Efuse User Guide

Version 1.0

Revision history

Version/Date	Modification
V1.0: 2023/12/14 Kouwanying	- The first version

Contents

1 Introduction

1.1 Purpose and scope

This document describes the feature, interface usage, and efuse management of OM6681 efuse.

1.2 Acronyms and abbreviations

The table below defines the acronyms and abbreviations used in this document.

Acronym	Definition

Table 1-1 Definitions and acronyms

1.3 References

2 Overview

Efuse is a one-time programmable memory. OM6681 uses Efuse controller to burn and read efuse memory transparently. The Efuse controller is connected to the system APB bus.

2.1 Feature

- Program 1bytes data in one programming operation is supported. Don't support burst program;
- Empty bits are "0", only "1" need to be programmed;
- Transparent random address access to the efuse memory cells via the APB slave memory, support bytes read;
- Support interrupt and query status function after program operation;
- Efuse memory offset address: 0x0000~0x01FF; efuse register address: 0xF000~0xFFFF
- Program speed: The single-bit burning speed is about 1bit/12us. Since AVDD requires 30us pull up time for each bytes of burning, the actual single-bit burning speed will be greater than 1bit/12us. (For example, when burning 0xff, the total time is about 8*12us+30us, and the single-bit burning speed is about 8*12+30/8=16us)

2.2 Note

Efuse memory The same address can only be burned once, repeated burning may damage the data content; Therefore, the software needs to avoid repeatedly burning the same address; For example, if the DATA in address 0 is 0x0F and you want to change the content to 0xFF, PROG DATA=0xF0 is burned.;

3 API Introduction

3.1 Register Structure

```
4     typedef struct {
5         __I  uint8_t    READ_DATA[512];            //0x400B0000
6              uint8_t    RSVD[0xF000-512];
7         __IO uint32_t   PROGRAM_ENABLE;            //0x400BF000
8         __IO uint32_t   PROGRAM_START;
9         __IO uint32_t   AVDD_TIMING_CFG;
10        __IO uint32_t   PROGRAM_ADDRESS;
11        __IO uint32_t   PROGRAM_DATA;
12        __IO uint32_t   EFUSE_RSVD;
13        __IO uint32_t   PROGRAM_CFG0;
14        __IO uint32_t   PROGRAM_CFG1;
15        __IO uint32_t   PROGRAM_CFG2;
16        __IO uint32_t   READ_CFG;
17        __IO uint32_t   PROGRAM_INTR;
18        __I  uint32_t   STATUS;
19    } OM_EFUSE_Type;
20
```

3.2 Library Function

3.2.1 eFuse_init

Function Name	eFuse_init
Function prototype	None
Function description	Efuse intialization
Input parameter	None
Return value	None

3.2.2 eFuse_write

Function Name	eFuse_write
Function prototype	om_error_t eFuse_write(uint16_t addr, uint8_t *data, uint16_t data_len);
Function description	Write data to efuse by polling mode

Input parameter 1	addr: EFUSE address
Input parameter 2	data: Pointer with data to write to EFUSE
Input parameter 3	data_len: Number of data bytes to write
Return value	status: - OM_ERROR_OK: Nothing more to do - OM_ERROR_OUT_OF_RANGE: Max write address out of efuse size

3.2.3 eFuse_write_int

Function Name	eFuse_write_int
Function prototype	om_error_t eFuse_write_int(uint16_t addr, uint8_t *data, uint16_t data_len);
Function description	Write data to efuse by interrupt mode
Input parameter 1	addr: EFUSE address
Input parameter 2	data: Pointer with data to write to EFUSE
Input parameter 3	data_len: Number of data bytes to write
Return value	status: - OM_ERROR_OK: Nothing more to do - OM_ERROR_OUT_OF_RANGE: Max write address out of efuse size

3.2.4 eFuse_read

Function Name	eFuse_read
Function prototype	uint16 eFuse_read(uint16_t addr, uint8_t *data, uint16_t data_len);
Function description	Read data from efuse memory
Input parameter 1	addr: EFUSE address
Input parameter 2	data: Pointer with data to write to EFUSE
Input parameter 3	data_len: Number of data bytes to read
Return value	Number of data bytes read completed

3.2.5 eFuse_control

Function Name	eFuse_control
Function prototype	void *eFuse_control(eFuse_control_t control, void *argu);
Function description	Write data to efuse by polling mode
Input parameter 1	control: Control operation

Input parameter 2	argu: Control argument, Bot used, always set NULL.
Return value	control operation status

3.2.6 EFUSE_PROGRAM_EN

Function Name	EFUSE_PROGRAM_EN
Function prototype	EFUSE_PROGRAM_EN(ctrl)
Function description	Efuse program enable; Excute EFUSE_PROGRAM_EN(1) Before Write data to efuse and Excute EFUSE_PROGRAM_EN(0) after Write data to efuses
Input parameter 1	ctrl: enable ssfuse program(1), disable efuse program(0)

4 Efuse manger

4.1 TLV Format

In the om6681A, efuse provides 512 bytes for users to use. In isp mode, the host can send corresponding commands to specify the efuse address for reading and writing, or the TLV format (tag+ length + content), as shown in the following figure:

tag (1B)	len (1B)	data (len B)

Table 2-1 TLV format

4.2 Efuse Tags

At present, efuse has been defined in tag as follows:

Tags type	value	length	Parameter description
EFUSE_TAG_HUK	0x01	32B	Hardware unique key
EFUSE_TAG_IFLASH_CONFIG	0x02	42B	Iflash configuration parameter. If this tag is not present, the internal flash will be initialized with the default parameter
EFUSE_TAG_HASH_OF_PUB_KEY	0x03	32B	hash of public key
EFUSE_TAG_SECURE_BOOT_ON	0x04	1B	0：closesecure boot 1 or without this tag：open secure boot
EFUSE_TAG_SWD_ON·	0x05	1B	0x55：close swd others：open swd
EFUSE_TAG_CP	0x06		CP data
EFUSE_TAG_FT	0x07		FT data
EFUSE_TAG_OFLASH_CONFIG	0x08	42B	External flash parameter Settings. Without this tag, the external flash will not be initialized
EFUSE_TAG_BOOT_FROM_OFLASH	0x09	1B	This tag is used when internal flash has problem. 0x55： Initializes the external flash pin and jumps to the external flash executive. 0x56： Initializes the external flash pin and attempts to read the flash id. If the read succeeds, jump to the

			external flash. Otherwise, the isp mode is entered.
EFUSE_TAG_ROM_CONFIG	0x0A	1B	0x1：cpu clk and ahb clk frequency factor is set as 1，apb clk frequency factor is set as 2 others：Frequency factor use default value

Table 2-2 efuse defined tags

4.2 EFUSE_TAG_xFLASH_CONFIG Parameters

parameters	length	parameter description
feature	1B	bit7 = 0：enter QPI mode -bit1 = 1：enter QPI mode -bit0 = 1：enter 32-bit address mode bit7 = 1：enter OPI mode -bit1 = 1：进入 OPI mode
clk div	1B	The frequency division factor of the flash working clock
dly sample	1B	Sampling delay
dly tx val	1B	Delay tx parameter
dly rx val	1B	Delay rx parameter
dtr en	1B	DTR enable
rd cmd	1B	Read command
wr cmd	1B	Write command
power_on_10us	2B	The time wait for the flash voltage to stabilize，unit is10us
key	32B	AES encrypt key，and nonce used key's before 16bytes In Efuse's flash_config tag，If this parameter is displayed, flash encryption is enabled. Otherwise, flash encryption is not enabled

Table 2-3 EFUSE_TAG_xFLASH_CONFIG Parameter description

s