2.20 uint32_t ctl1

Control Register 1

3.30.2.21 uint32_t bspi

BPI/SPI Configuration Options Register

The documentation for this struct was generated from the following file:

- xvsec.h

3.31 xvsec_mcap_ctl_reg_t Struct Reference

`#include <xvsec.h>`

Data Fields

- uint32_t enable: 1
- uint32_t rd_enable: 1
- uint32_t **reserved01**: 2
- uint32_t reset: 1
- uint32_t module_reset: 1
- uint32_t **reserved02**: 2
- uint32_t req4mcap_pcie: 1
- uint32_t **reserved03**: 3
- uint32_t cfg_desgn_sw: 1
- uint32_t **reserved04**: 3
- uint32_t wr_reg_enable: 1
- uint32_t **reserved05**: 15

3.31.1 Detailed Description

MCAP Control Register Fields.

3.31.2 Field Documentation

3.31.2.1 uint32_t enable

MCAP Module Enable

3.31.2.2 uint32_t rd_enable

MCAP Read Enable to perform FPGA CFG Read operation

3.31.2.3 uint32_t reset

MCAP Configurable Region Reset

3.31.2.4 uint32_t module_reset

MCAP Module Reset

3.31.2.5 uint32_t req4mcap_pcie

Request for gaining access to configurable region

3.31.2.6 uint32_t cfg_desgn_sw

MCAP Design Switch : Must be SET after loading bitstream

3.31.2.7 uint32_t wr_reg_enable

MCAP Write Register Enable

The documentation for this struct was generated from the following file:

- xvsec.h

3.32 xvsec_mcap_regs_t Struct Reference

```
#include <xvsec.h>
```

Data Fields

- uint32_t cap_header
- uint32_t vendor_header
- uint32_t fpga_jtag_id
- uint32_t fpga_bitstream_ver
- uint32_t status_reg
- uint32_t control_reg
- uint32_t write_data_reg
- uint32_t read_data_reg [4]

3.32.1 Detailed Description

MCAP Register set.

3.32.2 Field Documentation

3.32.2.1 uint32_t cap_header

Extended capability header register

3.32.2.2 uint32_t vendor_header

Vendor Specific header register

3.32.2.3 uint32_t fpga_jtag_id

FPGA JTAG ID register

3.32.2.4 uint32_t fpga_bitstream_ver

FPGA bit-stream version register

3.32.2.5 uint32_t status_reg

Status Register

3.32.2.6 uint32_t control_reg

Control Register

3.32.2.7 uint32_t write_data_reg

Write Data Register

3.32.2.8 uint32_t read_data_reg[4]

Read Data Register: 4 data words

The documentation for this struct was generated from the following file:

- xvsec.h

3.33 xvsec_mcap_sts_reg_t Struct Reference

`#include <xvsec.h>`

Data Fields

- uint32_t err: 1
- uint32_t eos: 1
- uint32_t **reserved01**: 2
- uint32_t read_complete: 1
- uint32_t read_count: 3
- uint32_t fifo_ovfl: 1
- uint32_t **reserved02**: 3
- uint32_t fifo_occu: 4
- uint32_t **reserved03**: 8
- uint32_t req4mcap_rel: 1
- uint32_t **reserved04**: 7

3.33.1 Detailed Description

MCAP Status Register Fields.

3.33.2 Field Documentation

3.33.2.1 uint32_t err

Error

3.33.2.2 uint32_t eos

End of Startup Signal

3.33.2.3 uint32_t read_complete

MCAP Read complete flag

3.33.2.4 uint32_t read_count

MCAP Read word count

3.33.2.5 uint32_t fifo_ovfl

MCAP Write Buffer FIFO overflow

3.33.2.6 uint32_t fifo_occu

MCAP Write Buffer FIFO Occupency

3.33.2.7 uint32_t req4mcap_rel

MCAP Request for Release Flag

The documentation for this struct was generated from the following file:

- xvsec.h

3.34 xvsec_ops Struct Reference

Data Fields

- enum xvsec_operation **op**
- int(∗ **execute**)(xvsec_handle_t ∗xvsec_handle, struct args ∗args)

The documentation for this struct was generated from the following file:

- main.h

3.35 xvsec_user_context_t Struct Reference

Data Fields

- xvsec_handle_t ∗ **handle**
- int **fd**
- pthread_mutex_t **mutex**

The documentation for this struct was generated from the following file:

- xvsec_int.h

Chapter 4

File Documentation

4.1 xvsec.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <stdbool.h>
#include <string.h>
#include <pthread.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <sys/ioctl.h>
#include <byteswap.h>
```

Data Structures

- struct xvsec_cap_t
- struct xvsec_cap_list_t
- struct xvsec_mcap_sts_reg_t
- struct xvsec_mcap_ctl_reg_t
- struct xvsec_mcap_regs_t
- struct xvsec_fpga_cfg_regs_t

Macros

- #define XVSEC_SUCCESS (0)
- #define XVSEC_FAILURE (-1)
- #define XVSEC_ERR_NULL_POINTER (-2)
- #define XVSEC_ERR_INVALID_PARAM (-3)
- #define XVSEC_ERR_MUTEX_LOCK_FAIL (-4)
- #define XVSEC_ERR_MUTEX_UNLOCK_FAIL (-5)
- #define XVSEC_ERR_MEM_ALLOC_FAILED (-6)
- #define XVSEC_ERR_OPERATION_NOT_SUPPORTED (-7)

- #define XVSEC_ERR_BITSTREAM_PROGRAM (-8)
- #define XVSEC_ERR_LINUX_SYSTEM_CALL (-9)
- #define XVSEC_MAX_DEVICES_LIMIT_REACHED (-10)
- #define XVSEC_ERR_INVALID_FILE_FORMAT (-11)
- #define XVSEC_ERR_INVALID_OFFSET_ACCESS_COMBO (-12)
- #define XVSEC_ERR_CAPABILITY_ID_MISSING (-13)
- #define XVSEC_ERR_CAPABILITY_NOT_SUPPORTED (-13)
- #define XVSEC_ERR_INVALID_FPGA_REG_NUM (-14)
- #define XVSEC_ERR_INVALID_OFFSET (-15)
- #define MAX_CAPS_SUPPORTED (10)

Typedefs

- typedef uint64_t xvsec_handle_t
- typedef enum access_type_t **access_type_t**
- typedef struct xvsec_cap_t **xvsec_cap_t**
- typedef struct xvsec_cap_list_t **xvsec_cap_list_t**
- typedef struct xvsec_mcap_sts_reg_t **xvsec_mcap_sts_reg_t**
- typedef struct xvsec_mcap_ctl_reg_t **xvsec_mcap_ctl_reg_t**
- typedef struct xvsec_mcap_regs_t **xvsec_mcap_regs_t**
- typedef struct xvsec_fpga_cfg_regs_t **xvsec_fpga_cfg_regs_t**

Enumerations

- enum access_type_t {
  ACCESS_BYTE = 0,
  ACCESS_SHORT,
  ACCESS_WORD }

Functions

- int xvsec_lib_init (int max_devices)
- int xvsec_lib_deinit (void)
- int xvsec_open (uint16_t bus_no, uint16_t dev_no, xvsec_handle_t *handle)
- int xvsec_close (xvsec_handle_t *handle)
- int xvsec_get_cap_list (xvsec_handle_t *handle, xvsec_cap_list_t *cap_list)
- int xvsec_show_device (xvsec_handle_t *handle)
- int xvsec_mcap_reset (xvsec_handle_t *handle)
- int xvsec_mcap_module_reset (xvsec_handle_t *handle)
- int xvsec_mcap_full_reset (xvsec_handle_t *handle)
- int xvsec_mcap_get_data_registers (xvsec_handle_t *handle, uint32_t data[4])
- int xvsec_mcap_get_registers (xvsec_handle_t *handle, xvsec_mcap_regs_t *mcap_regs)
- int xvsec_mcap_get_fpga_registers (xvsec_handle_t *handle, xvsec_fpga_cfg_regs_t *fpga_cfg_regs)
- int xvsec_mcap_access_config_reg (xvsec_handle_t *handle, uint16_t offset, void *data, access_type_t access, bool write)
- int xvsec_mcap_access_fpga_config_reg (xvsec_handle_t *handle, uint16_t offset, void *data, bool write)
- int xvsec_mcap_configure_fpga (xvsec_handle_t *handle, char *partial_cfg_file, char *bitfile)

Variables

- const char * error_codes [ ]

4.1.1 Detailed Description

This file contains the interface declarations of XVSEC User Space Library.

4.1.2 Macro Definition Documentation

4.1.2.1 #define XVSEC_SUCCESS (0)

XVSEC API Return Value : Indicates Success of API call

4.1.2.2 #define XVSEC_FAILURE (-1)

XVSEC API Return Value : Indicates Failure of API call

4.1.2.3 #define XVSEC_ERR_NULL_POINTER (-2)

XVSEC API Return Value : Indicates one or more NULL pointer passed as parameter

4.1.2.4 #define XVSEC_ERR_INVALID_PARAM (-3)

XVSEC API Return Value : Indicates one or more invalid parameters passed

4.1.2.5 #define XVSEC_ERR_MUTEX_LOCK_FAIL (-4)

XVSEC API Return Value : Indicates mutex lock failed

4.1.2.6 #define XVSEC_ERR_MUTEX_UNLOCK_FAIL (-5)

XVSEC API Return Value : Indicates mutex unlock failed while trying to release

4.1.2.7 #define XVSEC_ERR_MEM_ALLOC_FAILED (-6)

XVSEC API Return Value : Indicates memory allocation failed

4.1.2.8 #define XVSEC_ERR_OPERATION_NOT_SUPPORTED (-7)

XVSEC API Return Value : Indicates the requested operation is not supported

4.1.2.9 #define XVSEC_ERR_BITSTREAM_PROGRAM (-8)

XVSEC API Return Value : Indicates BitStream programming Failed

4.1.2.10 #define XVSEC_ERR_LINUX_SYSTEM_CALL (-9)

XVSEC API Return Value : Indicates one of the linux system call Failed

4.1.2.11 #define XVSEC_MAX_DEVICES_LIMIT_REACHED (-10)

XVSEC API Return Value : Indicates No more devices present to open

4.1.2.12 #define XVSEC_ERR_INVALID_FILE_FORMAT (-11)

XVSEC API Return Value : Indicates unsupported file format provided

4.1.2.13 #define XVSEC_ERR_INVALID_OFFSET_ACCESS_COMBO (-12)

XVSEC API Return Value : Indicates Offset and Access combination is incorrect (Ex : u16 needs an even offset, u32 needs an offset divisible by 4

4.1.2.14 #define XVSEC_ERR_CAPABILITY_ID_MISSING (-13)

XVSEC API Return Value : Indicates Capability ID is not provided

4.1.2.15 #define XVSEC_ERR_CAPABILITY_NOT_SUPPORTED (-13)

XVSEC API Return Value : Indicates Capability ID is not supported

4.1.2.16 #define XVSEC_ERR_INVALID_FPGA_REG_NUM (-14)

XVSEC API Return Value : Indicates FPGA CFG Register number provided is invalid (or) access to the requested register is prohibited

4.1.2.17 #define XVSEC_ERR_INVALID_OFFSET (-15)

XVSEC API Return Value : Indicates Provided VSEC Offset is invalid

4.1.2.18 #define MAX_CAPS_SUPPORTED (10)

Maximum supported Capability ID by the Library

4.1.3 Typedef Documentation

4.1.3.1 typedef uint64_t xvsec_handle_t

Unique XVSEC handle per device. This handle is needed for all XVSEC operations

4.1.4 Enumeration Type Documentation

4.1.4.1 enum access_type_t

Register/Offset Access Type(Byte/Short/Word)

Enumerator

ACCESS_BYTE 8 bits access (read/write)

ACCESS_SHORT 16 bits access (read/write)

ACCESS_WORD 32 bits access (read/write)

4.1.5 Function Documentation

4.1.5.1 int xvsec_lib_init (int *max_devices*)

xvsec_lib_init() - Initializes the XVSEC Library by allocating memory to support requested number of devices

Parameters

in	*max_devices*	Maximum devices to support

Returns

XVSEC_SUCCESS : Success
XVSEC_ERR_MEM_ALLOC_FAILED : Failure
XVSEC_ERR_LINUX_SYSTEM_CALL : Filure

4.1.5.2 int xvsec_lib_deinit (void)

xvsec_lib_deinit() - De-initializes the XVSEC Library by freeing allocated memory and clearing the context

Parameters

none	

Returns

XVSEC_SUCCESS : Success
XVSEC_ERR_LINUX_SYSTEM_CALL : Failure

4.1.5.3 int xvsec_open (uint16_t *bus_no*, uint16_t *dev_no*, xvsec_handle_t * *handle*)

xvsec_open() - Opens XVSEC character device which is dedicated to the given bus number and device number and returns a unique handle to access the device

Parameters

in	*bus_no*	PCIe bus number on which device sits
in	*dev_no*	Device number in the PCIe bus
out	*handle*	Unique handle returned to access the device

Returns

XVSEC_SUCCESS : Success
XVSEC_ERR_INVALID_PARAM : Failure
XVSEC_MAX_DEVICES_LIMIT_REACHED : Failure
XVSEC_ERR_LINUX_SYSTEM_CALL : Failure
XVSEC_ERR_OPERATION_NOT_SUPPORTED : Failure

4.1.5.4 int xvsec_close (xvsec_handle_t * *handle*)

xvsec_close() - Closes XVSEC character device of provided handle

Parameters

in	*handle*	Unique handle to access the device

Returns

XVSEC_SUCCESS : Success
XVSEC_ERR_NULL_POINTER : Failure
XVSEC_ERR_OPERATION_NOT_SUPPORTED : Failure
XVSEC_ERR_LINUX_SYSTEM_CALL : Failure

4.1.5.5 int xvsec_get_cap_list (xvsec_handle_t * *handle*, xvsec_cap_list_t * *cap_list*)

xvsec_get_cap_list() - Returns the supported VSEC capabilities of the given handle

Parameters

in	*handle*	Unique handle to access the device
out	*cap_list*	Supported capability list

Returns

XVSEC_SUCCESS : Success
XVSEC_ERR_INVALID_PARAM : Failure
XVSEC_ERR_OPERATION_NOT_SUPPORTED : Failure
XVSEC_ERR_LINUX_SYSTEM_CALL : Failure

4.1.5.6 int xvsec_show_device (xvsec_handle_t * *handle*)

xvsec_show_device() - Shows the device information of the given handle

Parameters

in	*handle*	Unique handle to access the device

Returns

XVSEC_SUCCESS : Success
XVSEC_ERR_INVALID_PARAM : Failure
XVSEC_ERR_OPERATION_NOT_SUPPORTED : Failure
XVSEC_ERR_LINUX_SYSTEM_CALL : Failure

4.1.5.7 int xvsec_mcap_reset (xvsec_handle_t * *handle*)

xvsec_mcap_reset() - Resets the configuration logic of the given handle

Parameters

in	*handle*	Unique handle to access the device

Returns

XVSEC_SUCCESS : Success
XVSEC_ERR_INVALID_PARAM : Failure
XVSEC_ERR_OPERATION_NOT_SUPPORTED : Failure
XVSEC_ERR_LINUX_SYSTEM_CALL : Failure

4.1.5.8 int xvsec_mcap_module_reset (xvsec_handle_t * *handle*)

xvsec_mcap_module_reset() - Resets the MCAP module of the given handle

Parameters

in	*handle*	Unique handle to access the device

Returns

XVSEC_SUCCESS : Success
XVSEC_ERR_INVALID_PARAM : Failure
XVSEC_ERR_OPERATION_NOT_SUPPORTED : Failure
XVSEC_ERR_LINUX_SYSTEM_CALL : Failure

4.1.5.9 int xvsec_mcap_full_reset (xvsec_handle_t * *handle*)

xvsec_mcap_full_reset() - Resets bothconfiguration logic & MCAP module of the given handle

Parameters

in	*handle*	Unique handle to access the device

Returns

XVSEC_SUCCESS : Success
XVSEC_ERR_INVALID_PARAM : Failure
XVSEC_ERR_OPERATION_NOT_SUPPORTED : Failure
XVSEC_ERR_LINUX_SYSTEM_CALL : Failure

4.1.5.10 int xvsec_mcap_get_data_registers (xvsec_handle_t * *handle*, uint32_t *data[4]*)

xvsec_mcap_get_data_registers() - Returns the MCAP read data registers

Parameters

in	*handle*	Unique handle to access the device
out	*data[]*	MCAP read data register values

Returns

XVSEC_SUCCESS : Success
XVSEC_ERR_INVALID_PARAM : Failure
XVSEC_ERR_OPERATION_NOT_SUPPORTED : Failure
XVSEC_ERR_LINUX_SYSTEM_CALL : Failure

4.1.5.11 int xvsec_mcap_get_registers (xvsec_handle_t * *handle*, xvsec_mcap_regs_t * *mcap_regs*)

xvsec_mcap_get_registers() - Returns the MCAP register set

Parameters

in	*handle*	Unique handle to access the device
out	*mcap_regs*	MCAP register values

Returns

XVSEC_SUCCESS : Success
XVSEC_ERR_INVALID_PARAM : Failure
XVSEC_ERR_OPERATION_NOT_SUPPORTED : Failure
XVSEC_ERR_LINUX_SYSTEM_CALL : Failure

4.1.5.12 int xvsec_mcap_get_fpga_registers (xvsec_handle_t * *handle*, xvsec_fpga_cfg_regs_t * *fpga_cfg_regs*)

xvsec_mcap_get_fpga_registers() - Returns the FPGA configuration register set

Parameters

in	*handle*	Unique handle to access the device
out	*fpga_cfg_regs*	FPGA configuration register values

Returns

XVSEC_SUCCESS : Success
XVSEC_ERR_INVALID_PARAM : Failure
XVSEC_ERR_OPERATION_NOT_SUPPORTED : Failure
XVSEC_ERR_LINUX_SYSTEM_CALL : Failure

4.1.5.13 int xvsec_mcap_access_config_reg (xvsec_handle_t * *handle*, uint16_t *offset*, void * *data*, access_type_t *access*, bool *write*)

xvsec_mcap_access_config_reg() - Performs read/write operations on the MCAP register set

Parameters

in	*handle*	Unique handle to access the device
in	*offset*	Register address offset from MCAP Base address

4.1.5.14 int xvsec_mcap_access_fpga_config_reg (xvsec_handle_t * *handle*, uint16_t *offset*, void * *data*, bool *write*)

xvsec_mcap_access_fpga_config_reg() - Performs read/write operations on the FPGA configuration register set

Parameters

in	*handle*	Unique handle to access the device
in	*offset*	Register address offset from MCAP Base address

4.1.5.15 int xvsec_mcap_configure_fpga (xvsec_handle_t * *handle*, char * *partial_cfg_file*, char * *bitfile*)

xvsec_mcap_configure_fpga() - Performs bitstream programming on FPGA

Parameters

in	*handle*	Unique handle to access the device
in	*partial_cfg_file*	Partial Clear bitstream file
in	*bitfile*	Bitstream file

Returns

XVSEC_SUCCESS : Success
XVSEC_ERR_INVALID_PARAM : Failure
XVSEC_ERR_OPERATION_NOT_SUPPORTED : Failure
XVSEC_ERR_LINUX_SYSTEM_CALL : Failure
XVSEC_ERR_INVALID_FPGA_REG_NUM : Failure

4.1.6 Variable Documentation

4.1.6.1 const char* error_codes[ ]

Error codes in human readable form

4.2 xvsec_cdev.h File Reference

Data Structures

- struct mcap_regs
- struct bitstream_file
- struct cfg_data
- struct fpga_cfg_reg
- struct fpga_cfg_regs
- struct device_info
- struct xvsec_capabilities

Macros

- #define XVSEC_IOC_MAGIC 'm'
- #define XVSEC_MINOR_BASE (0)
- #define XVSEC_MINOR_COUNT (1)
- #define XILINX_VENDOR_ID (uint16_t)(0x10ee)
- #define MCAP_EXT_CAP_ID (uint16_t)(0x000B)
- #define MAX_CAPABILITIES_SUPPORTED 10
- #define IOC_XVSEC_GET_CAP_LIST _IOW(XVSEC_IOC_MAGIC, 0, struct xvsec_capabilities *)
- #define IOC_MCAP_RESET _IO(XVSEC_IOC_MAGIC, 1)
- #define IOC_MCAP_MODULE_RESET _IO(XVSEC_IOC_MAGIC, 2)
- #define IOC_MCAP_FULL_RESET _IO(XVSEC_IOC_MAGIC, 3)
- #define IOC_MCAP_GET_DATA_REGISTERS _IOR(XVSEC_IOC_MAGIC, 4, uint32_t *)
- #define IOC_MCAP_GET_REGISTERS _IOR(XVSEC_IOC_MAGIC, 5, struct mcap_regs *)
- #define IOC_MCAP_GET_FPGA_REGISTERS _IOR(XVSEC_IOC_MAGIC, 6, struct mcap_regs *)
- #define IOC_MCAP_PROGRAM_BITSTREAM _IOWR(XVSEC_IOC_MAGIC, 7, struct bitstream_file *)
- #define IOC_MCAP_READ_DEV_CFG_REG _IOWR(XVSEC_IOC_MAGIC, 8, struct cfg_data *)
- #define IOC_MCAP_WRITE_DEV_CFG_REG _IOWR(XVSEC_IOC_MAGIC, 9, struct cfg_data *)
- #define IOC_MCAP_READ_FPGA_CFG_REG _IOWR(XVSEC_IOC_MAGIC, 10, struct fpga_cfg_reg *)
- #define IOC_MCAP_WRITE_FPGA_CFG_REG _IOWR(XVSEC_IOC_MAGIC, 11, struct fpga_cfg_reg *)
- #define IOC_GET_DEVICE_INFO _IOWR(XVSEC_IOC_MAGIC, 12, struct device_info *)

Enumerations

- enum bitstream_program_status { MCAP_BITSTREAM_PROGRAM_SUCCESS = 0, MCAP_BITSTREAM_PROGRAM_FAILURE = 1 }

4.2.1 Detailed Description

This file contains the interface declarations of XVSEC driver.

4.2.2 Macro Definition Documentation

4.2.2.1 #define XVSEC_IOC_MAGIC 'm'

XVSEC ioctl magic character

4.2.2.2 #define XVSEC_MINOR_BASE (0)

XVSEC char device first minor number

4.2.2.3 #define XVSEC_MINOR_COUNT (1)

XVSEC char device total minor numbers count

4.2.2.4 #define XILINX_VENDOR_ID (uint16_t)(0x10ee)

XILINX PCIe vendor ID

4.2.2.5 #define MCAP_EXT_CAP_ID (uint16_t)(0x000B)

XILINX PCIe vendor specific capability ID

4.2.2.6 #define MAX_CAPABILITIES_SUPPORTED 10

Maximum Supported capabilities by the driver

4.2.2.7 #define IOC_XVSEC_GET_CAP_LIST _IOW(XVSEC_IOC_MAGIC, 0, struct xvsec_capabilities *)

ioctl code for retrieving the XVSEC capability list

4.2.2.8 #define IOC_MCAP_RESET _IO(XVSEC_IOC_MAGIC, 1)

ioctl code for performing MCAP configuration logic reset

4.2.2.9 #define IOC_MCAP_MODULE_RESET _IO(XVSEC_IOC_MAGIC, 2)

ioctl code for performing MCAP Module reset

4.2.2.10 #define IOC_MCAP_FULL_RESET _IO(XVSEC_IOC_MAGIC, 3)

ioctl code for performing both configuration logic & Module reset

4.2.2.11 #define IOC_MCAP_GET_DATA_REGISTERS _IOR(XVSEC_IOC_MAGIC, 4, uint32_t *)

ioctl code for retrieving the MCAP Read Data Registers

4.2.2.12 #define IOC_MCAP_GET_REGISTERS _IOR(XVSEC_IOC_MAGIC, 5, struct mcap_regs *)

ioctl code for retrieving the MCAP Registers

4.2.2.13 #define IOC_MCAP_GET_FPGA_REGISTERS _IOR(XVSEC_IOC_MAGIC, 6, struct mcap_regs *)

ioctl code for retrieving the FPGA configuration Registers

4.2.2.14 #define IOC_MCAP_PROGRAM_BITSTREAM _IOWR(XVSEC_IOC_MAGIC, 7, struct bitstream_file *)

ioctl code for programming the bitstream

4.2.2.15 #define IOC_MCAP_READ_DEV_CFG_REG _IOWR(XVSEC_IOC_MAGIC, 8, struct cfg_data *)

ioctl code for reading an MCAP VSEC register

4.2.2.16 #define IOC_MCAP_WRITE_DEV_CFG_REG _IOWR(XVSEC_IOC_MAGIC, 9, struct cfg_data *)

ioctl code for Writing to an MCAP VSEC register

4.2.2.17 #define IOC_MCAP_READ_FPGA_CFG_REG _IOWR(XVSEC_IOC_MAGIC, 10, struct fpga_cfg_reg *)

ioctl code for reading an FPGA CFG register

4.2.2.18 #define IOC_MCAP_WRITE_FPGA_CFG_REG _IOWR(XVSEC_IOC_MAGIC, 11, struct fpga_cfg_reg *)

ioctl code for writing to an FPGA CFG register

4.2.2.19 #define IOC_GET_DEVICE_INFO _IOWR(XVSEC_IOC_MAGIC, 12, struct device_info *)

ioctl code for retrieving the Device information

4.2.3 Enumeration Type Documentation

4.2.3.1 enum bitstream_program_status

program status(Success/Failure)

Enumerator

MCAP_BITSTREAM_PROGRAM_SUCCESS Programming Success Indication

MCAP_BITSTREAM_PROGRAM_FAILURE Programming Failure Indication

Index