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

Release 2.0
December 1, 2001

AMD
1 AMD Place, Sunnyvale, CA 94088-3453

CFI SPECIFICATION AGREEMENT

CFI SPECIFICATION

Revision Record		
Edition	**Date Published**	**Revised Contents**
1.00	07/25/1996	First Draft Release
2.00	12/01/2001	Second Draft Release

Notice

Table of Contents

1. CFI Introduction

1.1. Purpose

The Common Flash Interface (CFI) specification outlines device and host system software interrogation handshake that 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 that 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 bus-width), 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 de-selection 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

	# of	First Bus Cycle			Second Bus Cycle		
Command	Cycles	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 bus-width
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.

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 bus width addresses	Command data	Command address location in bytes	Command data with byte addressing
x8 device / x8 mode	55h [2]	98h	55h	98h
x16 device / x16 mode	55h [2]	0098h	AAh	AAh: 98h ABh: 00h
x16 device / x8 mode	N/A [1]	N/A [1]	AAh	AAh: 98h
x32 device / x32 mode	55h [2]	00000098h	154h	154h: 98h 155h: 00h 156h: 00h 157h: 00h
x32 device / x16 mode	N/A [3]	N/A [3]	154h	154h: 98h 155h: 00h
x32 device / x8 mode	N/A [1]	N/A [1]	154h	154h: 98h

Notes:

1. 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 or double addressing where the lower addresses are not toggled by the system is "Not Applicable" for x8-Configured devices.
2. Flash devices may or may not have address sensitive query commands. Device drivers should always supply 55h on the address bus (55h, AAh, and 154h respectively for command address location in bytes on x8, x16 and x32 data bus configured devices) 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. Same as note 1 above but change x8 references to x16.

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
$A_8A_7A_6A_5A_4A_3A_2A_1$	A_{16} - A_1 : D_{15} - D_0	$A_7A_6A_5A_4A_3A_2A_1A_0$	A_{15} - A_0 : D_7 - D_0
$A_7A_6A_5A_4A_3A_2A_1A_0$	A_{15} - A_0 : D_{15} - D_0	$A_6A_5A_4A_3A_2A_1A_0A_{-1}$	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 A_0 or A_{-1}

Table 3.1.3 Example of Query Command Sequence of a x16/x32 Capable Device with an address sensitive Query Command

Binary Address	-- x32 Mode (WORD#=1) Address : Data	Binary Address	-- x16 Mode (WORD#=0) Address : Data
$A_9A_8A_7A_6A_5A_4A_3A_2$	A_{33} – A_2 : D_{31} - D_0	$A_8A_7A_6A_5A_4A_3A_2A_1$	A_{16} – A_1 : D_{15} - D_0
$A_8A_7A_6A_5A_4A_3A_2A_1$	A_{32} – A_1 : D_{31} - D_0	$A_7A_6A_5A_4A_3A_2A_1A_0$	A_{15} – A_0 : D_{15} - D_0
$A_7A_6A_5A_4A_3A_2A_1A_0$	A_{31} – A_0 : D_{31} - D_0	$A_6A_5A_4A_3A_2A_1A_0A_{-1}$	A_{14} – A_{-1} : D_{15} - D_0
0 1 0 1 0 1 0 1	0055h : 00000098h	1 0 1 0 1 0 1 0	00AAh : 0098h

Note: Address examples provided for devices with least significant byte address of A_1, A_0, 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 “d-word” 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, an 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-bus-width-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 d-word 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 pin-out) 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 bus-width.

Table 3.2.1 Summary of Query Structure Output as a Function of Device and Mode

Device type / mode	Query start location in maximum device bus-width addresses	Query data with maximum device Bus-width addressing "x" = ASCII equivalent	Query data with word addressing	Query start address in byte addressing	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 *[Note 1]*	N/A *[Note 1]*	-	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 / x16 mode	N/A *[Note 2]*	N/A *[Note 2]*	20h: 0051h "Q" 21h: 0051h "Q" 22h: 0052h "R" 23h: 0052h "R"	40h	40h: 51h "Q" 41h: 00h null 42h: 51h "Q" 43h: 00h null 44h: 52h "R" 45h: 00h null 46h: 52h "R"
x32 device / x8 mode	N/A *[Note 1]*	N/A *[Note 1]*		40h	40h: 51h "Q" 41h: 51h "Q" 42h: 51h "Q" 43h: 51h "Q" 44h: 52h "R"

Note 1: 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 or double-word addressing where these lower addresses are not toggled by the system is "Not Applicable" for x8-configured devices.
Note 2: Same as note 1 above but change x8 references to x16.

Table 3.2.2 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	
$A_6A_5A_4A_3A_2A_1$	A_{16} - A_1 : D_{15} - D_0		$A_5A_4A_3A_2A_1A_0$	A_7 - A_0 : D_7 - D_0	
$A_5A_4A_3A_2A_1A_0$	A_{15} - A_0 : D_{15} - D_0		$A_4A_3A_2A_1A_0A_{-1}$	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_{LO}	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 A_0 or A_{-1}

Table 3.2.3 Example of Query Structure Output of a x16/x32 Capable Device

Binary Address	-- x32 Mode (WORD#=1) Address : Data		Binary Address	-- x16 Mode (WORD#=0) Address : Data	
$A_7A_6A_5A_4A_3A_2$	A_{33} – A_2 : D_{31} - D_0		$A_6A_5A_4A_3A_2A_1$	A_{16} – A_1 : D_{15} - D_0	
$A_6A_5A_4A_3A_2A_1$	A_{32} – A_1 : D_{31} - D_0		$A_5A_4A_3A_2A_1A_0$	A_{15} – A_0 : D_{15} - D_0	
$A_5A_4A_3A_2A_1A_0$	A_{31} – A_0 : D_{31} - D_0		$A_4A_3A_2A_1A_0A_{-1}$	A_{14} – A_{-1} : D_{15} - 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_{LO}	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 word address of A_1, A_0, or A_{-1}

AltVendor = Alernate Vendor
PrVendor = Primary Vendor
TblAdr = Table Address

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:

Table 3.3.1 Query Structure Overview

Offset	Sub-section Name	Description
00h	*Reserved*	*Reserved for vendor-specific information*
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 *(optional)*
A	Alternate Vendor-specific Extended Query table	Vendor-defined additional information specific to the Alternate Vendor Algorithm *(optional)*

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.

Table 3.3.2 CFI Query Identification String

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 *[Refer to CFI Publication 100]*
15h	02h value = ***P***	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 *[Refer to CFI Publication 100]* Note: ID Code = 0000h means that no alternate algorithm is employed
19h	02h value = ***A***	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 word-width 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, all or some 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.

Table 3.3.3 CFI Query System Interface Information

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 microsecond
20h	01h	Typical timeout for minimum-size buffer write, 2^N microsecond (if supported; 00h=not supported)
21h	01h	Typical timeout per individual block erase, 2^N millisecond
22h	01h	Typical timeout for full chip erase, 2^N millisecond (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.

Table 3.3.4 Device Geometry Definition

Offset	**Length (bytes)**	**Description**
27h	01h	Device Size = 2^n in number of bytes.
28h	02h	Flash Device Interface description *[Refer to Appendix]*
2Ah	02h	Maximum number of bytes in multi-byte write = 2^n.
2Ch	01h	Number of Erase Block Regions within device **bits 7-0 = x** = number of Erase Block Regions Notes: 1. x = 0 means no erase blocking, i.e. the device erases at once in "bulk." 2. 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. 3. By definition, symmetrical block devices have only one blocking region.
2Dh	04h	Erase Block Region Information **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 **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 Notes: 1. Erase Blocks always start at address 0 of the Bottom Boot or Uniform Low Version of the Flash Device
31h - (k-1)	04h per entry	Additional Erase Block Region Information, 4 bytes per region Notes: 1. The total number of blocks times individual block size must add up to the device size. 2. 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 ≥ 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).

Table 3.3.5a 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

Table 3.3.5b 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.

Note: ***This is the original CFI Primary Vendor-Specific Extended Query Table definition agreed to by AMD and Fujitsu and was implemented in the initial CFI device (Am29LV160B).***

CFI Primary Vendor-Specific Extended Query Table
Major.Minor Version = 1.0
Table 4.0 **(x8/x16 Device in x8-Mode Used as an Example)**

Addresses (Word Mode)	Addresses (Byte Mode)	Data	Description
(P)h	(P*2)h	50h 52h 49h	Query-unique ASCII string "PRI" (3 bytes)
(P+3)h	((P*2)+6)h	31h	CFI Table - Major version number, ASCII (1 byte)
(P+4)h	((P*2)+8)h	30h	CFI Table - Minor version number, ASCII (1 byte)
(P+5)h	((P*2)+A)h	00h	Address Sensitive Unlock 0 = Required, 1 = Not Required
(P+6)h	((P*2)+C)h	02h	Erase Suspend (1 byte) 00 = Not Supported, 01 = To Read Only, 02 = To Read and Write
(P+7)h	((P*2)+E)h	01h	Sector Protect (1 byte) 00 = Not Supported, X = Number of sectors per group
(P+8)h	((P*2)+10)h	01h	Temporary Sector Unprotect (1 byte) 00 = Not Supported, 01 = Supported
(P+9)h	((P*2)+12)h	04h	Sector Protect/Unprotect scheme (1 byte) 01 = 29F040 mode, 02 = 29F016 mode, 03 = 29F400 mode, 04 = 29LV800A mode *[Refer to Appendix]*
(P+A)h	((P*2)+14)h	00h	Simultaneous Operation (1 byte) 00 = Not Supported, 01 = Supported
(P+B)h	((P*2)+16)h	00h	Burst Mode Type (1 byte) 00 = Not Supported, 01 = Supported
(P+C)h	((P*2)+18)h	00h	Page Mode Type (1 byte) 00 = Not Supported, 01 = 4 Word Page, 02 = 8 Word Page

Note: ***This example of the CFI Primary Vendor-Specific Extended Query Table definition was taken from Am29LV320D.***

CFI Primary Vendor-Specific Extended Query Table
Major.Minor Version = 1.1
Table 4.1 (x8/x16 Device in x8-Mode Used as an Example)

Addresses (Word Mode)	Addresses (Byte Mode)	Data	Description
(P)h	(P*2)h	50h 52h 49h	Query-unique ASCII string "PRI" (3 bytes)
(P+3)h	((P*2)+6)h	31h	Major version number, ASCII (1 byte)
(P+4)h	((P*2)+8)h	31h	Minor version number, ASCII (1 byte)
(P+5)h	((P*2)+A)h	00h	Address Sensitive Unlock (DQ1, DQ0) 00 = Required, 01 = Not Required Process Technology (DQ7 – DQ2)
(P+6)h	((P*2)+C)h	02h	Erase Suspend (1 byte) 00 = Not Supported, 01 = To Read Only, 02 = To Read and Write
(P+7)h	((P*2)+E)h	04h	Sector Protect (1 byte) 00 = Not Supported, X = Number of sectors per group
(P+8)h	((P*2)+10)h	01h	Temporary Sector Unprotect (1 byte) 00 = Not Supported, 01 = Supported
(P+9)h	((P*2)+12)h	04h	Sector Protect/Unprotect scheme (1 byte) 01 = 29F040 mode, 02 = 29F016 mode, 03 = 29F400 mode, 04 = 29LV800A mode *[Refer to Appendix]*
(P+A)h	((P*2)+14)h	00h	Simultaneous Operation (1 byte) 00 = Not Supported, X = Number of sectors in bank 2 (uniform sector bank)
(P+B)h	((P*2)+16)h	00h	Burst Mode Type (1 byte) 00 = Not Supported, 01 = Supported
(P+C)h	((P*2)+18)h	00h	Page Mode Type (1 byte) 00 = Not Supported, 01 = 4 Word Page, 02 = 8 Word Page
(P+D)h	((P*2)+1A)h	B5h	ACC [Acceleration] Supply Minimum (1 byte) Bits 7 – 4 = HEX Value in Volts Bits 3 – 0 = BCD Value in 100 Millivolts
(P+E)h	((P*2)+1C)h	C5h	ACC [Acceleration] Supply Maximum (1 byte) Bits 7 – 4 = HEX Value in Volts Bits 3 – 0 = BCD Value in 100 Millivolts
(P+F)h	((P*2)+1E)h	0Xh	Top/Bottom Boot Sector Flag (1 byte) 02 = Bottom boot device, 03 = Top boot device, 04 = Uniform / Bottom WP Protect, 05 = Uniform, Top WP Protect NOTE: Because early version CFI devices (prior to table revision 1.1) did not support this field: If number of erase block regions > 1: If this field is 00h: refer to device ID code for Top/Bottom boot version of Am29LV160 or Am29LV116

Note: ***This example of the CFI Primary Vendor-Specific Extended Query Table definition was taken from Am29DL322D.***

CFI Primary Vendor-Specific Extended Query Table
Major.Minor Version = 1.2
Table 4.2 **(x8/x16 Device in x8-Mode Used as an Example)**

Addresses (Word Mode)	Addresses (Byte Mode)	Data	Description
(P)h	(P*2)h	50h 52h 49h	Query-unique ASCII string "PRI" (3 bytes)
(P+3)h	((P*2)+6)h	31h	Major version number, ASCII (1 byte)
(P+4)h	((P*2)+8)h	32h	Minor version number, ASCII (1 byte)
(P+5)h	((P*2)+A)h	00h	Address Sensitive Unlock (DQ1, DQ0) 00 = Required, 01 = Not Required Process Technology (DQ7 – DQ2)
(P+6)h	((P*2)+C)h	02h	Erase Suspend (1 byte) 00 = Not Supported, 01 = To Read Only, 02 = To Read and Write
(P+7)h	((P*2)+E)h	01h	Sector Protect (1 byte) 00 = Not Supported, X = Number of sectors per group
(P+8)h	((P*2)+10)h	01h	Temporary Sector Unprotect (1 byte) 00 = Not Supported, 01 = Supported
(P+9)h	((P*2)+12)h	04h	Sector Protect/Unprotect scheme (1 byte) 01 = 29F040 mode, 02 = 29F016 mode, 03 = 29F400 mode, 04 = 29LV800A mode *[Refer to Appendix]*
(P+A)h	((P*2)+14)h	XXh	Simultaneous Operation (1 byte) 00 = Not Supported, XX = Number of Sectors in bank 2 (uniform sector bank)
(P+B)h	((P*2)+16)h	00h	Burst Mode Type (1 byte) 00 = Not Supported, 01 = Supported
(P+C)h	((P*2)+18)h	00h	Page Mode Type (1 byte) 00 = Not Supported, 01 = 4 Word Page, 02 = 8 Word Page
(P+D)h	((P*2)+1A)h	85h	ACC [Acceleration] Supply Minimum (1 byte) Bits 7 – 4 = HEX Value in Volts Bits 3 – 0 = BCD Value in 100 Millivolts
(P+E)h	((P*2)+1C)h	95h	ACC [Acceleration] Supply Maximum (1 byte) Bits 7 – 4 = HEX Value in Volts Bits 3 – 0 = BCD Value in 100 Millivolts
(P+F)h	((P*2)+1E)h	0Xh	Top/Bottom Boot Sector Flag (1 byte) 00 = Device without WP control 01 = 8x8kb Sectors at top and bottom with WP control 02 = Bottom boot device, 03 = Top boot device, 04 = Uniform / Bottom WP Protect, 05 = Uniform, Top WP Protect If number of erase block regions = 1: Ignore this field
(P+10)h	((P*2)+20)h	00h	Program Suspend 00 = Not Supported 01 = Supported

Note: ***This example of the CFI Primary Vendor-Specific Extended Query Table definition was taken from Am29LV640MU.***

CFI Primary Vendor-Specific Extended Query Table
Major.Minor Version = 1.3
Table 4.3 (x8/x16 Device in x8-Mode Used as an Example)

Addresses (Word Mode)	Addresses (Byte Mode)	Data	Description
(P)h	(P*2)h	50h 52h 49h	Query-unique ASCII string “PRI” (3 bytes)
(P+3)h	((P*2)+6)h	31h	Major version number, ASCII (1 byte)
(P+4)h	((P*2)+8)h	33h	Minor version number, ASCII (1 byte)
(P+5)h	((P*2)+A)h	08h	Address Sensitive Unlock (DQ1, DQ0) 00 = Required, 01 = Not Required Process Technology (DQ5 – DQ2) 0000 = CS49 0001 = CS59 0010 = CS99
(P+6)h	((P*2)+C)h	02h	Erase Suspend (1 byte) 00 = Not Supported, 01 = To Read Only, 02 = To Read and Write
(P+7)h	((P*2)+E)h	04h	Sector Protect (1 byte) 00 = Not Supported, X = Number of sectors per group
(P+8)h	((P*2)+10)h	01h	Temporary Sector Unprotect (1 byte) 00 = Not Supported, 01 = Supported
(P+9)h	((P*2)+12)h	04h	Sector Protect/Unprotect scheme (1 byte) *[Refer to Appendix]* 01 = 29F040 mode, 02 = 29F016 mode, 03 = 29F400 mode, 04 = 29LV800A mode 05 = 29BDS640 mode (Software Command Locking) 06 = 29BDD160 mode (New Sector Protect) 07 = 29PDL128 mode = (New Sector Protect) + 29LV800A mode
(P+A)h	((P*2)+14)h	00h	Simultaneous Operation (1 byte) 00 = Not Supported, XX = Total number of sectors in all banks except Boot Bank
(P+B)h	((P*2)+16)h	00h	Burst Mode Type (1 byte) 00 = Not Supported, 01 = Supported
(P+C)h	((P*2)+18)h	01h	Page Mode Type (1 byte) 00 = Not Supported, 01 = 4 Word Page, 02 = 8 Word Page
(P+D)h	((P*2)+1A)h	B5h	ACC [Acceleration] Supply Minimum (1 byte) Bits 7 – 4 = HEX Value in Volts Bits 3 – 0 = BCD Value in 100 Millivolts
(P+E)h	((P*2)+1C)h	C5h	ACC [Acceleration] Supply Maximum (1 byte) Bits 7 – 4 = HEX Value in Volts Bits 3 – 0 = BCD Value in 100 Millivolts
(P+F)h	((P*2)+1E)h	0Xh	Top/Bottom Boot Sector Flag (1 byte) 00 = Device without WP control 01 = 8x8kb Sectors at top and bottom with WP control 02 = Bottom boot device 03 = Top boot device 04 = Uniform, Bottom WP Protect 05 = Uniform, Top WP Protect If number of erase block regions = 1: Ignore this field

(P+10)h	((P*2)+20)h	01h	Program Suspend 00 = Not Supported 01 = Supported
(P+11)h	((P*2)+22)h	00h	RESERVED FOR FUTURE USE
(P+12)h	((P*2)+24)h	00h	RESERVED FOR FUTURE USE
(P+13)h	((P*2)+26)h	00h	RESERVED FOR FUTURE USE
(P+14)h	((P*2)+28)h	00h	RESERVED FOR FUTURE USE
(P+15)h	((P*2)+3A)h	00h	RESERVED FOR FUTURE USE
(P+16)h	((P*2)+3C)h	00h	RESERVED FOR FUTURE USE
(P+17)h	((P*2)+3E)h	00h	Bank Organization (1 byte) 00 = If data at 4Ah is zero XX = Number of banks
(P+18)h	((P*2)+40)h	00h	Bank 1 Region Information (1 byte) = Number of sectors in Bank 1
(P+19)h	((P*2)+42)h	00h	Bank 2 Region Information (1 byte) = Number of sectors in Bank 2
(P+1A)h	((P*2)+44)h	00h	Bank 3 Region Information (1 byte) = Number of sectors in Bank 3
(P+1B)h	((P*2)+46)h	00h	Bank 4 Region Information (1 byte) = Number of sectors in Bank 4
(P+1C)h	((P*2)+48)h	00h	RESERVED FOR FUTURE USE
(P+1D)h	((P*2)+4A)h	00h	RESERVED FOR FUTURE USE
(P+1E)h	((P*2)+4C)h	00h	RESERVED FOR FUTURE USE
(P+1F)h	((P*2)+4E)h	00h	RESERVED FOR FUTURE USE

APPENDIX

Device Interface

Note: April 2000 – x16/x32 devices will be represented by hex value 0005h as requested by Intel in order to make them more software friendly. Changes will be made to the CFI drivers so that a bit-wise switch is created to represent different data widths. The following is the actual bit-wise switch:

Bit 0 – represents x8
Bit 1 – represents x16
Bit 2 – represents x32 … And so on

These bits are created by the CFI drivers and not by what is read out of location 28h. If a device is a x8/x16 device the resulting bit-wise pattern to be read out of the CFI is 0010b, since 0001b is added to the bit values by the CFI drivers to create 0011b. In this case:

Bit 0 – 1 (TRUE) because it supports x8
Bit 1 – 1 (TRUE) because it supports x16
Bit 2 – 0 (FALSE) because it does not support x32

For example, if we take the description for an x16/x32 device (0005h) and we convert that to binary we get 0101b. If we add one to this value we get a bit pattern that looks like this: 0110b. This bit-wise switch indicates that the device a x16/x32 device.

Bit 0 – 1 (FALSE) because it does not supports x8
Bit 1 – 1 (TRUE) because it supports x16
Bit 2 – 0 (TRUE) because it supports x32

Sector Protect/Unprotect Schemes

August 2001 – The Software Command Locking and New Sector Protection schemes were added to newer parts.

29F040 mode

External Programmer

- Programmer Protect >> apply voltage VID (12V) on pins A9 and OE#, VIL on CE#
- 98403 and 98406 Die
  Programmer Unprotect >> apply voltages VID on OE#, VIH on A5, VIL on A9
- 98401 Die
  Programmer Unprotect >> apply voltages VID on OE#, CE#, and A9, VIH on A12 and A6
- Addresses A18:A16 are used to identify one of the eight sectors

29F016 mode

External Programmer

- Programmer Protect >> apply voltage VID (12V) on pins A9 and OE#, VIH on A1 and RESET#, and VIL on CE#, A6, and A0
- Programmer Unprotect >> apply voltages VID on A9 and OE#, VIH on RESET#, A6, and A1, VIL on CE# and A0
- Addresses A20:A18 are used to identify one of the eight sector groups

29F400 mode

Am29F400 implements only the "Temporary Sector Unprotect" device operation

Device Bus Operation

- Temporary Sector Unprotect >> apply voltage VID on the RESET# pin

External Programmer

- Programmer Protect >> apply voltage VID on pins A9 and OE#

29LV800 mode

Am29LV800 implements the "Sector Protect" and "Sector Unprotect" device operations, in addition to Am29F400.

29BDS640 mode (Software Command Locking)

Am29BDS640 implements the Software Command Locking scheme. The Software Command Locking scheme implements VID functionality at the Vcc-level. This scheme uses the Sector Lock/Unlock command definition to lock and unlock sectors through software without having to use the high voltage method apparent in older devices.

29BDD160 mode (New Sector Protect)

Am29BDD160 implements the New Sector Protect scheme. The New Sector Protect scheme involves two parts: the "Persistent Sector Protection" and "Password Sector Protection" methods. The Persistent Sector Protection method allows for persistent or dynamic protection of individual sectors or sector groups. The Password Sector Protection method acts as a One-Time Programmable scheme where a password has to be given for any modification of protection status of the sectors or sector groups.

29PDL128 mode (New Sector Protect + 29LV800)

Am29PDL128 implements the New Sector Protect scheme with the security features from 29LV800 incorporated also.