FTL Media Driver for CFI and SFDP SPI NOR User Guide

Version 1.50

For use with FTL Media Driver for CFI and SFDP SPI NOR versions 1.13 and above

Date: 18-Aug-2017 12:34

Table of Contents

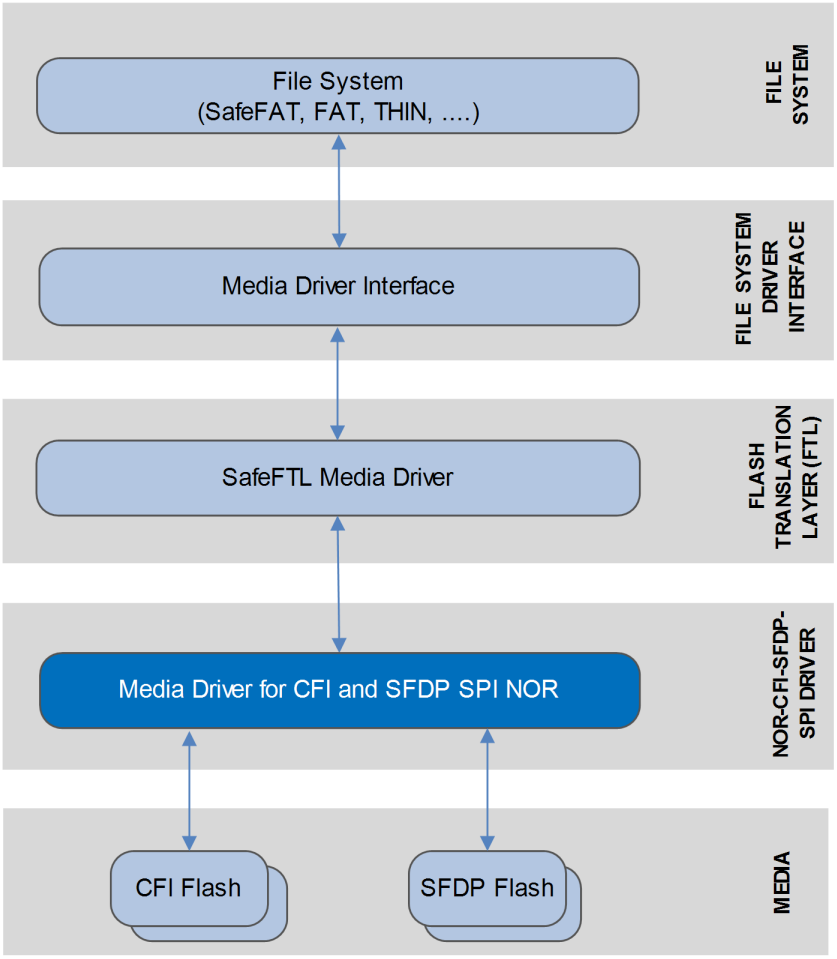
1 System Overview

1.1 Introduction

This guide is for those who wish to implement a NOR flash driver, to which multiple NOR flash drivers of the following types can be connected:

- Common Flash memory Interface (CFI) – an open standard for flash interfaces.
- Serial Flash Discoverable Parameter (SFDP) – a standard for serial NOR flash devices.

The diagram below shows a typical system architecture including a file system, media driver and media. As an example, this shows two drives of each type attached.



This guide covers all aspects of configuration and use. Read it thoroughly before implementing a driver.

This NOR flash driver fully conforms to the *HCC Media Driver Interface Specification*.

1.2 Feature Check

For a full list of SafeFTL features, see the *HCC SafeFTL User Guide*.

The features especially relevant to this system are as follows:

- Supports NOR flash connected by Serial Peripheral Interface (SPI).
- Supports parallel connected NOR flash (CFI only).
- Supports all common NOR devices.
- Can handle any array of NOR flash.
- The flash can be divided into two parts.
- Supports the Common Flash memory Interface (CFI) standard for NOR flash devices.
- Supports the Serial Flash Discoverable Parameter (SFDP) standard for serial NOR flash devices.

1.3 Packages and Documents

Packages

The table below lists the packages that you need in order to use this module:

Package	Description
media_drv_base	The base media driver package that include the framework for all media drivers to use.
media_drv_ftl_base	The base SafeFTL package.
media_drv_ftl_nor_base	The base flash driver used by all NOR flash devices. This enables the NOR flash to be managed by SafeFTL.
media_drv_ftl_nor_cfi_sfdp_spi	The media driver for CFI and SFDP SPI NOR flash package described in this document.
psp_template_spi	The SPI Platform Support Package (PSP).

Documents

For an overview of HCC file systems and flash management technologies, see Product Information on the main HCC website.

Readers should note the points in the HCC Documentation Guidelines on the HCC documentation website.

HCC Firmware Quick Start Guide

This document describes how to install packages provided by HCC in the target development environment. Also follow the *Quick Start Guide* when HCC provides package updates.

HCC Source Tree Guide

This document describes the HCC source tree. It gives an overview of the system to make clear the logic behind its organization.

HCC Media Driver Interface Guide

Specification for upper layer interface that the SafeFTL uses so that SafeFTL can be used as a set of drives by any file system using this media driver interface standard.

HCC SafeFTL User Guide

The base SafeFTL document.

HCC FTL NOR Base Flash Driver User Guide

This describes the NOR base flash driver that handles the NOR flash driver.

HCC FTL Media Driver for CFI and SFDP SPI NOR User Guide

This is this document.

1.4 Change History

This section describes past changes to this manual.

- To view or download earlier manuals, see Archive: FTL Media Driver for CFI and SFDP SPI NOR User Guide.
- For the history of changes made to the package code itself, see History: media_drv_ftl_nor_cfi_sfdp_spi.

The current version of this manual is 1.50. The full list of versions is as follows:

Manual version	Date	Software version	Reason for change
1.50	2017-08-18	1.13	Updated *Packages* list.
1.40	2017-06-21	1.13	New *Change History* format.
1.30	2016-01-05	1.13	Added legacy mode configuration option.
1.20	2015-09-22	1.11	Added new configuration options.
1.10	2015-05-08	1.10	Various small changes.
1.00	2015-04-27	1.10	First online version.

2 Source File List

This section describes all the source code files included in the system. These files follow the HCC Embedded standard source tree system, described in the *HCC Source Tree Guide*. All references to file pathnames refer to locations within this standard source tree, not within the package you initially receive.

> **Note:** Do not modify any files except the configuration file and PSP files.

2.1 API Header File

The file **src/api/api_ftl_nor_cfi_sfdp.h** is the only file that should be included by an application using this module. For details of the single API function, see Application Programming Interface.

2.2 Configuration File

The file **src/config/config_ftl_nor_cfi_sfdp.h** contains all the configurable parameters. Configure these as required. For details of these options, see Configuration Options.

2.3 Source Code

The file **src/media-drv/ftl/drivers/nor/cfi_sfdp/cfi_sfdp_spi.c** is the main source code file. **This file should only be modified by HCC.**

2.4 Platform Support Package (PSP) Files

These files in the directory **src/psp/target/ftl_nor_cfi_sfdp** provide the psp_ftl_nor_cfi_sfdp_delay() function and other elements the core code needs to use, depending on the hardware. Modify these files as required for your hardware.

> **Note:** You must modify these PSP implementations for your specific microcontroller and development board; see PSP Porting for details.

File	Description
psp_ftl_nor_cfi_sfdp.c	Source code.
psp_ftl_nor_cfi_sfdp.h	Header file.

2.5 Version File

The file **src/version/ver_ftl_nor_cfi_sfdp.h** contains the version number of this module. This version number is checked by all modules that use this module to ensure system consistency over upgrades.

3 Configuration Options

Set the system configuration options in the file **src/config/config_ftl_nor_cfi_sfdp.h**. This section lists the available configuration options and their default values**.**

NOR_CFI_SFDP_PAGE_SIZE

This is purely logical on NOR flashes. The default is 256.

NOR_CFI_SFDP_BLOCK_THRESHOLD

The number of blocks below which a 4K sector size is preferred. The default is 128.

NOR_CFI_SFDP_RESERVED_BLOCKS

The number of reserved blocks. FTL will not access blocks below this index. The default is 0.

NOR_CFI_SFDP_STATIC_LIMIT

The maximum value that the difference between the maximum and minimum wear count can be. The default is 1024.

NOR_CFI_SFDP_STATIC_COUNT

The number of merge operations after which static wear checking must be run. The default is 128.

NOR_CFI_SFDP_BLOCK_SIZE

The block size. This parameter is only used if flash identification is not possible by using SFDP or CFI. The default is 4096.

NOR_CFI_SFDP_PAGE_PER_BLOCK

The number of pages per erasable block. This parameter is only used if flash identification is not possible by using SFDP or CFI. The default is 16.

NOR_CFI_SFDP_NUM_BLOCKS

The number of erasable blocks in the target flash array. This parameter is only used if flash identification is not possible by using SFDP or CFI. The default is 4096.

NOR_CFI_SFDP_TBPARM_TOP

If this is set to non-zero, the driver checks whether the TBPARM bit is set (meaning the device is already configured to have parameter sectors on the top of the address space). The default is 1.

If this is set to zero, the driver sets this OTP bit.

NOR_CFI_SFDP_LEGACY_MODE

This has two settings:

- 0 – automatic map and log block calculation. This is the default and is recommended for new designs.
- 1 – the layout (map and log block numbers) is compatible with **media_drv_ftl_nor_cfi_sfdp_spi** version 1.5.

NOR_CFI_SFDP_SPI_UNIT

The ID of the SPI to use. The default is 0.

NOR_CFI_SFDP_SPI_BAUDRATE

The SPI clock frequency in Hz. The default is 50000000.

NOR_CFI_SFDP_TMO_ERASE_MS

The erase timeout in milliseconds. The default is 3000.

NOR_CFI_SFDP_TMO_PROGRAM_MS

The program timeout in milliseconds The default is 500.

NOR_CFI_SFDP_TMO_WREN_MS

The write enable timeout in milliseconds. The default is 5.

NOR_CFI_SFDP_TMO_WRITE_SR_MS

The write status register timeout in milliseconds. The default is 8.

NOR_CFI_SFDP_DIVIDED

Set this to 1 if NOR is split into two separate media drives. The default is 0.

NOR_CFI_SFDP_BLOCK_NUM1

If the flash is divided into two parts, the number of blocks in the first part . The default is 0x80.

NOR_CFI_SFDP_DRV_NUM

This is automatically set to 2 if NOR_CFI_SFDP_DIVIDED is set, otherwise it is 1.

4 Design Overview

4.1 Chip Identification

In order to identify the NOR chip parameters, the driver tries to do the following:

1. Get the CFI header and parameters.
2. If CFI data is not found, get the SFDP header.
3. If both the above fail, use the configuration parameters in **config_ftldriver_nor_sfdp.h**.

After parameter identification, the driver reports the relevant parameters (sector size, sectors/block, and so on) to the upper layer by using **nor_getphy()**.

4.2 CFI Limitations

The limitations are as follows:

- The driver only supports CFI-compatible devices which have one or two Erase blocks. For example, devices having 32*4KB blocks and 64KB blocks in the remaining address space are supported.
- For devices having two Erase blocks, the driver does not use the Erase blocks it has found first.
- The driver assumes that CFI-compatible devices that have two Erase Blocks can be programmed to locate the unused Erase blocks at the top of the memory map. It also assumes that this programming can be performed as described in the data sheet of the Spansion® S25FL128S and S25FL256S devices. (This sets the OTP bit TBPARM in the configuration register by using the WRR command 0x01).

4.3 SFDP Limitations

The Spansion® S25FL512S chip supports both CFI and SFDP. However, it stores the starting address of its SFDP Parameter Table in an incorrect format: in terms of DWORDS, not in terms of bytes.

Therefore, a plausibility check has been introduced to check whether the starting address (located in the SFDP header) is valid. This check looks for the density information in the parameter table pointed to by the starting address. In case it finds 0xFFFFFFFF, it assumes that the starting address is stored the way it is stored in S25FL512S, and calculates a corrected starting address by multiplying the address by 4.

This method, however, does not guarantee that a valid starting address will be used.

4.4 Four Byte Addressing

Devices with density > 128Mbits must be addressed using four address bytes instead of three. The driver assumes that four byte addressing can be entered in one of two ways:

- As described in the datasheet of the Micron® N25Q512x devices: by issuing the ENTER 4-BYTE ADDRESS command (0xB7) for SFDP compatible devices.

- As described in the datasheet of the Spansion® S25FL128S and S25FL256S devices: by issuing the Bank Register Write (0x17) command with EXTADD (bit7) = 1 in the data.

4.5 Chips that do not Support SFDP

The following devices are declared to support SFDP but the SFDP header cannot be accessed in them. Therefore, in case both CFI and SFDP identification has failed, the driver checks the ID of the inserted device and, if it is the ID of one of the following devices, the initialization succeeds:

- Micron® N25Q032A13ESF40F
- Micron® N25Q064A13ESF40F

4.6 Chips that Support SFDP

The driver has been tested with the following serial NOR flash chips:

Micron®

- N25Q128A13GSF40F
- N25Q256A13ESF40E
- N25Q512A13GSF40E
- N25Q00AA13G1240E (supports SFDP, but contains an incorrect density parameter)
- N25Q032A13ESF40F (does not support SFDP)
- N25Q064A13ESF40F (does not support SFDP)

Spansion®

- S25FL128SAGMFIG0
- S25FL256SAGMFIG0
- S25FL512SAGMFIG1

Macronix

- MX25L12835FMI-10G
- MX25L12835EMI-10G
- MX25L25635EMI-10G

Winbond

- W25Q32FVSFIG
- W25Q64FVSFIG
- W25Q128FVFSG
- W25Q256FVFIG

5 Application Programming Interface

This section describes the single Application Programming Interface (API) function, the structures it uses, and the error codes.

5.1 nor_cfi_sfdp_init

Use this function to initialize the FTL media driver interface.

Format

```
t_ftl_ret nor_cfi_sfdp_init (
    uint32_t               driver_param,
    t_ftl_nor_driver * *   pps_ftl_nor_driver )
```

Arguments

Argument	Description	Type
driver_param	The number of the drive to query.	uint32_t
pps_ftl_nor_driver	A pointer to a *t_ftl_driver* structure.	t_ftl_nor_driver * *

Return values

Return value	Description
NOR_ST_OK	Successful execution.
NOR_ST_ERROR	Operation failed.

5.2 Types and Definitions

t_ftl_nor_phy

The *t_ftl_nor_phy* structure describes the real media under the FTL-NOR driver.

Element	Type	Description
n_pageperblock_nor	uint32_t	RAM NOR page per block.
sz_page_nor	uint32_t	The RAM NOR page size.

t_ftl_nor_driver

The **nor_cfi_sfdp_init()** function returns a pointer to a *t_ftl_nor_driver* structure.

Element	Type	Description
user_data	uint32_t	User-defined data.
pf_nor_getphy	(* pf_nor_getphy)	A pointer to the **nor_getphy()** function.
pf_nor_read_page	(* pf_nor_read_page)	A pointer to the **nor_read_page()** function.
pf_nor_write_page	(* pf_nor_write_page)	A pointer to the **nor_write_page()** function.
pf_nor_erase_block	(* pf_nor_erase_block)	A pointer to the **nor_erase_block()** function.

5.3 Error Codes

The possible return codes are shown in the table below:

Code	Value	Description
NOR_ST_OK	0	Successful execution.
NOR_ST_ERROR	1	Operation failed.
NOR_ST_TIMEOUT	2	Operation timed out.

6 Integration

This section describes all aspects of the module that require integration with your target project. This includes porting and configuration of external resources.

6.1 PSP Porting

The Platform Support Package (PSP) is designed to hold all platform-specific functionality, either because it relies on specific features of a target system, or because this provides the most efficient or flexible solution for the developer.

The module makes use of the following standard PSP functions. These are described in the *HCC SPI Driver PSP User Guide*.

Function	Package	Element	Description
psp_spi_init()	psp_base	psp_spi	Initializes the SPI port.
psp_spi_start()	psp_base	psp_spi	Starts the SPI port.
psp_spi_cs_hi()	psp_base	psp_spi	Sets chip select high.
psp_spi_cs_lo()	psp_base	psp_spi	Sets chip select low.
psp_spi_get_baudrate()	psp_base	psp_spi	Gets the baudrate.
psp_spi_set_baudrate()	psp_base	psp_spi	Sets the baudrate.
psp_spi_rx()	psp_base	psp_spi	Receives a number of bytes.
psp_spi_tx1()	psp_base	psp_spi	Transmits one byte.
psp_spi_lock()	psp_base	psp_spi	Locks the SPI for the specific unit. This can be useful if multiple units are attached to the same SPI bus.
psp_spi_unlock()	psp_base	psp_spi	Unlocks the SPI for the specific unit. This can be useful if multiple units are attached to the same SPI bus.

The module makes use of the following function provided by the PSP. Its design makes it easy for you to port it to work with your hardware solution. The package includes a sample in the **src/psp/target /ftl_nor_cfi_sfdp/psp_ftl_nor_cfi_sfdp.c** file.

Macro	Description
psp_ftl_nor_cfi_sfdp_delay()	Waits at least 100 ns, which is the chip select/deselect time.

This function is described in the following section.

psp_ftl_nor_cfi_sfdp_delay

This function is provided by the PSP to wait at least 100 ns, which is the chip select/deselect time.

It is designed for you to port it to work with your hardware solution. The package includes a sample in the **psp_ftl_nor_cfi_sfdp.c** file.

Format

```
void psp_ftl_nor_cfi_sfdp_delay ( void )
```

Arguments

None.

Return Values

None.