FLASH SECURITY USER'S GUIDE

1. Relevant Devices

This application note applies to the following devices:

C8051F000, C8051F001, C8051F002, C8051F005, C8051F006, C8051F010, C8051F011, C8051F012, C8051F015, C8051F016, C8051F017, C8051F206, C8051F220, C8051F221, C8051F226, C8051F230, C8051F231, and C8051F236.

Introduction

Silicon Labs Integrated Devices feature in-system programmable FLASH memory for convenient, upgradable code storage. The FLASH may be programmed via the JTAG interface or by application code for maximum flexibility. Proprietary information in the form of code and constants are often stored in FLASH memory. Silicon Labs provides security options at the disposal of the designer to prevent unauthorized access to information stored in FLASH memory.

Silicon Labs integrated devices provide FLASH security options to:

1. Prevent unauthorized access of intellectual property in the form of code and constants stored in FLASH.

2. Prevent inadvertent modification of code by the end-user.

3. Prevent code modification due to abnormal system conditions (e.g., low-voltage supply conditions to the device).

Silicon Labs devices offer security options to prevent unauthorized access of the FLASH via the JTAG port and application software loaded by the end-user. *FLASH Program Memory Security Bytes* are used to prevent access via the JTAG interface, and a *Software Read Limit* (available on most Silicon Labs devices) is to prevent unauthorized access through application software. This application note discusses the operation and use of the FLASH security options.

Key Points

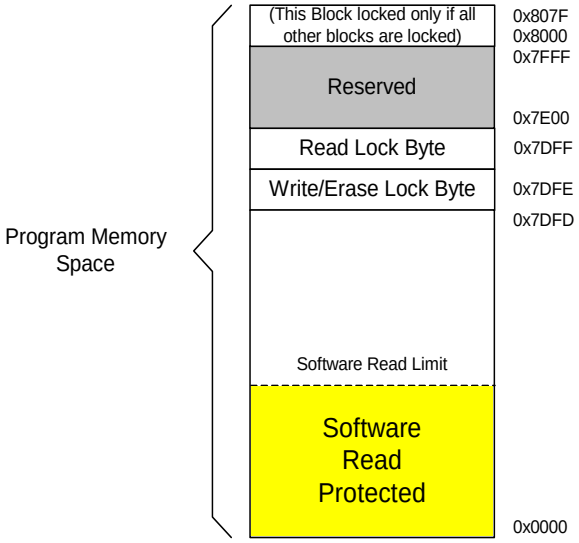
- FLASH memory can be protected from access across the JTAG interface by setting bits in the FLASH Security Bytes to '0'.
- FLASH memory can also be protected from *read* accesses by software by setting a Software Read Access Limit. (Used to allow the end-user to access some portions of FLASH memory.)
- FLASH memory protected from software access should also be protected from JTAG access using the FLASH Security Bytes.
- When protecting FLASH, the FLASH page containing the FLASH Security Bytes should also be protected. (FLASH cannot be unlocked using software).
- If the end-user does not need access to FLASH memory, the entire FLASH memory can be protected by simply locking the entire FLASH memory from JTAG access (Software Read Limit is not needed in this case as the end-user cannot download software to access proprietary information).

Preventing FLASH Access Via the JTAG Interface.

One of the two ways to read, write, and erase the FLASH memory is via the JTAG interface (see application note “AN005 - Programming FLASH Through the JTAG Interface”.) The *FLASH Program Memory Security Bytes* located in the FLASH memory are used to prevent both read and/or write/erase operations of any or all of the 512-byte memory blocks **via the JTAG interface**.

The FLASH security bytes are located in the FLASH memory as shown in Figure 1 and Figure 2 below. To protect a FLASH memory block from unauthorized read or write/erase operations across the JTAG interface, refer to the memory block chart (also located in each device’s data sheet).

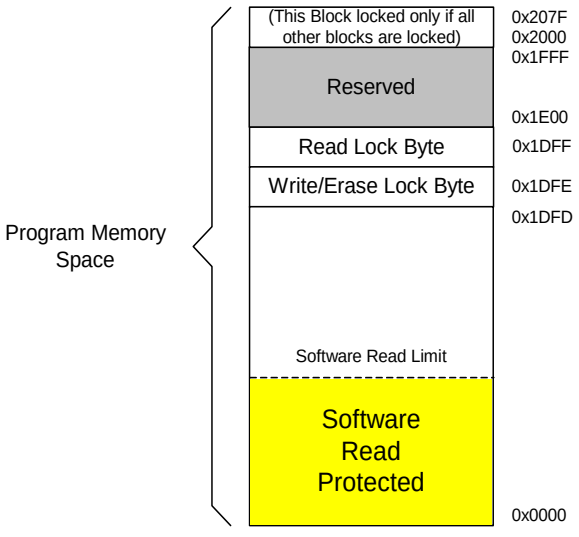
Attempting a read operation on a byte in a read-locked sector will return a value of ‘0’ and will set the FAIL bit in the FLASHDAT register to ‘1’, indicating a FLASH operation failure. (Please see



Read and Write/Erase Security Bits.
(Bit 7 is MSB.)

Bit	Memory Block
7	0x7000 - 0x7DFD
6	0x6000 - 0x6FFF
5	0x5000 - 0x5FFF
4	0x4000 - 0x4FFF
3	0x3000 - 0x3FFF
2	0x2000 - 0x2FFF
1	0x1000 - 0x1FFF
0	0x0000 - 0x0FFF

Figure 1. FLASH Security Bytes The C8051F0xx Family of Devices



Read and Write/Erase Security Bits.
(Bit 7 is MSB.)

Bit	Memory Block
7	0x1C00 - 0x1DFD
6	0x1800 - 0x1BFF
5	0x1400 - 0x17FF
4	0x1000 - 0x13FF
3	0x0C00 - 0x0FFF
2	0x0800 - 0x0BFF
1	0x0400 - 0x07FF
0	0x0000 - 0x03FF

Figure 2. FLASH Security Bytes For The C8051F2xx Family of Devices

application note, “AN005 - Programming FLASH Through the JTAG Interface” for more information about how to read FLASH data via the JTAG interface). Clearing a bit to logic ‘0’ in the Read Lock Byte will prevent the corresponding block of FLASH memory from being read via the JTAG interface.

Attempting a write or erase operation on a byte in a write/erase-locked sector will be ignored and will set the FAIL bit in the FLASHDAT register to ‘1’ indicating a FLASH operation failure. Clearing a bit to logic 0 in the write/erase lock byte will prevent the corresponding block of FLASH memory from write/erase operations via the JTAG interface. Clearing an entire security byte to 0x00 will protect the entire FLASH code space from that respective operation across the JTAG interface.

NOTE: The FLASH Security bytes prevent access via JTAG only -- software can still access JTAG locked blocks! To prevent unauthorized access, the application should lock the entire FLASH memory. Locking all memory bytes will prevent an end-user from downloading code to unlocked memory space and using software to access information in the locked space. If an application must leave some memory unlocked, but the designer still wants to prevent access to some FLASH memory, the *FLASH Access Limit* feature should be used in conjunction with the security bytes. In an application that locks some blocks of FLASH memory yet leaves some blocks unlocked for the end-user, the block containing security bytes should always be write/erase locked to prevent unlocking the protected FLASH by erasing the FLASH page containing the security bytes.

Device Erase

Performing a JTAG erase operation using the address of either the read lock byte or erase/write lock byte will automatically initiate erasure of the entire FLASH program space (with the exception of the RESERVED area.) **This can only be performed via the JTAG interface, and not by software**. If software attempts to erase any byte in the FLASH page containing the lock bytes, the erase operation is ignored. If a non-security byte in the memory block that contains the FLASH security bytes is addressed to perform a FLASH erase, only that 512-byte page will be erased (including the security bytes.)

Preventing FLASH Access Via Software

Note: The Software Access Read Limit security option discussed in the following section is not available on the C8051F000/01/02 and C8051F010/11/12. In these devices, the entire FLASH user space should be read and write/erase locked using FLASH security bytes to protect intellectual property.

Silicon Labs devices’ FLASH memory may be accessed via application software (see application note, “AN009 - Writing to FLASH from Application Code.”) This facilitates maximum flexibility in application design including the implementation of bootloading software, but does give a way for the end-user to access FLASH memory that has been locked from JTAG access (unless ALL of the FLASH memory is locked.) For this reason, Silicon Labs devices feature a FLASH access *Software Read Limit* to restrict access via downloaded application code. Used in conjunction with the security bytes to prevent JTAG access, the software read limit allows the application to prevent software access to some FLASH memory, while leaving some FLASH accessible to the end-user.

The FLASH software access limit works as follows. The designer defines an address as an access limit. FLASH memory from address 0x0000 up to and including the address defined as the *software read limit* is protected from software access. If code loaded into the FLASH above the software access limit address attempts to execute a MOVC instruction with a source address in the software read protected address space, a data value of 0x00

will be returned. Code loaded into FLASH in the software protected space (below the FLACL boundary) is not restricted from executing. FLASH memory above the software access limit address boundary may be used as normal (i.e., read and write/erase operations may be performed by software), but may not write or erase code below the FLACL boundary. Thus, the application can protect code from unauthorized access, yet still leave FLASH memory usable to the end-user.

NOTE: Software read protected FLASH should also be locked using the security bytes to prevent JTAG access to the protected memory blocks. (When locking only certain memory blocks, the memory block containing the security bytes should always be locked from JTAG access as well to prevent the end-user from unlocking FLASH memory.)

Setting the Software Read Limit

The software read limit is set using the FLASH Access Limit special function register (FLACL). The **upper byte** of the desired software access limit address (the highest address the designer wishes to have software access protection) is moved into the FLACL register. The lower byte of the address will be 0x00. (See Figure 3 below.) Thus, if the FLACL register is assigned the value 0x40, then the software access limit address will be 0x4000. Thus, all code in memory from 0x0000 to 0x4000 (including 0x4000) will not be accessible via software executing above this address. Code executing above the FLACL boundary may perform *jump* and *call* instructions into protected memory space below the FLACL boundary. Only MOVX and MOVC operations are prevented by the Software Read Limit. To prevent access, the FLASH Security Bytes should also be used to prevent JTAG access of the memory blocks at and below 0x4000 for total protection (see the previous section, “Preventing FLASH Access Via the JTAG Interface.)

If the application does not require FLASH memory to be programmable for the end-user, then it is best to lock the entire FLASH memory using the security bytes, and the software access read limit is not needed (the end-user will not be able to download code to FLASH to access protected information.)

FLASH Write and Erase Enable Bits

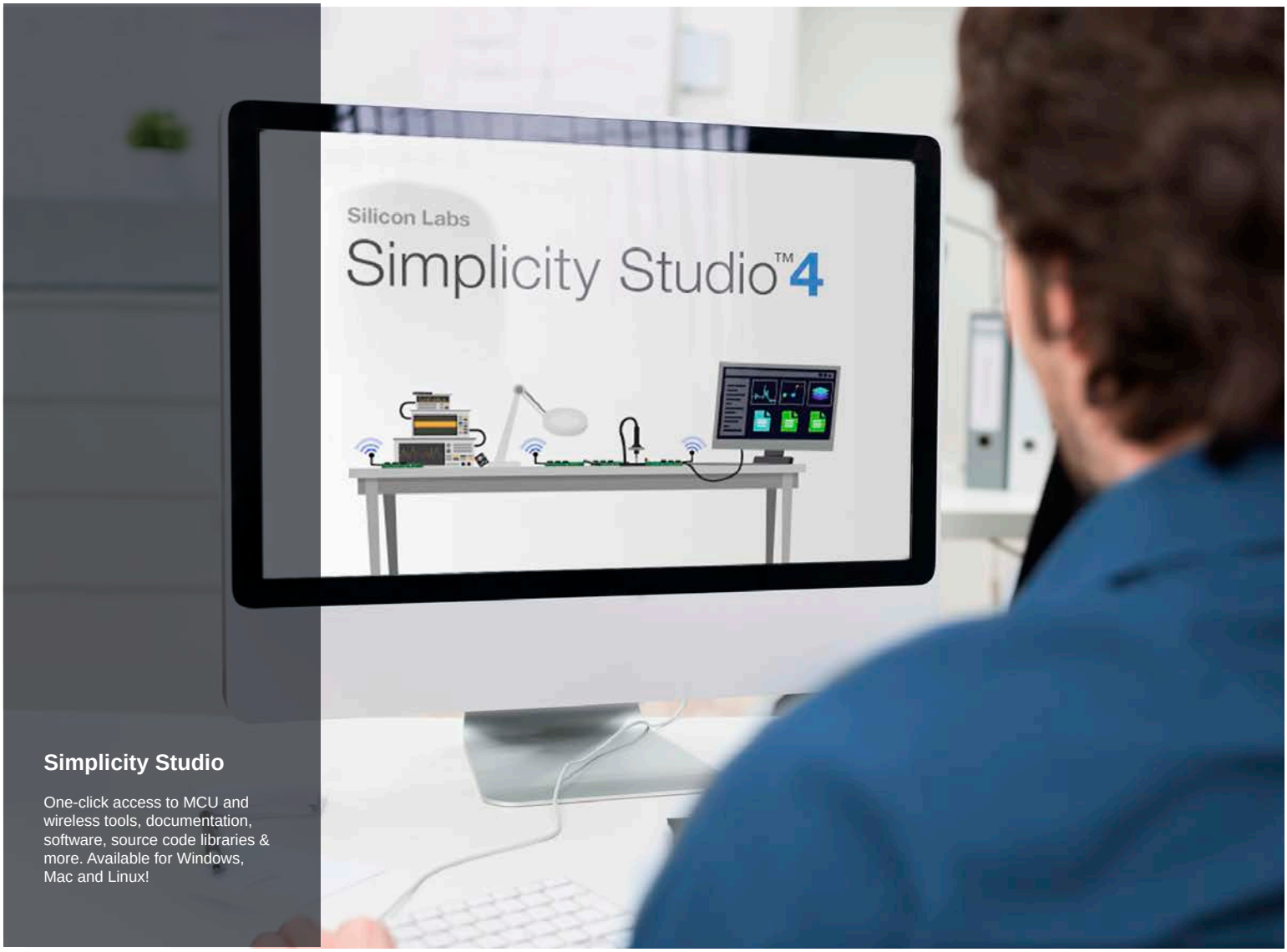
One function of FLASH security is to prevent inadvertent modification of code. Silicon Labs FLASH write and erase operations cannot occur via software unless they are enabled using the Program Store Write Enable (PSWE) and Program Store Erase Enable (PSEE) bits. In order to write to the FLASH memory, the PSWE bit must be set to 1. When the PSWE bit is set to ‘1’, the MOVX instruction writes to FLASH memory instead of XRAM (the default target). In order to erase a page of FLASH memory, the PSEE **and** PSWE bits must be set to ‘1’. When the PSEE bit is set to ‘1’, the

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xB7

Bits 7-0: FLACL: Flash Memory Read Limit.
This register holds the high byte of the 16-bit program memory read limit address. The entire 16-bit access limit address value is calculated as 0xNN00 where NN is replaced by contents of FLACL. A write to this register sets the Flash Access Limit. Any subsequent writes are ignored until the next reset.

Figure 3. FLACL: Flash Access Limit Special Function Register

FLASH control logic interprets a FLASH *write* operation as an *erase* operation. The PSEE and PSWE bits aid in preventing inadvertent write and erase modifications when they are not intended. Of course, this does not perform the function of protecting intellectual property access by an unauthorized end-user, as the PSWE and PSEE bits are always accessible. Always use the software read access limit and/or FLASH security bytes for protection of intellectual property.



Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com