Application Report
SPNA094–February 2006

Modifying the Flash Protection Keys of Catalog TMS470 Devices

John Mangino TMS470 Applications

ABSTRACT

The purpose of this document is to give an example of how to reprogram the flash protection keys on the TMS470 devices. These are intended as a reference to enable the user to modify the flash protection keys and unlock the part using the IAR tools. The reference design includes TMS470 software for use with the IAR Embedded Workbench.

Contents

1 Introduction

This application note introduces an understanding of the flash protection keys, modifying them and unlocking the TMS470 Flash devices. These examples are using the IAR Embedded Workbench and have information for code generation specific to the IAR tools.

Note: Note: Do not overwrite the Flash Protection Keys and the Memory Security Module (MSM) Keys. If the keys are rewritten and the data is not known, the part cannot be reprogrammed or accessed in the case of the MSM.

1.1 Flash Protection Key Overview

The TMS470 devices flash memory is protected from unauthorized erasures or writes with the four 32-bit protection keys. The CPU reads the four stored protection keys out of the flash bank one at a time and into a register in the flash module. After the CPU loads each key from the bank to the control logic, the CPU must load an identical user key into the FMPKEY control register. The CPU must load and match all four keys before any program or erase command can be sent to the flash module. To enable the module for programming, the CPU must load each stored key value from the bank to the control logic by performing a normal read access to one of the four protection key addresses in the Flash module. The CPU must then load the matching user key value into the FMPKEY register while in configuration mode. This process is repeated until all four keys are loaded and matched. The control logic monitors which keys have been matched, so the CPU can not gain write access until it loads and matches all four keys at least once without any intervening mismatch.

If the CPU writes a mismatching key at any time (that is, if the user key does not match the key that was most recently loaded from the bank to the control logic), then all key match states are cleared and the CPU must reload and rematch all four keys again to gain write access to the module. This feature can be used to disable write access after programming is completed.

After the CPU has successfully loaded and matched all four keys, flash write access is enabled until a device reset occurs or until the CPU writes a mismatching key to the FMPKEY register.

To store the key values, the CPU programs the key data into the bank by performing normal program and erase operations on these four protection key addresses. The key values are stored in the bank as ordinary data, so the CPU must provide the correct keys before it can perform any program or erasure of the key values.

When a new device is delivered to the customer, the keys will be all ones, so keys of all ones should be used to enable flash writes for the first time. Once the keys are changed in the Flash bank, the CPU must deliberately write a mismatching key value to FMPKEY in order to disable further programming until the new key values have been loaded and matched. In other words, the flash module remains enabled for the remainder of this programming session even though the keys have been modified in flash.

The difference between flash protection key read accesses and other reads is that the key data does not propagate to the CPU data bus until the correct keys are written and the user has taken all required steps to gain programming access. This is intended to prevent unauthorized discovery of the stored keys by reading them out via the CPU: only a user who already knows the keys can gain access to them. The availability of this protection feature and the location of the four protection keys depend on the specific device being used (as specified in the specific TMS470R1x device data sheet). If this feature is not available, then the four protection key addresses in the module are available for normal memory access.

The flash protection keys are located in the last 4 32 bit words of the first sector of the flash in the tms470 devices.

2 Flash Protection Key Modification

The flash protection keys are modified by writing to the last four words of the first sector in flash. The process flow is:

1. Match the current flash protection keys to unlock the flash
2. Erase the flash
3. Write the new keys

2.1 Matching the Keys

As with any writes to flash, the flash must be unlocked by matching these keys. The following two sections discuss using the IAR workbench and the Flash API routines to unlock the flash.

2.1.1 IAR Workbench

The IAR workbench has a command line located in **Options > Debugger > Download** pane of the tool. Editing the Flash loader over view with the following command will unlock the flash.

```
--flashkey0 0xFFFFFFFF --flashkey1 0xFFFFFFFF --flashkey2 0xFFFFFFFF --flashkey3 0xFFFFFFFF
--clock12000
```

The **—flashkeyx** command insert the existing flash keys to unlock the flash for reprogramming. 0xFFFFFFFF is the default of of the keys and it is not necessary to enter these keys because the workbench automatically inserts them when programming the flash. However is the keys are changed then the existing keys must be entered.

Note: The **—clock** command sets the delay factor for flash programming. On the A256 EVM a 12-MHz system clock frequency is used for reprogramming. The 12000 entry represents 12MHz and creates a delay factor of 6. See the Flash API documentation for a full explanation of the clock frequency and flash programming.

2.1.2 Flash API modules

The Flash API Modules use the Match_Key_B routine to unlock the flash. See the TMS470 FLASH API Modules (SPRC236.zip) and the Flash API Modules reference (SPNU257) for details on this module. Below is sample code to use this module to unlock the flash.

```
//-----------------------------------------------------------------------------
// Match the Flash protection keys
//
// Returns Success or Error
//-----------------------------------------------------------------------------
#pragma location="API_SEGMENT"
static unsigned char FlashUnlock( void)
{
  UINT32 Keys[] = {0xFFFFFFFF,0xFFFFFFFF,0xFFFFFFFF,0xFFFFFFFF};
#ifdef FLASH_CFG
  Flash_Match_Key_B((UINT32 *)0x00001ff0UL,(UINT32 *)Keys,( FLASH_ARRAY_ST )0xFFE88000UL);
#endif
  return( FLASH_OK );
}
```

2.1.3 Flash Protection Key Locations

The flash protection keys are located in the last four 32-bit words of the first sector of the flash in the tms470 devices.

DEVICE	FLASH PROTECTION KEY LOCATION
TMS470R1A64	0x00001FF0 – 0x00001FFF
TMS470R1A128	0x00001FF0 – 0x00001FFF
TMS470R1A256	0x00001FF0 – 0x00001FFF
TMS470R1A288	0x00001FF0 – 0x00001FFF
TMS470R1A384	0x00001FF0 – 0x00001FFF
TMS470R1B512	0x00003FF0 – 0x00003FFF
TMS470R1B768	0x00003FF0 – 0x00003FFF
TMS470R1B1M	0x0000FFF0 – 0x0000FFFF

2.2 Modifying the Flash Protection Keys

The flash protection keys are modified by writing to the flash protection key locations in flash.

2.2.1 IAR Workbench

The IAR workbench has a command line located in Options > Debugger > Download pane of the tool. Editing the Flash loader over view with the following command will unlock the flash.

--flashkey0 0xFFFFFFFF --flashkey1 0xFFFFFFFF --flashkey2 0xFFFFFFFF --flashkey3 0xFFFFFFFF --allownewkeys --clock12000

The **—flashkeyx** command inserts the existing flash keys to unlock the flash for reprogramming. 0xFFFFFFFF is the default of of the keys and it is not necessary to enter these keys because the workbench automatically inserts them when programming the flash. However is the keys are changed then the existing keys must be entered.

The **allownewkeys** command signals the IAR workbench to allow the flash protection key locations to be modified.

Note: The **—clock**” command sets the delay factor for flash programming. On the A256 EVM a 12 MHz system clock frequency is used for reprogramming. The 12000 entry represents 12MHz and creates a delay factor of 6. See the Flash API documentation for a full explanation of the clock frequency and flash programming.

The C program below assigns the FlashKeys array to the FLASHKEYS segment defined in the linker file that points to the flash protection key location for the specific device. The **_root** directive prevents the compiler from optimizing out the FlashKeys variable.

```
//------------------------------------------------------------------------------
// This module invokes the API Routines
//------------------------------------------------------------------------------
#pragma location="FLASHKEYS"
__root const unsigned long FlashKeys[4] =
{
  0xFFFFFFFF,
  0xFFFFFFFF,
  0xFFFFFFFF,
  0xFFFFFFFF
};
```

The Flash Key Segment is defined below in the example segment. This segment is added to the linker files and is device specific.

```
//////////////////////////////////////////////////////////////////////////
// Flash Key Segment
//////////////////////////////////////////////////////////////////////////
-Z(CODE)FLASHKEYS=(ROMSTART+0x1FF0):+0x10
```

2.2.2 Flash API Modules

When using the Flash API modules the user creates the routine to write the flash. Once the part is unlocked, the flash memory is open and the flash protection key locations can be over written as ordinary flash memory. See Flash API documentation and examples.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265