

C*ommon* F*lash Memory* I*nterface* **Specification**

Release 1.0
July 25, 1996

Intel Corporation
1900 Prairie City Rd, Folsom CA 95630-9598

CFI SPECIFICATION AGREEMENT

This is a royalty-free, reciprocal patent license for parties wishing to adopt the CFI Specification in their products. By making use of this specification, you (“User”) are agreeing to be bound by the terms of this agreement. If you do not agree to them, then you have no license to use the specification, and you should destroy these materials or return them to the CFI Promoter from whom this Agreement was obtained.

“**CFI Specification**” means a revision of the “*Common Flash Interface Specification*,” numbered 1.0 or greater, published and made available for industry licensing by the CFI Promoters.

“**CFI Promoters**” means Intel Corporation, Advanced Micro Devices, Fujitsu Limited, and Sharp Corporation.

Agreement: Effective as of User’s acceptance of this Agreement, and subject to its terms and conditions CFI Promoters and User agree as follows:

License: CFI Promoters and User each grant to the other and its subsidiaries, under any claim of a patent or patent application otherwise infringed, a non-exclusive, royalty-free, non-transferable, world-wide license, without rights to sublicense, to make or have made such party’s products which comply with the CFI Specification solely in connection with meeting the CFI Specification, and to use, sell, offer to sell, and import such products, where infringement of such claims would not have occurred but for the incorporation of the CFI Specification in such products, and there is no feasible alternative to such infringement.

No Other Licenses. Except for the rights expressly provided by this Agreement, neither party grants or receives, by implication, or estoppel, or otherwise, any rights under any patents or other intellectual property rights.

LIMITATION OF LIABILITY: The CFI Specification is provided "AS IS" without warranty of any kind. THE CFI PROMOTERS OFFER NO WARRANTY EITHER EXPRESS OR IMPLIED INCLUDING THOSE OF MERCHANTABILITY, NONINFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY OR FITNESS FOR A PARTICULAR PURPOSE. THE CFI PROMOTERS SHALL NOT BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR OTHER LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE CFI SPECIFICATION, EVEN IF THE CFI PROMOTERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME JURISDICTIONS PROHIBIT THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY.

TERMINATION OF THIS LICENSE: The CFI Promoters may terminate this license at any time if you are in breach of any of its terms and conditions. Upon termination, you will immediately destroy the CFI Specification or return all copies of the same you have made to the CFI Promoters.

U.S. GOVERNMENT RESTRICTED RIGHTS: The CFI Specification is provided with "RESTRICTED RIGHTS." Use, duplication or disclosure by the Government is subject to restrictions set forth in FAR52.227-14 and DFAR252.227-7013 *et seq.* or its successor. Use of the CFI Specification by the Government constitutes acknowledgment of CFI Promoters’ rights in them .

APPLICABLE LAWS: Any claim arising under or relating to this Agreement shall be governed by the laws of Delaware. You may not export the CFI Specification or products in compliance thereto in violation of applicable export laws.

CFI SPECIFICATION

Release 1.0

| Revision Record | | |
|------------------------|-----------------------|-------------------------|
| Edition | Date Published | Revised Contents |
| 1.00 | 07/25/1996 | First Draft Release |

Intel CFI Program Office

c/o Alan Hanson
Intel Corporation
1900 Prairie City Road, Mailstop FM3-123
Folsom, Ca. 95630-9598

Notice

This Specification is hereby provided to you for your use subject to the terms of the enclosed CFI Specification Agreement.

Intel retains the right to make changes to this document at any time, without notice.

Table of Contents

- 1. Introduction**
 - 1.1. Purpose**
 - 1.2. Scope**
- 2. CFI Overview**
 - 2.1. CFI Operational Summary**
- 3. CFI Hardware Interface**
 - 3.1. CFI Query Command Interface**
 - 3.2. Query Structure Output**
 - 3.3. CFI Query Structure**
- 4. Extensibility**

1. CFI Introduction

1.1. Purpose

The Common Flash Interface (CFI) specification outlines device and host system software interrogation handshake which allows specific vendor-specified software algorithms to be used for entire families of devices. This allows device-independent, JEDEC ID-independent, and forward- and backward-compatible software support for the specified flash device families. It allows flash vendors to standardize their existing interfaces for long-term compatibility.

1.2. Scope

This release of the specification defines the basic Query interface for CFI-compliant devices. This allows parameterization of known and future flash Read/Write/Erase control interfaces. This Query structure attempts to define all the critical parameters relevant to a broad base of flash memory devices. The CFI specification will not specify detail command sets, status polling methods, and software algorithms of individual flash vendors. A 16-bit ID code is assigned to specific manufacturers' interfaces, and it is up to that manufacturer to provide these detailed specifications.

2. CFI Overview

2.1. CFI Operational Summary

After the Query command code has been issued, the device enters the Query mode, allowing read output of the CFI Query data structure. The CFI Query data structure contains a 16-bit Command Set and Control Interface ID code which specifies a vendor-specific control interface for a family of flash devices. Query also contains general, common flash memory parameters and vendor-specified data areas. These provide all the necessary information for controlling Read/Write/Erase operations of a particular family of flash devices according to a vendor-specified interface. Any additional information not covered in the common CFI Query data structure is located in vendor-specific extended Query tables, the address location(s) of which is (are) contained in the general CFI Query structure.

3. CFI Hardware Interface

3.1. CFI Query Command Interface

The CFI Query structure is accessed similar to the existing “ID Mode” or “JEDEC ID” access for nonvolatile memories, but uses a different, non-conflicting command code. The Query access command is 98h, while the JEDEC ID mode access mode is 90h. The Query addressing is always relative to the device word (largest supported) with data always presented on the lowest order byte (D7 - D0 outputs).

Nonvolatile memory devices are assumed to power up in a read-only state. Independent of that assumption, the Query structure contents must be able to be read at the specific address locations following a single system write cycle where: 1) a 98h Query command code is written to 55h address location within the device’s address space (in maximum device buswidth), and 2) the device is in any valid read state, such as “Read Array” or “Read ID Data.” Other device states may exist within a long sequence of commands or data input; such sequences must first be completed or terminated before the writing of the 98h Query command code will result in valid Query data structure output.

Note that for devices wider than 8 bits, the valid Query access code has all zeroes (0’s) in upper bytes of the data bus. Thus the 16-bit Query command code is 0098h and the 32-bit Query command code is 00000098h.

A CFI-compliant device must allow selection and deselection of the Query output mode to and from normal read array operation with a single command write cycle so that the desired data are accessible in the second of two active bus cycles, i.e. bus cycles in which the devices Chip Enable(s) are active.

Table 3.1 Command Write Cycles for Query Select & Deselect

| Command | # of Cycles | First Bus Cycle | | | Second Bus Cycle | | |
|------------|-------------|-----------------|---------|---------|------------------|---------|------|
| | | Oper | Address | Data | Oper | Address | Data |
| Read Array | ≥ 2 | Write | X | FFh/F0h | Read | AA | AD |
| Query | ≥ 2 | Write | 55h | 98H | Read | QA | QD |

Notes:

1. "Address" is the location in maximum device buswidth
2. Flash devices may or may not have address sensitive query commands. Device drivers should always supply 55h on the address bus and 98h on the data bus to enter query mode, however Flash devices may choose to ignore the address bus and enter query mode if 98h is seen on the data bus only
3. A flash vendor must define other command sequences for other mode accesses as part of the Vendor-specific Algorithm and Control Interface specification referenced by the appropriate CFI ID code. Access to and from Query and Read Array modes from any other mode may require additional command sequences.
4. Abbreviations for inputs and outputs of the second cycle refer to address/data for the normal flash array (AA, AD) and Query structure (QA, QD), which may be accessed in random order.

Table 3.1.1 Summary of Command Sequence as a function of device and mode

| Device type / mode | Command location in maximum device buswidth addresses | Command data | Command address location in bytes | Command data with byte addressing |
|-----------------------|---|--------------|-----------------------------------|--|
| x8 device / x8 mode | 55h | 98h | 55h | 98h |
| x16 device / x16 mode | 55h | 0098h | AAh | AAh: 98h ABh: 00h |
| x16 device / x8 mode | N/A * | N/A * | AAh | AAh: 98h |
| x32 device / x32 mode | 55h | 00000098h | 154h | 154h: 98h 155h: 00h 156h: 00h 157h: 00h |
| x32 device / x8 mode | N/A * | N/A * | 154h | 154h: 98h |

Notes:

- * The system must drive the lowest order addresses to access all the device's array data when the device is configured in x8 mode. Therefore, word addressing where these lower addresses not toggled by the system is "Not Applicable" for x8-configured devices.
- Flash devices may or may not have address sensitive query commands. Device drivers should always supply 55h on the address bus and 98h on the data bus to enter query mode, however Flash devices may choose to ignore the address bus and enter query mode if 98h is seen on the data bus only

Table 3.1.2 Example of Query Command Sequence of a x8/x16 Capable Device with an address sensitive Query Command

| Binary Address | -- x16 Mode (BYTE#=1) Address : Data | Binary Address | -- x8 Mode (BYTE#=0) Address : Data |
|---|---|--|---|
| A ₈ A ₇ A ₆ A ₅ A ₄ A ₃ A ₂ A ₁ | A ₁₆ - A ₁ : D ₁₅ - D ₀ | A ₇ A ₆ A ₅ A ₄ A ₃ A ₂ A ₁ A ₀ | A ₁₅ - A ₀ : D ₇ - D ₀ |
| A ₇ A ₆ A ₅ A ₄ A ₃ A ₂ A ₁ A ₀ | A ₁₅ - A ₀ : D ₁₅ - D ₀ | A ₆ A ₅ A ₄ A ₃ A ₂ A ₁ A ₀ A ₋₁ | A ₁₄ - A ₋₁ : D ₇ - D ₀ |
| 0 1 0 1 0 1 0 1 | 0055h : 0098h | 1 0 1 0 1 0 1 0 | 00AAh : 98h |

Note: Address examples provided for devices with least significant byte address of A₀ or A₋₁

3.2. Query Structure Output

Query data are always presented on the lowest-order data outputs (D7 - D0) only. The numerical offset value is the address relative to the maximum bus width supported by the device. The Query table device starting address is a 10h byte address for a byte-wide (x8) device, 10h word address for word-wide (x16) device, 10h “dword” address for a x32 device, etc.

Thus for the byte-wide (x8) device, the first 2 bytes of the Query structure, “Q” and “R” in ASCII, appear at device addresses 10h and 11h, which is the same as the absolute byte address. These same data appear on the low byte at word addresses 10h and 11h in a word-wide (x16) device. A CFI-compliant device must output 00H data on upper bytes. Thus, a x16 device outputs ASCII “Q” in the low byte (D7-D0) and 00h in the high byte (D15-D8). The same logic extends to x32 and larger devices, such that: 1) the data are presented in the lowest byte, 2) the data are addressed in maximum-buswidth-relative addresses, and 3) the upper bytes in each data word are filled with 00h data. Thus outputs D31 - D8 of a x32 device present 00h data during Query read, starting at dword address 10h or byte-relative address 40h.

In devices that are x8/x16 capable, the x8 data is still presented in word-relative (16-bit) addresses. However, the “fill data” (00h) is not the same as driven by the upper bytes in the x16 mode. As in x16 mode, the byte address (A_0 or A_{-1} depending on pinout) is ignored for Query output so that the “odd byte address” (A_0 or A_{-1} high) repeats the “even byte address” data (A_0 or A_{-1} low). Therefore, in x8 mode using byte addressing, such devices will output the sequence “Q”, “Q”, “R”, “R”, “Y”, “Y”, and so on, beginning at byte-relative address 20h (which is equivalent to word offset 10h in x16 mode). Again, this is extensible to x32 and wider devices in that byte addresses are ignored during Query output in x8 mode such that: 1) Query data appears to repeat at each byte address within a word and, 2) the Query data starts at the byte address 10h times the number of bytes of maximum device buswidth.

Table 3.2. Summary of Query Structure Output as a Function of Device and Mode

| Device type / mode | Query start location in maximum device buswidth addresses | Query data with maximum device buswidth addressing "x" = ASCII equivalent | Query start address in bytes | Query data with byte addressing |
|-----------------------|---|---|------------------------------|---|
| x8 device /x8 mode | 10h | 10h: 51h "Q" 11h: 52h "R" 12h: 59h "Y" | 10h | 10h: 51h "Q" 11h: 52h "R" 12h: 59h "Y" |
| x16 device / x16 mode | 10h | 10h: 0051h "Q" 11h: 0052h "R" 12h: 0059h "Y" | 20h | 20h: 51h "Q" 21h: 00h null 22h: 52h "R" |
| x16 device / x8 mode | N/A * | N/A * | 20h | 20h: 51h "Q" 21h: 51h "Q" 22h: 52h "R" |
| x32 device / x32 mode | 10h | 10h: 00000051h "Q" 11h: 00000052h "R" 12h: 00000059h "Y" | 40h | 40h: 51h "Q" 41h: 00h null 42h: 00h null 43h: 00h null 44h: 52h "R" |
| x32 device / x8 mode | N/A * | N/A * | 40h | 40h: 51h "Q" 41h: 51h "Q" 42h: 51h "Q" 43h: 51h "Q" 44h: 52h "R" |

* NOTE: The system must drive the lowest order addresses to access all the device's array data when the device is configured in x8 mode. Therefore, word addressing where these lower addresses not toggled by the system is "Not Applicable" for x8-configured devices.

Table 3.3. Example of Query Structure Output of a x8/x16 Capable Device

| Binary Address | -- x16 Mode (BYTE#=1) Address : Data | Binary Address | -- x8 Mode (BYTE#=0) Address : Data |
|---|---|--|--|
| A ₆ A ₅ A ₄ A ₃ A ₂ A ₁ | A ₁₆ - A ₁ : D ₁₅ - D ₀ | A ₅ A ₄ A ₃ A ₂ A ₁ A ₀ | A ₇ - A ₀ : D ₇ - D ₀ |
| A ₅ A ₄ A ₃ A ₂ A ₁ A ₀ | A ₁₅ - A ₀ : D ₁₅ - D ₀ | A ₄ A ₃ A ₂ A ₁ A ₀ A ₋₁ | A ₆ - A ₋₁ : D ₇ - D ₀ |
| 0 1 0 0 0 0 | 0010h: 0051h " Q" | 1 0 0 0 0 0 | 20h: 51h "Q" |
| 0 1 0 0 0 1 | 0011h: 0052h " R" | 1 0 0 0 0 1 | 21h: 51h "Q" |
| 0 1 0 0 1 0 | 0012h: 0059h " Y" | 1 0 0 0 1 0 | 22h: 52h "R" |
| 0 1 0 0 1 1 | 0013h: P_ID _{LO} PrVendor | 1 0 0 0 1 1 | 23h: 52h "R" |
| 0 1 0 1 0 0 | 0014h: P_ID _{HI} ID # | 1 0 0 1 0 0 | 24h: 59h "Y" |
| 0 1 0 1 0 1 | 0015h: P_ADR _{LQ} PrVendor | 1 0 0 1 0 1 | 25h: 59h "Y" |
| 0 1 0 1 1 0 | 0016h: P_ADR _{HI} TblAdr | 1 0 0 1 1 0 | 26h: P_ID _{LO} PrVendor |
| 0 1 0 1 1 1 | 0017h: A_ID _{LO} AltVendor | 1 0 0 1 1 1 | 27h: P_ID _{LO} ID # |
| 0 1 1 0 0 0 | 0018h: A_ID _{HI} ID # | 1 0 1 0 0 0 | 28h: P_ID _{HI} " |
| ... | ... | ... | ... |

Note: Address examples provided for devices with least significant byte address of A₀ or A₋₁

3.3. CFI Query Structure

3.3.1. Query Structure Overview

The Query command causes the flash component to display the CFI Query structure or “database.” The structure sub-sections and address locations are summarized as follows:

| Offset | Sub-section Name | Description |
|----------|--|---|
| 00h | <i>Reserved</i> | <i>Reserved for vendor-specific information</i> |
| 10h | CFI Query Identification String | Command set ID and vendor data offset |
| 1Bh | System Interface Information | Device timing & voltage information |
| 27h | Device Geometry Definition | Flash device layout |
| P | Primary Vendor-specific Extended Query table | Vendor-defined additional information specific to the Primary Vendor Algorithm(<i>optional</i>) |
| A | Alternate Vendor-specific Extended Query table | Vendor-defined additional information specific to the Alternate Vendor Algorithm(<i>optional</i>) |

The following sections describe the Query structure sub-sections in detail.

3.3.2. CFI Query Identification String

The Identification String provides verification that the component supports the Common Flash Interface specification. Additionally, it indicates which version of the spec and which Vendor-specified command set(s) is(are) supported.

| Offset | Length (bytes) | Description |
|--------|-------------------------|---|
| 10h | 03h | Query-unique ASCII string "QRY" |
| 13h | 02h | Primary Vendor Command Set and Control Interface ID Code 16-bit ID code defining a specific Vendor-specified algorithms <i>[Refer to CFI Publication 100]</i> |
| 15h | 02h value = <i>P</i> | Address for Primary Algorithm extended Query table Note: Address 0000h means that no extended table exists |
| 17h | 02h | Alternate Vendor Command Set and Control Interface ID Code second vendor-specified algorithm supported by the device <i>[Refer to CFI Publication 100]</i> Note: ID Code = 0000h means that no alternate algorithm is employed |
| 19h | 02h value = <i>A</i> | Address for Alternate Algorithm extended Query table Note: Address 0000h means that no alternate extended table exists |

Notes:

1. Refer to Query Data Output section of Device Hardware interface for the detailed definition of offset address as a function of device wordwidth and mode.
2. The CFI specification allows for replacement of the standard Query table contents. When the Vendor Primary Algorithm extended Query table address points to any address between 10h and 32h, the standard Query table contents are assumed to be "replaced" at those address. Thus, some or all of the standard Query may be replaced. For example, a Vendor Primary Algorithm extended Query table address of 28h means that the standard Device Geometry definition has been replaced. The System Interface information at locations 1Bh to 27h may be assumed valid, but the ultimate definition must be spelled out in the appropriate specification for the particular vendor algorithm.

3.3.3. CFI Query System Interface Information

The following device information can be useful in optimizing system interface software.

| Offset | Length (bytes) | Description |
|--------|----------------|--|
| 1Bh | 01h | Vcc Logic Supply Minimum Write/Erase voltage bits 7 - 4 BCD value in volts bits 3 - 0 BCD value in 100 millivolts |
| 1Ch | 01h | Vcc Logic Supply Maximum Write/Erase voltage bits 7 - 4 BCD value in volts bits 3 - 0 BCD value in 100 millivolts |
| 1Dh | 01h | Vpp [Programming] Supply Minimum Write/Erase voltage bits 7 - 4 HEX value in volts bits 3 - 0 BCD value in 100 millivolts Note: This value must be 0000h if no Vpp pin is present |
| 1Eh | 01h | Vpp [Programming] Supply Maximum Write/Erase voltage bits 7 - 4 HEX value in volts bits 3 - 0 BCD value in 100 millivolts Note: This value must be 0000h if no Vpp pin is present |
| 1Fh | 01h | Typical timeout per single byte/word write (buffer write count = 1), 2^N u-sec |
| 20h | 01h | Typical timeout for maximum-size buffer write, 2^N u-sec (if supported; 00h=not supported) |
| 21h | 01h | Typical timeout per individual block erase, 2^N m-sec |
| 22h | 01h | Typical timeout for full chip erase, 2^N m-sec (if supported; 00h=not supported) |
| 23h | 01h | Maximum timeout for byte/word write, 2^N times typical (offset 1Fh) |
| 24h | 01h | Maximum timeout for buffer write, 2^N times typical (offset 20h) (00h=not supported) |
| 25h | 01h | Maximum timeout per individual block erase, 2^N times typical (offset 21h) |
| 26h | 01h | Maximum timeout for chip erase, 2^N times typical (offset 22h) (00h=not supported) |

3.3.4. Device Geometry Definition

This field provides critical details of the flash device geometry.

| Offset | Length (bytes) | Description |
|-------------|----------------|--|
| 27h | 01h | Device Size = 2^n in number of bytes. |
| 28h | 02h | Flash Device Interface description [<i>Refer to CFI Publication 100</i>] |
| 2Ah | 02h | Maximum number of bytes in multi-byte write = 2^2 |
| 2Ch | 01h | <p>Number of Erase Block Regions within device bits 7-0 = x = number of Erase Block Regions</p> <p>Notes:</p> <ol style="list-style-type: none"> x = 0 means no erase blocking, i.e. the device erases at once in "bulk." x specifies the number of regions within the device containing one or more contiguous Erase Blocks of the same size. For example, a 128KB device (1Mb) having blocking of 16KB, 8KB, four 2KB, two 16KB, and one 64KB is considered to have 5 Erase Block Regions. Even though two regions both contain 16KB blocks, the fact that they are not contiguous means they are separate Erase Block Regions. By definition, symmetrically block devices have only one blocking region. |
| 2Dh | 04h | <p>Erase Block Region Information</p> <p>bits 31- 16 = z, where the Erase Block(s) within this Region are (z) times 256 bytes in size. The value z = 0 is used for 128-byte block size. e.g. for 64KB block size, z = 0100h = 256 => 256 * 256 = 64K</p> <p>bits 15 - 0 = y, where y+1 = Number of Erase Blocks of identical size within the Erase Block Region: e.g. y = D15-D0 = FFFFh => y+1 = 64K blocks [maximum number] y = 0 means no blocking (# blocks = y+1 = "1 block") Note: y = 0 value must be used with # of block regions of one as indicated by (x) = 0</p> |
| 31h - (k-1) | 04h per entry | <p>additional Erase Block Region Information, 4 bytes per region</p> <p>Notes:</p> <ol style="list-style-type: none"> The total number of blocks times individual block size must add up to the device size. The address K is next available Query address at end of the Device Geometry structure. It is the first possible starting address of the optional vendor-specific Query table(s) (i.e. Address "P," the Primary Vendor-specific extended Query table offset, must be $\geq k$). This address is equal to the CFI offset address 2Eh (address of Erase Block Region #1) plus $4*x$, i.e. 4 bytes per entry times the number of Erase Block Region entries. |

3.3.5. Optional Vendor-Specific Extended Query Tables

Certain flash features and commands may be optional in a vendor-specific algorithm specification. The optional vendor-specific Query table(s) may be used to specify this and other types of information. These structures are defined solely by the flash vendor(s).

Primary Vendor-Specific Extended Query Table

| Offset | Length (bytes) | Description |
|--------|----------------|---|
| (P)h | 03h | Primary Algorithm extended Query table unique ASCII string "PRI" |
| (P+3)h | 01h | Major version number, ASCII |
| (P+4)h | 01h | Minor version number, ASCII |
| (P+5)h | variable | Vendor-specific extended Query table contents for Primary Algorithm |

Alternate Vendor-Specific Extended Query Table

| Offset | Length (bytes) | Description |
|--------|----------------|---|
| (A)h | 03h | Alternate Algorithm extended Query table unique ASCII string "ALT" |
| (A+3)h | 01h | Major version number, ASCII |
| (A+4)h | 01h | Minor version number, ASCII |
| (A+5)h | variable | Vendor-specific extended Query table contents for Alternate Algorithm |

4. Extensibility

The CFI specification supports extensibility for future device characteristics through the vendor-specific extended Query table(s). Anything not defined in the common CFI Query database is to be defined in the vendor extended tables, with the detailed structure of such tables defined by the major and minor vendor revision numbers and the associated vendor-supplied Command Set and Control Interface specification.