CEC1302

CEC1302 ROM Description Addendum

INTRODUCTION

This document describes the functionality provided by the ROM code in the CEC1302.

This document includes the following topics:

Audience

This document is written for developers.

References

The following documents should be referenced when using this addendum. See your Microchip representative for availability.

- CEC1302 Data Sheet
- CEC1302 Crypto API User's Guide
- CEC1302 Peripheral API User's Guide
- CEC1302 CLIB/PLIB Releases

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com**. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000000A is version A of document DS30000000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include -literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

1.0 CONVENTIONS

The first table defines common terminology used in the documentation. The second table defines the register bit access type notation used in the documentation. These are the access types that are supported.

Term	Definition
Block	Used to identify or describe the logic or IP Blocks implemented in the device.
Reserved	Reserved registers and bits defined in the following table are read only values that return 0 when read. Writes to these reserved registers have no effect.
Test	Test locations should not be modified from their default value. Changing a Test register may cause unwanted results. Unless otherwise specified, a Test bit field, when written, should be written with its current value.
b	The letter ‘b’ following a number denotes a binary number.
h	The letter ‘h’ following a number denotes a hexadecimal number.

Register access notation is in the form “Read / Write”. A Read term without a Write term means that the bit is read-only and writing has no effect. A Write term without a Read term means that the bit is write-only, and assumes that reading returns all zeros.

Register Bit Type Notation	Register Bit Description
R	**Read:** A register or bit with this attribute can be read.
W	**Write:** A register or bit with this attribute can be written.
RS	**Read to Set:** This bit is set on read.
RC	**Read to Clear:** Content is cleared after the read. Writes have no effect.
WC	**Write One to Clear:** writing a one clears the value. Writing a zero has no effect.
WZC	**Write Zero to Clear:** writing a zero clears the value. Writing a one has no effect.
WS	**Write One to Set:** writing a one sets the value to 1. Writing a zero has no effect.
WZS	**Write Zero to Set:** writing a zero sets the value to 1. Writing a one has no effect.

2.0 CEC1302 ROM DESIGN

The CEC1302 ROM's purpose is to load application EC firmware from an external SPI flash device into internal SRAM, verify its authenticity and launch the application firmware. In addition, the ROM includes a set of APIs for loading firmware from the SPI flash device into SRAM at runtime and cryptographic operations.

The CEC1302 includes two master-only General Purpose SPI controllers. Each SPI port has two dedicated chip select pins, implemented with GPIOs. The controller is capable of half duplex (single bi-directional pin), full duplex (one transmit and one receive pin) and double input data rate (transmit pin becomes second receive pin during data reception phase).

The ROM EC firmware loader will support SPI frequencies of [48, 24, 16, 12] MHz. The runtime API will allow application firmware to use the GP-SPI controllers at any of the four supported frequencies.

The GP-SPI controller allows configuration of both transmit and receive clock edge sampling. The ROM application firmware loader function will use Mode 00, in which both transmit and receive data occur on rising clock edges.

2.1 ROM Startup on Reset

On VCC1 power-on reset, the EC will perform the following functions:

1. Basic configuration of device:
   a) Disable interrupts in the EC (except NMI and Hard Error)
   b) Enable the EC FPU
   c) Set EC clock frequency to maximum (48MHz)
2. Execute C startup code
   a) Setup stack
   b) Load non-zero global variable values from table
   c) Call C main() function
3. C main() processing
   a) Clear EC Subsystem AHB Error register.
   b) Route interrupts to NVIC.
   c) Enable 16-bit Basic Timer 0 for 10 us tick time, count down, and no interrupt.
   d) Power on DMA block.
4. Select the SPI port for loading. See **Section 2.2 “Selection of SPI Port”**
5. SPI Load of Application Firmware. See **Section 3.0 “SPI Load of Firmware”**

2.2 Selection of SPI Port

The Boot ROM power-on firmware attempts to boot the EC using data from either a private SPI Flash device attached to SPI Controller 1, or from the public SPI Flash attached to SPI Controller 0 and shared with the Host core logic.

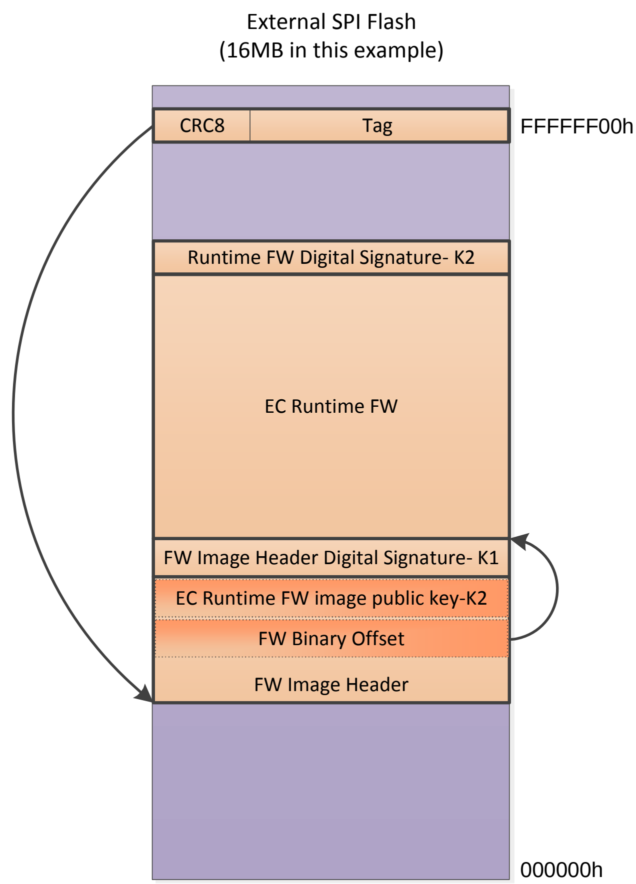
3.0 SPI LOAD OF FIRMWARE

In order to boot the EC from an external SPI Flash, the Boot ROM firmware can access several regions in the SPI Flash:

1. A 4-byte Tag that identifies the position of the FW Image Header in the Flash
2. A 320-byte FW Image Header
3. A 256-byte DER encoded PKCS#1 v1.5 Signature of the FW Image Header
4. The EC Runtime FW body, which is an integral multiple of 64 bytes
5. A 256-byte DER encoded PKCS#1 v1.5 Signature of the EC Runtime FW

The following figure illustrates the layout of the regions in an example SPI Flash:

FIGURE 3-1: SPI FLASH REGION LAYOUT EXAMPLE



The regions are described in more detail in the next sections.

The SPI Load operation proceeds as follows:

1. Wait for the Chip Select 0 input for the shared SPI controller (SHD_CS0#) to be high. Since the Chip Select should be pulled high to the power rail that powers the SPI Flash devices, the EC uses this test to verify that the SPI Flash devices have power.
2. Examine the Private SPI Chip Select (PVT_CS0#). This is used as strap to determine if there is a Private SPI Flash in the system. If PVT_CS0# is high, then the following step is executed:
   a) Configure the private SPI Controller for 12MHz operation and execute the Load Sequence targeting the Private SPI Flash
3. If the Load Sequence on the Private SPI Flash fails, or the strap option indicates that there is no Private SPI Flash, and RSMRST# is asserted, configure the private SPI Controller for 12MHz operation and execute the Load Sequence on the Shared SPI Flash.

4. If the Load Sequence on the Shared SPI Flash fails the ROM code quits

The Shared SPI Bus will only be checked if RSMRST# is active Low. If RSMRST# is high then the chipset owns the shared SPI bus and the EC must keep its shared SPI signals tri-stated.

The Load Sequence attempts to load a valid firmware image in the SRAM and then execute it.

The ROM tries to locate a valid image in up to 4 locations. The locations are searched in the following order:

1. Private SPI Controller, Tag located at 0xFFFFFF00 in the SPI Flash, Chip Select 0,
2. Private SPI Controller, Tag located at 0xFFFFFF04 in the SPI Flash, Chip Select 0.
3. Shared SPI Controller, Tag located at 0xFFFFFF00 in the SPI Flash, Chip Select 0,
4. Shared SPI Controller, Tag located at 0xFFFFFF04 in the SPI Flash, Chip Select 0.

The read of the Tag section is suppressed and the load fails if the Tag section is not valid. The Tag is considered valid if:

1. The CRC in the Tag is correctly generated

Once a location in the SPI Flash is selected, the Load Sequence loads the Firmware Image Header. The validity of the header is checked once it is loaded, and if the validity check fails, no further reads from the SPI Flash are performed and the load fails. The Firmware Image Header is valid if:

1. The first 4 bytes of the header are "SMSC"
2. The SPI speed and SPI read command are valid
3. The Load Address, Payload Length and Payload entry point are all valid with respect to the SRAM

If the Header is valid, the ROM code proceeds to copy the EC Runtime Binary Image from the SPI Flash into SRAM. The validity of the Binary Image is checked once it is loaded, and if the validity check fails, the load fails. The EC Runtime Binary Image is valid if:

1. The signature of the FW Image Header was correctly signed by the customer
2. The EC Runtime FW Binary Image was correctly signed using the public key contained in the FW Image Header

Once a validated EC Runtime FW Binary Image is loaded into SRAM, the ROM code exits the ROM by:

1. Clearing all SRAM used by Boot ROM functions, including the Cryptographic RAM
2. Leaving the SPI controllers in their power-on default state
3. Jumping to an entry point in the SRAM defined by the Header

3.1 Tag

The Tag consists of 32-bits that contain a pointer to the EC code image and its header. The Load Sequence first checks for a tag at offset 0xFFFFFF00 (256 bytes below the last location in the flash) in the SPI flash chip connected to the currently selected SPI controller, using Chip Select 0. If the data stored at offset 0xFFFFFF00 fail the Tag validation, the FW Image Header validation, or the EC Runtime FW Binary Image validation, the Load Sequence then checks for a valid tag at offset 0xFFFFFF04. If the second validation also fails, the ROM code concludes that there is not a valid EC Runtime FW Binary Image in the SPI Flash.

The format of the Tag is:

1. Tag Bits [22:0] correspond to bits [30:8] of the address in the SPI flash device. The Header is therefore always located on a 256 byte boundary in the SPI Flash
2. Tag Bit [23] indicates the SPI chip select.

   0b = chip select 0,
   1b = chip select 1.

   This means the Header and EC binary can be located in a different SPI Flash part than the Tag.
3. Tag Bits [31:24] contain a CRC8 checksum of bits [23:0] of the Tag. The CRC uses CRC8-ITU. If the CRC check fails the tag is considered invalid.

The Tag is read at a SPI Flash data rate of 12MHz, using the standard Read command 0x3.

3.2 Firmware Image Header

The FW Image Header is located in a SPI Flash device selected by CS0# or CS1#, as determined by the Tag, on the selected SPI controller. The Header is located at an offset in the SPI Flash that is on a 256-byte boundary. The Header is read at a SPI Flash data rate of 12MHz, using the standard Normal Read command 0x3.

The FW Image Header is validated by calculating a SHA-256 Hash on the header (offsets 0x0 through 0x13F), decrypting the 256-byte Signature located at offset 0x140, and verifying that the calculated hash and the hash that was encrypted in the Signature match.

The following tables define the format of the Firmware Image Header:

TABLE 3-1: FIRMWARE IMAGE HEADER FORMAT

Byte Offset	Definition	Comment
0x00	ASCII ‘C’	0x43
0x01	ASCII ‘S’	0x53
0x02	ASCII ‘M’	0x4D
0x03	ASCII ‘S’	0x53
0x04	Header Version	0x00
0x05	Reserved	Must be 0
0x06	Bits[1:0] SPI Clock Speed	0 = 48 MHz 1 = 24 MHz 2 = 16 MHz 3 = 12 MHz
	Bits[7:2]	Must be 0
0x07	Flash Read Command	See Table 3-2, "Flash Read Command Options"
0x08 to 0x0B	Load Address b[31:0] Little-Endian: Offset 0x08 = b[7:0] Offset 0x09 = b[15:8] Offset 0x0A = b[23:16] Offset 0x0B = b[31:24]	Start address in SRAM where the EC Firmware Image will be loaded.
0x0C to 0x0F	Entry Address b[31:0] Little-Endian: Offset 0x0C = b[7:0] Offset 0x0D = b[15:8] Offset 0x0E = b[23:16] Offset 0x0F = b[31:24]	EC Firmware Entry Point. ROM jumps to this address on successful load and verify.
0x10 to 0x11	FW Binary Length Little-Endian: Offset 0x10 = b[7:0] Offset 0x11 = b[15:8]	Units of 64 bytes
0x12 to 0x13	Reserved	Must be 0
0x14 to 0x17	FW Binary SPI Location Offset 0x14 = b[7:0] Offset 0x15 = b[15:8] Offset 0x16 = b[23:16] Offset 0x17 = b[31:24] Bits[5:0] must be 0	Unsigned offset in bytes from beginning of header in SPI of the FW payload. FW Payload is the code image plus signature. The offset must be evenly divisible by 64. Stored little-Endian.
0x18 to 0x1F	Reserved	Must be 0
0x20 to 0x27	RSA Public Key 2 Exponent Offset[23:20] = b[31:0] Offset[27:24] = b[63:32]	RSA Public Key Exponent of PK2. NOTE: Exponent length is 8 bytes (64 bits). Stored little-endian.
0x28 to 0x2F	Reserved	Must be 0

TABLE 3-1: FIRMWARE IMAGE HEADER FORMAT (CONTINUED)

Byte Offset	Definition	Comment
0x30 to 0x12F	RSA Public Key 2 Modulus Offset[33:30] = b[31:0] Offset[37:34] = b[63:32] Offset[3B:38] = b[95:64] … Offset[12F:12C] = b[2047:2016]	RSA Public Key Modulus of PK2. NOTE: Modulus length is 256 bytes (2048 bits). Stored Little-Endian.
0x130 to 0x13F	Reserved	Must be 0
0x140 to 0x23F	Signature Offset[143:140] = b[31:0] Offset[147:144] = b[63:32] Offset[14B:148] = b[95:64] … Offset[23F:23C] = b[2047:2016]	256-byte signature of Header, encrypted with a 2048-bit Private key. It is stored Little-Endian.

TABLE 3-2: FLASH READ COMMAND OPTIONS

Command	SPI Read Command	Description
0	0x03	Normal Read
1	0x0B	Fast Read
2	0x3B	Fast read with double data rate return
>2	Reserved	

3.3 FW Image Header Digital Signature – K1

The FW Image Header Digital Signature is a DER encoded PKCS#1 v1.5 digital signature. The signature comprises a 256-bit SHA256 hash of the 320 byte FW Image Header, encrypted using RSA-2048, with the private key that corresponds to the public key stored in eFUSE memory. It is stored Little-Endian.

3.4 EC Runtime Binary Image

The EC Runtime FW Binary Image contains two contiguous components: the first is the firmware image that is loaded into SRAM and the second is the 256-byte signature that immediately follows the SRAM data. The Firmware Image is stored on the same SPI Flash as the Header. The Firmware image must start on a 64-byte boundary in the SPI Flash, and consists of an integral number of 64-byte blocks. If the firmware image does not end on a 64-byte boundary, it must be padded with zeros.

The Load Sequence configures the SPI Controller for a read operation as determined by the Flash Read Command field of the Header (as defined by Table 2 Flash Read Command Options).

After the image is read into SRAM, the Load Sequence calculates SHA-256 hash of the entire image.

The EC Firmware Binary Image signature is decrypted using the RSA Public Key 2 (modulus and exponent) contained in the Header. The EC Firmware Binary Image is validated by decrypting the 256-byte Signature located after the SRAM image in the SPI Flash and verifying that the calculated hash and the hash that was encrypted in the Signature match.

3.5 Runtime FW Digital Signature – K2

The Runtime FW Digital Signature is a DER encoded PKCS#1 v1.5 digital signature. The signature comprises a 256-bit SHA256 hash of the EC Runtime FW Binary Image, encrypted using RSA-2048, with the private key that corresponds to the public key stored in the RSA Public Key fields of the FW Image Header. It is stored Little-Endian.

4.0 LOAD FAILURES

If the ROM code fails to load a valid EC Runtime Binary Image for any reason, the ROM code exits. The ROM code writes to the ROM Event Log portion of SRAM and resets the part, using the Watchdog Timer.

4.1 ROM Event Log

A log of ROM processing from POR/Reset will be stored in the last 16 bytes of data SRAM space (0x11FFF0 - 0x11FFFF). The ROM will log strap detection and various states of the SPI read and verification process.

The ROM Event Log format is defined in the following table:

TABLE 4-1: ROM EVENT LOG FORMAT

Byte	Bits	Description
0x11FFF0	[7:4]	Current number of WDT resets
	[3]	Private SPI (SPI-1) Chip Select 0 pin sampled 0b = Not stable high 1b = High for 200us sample period
	[2]	SPI Voltage Rail (SPI-0 CS0 pin) 0b = SPI Voltage Rail down (low) 1b = SPI Voltage Rail Up (high for 500 us)
	[1]	RSMRST# pin state 0b = RSMRST# inactive (High) 1b = RSMRST# active (Low for 200us)
	[0]	1 = Halt due to Watch Dog Timer resetting the system 15 times.
0x11FFF1	[7:0]	Reserved
0x11FFF2	[7:0]	Bits[3:0] = SPI0 Tag 0 Read attempts Bit[4] = Good Tag has bit[23]==1 use CS1
0x11FFF3	[7:0]	SPI0 CS0/CS1 Tag 0 State see Table 4-2, "Log State Values"
0x11FFF4	[7:0]	Bits[3:0] = SPI0 Tag 1 Read attempts Bit[4] = Good Tag has bit[23]==1 use CS1
0x11FFF5	[7:0]	SPI0 CS0/CS1 Tag 1 State see Table 4-2, "Log State Values"
0x11FFF6	[7:0]	Bits[3:0] = SPI1 Tag 0 Read attempts Bit[4] = Good Tag has bit[23]==1 use CS1
0x11FFF7	[7:0]	SPI1 CS0/CS1 Tag 0 State see Table 4-2, "Log State Values"
0x11FFF8	[7:0]	Bits[3:0] = SPI1 Tag 1 Read attempts Bit[4] = Good Tag has bit[23]==1 use CS1
0x11FFF9	[7:0]	SPI1 CS0/CS1 Tag1 State see Table 4-2, "Log State Values"
0x11FFFA	[7:0]	Reserved
0x11FFFB	[7:0]	Reserved
0x11FFFC	[7:0]	Reserved
0x11FFFD	[7:0]	Reserved
0x11FFFE	[7:0]	Reserved
0x11FFFF	[7:0]	Reserved

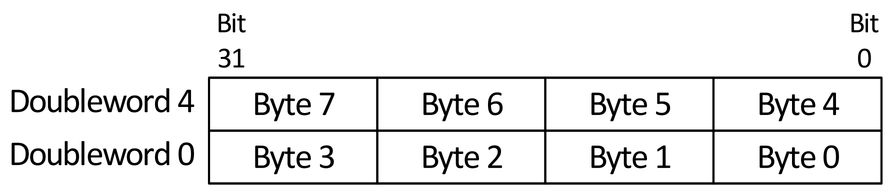
TABLE 4-2: LOG STATE VALUES

Value	Description
0x00	Device (SPIx TAGy) Loader state machine not entered
0x01	SPI Read of Header and RSA signature successful
0x02	Header Title OK (‘SMSC’ in first 4 bytes)
0x03	Header RSA Signature Decryption successful
0x04	Header RSA Signature Authentication OK
0x05	Header Payload Length OK
0x06	Header Payload Load Address Aligned on 64-byte Boundary OK
0x07	Header Content Check OK
0x08	SPI Read of Payload RSA signature successful
0x09	Payload RSA Signature Decryption successful
0x0A	SPI Read of Payload successful
0x0B	Payload RSA signature Authentication OK
0x0C	ROM Launching Payload
0x0D – 0xFF	Unused

5.0 SHA-256 HASHING AND DATA ORDER

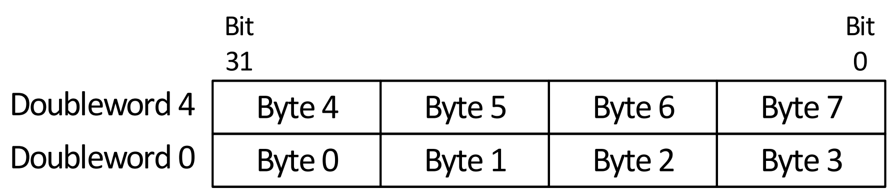
The CEC1302, along with PCs based on the Intel x86 architecture, employ a “Little-Endian” byte order: memory addresses bytes, and within a 32-bit doubleword, the least significant bit. The following figure illustrates Little-Endian byte order:

FIGURE 5-1: LITTLE ENDIAN BYTE ORDER



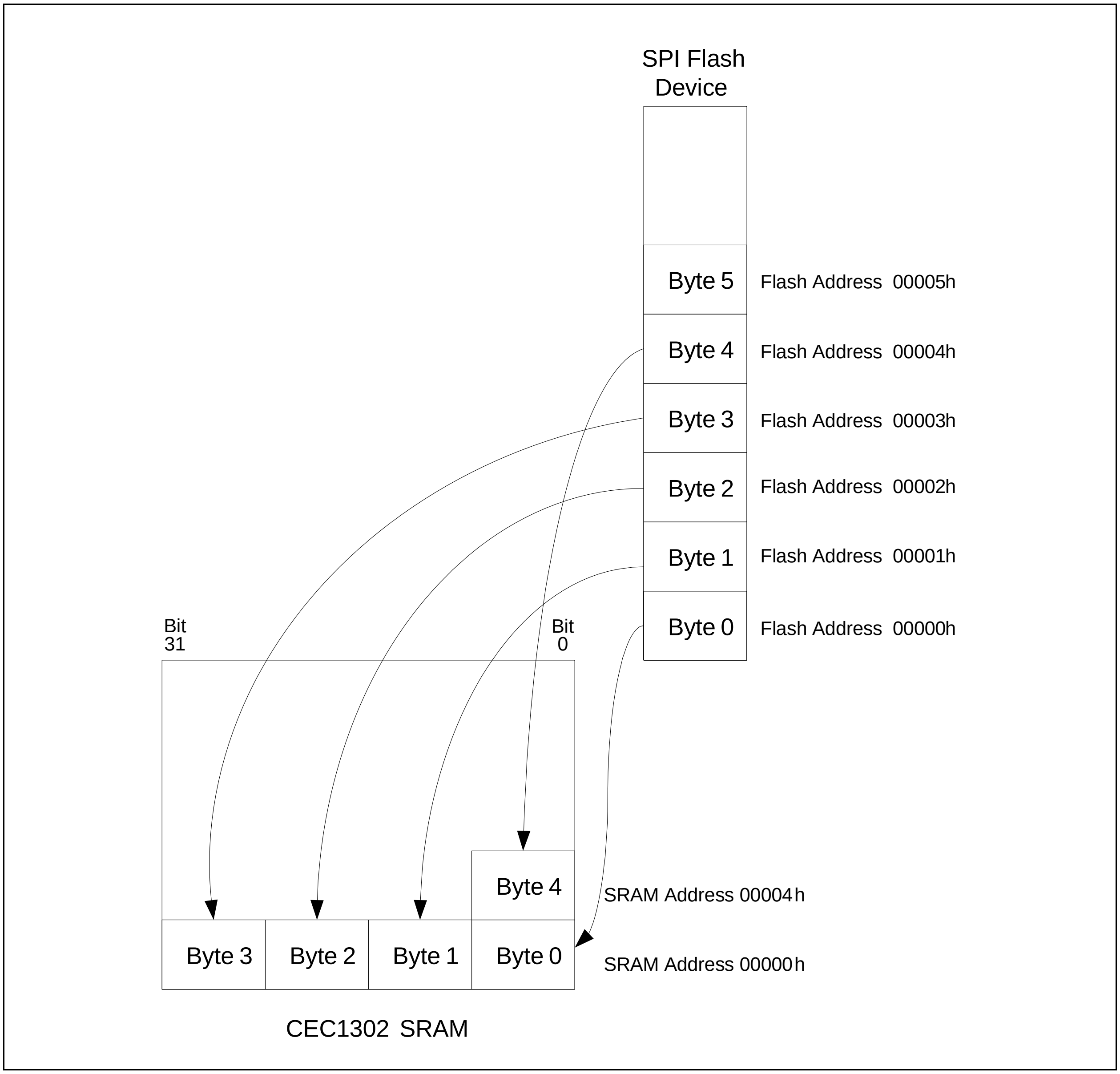
Other systems, in particular networking systems, use “Big Endian” byte order, in which byte 0 within a 32-bit doubleword addresses the most significant byte of the word and byte 3 addresses the least significant byte:

FIGURE 5-2: BIG ENDIAN BYTE ORDER



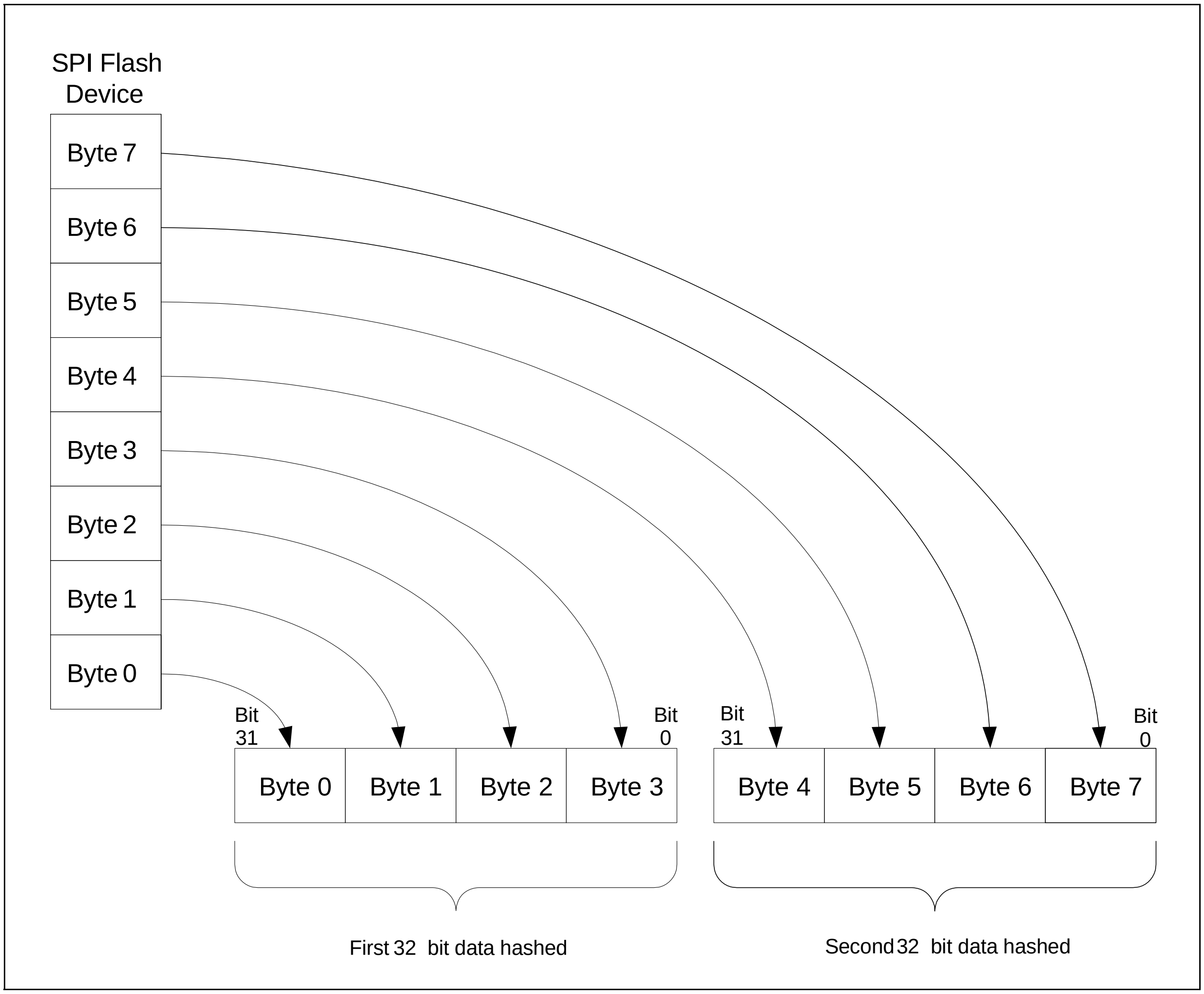
The boot ROM loads data from a byte-oriented SPI Flash into SRAM. The ROM firmware moves sequential bytes from the SPI Flash into the CEC1302 SRAM in a Little-Endian order, as shown in the following figure:

FIGURE 5-3: MAPPING SPI DATA TO SRAM



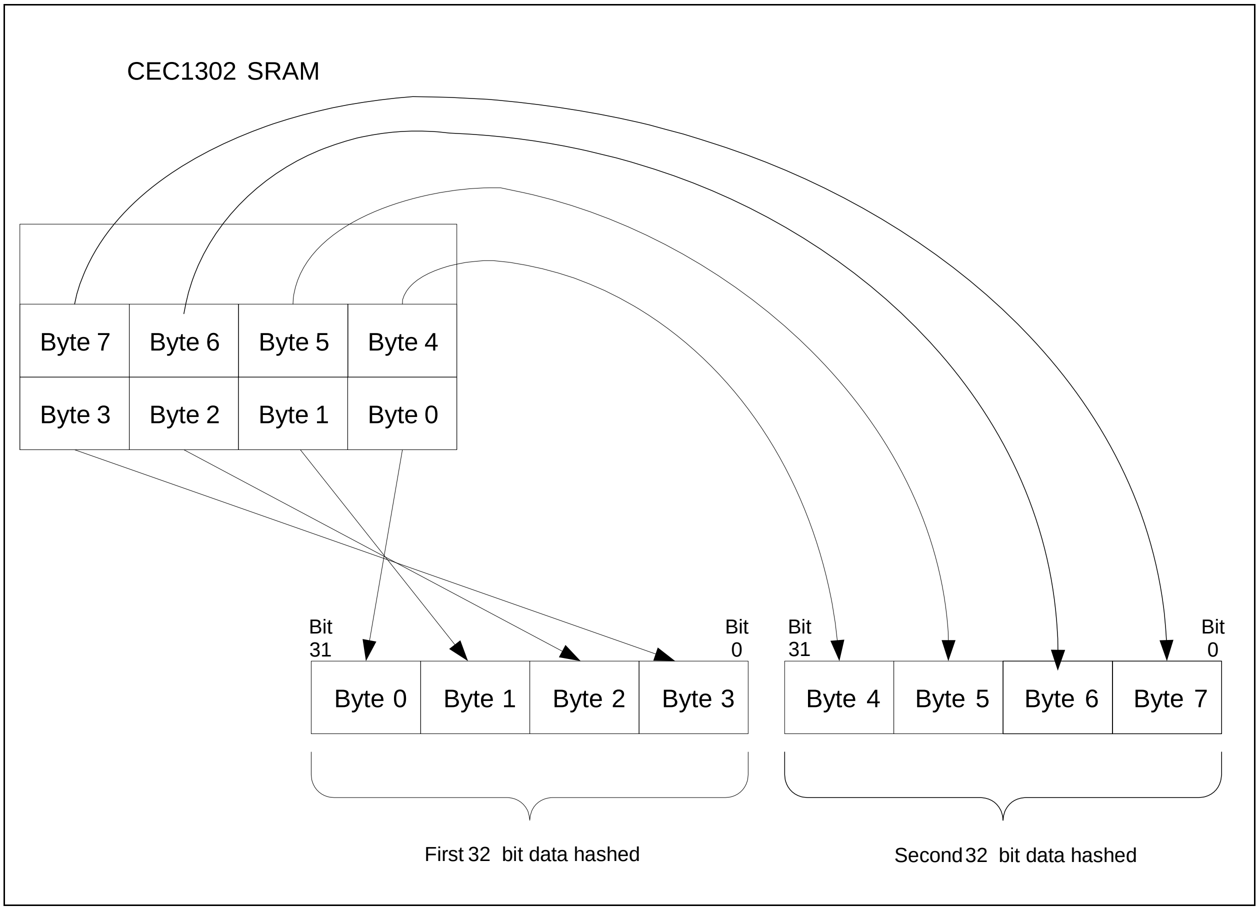
Because most network-oriented software tools that deal with SHA hashing are Big-Endian, most of the tools hash data that went into the SPI Flash device in the order shown in the following figure:

FIGURE 5-4: NETWORK-ORIENTED HASH



In order to be compatible with software tools that are used to hash the image that is loaded into the Flash, the CEC1302 automatically reverses the byte order of data in SRAM when sending it to the Hash Engine:

FIGURE 5-5: MAPPING FROM SRAM TO HASH



See the documents referenced in **Section 6.0 “ROM Runtime API”** for information on the APIs that are used for SHA-256.

6.0 ROM RUNTIME API

The CEC1302 ROM provides a number of application programming interfaces to assist the EC application firmware. The APIs that are provided for the device fall into the following categories:

- SPI Access
- Security
- Miscellaneous Functions

The full descriptions of these APIs are contained in these referenced documents:

- CEC1302 Crypto API User's Guide
- CEC1302 Peripheral API User's Guide.

Sample code is contained in the following document:

- CEC1302 CLIB/PLIB Releases

APPENDIX A: ADDENDUM REVISION HISTORY

TABLE A-1: REVISION HISTORY

Revision	Section/Figure/Entry	Correction
DS00002235A (07-14-16)	Document Release	

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://microchip.com/support

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

ISBN: 9781522407782

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

ASIA/PACIFIC

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7828

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Dusseldorf
Tel: 49-2129-3766400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Venice
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Poland - Warsaw
Tel: 48-22-3325737

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

06/23/16