AN1163
APPLICATION NOTE

Use of the Common Flash Memory Interface (CFI)

CONTENTS

INTRODUCTION

This application note describes the Common Flash Memory Interface (CFI) as implemented by STMicroelectronics Flash memories.

CFI is a published industry-wide protocol that enables compatibility between Flash memories. It allows designers to exercise some flexibility in their work, and end users (consumers) to take advantage of opportunities to upgrade equipment memory without the need to search for a specific product family or vendor.

The structure of CFI allows system software to interrogate the memory so that it can configure itself optimally for the reported memory size, speed and characteristics.

WHY USE CFI?

CFI provides the major benefits of simple upgrade capability and wide second-source availability. Both these factors are of interest to an OEM (Original Equipment Manufacturer) when they buy the product. They are of even greater importance to the end user who purchases a product. A product that uses CFI-compliant Flash will allow the end user to add and upgrade memory without the inconvenience of searching for a particular vendor or product family, making it substantially 'future proof'.

Upgrade Capability

Memory devices are the subject of rapid and continuing improvements in densities and performance. Products could very quickly become outdated and redundant unless a simple method were available to introduce memory upgrades as they became available.

It is common practice for hardware designers to ensure footprint and pin-for-pin compatibility in their products, and to design the board layout so that it allows sufficient flexibility to support future upgrade intentions.

In many cases, the principal difference between one product and another exists in the design of its proprietary system software and this is an area where it is less common for similar compatibility to exist in support of upgrade intentions. CFI compliant devices eliminate this as a potential source of problems by allowing newer and improved devices to be substituted for their older versions without any need to modify or update the system software.

Instead, the system software can interrogate the CFI compliant Flash and can configure itself automatically to use the features, timings and other characteristics of the device to take full advantage of any improvements in performance. For example, if your new device has a faster Block Erase cycle, the software will recognize this and optimize itself automatically.

CFI relieves system designers of the need to rework low-level software when they upgrade Flash memory. All they need do is to arrange for their software to accept CFI compliant Flash memory and then allow the software to upgrade itself.

Second Source Availability

Second source availability is a very important consideration. The convenience of a system that can use second source components is very important for the end user who does not want to source the Memory exclusively from one particular supplier or exclusively from one family of Flash memories.

In an ideal situation, the end user could confidently purchase any Flash memory product without consideration of their manufacturing source. Ideally Flash memory would following the practice exemplified by the design of camera film and computer floppy disks. CFI compliance is the mechanism that makes this possible.

Software reads the memory to ascertain the device manufacturer, the size, speed and the configuration of the memory, and any special command codes necessary. The software then uses this information to create the optimal interface environment between the system and the card. To gain full benefit from using CFI compliant memories, the system software must be able to adapt to the size and timing characteristics of the memory, and to use an vendor-specific command codes needed by the Flash.

ACCESS TO CFI: QUERY MODE

Not all memories inserted into a Flash memory socket will be CFI enabled, so the system software must determine whether the memory is CFI compliant.

For a CFI compliant memory it is possible to select the Query mode using a single command write cycle and switch back again to read array mode using a further single command write cycle. To enter the Query Mode of the CFI the software must write 98h to address 55h in the Flash memory's address space. Many Flash memories ignore the address and enter Query mode directly when 98h is written on the data bus.

Once in the CFI Query mode all of the CFI information can be read until a write cycle is inserted. From Query mode the memory should respond with the query data string (ASCII equivalent "Q", "R", "Y" starting at address 10h in the Flash memory address space). Memories that do no respond with the query data string are not CFI enabled.

Each product Data Sheet contains all of the information that can be found for that particular product.

CFI DATA STRUCTURE

From the CFI mode you can access the following information:

a. From Address 00h to 09h: Vendor Specific Information (refer to the memory's Data Sheet for further information). See Table 1.

b. From Address 10h to 1Ah: System Interface Information. See Table 2.

c. From Address 1Bh to 27h: Device Geometry Information. See Table 3.

Table 1. Vendor Specific Information

Address Offset	Length (bytes)	Description
10h	03h	Query Response string (ASCII “QRY”)
13h	02h	Primary vendor Command Set and Control Interface ID Code. A 16-bit code that defines a specific vendor-specified algorithm.
15h	02h	Address for the Primary Algorithm extended Query Table Note - Address location 0000h means that there is no extended Query Table
17h	02h	Alternative vendor Command Set and Control Interface ID Code The second vendor-specific algorithm supported by the device Note - ID code 0000h means that the device uses no alternative algorithm
19h	02h	Address for Alternative Algorithm extended Query table Note - Address 0000h means that no alternative extended table exists

Table 2. System Interface Information

Address Offset	Length (bytes)	Description
1Bh	01h	V_{CC} Logic Supply - Minimum Write/Erase voltage: bits 7 - 4 = BCD value in volts bits 3 - 0 = BCD value in 100mV
1Ch	01h	V_{CC} Logic Supply - Maximum Write/Erase voltage: bits 7 - 4 =BCD value in volts bits 3 - 0 = BCD value in 100mV
1Dh	01h	V_{PP} Programming Supply - Minimum Write/Erase voltage: bits 7 - 4 = HEX value in volts bits 3 - 0 = BCD value in 100mV This value must be 00h if the device has no Vpp pin.
1Eh	01h	V_{PP} Programming Supply - Maximum Write/Erase voltage: bits 7 - 4 = HEX value in volts bits 3 - 0 = BCD value in 100mV This value must be 00h if the device has no Vpp pin.
1Fh	01h	Typical timeout per single byte/word (buffer write count = 1), 2^N µs (if supported, otherwise 00h)
20h	01h	Typical timeout for maximum size buffer write, 2^N µs (if supported, otherwise 00h)
21h	01h	Typical timeout per individual block erase, 2^N ms (if supported, otherwise 00h)
22h	01h	Typical timeout for full chip erase, 2^N ms (if supported, otherwise 00h)
23h	01h	Maximum timeout for byte/word write, 2^N times typical (offset 1Fh) (if supported, otherwise 00h)
24h	01h	Maximum timeout for buffer write, 2^N times typical (offset 20h) (if supported, otherwise 00h)
25h	01h	Maximum timeout per individual block erase, 2^N times typical (offset 21h) (if supported, otherwise 00h)
26h	01h	Maximum timeout for chip erase, 2^N times typical (offset 22h) (if supported, otherwise 00h)

Table 3. Device Geometry Information

Address Offset	Length (bytes)	Description
27h	01h	Device size 2^N in bytes
28h	02h	Flash Device Interface Code description.
2Ah	02h	Maximum number of bytes in a multi-byte write operation 2^N
2Ch	01h	Number of Erase Block regions within the device Note: This parameter specifies the number of regions within the device that contain one or more contiguous Erase Blocks of the same size. The parameter will be zero if the device does not support Block Erase, i.e. the device erases all regions simultaneously.
2Dh	04h	Erase Block Region information bits 31 - 16 = *a*, where the Erase Blocks in this region are *a* times 256 bytes in size. The value *a* = 0 is used for 128-byte block size. bits 15 - 0 = *b*, where *b*+1 = the number of Erase Blocks of identical size within the Erase Block region.
31h to (*z*-1)h	04h per entry	Additional Erase Block information, with 4 bytes per region. Note: The total number of Blocks times the individual Block size must equal the device size. The address *z* is the next available Query address at the end of the device geometry structure. It is the first possible starting address of the optional vendor-specific Query table.

CONCLUSIONS

The Common Flash Interface (CFI) can be used to make your product, and your manufacturing process, more flexible by allowing upgrade capability and second source availability.

If you have any questions or suggestion concerning the matters raised in this document please send them to the following electronic mail address:

ask.memory@st.com (for general enquiries)

Please remember to include your name, company, location, telephone number and fax number.