

Linux XVSEC Driver

2000-0142

Generated by Doxygen 1.8.11

Contents

1	Data Structure Index	1
1.1	Data Structures	1
2	File Index	3
2.1	File List	3
3	Data Structure Documentation	5
3.1	args Struct Reference	5
3.2	bitstream_file Struct Reference	5
3.2.1	Detailed Description	6
3.2.2	Field Documentation	6
3.2.2.1	partial_clr_file	6
3.2.2.2	bitstream_file	6
3.2.2.3	status	6
3.3	cfg_data Struct Reference	6
3.3.1	Detailed Description	6
3.3.2	Field Documentation	6
3.3.2.1	access	6
3.3.2.2	offset	7
3.3.2.3	data	7
3.4	context Struct Reference	7
3.5	device_info Struct Reference	7
3.5.1	Detailed Description	8
3.5.2	Field Documentation	8

3.5.2.1	vendor_id	8
3.5.2.2	device_id	8
3.5.2.3	device_no	8
3.5.2.4	device_fn	8
3.5.2.5	subsystem_vendor	8
3.5.2.6	subsystem_device	8
3.5.2.7	class_id	8
3.5.2.8	is_msi_enabled	8
3.5.2.9	is_msix_enabled	8
3.5.2.10	cfg_size	9
3.6	file_priv Struct Reference	9
3.7	fpga_cfg_access_reg Struct Reference	9
3.8	fpga_cfg_reg Struct Reference	9
3.8.1	Detailed Description	10
3.8.2	Field Documentation	10
3.8.2.1	offset	10
3.8.2.2	data	10
3.9	fpga_cfg_reg_dump_args Struct Reference	10
3.10	fpga_cfg_regs Struct Reference	10
3.10.1	Detailed Description	11
3.10.2	Field Documentation	11
3.10.2.1	valid	11
3.10.2.2	crc	11
3.10.2.3	far	11
3.10.2.4	fdri	11
3.10.2.5	fdro	12
3.10.2.6	cmd	12
3.10.2.7	ctl0	12
3.10.2.8	mask	12
3.10.2.9	stat	12

3.10.2.10 lout	12
3.10.2.11 cor0	12
3.10.2.12 mfwr	12
3.10.2.13 cbc	12
3.10.2.14 idcode	12
3.10.2.15 axss	13
3.10.2.16 cor1	13
3.10.2.17 wbstar	13
3.10.2.18 timer	13
3.10.2.19 scratchpad	13
3.10.2.20 bootsts	13
3.10.2.21 ctl1	13
3.10.2.22 bspi	13
3.11 handle_t Struct Reference	14
3.12 help_args Struct Reference	14
3.13 ioctl_ops Struct Reference	14
3.14 list_caps_args Struct Reference	14
3.15 mcap_access_reg Struct Reference	15
3.16 mcap_data_dump_args Struct Reference	15
3.17 mcap_full_reset_args Struct Reference	15
3.18 mcap_module_reset_args Struct Reference	15
3.19 mcap_ops Struct Reference	16
3.20 mcap_program_bitstream Struct Reference	16
3.21 mcap_reg_dump_args Struct Reference	16
3.22 mcap_regs Struct Reference	16
3.22.1 Detailed Description	17
3.22.2 Field Documentation	17
3.22.2.1 valid	17
3.22.2.2 ext_cap_header	17
3.22.2.3 vendor_header	17

3.22.2.4	fpga_jtag_id	17
3.22.2.5	fpga_bit_ver	17
3.22.2.6	status_reg	17
3.22.2.7	control_reg	18
3.22.2.8	wr_data_reg	18
3.22.2.9	rd_data_reg	18
3.23	mcap_reset_args Struct Reference	18
3.24	type1_header Union Reference	18
3.25	verbose_args Struct Reference	19
3.26	xvsec_cap_list_t Struct Reference	19
3.26.1	Detailed Description	19
3.26.2	Field Documentation	19
3.26.2.1	no_of_caps	19
3.26.2.2	cap_info	19
3.27	xvsec_cap_t Struct Reference	19
3.27.1	Detailed Description	20
3.27.2	Field Documentation	20
3.27.2.1	cap_id	20
3.27.2.2	cap_name	20
3.28	xvsec_capabilities Struct Reference	20
3.28.1	Detailed Description	20
3.28.2	Field Documentation	20
3.28.2.1	no_of_caps	20
3.28.2.2	capability_id	21
3.28.2.3	capability_offset	21
3.29	xvsec_dev Struct Reference	21
3.30	xvsec_fpga_cfg_regs_t Struct Reference	21
3.30.1	Detailed Description	22
3.30.2	Field Documentation	22
3.30.2.1	crc	22

3.30.2.2	far	22
3.30.2.3	fdri	22
3.30.2.4	fdro	22
3.30.2.5	cmd	22
3.30.2.6	ctl0	22
3.30.2.7	mask	22
3.30.2.8	stat	22
3.30.2.9	lout	22
3.30.2.10	cor0	23
3.30.2.11	mfwr	23
3.30.2.12	cbc	23
3.30.2.13	idcode	23
3.30.2.14	axss	23
3.30.2.15	cor1	23
3.30.2.16	wbstar	23
3.30.2.17	timer	23
3.30.2.18	scratchpad	23
3.30.2.19	bootsts	23
3.30.2.20	ctl1	24
3.30.2.21	bspi	24
3.31	xvsec_mcap_ctl_reg_t Struct Reference	24
3.31.1	Detailed Description	24
3.31.2	Field Documentation	24
3.31.2.1	enable	24
3.31.2.2	rd_enable	24
3.31.2.3	reset	25
3.31.2.4	module_reset	25
3.31.2.5	req4mcap_pcie	25
3.31.2.6	cfg_desgn_sw	25
3.31.2.7	wr_reg_enable	25

3.32 xvsec_mcap_regs_t Struct Reference	25
3.32.1 Detailed Description	25
3.32.2 Field Documentation	26
3.32.2.1 cap_header	26
3.32.2.2 vendor_header	26
3.32.2.3 fpga_jtag_id	26
3.32.2.4 fpga_bitstream_ver	26
3.32.2.5 status_reg	26
3.32.2.6 control_reg	26
3.32.2.7 write_data_reg	26
3.32.2.8 read_data_reg	26
3.33 xvsec_mcap_sts_reg_t Struct Reference	26
3.33.1 Detailed Description	27
3.33.2 Field Documentation	27
3.33.2.1 err	27
3.33.2.2 eos	27
3.33.2.3 read_complete	27
3.33.2.4 read_count	27
3.33.2.5 fifo_ovfl	27
3.33.2.6 fifo_occu	27
3.33.2.7 req4mcap_rel	28
3.34 xvsec_ops Struct Reference	28
3.35 xvsec_user_context_t Struct Reference	28

4 File Documentation	29
4.1 xvsec.h File Reference	29
4.1.1 Detailed Description	31
4.1.2 Macro Definition Documentation	31
4.1.2.1 XVSEC_SUCCESS	31
4.1.2.2 XVSEC_FAILURE	31
4.1.2.3 XVSEC_ERR_NULL_POINTER	31
4.1.2.4 XVSEC_ERR_INVALID_PARAM	31
4.1.2.5 XVSEC_ERR_MUTEX_LOCK_FAIL	31
4.1.2.6 XVSEC_ERR_MUTEX_UNLOCK_FAIL	31
4.1.2.7 XVSEC_ERR_MEM_ALLOC_FAILED	31
4.1.2.8 XVSEC_ERR_OPERATION_NOT_SUPPORTED	31
4.1.2.9 XVSEC_ERR_BITSTREAM_PROGRAM	31
4.1.2.10 XVSEC_ERR_LINUX_SYSTEM_CALL	32
4.1.2.11 XVSEC_MAX_DEVICES_LIMIT_REACHED	32
4.1.2.12 XVSEC_ERR_INVALID_FILE_FORMAT	32
4.1.2.13 XVSEC_ERR_INVALID_OFFSET_ACCESS_COMBO	32
4.1.2.14 XVSEC_ERR_CAPABILITY_ID_MISSING	32
4.1.2.15 XVSEC_ERR_CAPABILITY_NOT_SUPPORTED	32
4.1.2.16 XVSEC_ERR_INVALID_FPGA_REG_NUM	32
4.1.2.17 XVSEC_ERR_INVALID_OFFSET	32
4.1.2.18 MAX_CAPS_SUPPORTED	32
4.1.3 Typedef Documentation	32
4.1.3.1 xvsec_handle_t	32
4.1.4 Enumeration Type Documentation	33
4.1.4.1 access_type_t	33
4.1.5 Function Documentation	33
4.1.5.1 xvsec_lib_init(int max_devices)	33
4.1.5.2 xvsec_lib_deinit(void)	33
4.1.5.3 xvsec_open(uint16_t bus_no, uint16_t dev_no, xvsec_handle_t *handle)	33

4.1.5.4	xvsec_close(xvsec_handle_t *handle)	34
4.1.5.5	xvsec_get_cap_list(xvsec_handle_t *handle, xvsec_cap_list_t *cap_list)	34
4.1.5.6	xvsec_show_device(xvsec_handle_t *handle)	34
4.1.5.7	xvsec_mcap_reset(xvsec_handle_t *handle)	35
4.1.5.8	xvsec_mcap_module_reset(xvsec_handle_t *handle)	35
4.1.5.9	xvsec_mcap_full_reset(xvsec_handle_t *handle)	35
4.1.5.10	xvsec_mcap_get_data_registers(xvsec_handle_t *handle, uint32_t data[4])	36
4.1.5.11	xvsec_mcap_get_registers(xvsec_handle_t *handle, xvsec_mcap_regs_t *mcap_regs)	36
4.1.5.12	xvsec_mcap_get_fpga_registers(xvsec_handle_t *handle, xvsec_fpga_cfg_regs_t *fpga_cfg_regs)	36
4.1.5.13	xvsec_mcap_access_config_reg(xvsec_handle_t *handle, uint16_t offset, void *data, access_type_t access, bool write)	37
4.1.5.14	xvsec_mcap_access_fpga_config_reg(xvsec_handle_t *handle, uint16_t offset, void *data, bool write)	37
4.1.5.15	xvsec_mcap_configure_fpga(xvsec_handle_t *handle, char *partial_cfg_file, char *bitfile)	37
4.1.6	Variable Documentation	38
4.1.6.1	error_codes	38
4.2	xvsec_cdev.h File Reference	38
4.2.1	Detailed Description	39
4.2.2	Macro Definition Documentation	39
4.2.2.1	XVSEC_IOC_MAGIC	39
4.2.2.2	XVSEC_MINOR_BASE	39
4.2.2.3	XVSEC_MINOR_COUNT	39
4.2.2.4	XILINX_VENDOR_ID	39
4.2.2.5	MCAP_EXT_CAP_ID	39
4.2.2.6	MAX_CAPABILITIES_SUPPORTED	39
4.2.2.7	IOC_XVSEC_GET_CAP_LIST	39
4.2.2.8	IOC_MCAP_RESET	40
4.2.2.9	IOC_MCAP_MODULE_RESET	40
4.2.2.10	IOC_MCAP_FULL_RESET	40
4.2.2.11	IOC_MCAP_GET_DATA_REGISTERS	40
4.2.2.12	IOC_MCAP_GET_REGISTERS	40
4.2.2.13	IOC_MCAP_GET_FPGA_REGISTERS	40
4.2.2.14	IOC_MCAP_PROGRAM_BITSTREAM	40
4.2.2.15	IOC_MCAP_READ_DEV_CFG_REG	40
4.2.2.16	IOC_MCAP_WRITE_DEV_CFG_REG	40
4.2.2.17	IOC_MCAP_READ_FPGA_CFG_REG	40
4.2.2.18	IOC_MCAP_WRITE_FPGA_CFG_REG	41
4.2.2.19	IOC_GET_DEVICE_INFO	41
4.2.3	Enumeration Type Documentation	41
4.2.3.1	bitstream_program_status	41

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

args	5
bitstream_file MCAP bitstream parameters for programming	5
cfg_data MCAP configuration parameters to perform read and writes	6
context	7
device_info PCIe device information for verbose option	7
file_priv	9
fpga_cfg_access_reg	9
fpga_cfg_reg FPGA configuration parameters to perform read and writes	9
fpga_cfg_reg_dump_args	10
fpga_cfg_regs FPGA configuration register set(See UG570 for more information)	10
handle_t	14
help_args	14
ioctl_ops	14
list_caps_args	14
mcap_access_reg	15
mcap_data_dump_args	15
mcap_full_reset_args	15
mcap_module_reset_args	15
mcap_ops	16
mcap_program_bitstream	16
mcap_reg_dump_args	16
mcap_regs MCAP register set	16
mcap_reset_args	18
type1_header	18
verbose_args	19
xvsec_cap_list_t Capability List	19
xvsec_cap_t Capability Information	19

xvsec_capabilities	
Xilinx Vendor Specific Capabilities	20
xvsec_dev	21
xvsec_fpga_cfg_regs_t	
FPGA Configuration Register set	21
xvsec_mcap_ctl_reg_t	
MCAP Control Register Fields	24
xvsec_mcap_regs_t	
MCAP Register set	25
xvsec_mcap_sts_reg_t	
MCAP Status Register Fields	26
xvsec_ops	28
xvsec_user_context_t	28

Chapter 2

File Index

2.1 File List

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

main.h	??
mcaps_ops.h	??
drv/version.h	??
libxvsec/version.h	??
tools/version.h	??
xvsec.h	
This file contains the interface declarations of XVSEC User Space Library	29
xvsec_cdev.h	
This file contains the interface declarations of XVSEC driver	38
xvsec_cdev_int.h	??
xvsec_int.h	??

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 bspl

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_design_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 Occupancy

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	<i>max_devices</i>	Maximum devices to support
----	--------------------	----------------------------

Returns

XVSEC_SUCCESS : Success
 XVSEC_ERR_MEM_ALLOC_FAILED : Failure
 XVSEC_ERR_LINUX_SYSTEM_CALL : Failure

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

<i>none</i>	
-------------	--

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	<i>bus_no</i>	PCIe bus number on which device sits
in	<i>dev_no</i>	Device number in the PCIe bus
out	<i>handle</i>	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	<i>handle</i>	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	<i>handle</i>	Unique handle to access the device
out	<i>cap_list</i>	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	<i>handle</i>	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	<i>handle</i>	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	<i>handle</i>	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 both configuration logic & MCAP module of the given handle

Parameters

in	<i>handle</i>	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	<i>handle</i>	Unique handle to access the device
out	<i>data[]</i>	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	<i>handle</i>	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	<i>handle</i>	Unique handle to access the device
out	<i>fpga_cfg_regs</i>	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	<i>handle</i>	Unique handle to access the device
in	<i>offset</i>	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	<i>handle</i>	Unique handle to access the device
in	<i>offset</i>	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	<i>handle</i>	Unique handle to access the device
in	<i>partial_cfg_file</i>	Partial Clear bitstream file
in	<i>bitfile</i>	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 fpga_cfg_reg *)`
- `#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

ACCESS_BYTE
 xvsec.h, [33](#)
ACCESS_SHORT
 xvsec.h, [33](#)
ACCESS_WORD
 xvsec.h, [33](#)
access
 cfg_data, [6](#)
access_type_t
 xvsec.h, [33](#)
args, [5](#)
axss
 fpga_cfg_regs, [12](#)
 xvsec_fpga_cfg_regs_t, [23](#)

bitstream_file, [5](#)
 bitstream_file, [6](#)
 partial_clr_file, [6](#)
 status, [6](#)
bitstream_program_status
 xvsec_cdev.h, [41](#)
bootsts
 fpga_cfg_regs, [13](#)
 xvsec_fpga_cfg_regs_t, [23](#)
bspi
 fpga_cfg_regs, [13](#)
 xvsec_fpga_cfg_regs_t, [24](#)

cap_header
 xvsec_mcap_regs_t, [26](#)
cap_id
 xvsec_cap_t, [20](#)
cap_info
 xvsec_cap_list_t, [19](#)
cap_name
 xvsec_cap_t, [20](#)
capability_id
 xvsec_capabilities, [20](#)
capability_offset
 xvsec_capabilities, [21](#)
cbc
 fpga_cfg_regs, [12](#)
 xvsec_fpga_cfg_regs_t, [23](#)
cfg_data, [6](#)
 access, [6](#)
 data, [7](#)
 offset, [6](#)
cfg_desgn_sw
 xvsec_mcap_ctl_reg_t, [25](#)
cfg_size
 device_info, [8](#)
class_id
 device_info, [8](#)
cmd
 fpga_cfg_regs, [12](#)
 xvsec_fpga_cfg_regs_t, [22](#)
context, [7](#)
control_reg
 mcap_regs, [17](#)
 xvsec_mcap_regs_t, [26](#)
cor0
 fpga_cfg_regs, [12](#)
 xvsec_fpga_cfg_regs_t, [22](#)
cor1
 fpga_cfg_regs, [13](#)
 xvsec_fpga_cfg_regs_t, [23](#)
crc
 fpga_cfg_regs, [11](#)
 xvsec_fpga_cfg_regs_t, [22](#)
ctl0
 fpga_cfg_regs, [12](#)
 xvsec_fpga_cfg_regs_t, [22](#)
ctl1
 fpga_cfg_regs, [13](#)
 xvsec_fpga_cfg_regs_t, [23](#)

data
 cfg_data, [7](#)
 fpga_cfg_reg, [10](#)
device_fn
 device_info, [8](#)
device_id
 device_info, [8](#)
device_info, [7](#)
 cfg_size, [8](#)
 class_id, [8](#)
 device_fn, [8](#)
 device_id, [8](#)
 device_no, [8](#)
 is_msi_enabled, [8](#)
 is_msix_enabled, [8](#)
 subsystem_device, [8](#)
 subsystem_vendor, [8](#)
 vendor_id, [8](#)
device_no
 device_info, [8](#)

enable
 xvsec_mcap_ctl_reg_t, [24](#)
eos

- xvsec_mcap_sts_reg_t, [27](#)
- err
 - xvsec_mcap_sts_reg_t, [27](#)
- error_codes
 - xvsec.h, [38](#)
- ext_cap_header
 - mcap_regs, [17](#)
- far
 - fpga_cfg_regs, [11](#)
 - xvsec_fpga_cfg_regs_t, [22](#)
- fdri
 - fpga_cfg_regs, [11](#)
 - xvsec_fpga_cfg_regs_t, [22](#)
- fdro
 - fpga_cfg_regs, [11](#)
 - xvsec_fpga_cfg_regs_t, [22](#)
- fifo_occu
 - xvsec_mcap_sts_reg_t, [27](#)
- fifo_ovfl
 - xvsec_mcap_sts_reg_t, [27](#)
- file_priv, [9](#)
- fpga_bit_ver
 - mcap_regs, [17](#)
- fpga_bitstream_ver
 - xvsec_mcap_regs_t, [26](#)
- fpga_cfg_access_reg, [9](#)
- fpga_cfg_reg, [9](#)
 - data, [10](#)
 - offset, [10](#)
- fpga_cfg_reg_dump_args, [10](#)
- fpga_cfg_regs, [10](#)
 - axss, [12](#)
 - bootsts, [13](#)
 - bspi, [13](#)
 - cbc, [12](#)
 - cmd, [12](#)
 - cor0, [12](#)
 - cor1, [13](#)
 - crc, [11](#)
 - ctl0, [12](#)
 - ctl1, [13](#)
 - far, [11](#)
 - fdri, [11](#)
 - fdro, [11](#)
 - idcode, [12](#)
 - lout, [12](#)
 - mask, [12](#)
 - mfwr, [12](#)
 - scratchpad, [13](#)
 - stat, [12](#)
 - timer, [13](#)
 - valid, [11](#)
 - wbstar, [13](#)
- fpga_jtag_id
 - mcap_regs, [17](#)
 - xvsec_mcap_regs_t, [26](#)
- handle_t, [14](#)
- help_args, [14](#)
- IOC_GET_DEVICE_INFO
 - xvsec_cdev.h, [41](#)
- IOC_MCAP_FULL_RESET
 - xvsec_cdev.h, [40](#)
- IOC_MCAP_GET_DATA_REGISTERS
 - xvsec_cdev.h, [40](#)
- IOC_MCAP_GET_FPGA_REGISTERS
 - xvsec_cdev.h, [40](#)
- IOC_MCAP_GET_REGISTERS
 - xvsec_cdev.h, [40](#)
- IOC_MCAP_MODULE_RESET
 - xvsec_cdev.h, [40](#)
- IOC_MCAP_PROGRAM_BITSTREAM
 - xvsec_cdev.h, [40](#)
- IOC_MCAP_READ_DEV_CFG_REG
 - xvsec_cdev.h, [40](#)
- IOC_MCAP_READ_FPGA_CFG_REG
 - xvsec_cdev.h, [40](#)
- IOC_MCAP_RESET
 - xvsec_cdev.h, [39](#)
- IOC_MCAP_WRITE_DEV_CFG_REG
 - xvsec_cdev.h, [40](#)
- IOC_MCAP_WRITE_FPGA_CFG_REG
 - xvsec_cdev.h, [40](#)
- IOC_XVSEC_GET_CAP_LIST
 - xvsec_cdev.h, [39](#)
- idcode
 - fpga_cfg_regs, [12](#)
 - xvsec_fpga_cfg_regs_t, [23](#)
- ioctl_ops, [14](#)
- is_msi_enabled
 - device_info, [8](#)
- is_msix_enabled
 - device_info, [8](#)
- list_caps_args, [14](#)
- lout
 - fpga_cfg_regs, [12](#)
 - xvsec_fpga_cfg_regs_t, [22](#)
- MAX_CAPABILITIES_SUPPORTED
 - xvsec_cdev.h, [39](#)
- MAX_CAPS_SUPPORTED
 - xvsec.h, [32](#)
- MCAP_BITSTREAM_PROGRAM_FAILURE
 - xvsec_cdev.h, [41](#)
- MCAP_BITSTREAM_PROGRAM_SUCCESS
 - xvsec_cdev.h, [41](#)
- MCAP_EXT_CAP_ID
 - xvsec_cdev.h, [39](#)
- mask
 - fpga_cfg_regs, [12](#)
 - xvsec_fpga_cfg_regs_t, [22](#)
- mcap_access_reg, [15](#)
- mcap_data_dump_args, [15](#)
- mcap_full_reset_args, [15](#)
- mcap_module_reset_args, [15](#)

- mcap_ops, [16](#)
- mcap_program_bitstream, [16](#)
- mcap_reg_dump_args, [16](#)
- mcap_regs, [16](#)
 - control_reg, [17](#)
 - ext_cap_header, [17](#)
 - fpga_bit_ver, [17](#)
 - fpga_htag_id, [17](#)
 - rd_data_reg, [18](#)
 - status_reg, [17](#)
 - valid, [17](#)
 - vendor_header, [17](#)
 - wr_data_reg, [18](#)
- mcap_reset_args, [18](#)
- mfwr
 - fpga_cfg_regs, [12](#)
 - xvsec_fpga_cfg_regs_t, [23](#)
- module_reset
 - xvsec_mcap_ctl_reg_t, [25](#)
- no_of_caps
 - xvsec_cap_list_t, [19](#)
 - xvsec_capabilities, [20](#)
- offset
 - cfg_data, [6](#)
 - fpga_cfg_reg, [10](#)
- partial_clr_file
 - bitstream_file, [6](#)
- rd_data_reg
 - mcap_regs, [18](#)
- rd_enable
 - xvsec_mcap_ctl_reg_t, [24](#)
- read_complete
 - xvsec_mcap_sts_reg_t, [27](#)
- read_count
 - xvsec_mcap_sts_reg_t, [27](#)
- read_data_reg
 - xvsec_mcap_regs_t, [26](#)
- req4mcap_pcie
 - xvsec_mcap_ctl_reg_t, [25](#)
- req4mcap_rel
 - xvsec_mcap_sts_reg_t, [27](#)
- reset
 - xvsec_mcap_ctl_reg_t, [24](#)
- scratchpad
 - fpga_cfg_regs, [13](#)
 - xvsec_fpga_cfg_regs_t, [23](#)
- stat
 - fpga_cfg_regs, [12](#)
 - xvsec_fpga_cfg_regs_t, [22](#)
- status
 - bitstream_file, [6](#)
- status_reg
 - mcap_regs, [17](#)
 - xvsec_mcap_regs_t, [26](#)
- subsystem_device
 - device_info, [8](#)
- subsystem_vendor
 - device_info, [8](#)
- timer
 - fpga_cfg_regs, [13](#)
 - xvsec_fpga_cfg_regs_t, [23](#)
- type1_header, [18](#)
- valid
 - fpga_cfg_regs, [11](#)
 - mcap_regs, [17](#)
- vendor_header
 - mcap_regs, [17](#)
 - xvsec_mcap_regs_t, [26](#)
- vendor_id
 - device_info, [8](#)
- verbose_args, [19](#)
- wbstar
 - fpga_cfg_regs, [13](#)
 - xvsec_fpga_cfg_regs_t, [23](#)
- wr_data_reg
 - mcap_regs, [18](#)
- wr_reg_enable
 - xvsec_mcap_ctl_reg_t, [25](#)
- write_data_reg
 - xvsec_mcap_regs_t, [26](#)
- XILINX_VENDOR_ID
 - xvsec_cdev.h, [39](#)
- XVSEC_ERR_BITSTREAM_PROGRAM
 - xvsec.h, [31](#)
- XVSEC_ERR_CAPABILITY_ID_MISSING
 - xvsec.h, [32](#)
- XVSEC_ERR_CAPABILITY_NOT_SUPPORTED
 - xvsec.h, [32](#)
- XVSEC_ERR_INVALID_FILE_FORMAT
 - xvsec.h, [32](#)
- XVSEC_ERR_INVALID_FPGA_REG_NUM
 - xvsec.h, [32](#)
- XVSEC_ERR_INVALID_OFFSET_ACCESS_COMBO
 - xvsec.h, [32](#)
- XVSEC_ERR_INVALID_OFFSET
 - xvsec.h, [32](#)
- XVSEC_ERR_INVALID_PARAM
 - xvsec.h, [31](#)
- XVSEC_ERR_LINUX_SYSTEM_CALL
 - xvsec.h, [31](#)
- XVSEC_ERR_MEM_ALLOC_FAILED
 - xvsec.h, [31](#)
- XVSEC_ERR_MUTEX_LOCK_FAIL
 - xvsec.h, [31](#)
- XVSEC_ERR_MUTEX_UNLOCK_FAIL
 - xvsec.h, [31](#)
- XVSEC_ERR_NULL_POINTER
 - xvsec.h, [31](#)
- XVSEC_ERR_OPERATION_NOT_SUPPORTED

- xvsec.h, 31
- XVSEC_FAILURE
 - xvsec.h, 31
- XVSEC_IOC_MAGIC
 - xvsec_cdev.h, 39
- XVSEC_MAX_DEVICES_LIMIT_REACHED
 - xvsec.h, 32
- XVSEC_MINOR_BASE
 - xvsec_cdev.h, 39
- XVSEC_MINOR_COUNT
 - xvsec_cdev.h, 39
- XVSEC_SUCCESS
 - xvsec.h, 31
- xvsec.h, 29
 - ACCESS_BYTE, 33
 - ACCESS_SHORT, 33
 - ACCESS_WORD, 33
 - access_type_t, 33
 - error_codes, 38
 - MAX_CAPS_SUPPORTED, 32
 - XVSEC_ERR_BITSTREAM_PROGRAM, 31
 - XVSEC_ERR_CAPABILITY_ID_MISSING, 32
 - XVSEC_ERR_CAPABILITY_NOT_SUPPORTED, 32
 - XVSEC_ERR_INVALID_FILE_FORMAT, 32
 - XVSEC_ERR_INVALID_FPGA_REG_NUM, 32
 - XVSEC_ERR_INVALID_OFFSET_ACCESS_C←OMBO, 32
 - XVSEC_ERR_INVALID_OFFSET, 32
 - XVSEC_ERR_INVALID_PARAM, 31
 - XVSEC_ERR_LINUX_SYSTEM_CALL, 31
 - XVSEC_ERR_MEM_ALLOC_FAILED, 31
 - XVSEC_ERR_MUTEX_LOCK_FAIL, 31
 - XVSEC_ERR_MUTEX_UNLOCK_FAIL, 31
 - XVSEC_ERR_NULL_POINTER, 31
 - XVSEC_ERR_OPERATION_NOT_SUPPORTED, 31
 - XVSEC_FAILURE, 31
 - XVSEC_MAX_DEVICES_LIMIT_REACHED, 32
 - XVSEC_SUCCESS, 31
- xvsec_close, 34
- xvsec_get_cap_list, 34
- xvsec_handle_t, 32
- xvsec_lib_deinit, 33
- xvsec_lib_init, 33
- xvsec_mcap_access_config_reg, 37
- xvsec_mcap_access_fpga_config_reg, 37
- xvsec_mcap_configure_fpga, 37
- xvsec_mcap_full_reset, 35
- xvsec_mcap_get_data_registers, 36
- xvsec_mcap_get_fpga_registers, 36
- xvsec_mcap_get_registers, 36
- xvsec_mcap_module_reset, 35
- xvsec_mcap_reset, 35
- xvsec_open, 33
- xvsec_show_device, 34
- xvsec_cap_list_t, 19
 - cap_info, 19
 - no_of_caps, 19
- xvsec_cap_t, 19
 - cap_id, 20
 - cap_name, 20
- xvsec_capabilities, 20
 - capability_id, 20
 - capability_offset, 21
 - no_of_caps, 20
- xvsec_cdev.h, 38
 - bitstream_program_status, 41
 - IOC_GET_DEVICE_INFO, 41
 - IOC_MCAP_FULL_RESET, 40
 - IOC_MCAP_GET_DATA_REGISTERS, 40
 - IOC_MCAP_GET_FPGA_REGISTERS, 40
 - IOC_MCAP_GET_REGISTERS, 40
 - IOC_MCAP_MODULE_RESET, 40
 - IOC_MCAP_PROGRAM_BITSTREAM, 40
 - IOC_MCAP_READ_DEV_CFG_REG, 40
 - IOC_MCAP_READ_FPGA_CFG_REG, 40
 - IOC_MCAP_RESET, 39
 - IOC_MCAP_WRITE_DEV_CFG_REG, 40
 - IOC_MCAP_WRITE_FPGA_CFG_REG, 40
 - IOC_XVSEC_GET_CAP_LIST, 39
 - MAX_CAPABILITIES_SUPPORTED, 39
 - MCAP_BITSTREAM_PROGRAM_FAILURE, 41
 - MCAP_BITSTREAM_PROGRAM_SUCCESS, 41
 - MCAP_EXT_CAP_ID, 39
 - XILINX_VENDOR_ID, 39
 - XVSEC_IOC_MAGIC, 39
 - XVSEC_MINOR_BASE, 39
 - XVSEC_MINOR_COUNT, 39
- xvsec_close
 - xvsec.h, 34
- xvsec_dev, 21
- xvsec_fpga_cfg_regs_t, 21
 - axss, 23
 - bootsts, 23
 - bspi, 24
 - cbc, 23
 - cmd, 22
 - cor0, 22
 - cor1, 23
 - crc, 22
 - ctl0, 22
 - ctl1, 23
 - far, 22
 - fdri, 22
 - fdro, 22
 - idcode, 23
 - lout, 22
 - mask, 22
 - mfwr, 23
 - scratchpad, 23
 - stat, 22
 - timer, 23
 - wbstar, 23
- xvsec_get_cap_list
 - xvsec.h, 34

- xvsec_handle_t
 - xvsec.h, [32](#)
- xvsec_lib_deinit
 - xvsec.h, [33](#)
- xvsec_lib_init
 - xvsec.h, [33](#)
- xvsec_mcap_access_config_reg
 - xvsec.h, [37](#)
- xvsec_mcap_access_fpga_config_reg
 - xvsec.h, [37](#)
- xvsec_mcap_configure_fpga
 - xvsec.h, [37](#)
- xvsec_mcap_ctl_reg_t, [24](#)
 - cfg_desgn_sw, [25](#)
 - enable, [24](#)
 - module_reset, [25](#)
 - rd_enable, [24](#)
 - req4mcap_pcie, [25](#)
 - reset, [24](#)
 - wr_reg_enable, [25](#)
- xvsec_mcap_full_reset
 - xvsec.h, [35](#)
- xvsec_mcap_get_data_registers
 - xvsec.h, [36](#)
- xvsec_mcap_get_fpga_registers
 - xvsec.h, [36](#)
- xvsec_mcap_get_registers
 - xvsec.h, [36](#)
- xvsec_mcap_module_reset
 - xvsec.h, [35](#)
- xvsec_mcap_regs_t, [25](#)
 - cap_header, [26](#)
 - control_reg, [26](#)
 - fpga_bitstream_ver, [26](#)
 - fpga_jtag_id, [26](#)
 - read_data_reg, [26](#)
 - status_reg, [26](#)
 - vendor_header, [26](#)
 - write_data_reg, [26](#)
- xvsec_mcap_reset
 - xvsec.h, [35](#)
- xvsec_mcap_sts_reg_t, [26](#)
 - eos, [27](#)
 - err, [27](#)
 - fifo_occu, [27](#)
 - fifo_ovfl, [27](#)
 - read_complete, [27](#)
 - read_count, [27](#)
 - req4mcap_rel, [27](#)
- xvsec_open
 - xvsec.h, [33](#)
- xvsec_ops, [28](#)
- xvsec_show_device
 - xvsec.h, [34](#)
- xvsec_user_context_t, [28](#)