DMI 信息的读取

版本	日期	说明
0.2	05-3-24	最初版本。

一. 参考资料

<<System Management BIOS Reference Specification>> （强烈建议对照此 spec 阅读本文）

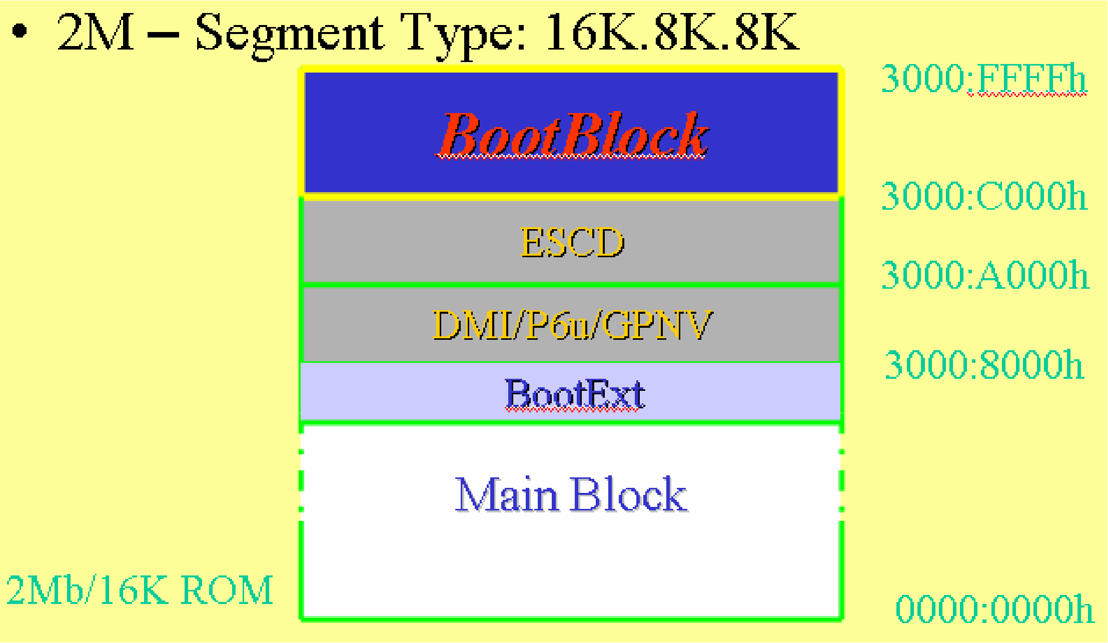
版本 Version 2.3 — 12 August 1998

使用的工具是 Debug 32 。

二. 什么是 DMI ？

个人理解就是一种定制出来的结构， 按照一定格式存放计算机中各种信息。这样，软件就可以很方便的读取这些信息。

这个信息通常存放在 BIOS 中，如图（ 1 ）



图（ 1 ）

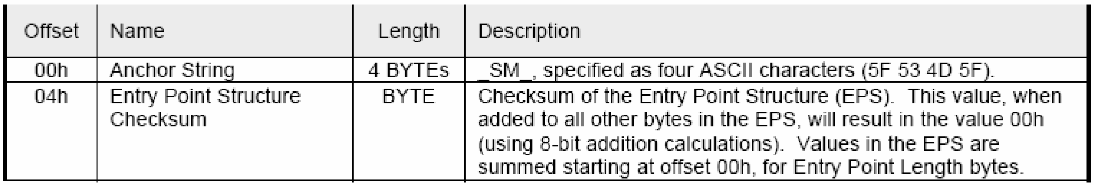
上面是 2M(单位是 bit)的 BIOS ROM 的基本格式。在启动的时候。BIOS 会将上面的 DMI 信息拷贝到内存中。使用各种方法读取的 DMI 信息实际上是在内存中。

三．DMI 的读取

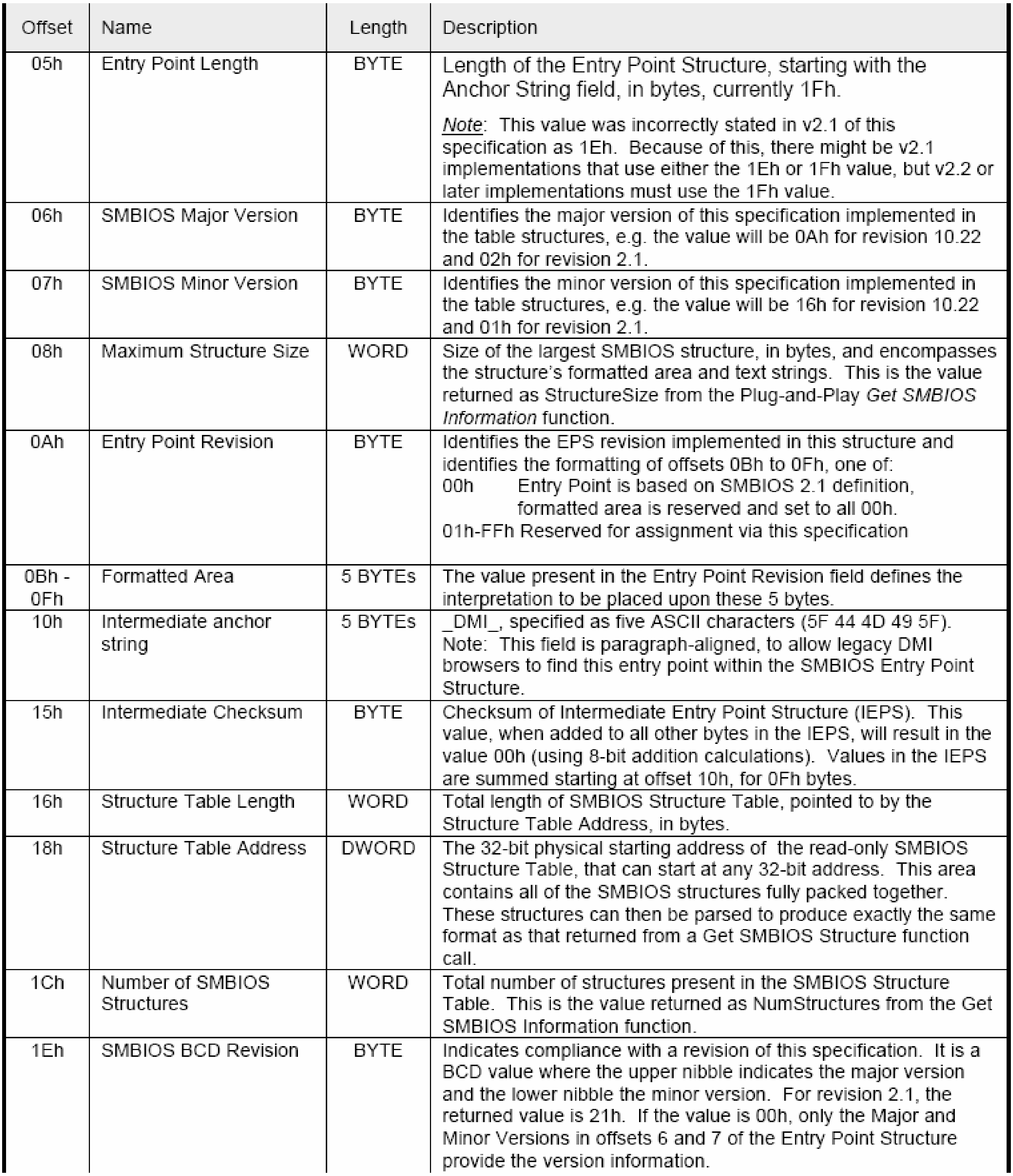
读取 DMI 信息有两种方法，一种是使用 SMBIOS 提供的中断；另外一种是在内存 F000 段搜索标志字符串。前者是 v2.0 规范及其之前版本定义的，后者是 v2.1 以及后继版本定义的。一般的电脑都支持这两种方法（至少要支持第一种方法）。

这篇文章只介绍使用第二种方法。

查阅规范，第 9 页：



Offset	Name	Length	Description
00h	Anchor String	4 BYTEs	_SM_, specified as four ASCII characters (5F 53 4D 5F).
04h	Entry Point Structure Checksum	BYTE	Checksum of the Entry Point Structure (EPS). This value, when added to all other bytes in the EPS, will result in the value 00h (using 8-bit addition calculations). Values in the EPS are summed starting at offset 00h, for Entry Point Length bytes.

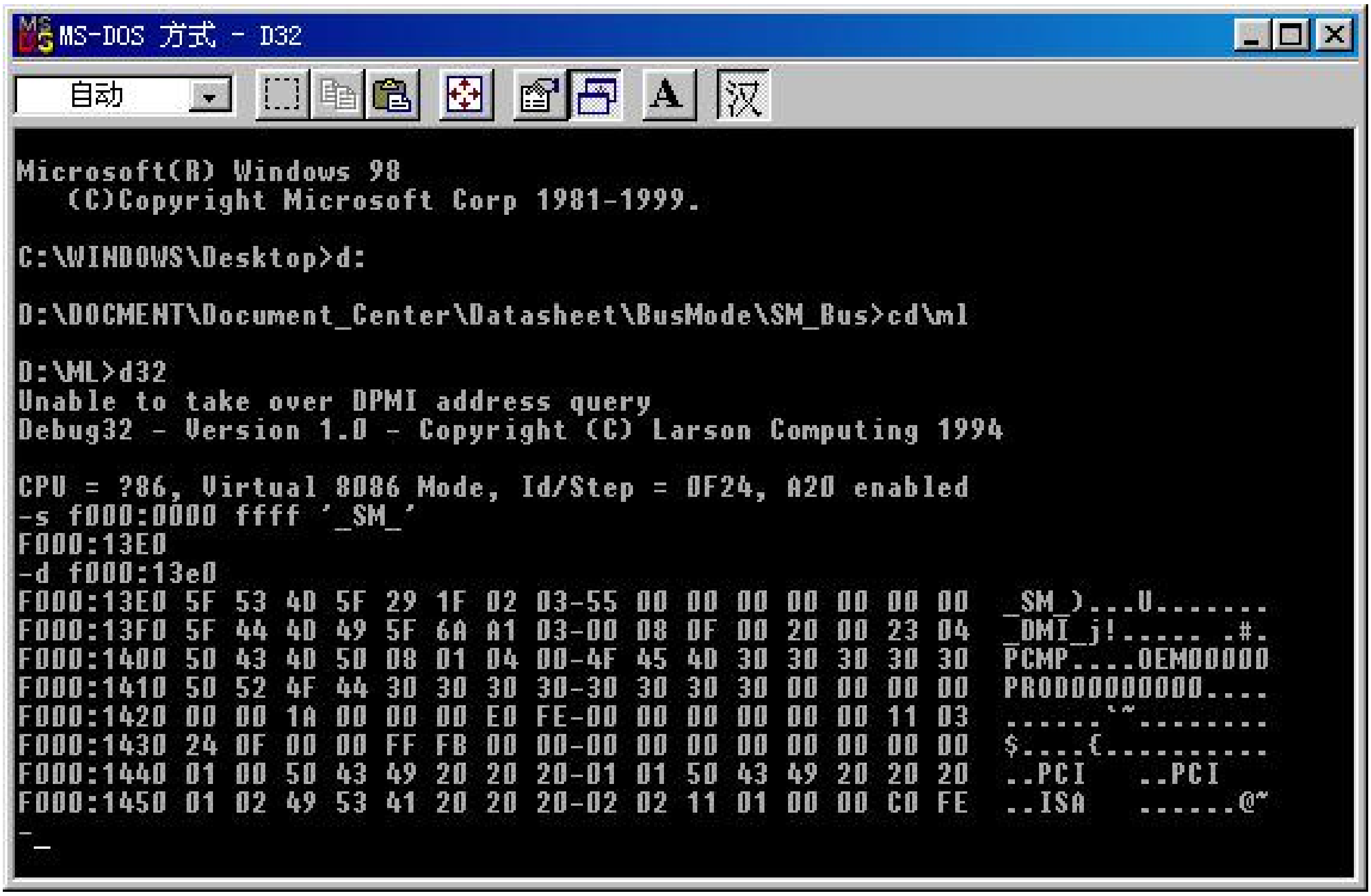


Offset	Name	Length	Description
05h	Entry Point Length	BYTE	Length of the Entry Point Structure, starting with the Anchor String field, in bytes, currently 1Fh. *Note*: This value was incorrectly stated in v2.1 of this specification as 1Eh. Because of this, there might be v2.1 implementations that use either the 1Eh or 1Fh value, but v2.2 or later implementations must use the 1Fh value.
06h	SMBIOS Major Version	BYTE	Identifies the major version of this specification implemented in the table structures, e.g. the value will be 0Ah for revision 10.22 and 02h for revision 2.1.
07h	SMBIOS Minor Version	BYTE	Identifies the minor version of this specification implemented in the table structures, e.g. the value will be 16h for revision 10.22 and 01h for revision 2.1.
08h	Maximum Structure Size	WORD	Size of the largest SMBIOS structure, in bytes, and encompasses the structure's formatted area and text strings. This is the value returned as StructureSize from the Plug-and-Play *Get SMBIOS Information* function.
0Ah	Entry Point Revision	BYTE	Identifies the EPS revision implemented in this structure and identifies the formatting of offsets 0Bh to 0Fh, one of: 00h Entry Point is based on SMBIOS 2.1 definition, formatted area is reserved and set to all 00h. 01h-FFh Reserved for assignment via this specification
0Bh - 0Fh	Formatted Area	5 BYTEs	The value present in the Entry Point Revision field defines the interpretation to be placed upon these 5 bytes.
10h	Intermediate anchor string	5 BYTEs	_DMI_, specified as five ASCII characters (5F 44 4D 49 5F). Note: This field is paragraph-aligned, to allow legacy DMI browsers to find this entry point within the SMBIOS Entry Point Structure.
15h	Intermediate Checksum	BYTE	Checksum of Intermediate Entry Point Structure (IEPS). This value, when added to all other bytes in the IEPS, will result in the value 00h (using 8-bit addition calculations). Values in the IEPS are summed starting at offset 10h, for 0Fh bytes.
16h	Structure Table Length	WORD	Total length of SMBIOS Structure Table, pointed to by the Structure Table Address, in bytes.
18h	Structure Table Address	DWORD	The 32-bit physical starting address of the read-only SMBIOS Structure Table, that can start at any 32-bit address. This area contains all of the SMBIOS structures fully packed together. These structures can then be parsed to produce exactly the same format as that returned from a Get SMBIOS Structure function call.
1Ch	Number of SMBIOS Structures	WORD	Total number of structures present in the SMBIOS Structure Table. This is the value returned as NumStructures from the Get SMBIOS Information function.
1Eh	SMBIOS BCD Revision	BYTE	Indicates compliance with a revision of this specification. It is a BCD value where the upper nibble indicates the major version and the lower nibble the minor version. For revision 2.1, the returned value is 21h. If the value is 00h, only the Major and Minor Versions in offsets 6 and 7 of the Entry Point Structure provide the version information.

这是 DMI 入口的格式，象一个链表的头节点一样，称作 SMBIOS Structure Table Entry Point,简称 EPS。具体含义如下：

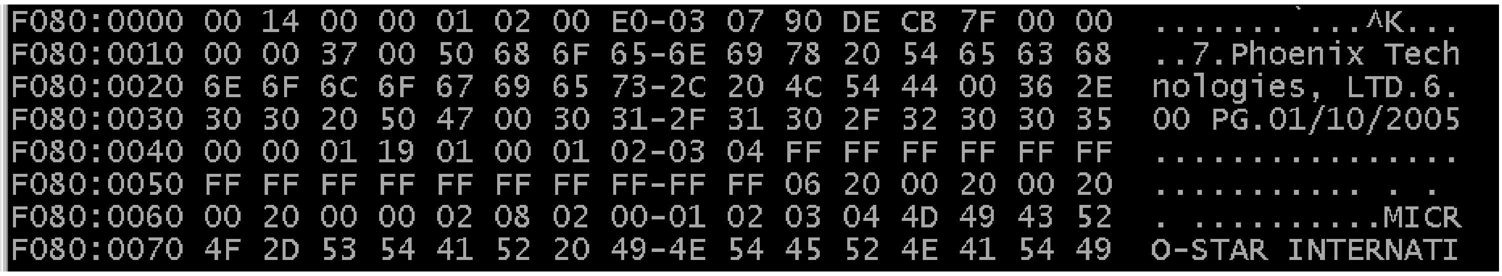
```
DB   '_SM_"              ;标志
DB    ?                  ;CheckSum,该值与其余 EPS 之和应该是 0
DB    01FH               ;EPSD 的长度,目前是 31 个字节
DB    ?                  ;主版本
DB    ?                  ;次版本
DW   ?                   ;最大的 SMBIOS Structure  长度
DB      0DH              ;EPS 版本
DB     dup（5） ?         ;这个是什么意思有什么用,我还不清楚
DB     '_DMI_'           ;中间的,'DMI'起始标志
DB      ?                ;中间的 CHECKSUM
DW      ?                ;以 byte 为单位 SMBIOS Structure Table  的总长度, ,起始位置
                          由下面的 Structure Table Address  指出.
DD      ?                ;4 字节长的只读的 SMBIOS Structure Table
DW     ?                  ;Structure  的数量
DB      ?                ;SMBIOS 本版,是 BCD 码的
```

找到了头,就能找到进入的位置.下面的例子是我的电脑.



```
Microsoft(R) Windows 98
   (C)Copyright Microsoft Corp 1981-1999.

C:\WINDOWS\Desktop>d:

D:\DOCMENT\Document_Center\Datasheet\BusMode\SM_Bus>cd\ml

D:\ML>d32
Unable to take over DPMI address query
Debug32 - Version 1.0 - Copyright (C) Larson Computing 1994

CPU = ?86, Virtual 8086 Mode, Id/Step = 0F24, A20 enabled
-s f000:0000 ffff '_SM_'
F000:13E0
-d f000:13e0
F000:13E0 5F 53 4D 5F 29 1F 02 03-55 00 00 00 00 00 00 00  _SM_)...U.......
F000:13F0 5F 44 4D 49 5F 6A A1 03-00 08 0F 00 20 00 23 04  _DMI_j!..... .#.
F000:1400 50 43 4D 50 08 01 04 00-4F 45 4D 30 30 30 30 30  PCMP....OEM00000
F000:1410 50 52 4F 44 30 30 30 30-30 30 30 30 00 00 00 00  PROD00000000....
F000:1420 00 00 1A 00 00 00 E0 FE-00 00 00 00 00 00 11 03  ......`~........
F000:1430 24 0F 00 00 FF FB 00 00-00 00 00 00 00 00 00 00  $....{..........
F000:1440 01 00 50 43 49 20 20 20-01 01 50 43 49 20 20 20  ..PCI   ..PCI
F000:1450 01 02 49 53 41 20 20 20-02 02 11 01 00 00 C0 FE  ..ISA   ......@~
-
 _
```

从 f000:0 的内存搜索标志,按照上面阅读,都可以解释完整.入口在内存 0F0800H 处.

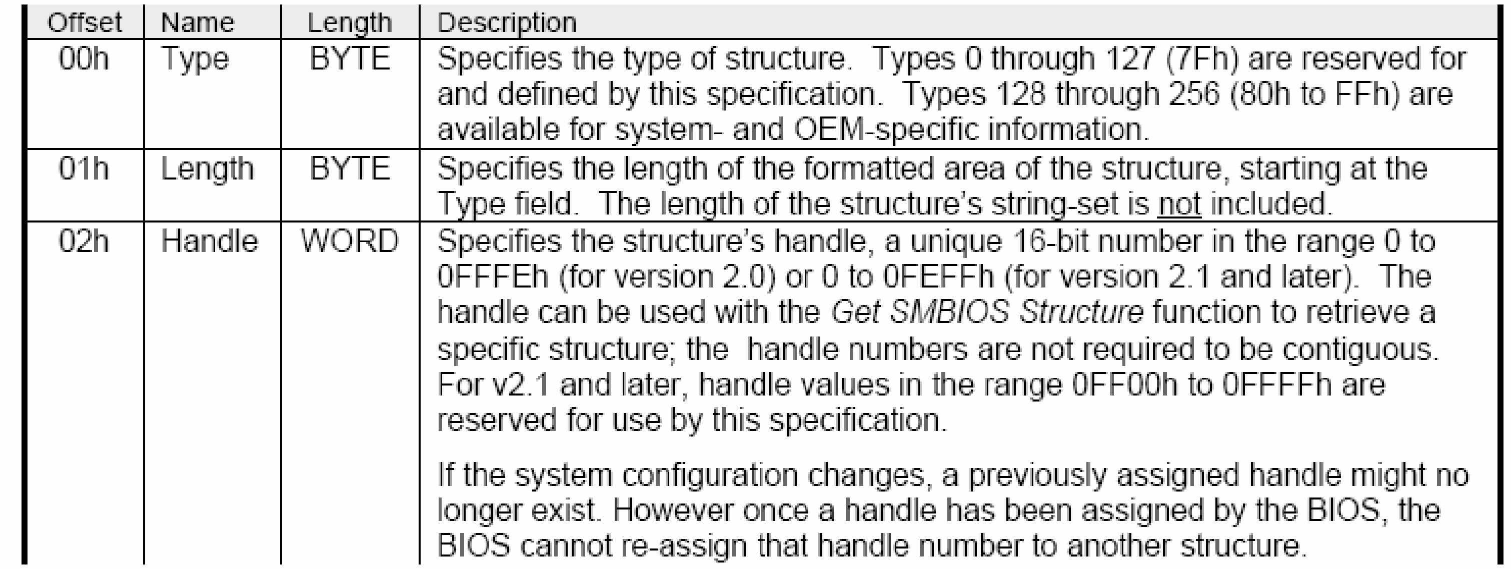


```
F080:0000 00 14 00 00 01 02 00 E0-03 07 90 DE CB 7F 00 00  .......`...^K...
F080:0010 00 00 37 00 50 68 6F 65-6E 69 78 20 54 65 63 68  ..7.Phoenix Tech
F080:0020 6E 6F 6C 6F 67 69 65 73-2C 20 4C 54 44 00 36 2E  nologies, LTD.6.
F080:0030 30 30 20 50 47 00 30 31-2F 31 30 2F 32 30 30 35  00 PG.01/10/2005
F080:0040 00 00 01 19 01 00 01 02-03 04 FF FF FF FF FF FF  ................
F080:0050 FF FF FF FF FF FF FF FF-FF FF 06 20 00 20 00 20  ........... . .
F080:0060 00 20 00 00 02 08 02 00-01 02 03 04 4D 49 43 52  . ..........MICR
F080:0070 4F 2D 53 54 41 52 20 49-4E 54 45 52 4E 41 54 49  O-STAR INTERNATI
```

这个结构在 spec 26 页有介绍

下面的问题就是每一个 structure 的读取了.

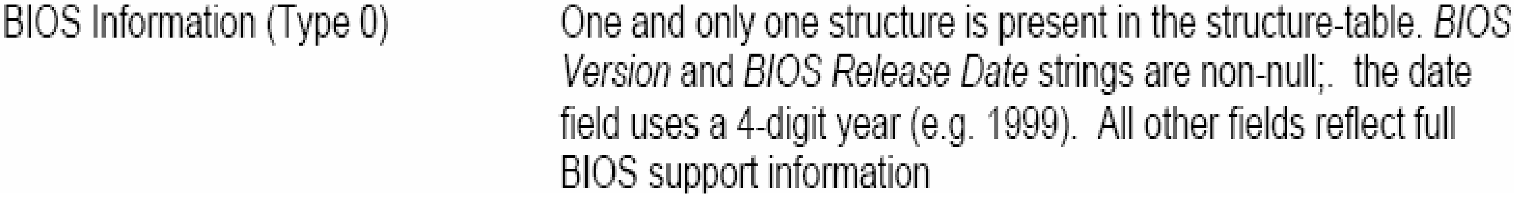
参考 spec



Offset	Name	Length	Description
00h	Type	BYTE	Specifies the type of structure. Types 0 through 127 (7Fh) are reserved for and defined by this specification. Types 128 through 256 (80h to FFh) are available for system- and OEM-specific information.
01h	Length	BYTE	Specifies the length of the formatted area of the structure, starting at the Type field. The length of the structure's string-set is not included.
02h	Handle	WORD	Specifies the structure's handle, a unique 16-bit number in the range 0 to 0FFFEh (for version 2.0) or 0 to 0FEFFh (for version 2.1 and later). The handle can be used with the *Get SMBIOS Structure* function to retrieve a specific structure; the handle numbers are not required to be contiguous. For v2.1 and later, handle values in the range 0FF00h to 0FFFFh are reserved for use by this specification. If the system configuration changes, a previously assigned handle might no longer exist. However once a handle has been assigned by the BIOS, the BIOS cannot re-assign that handle number to another structure.

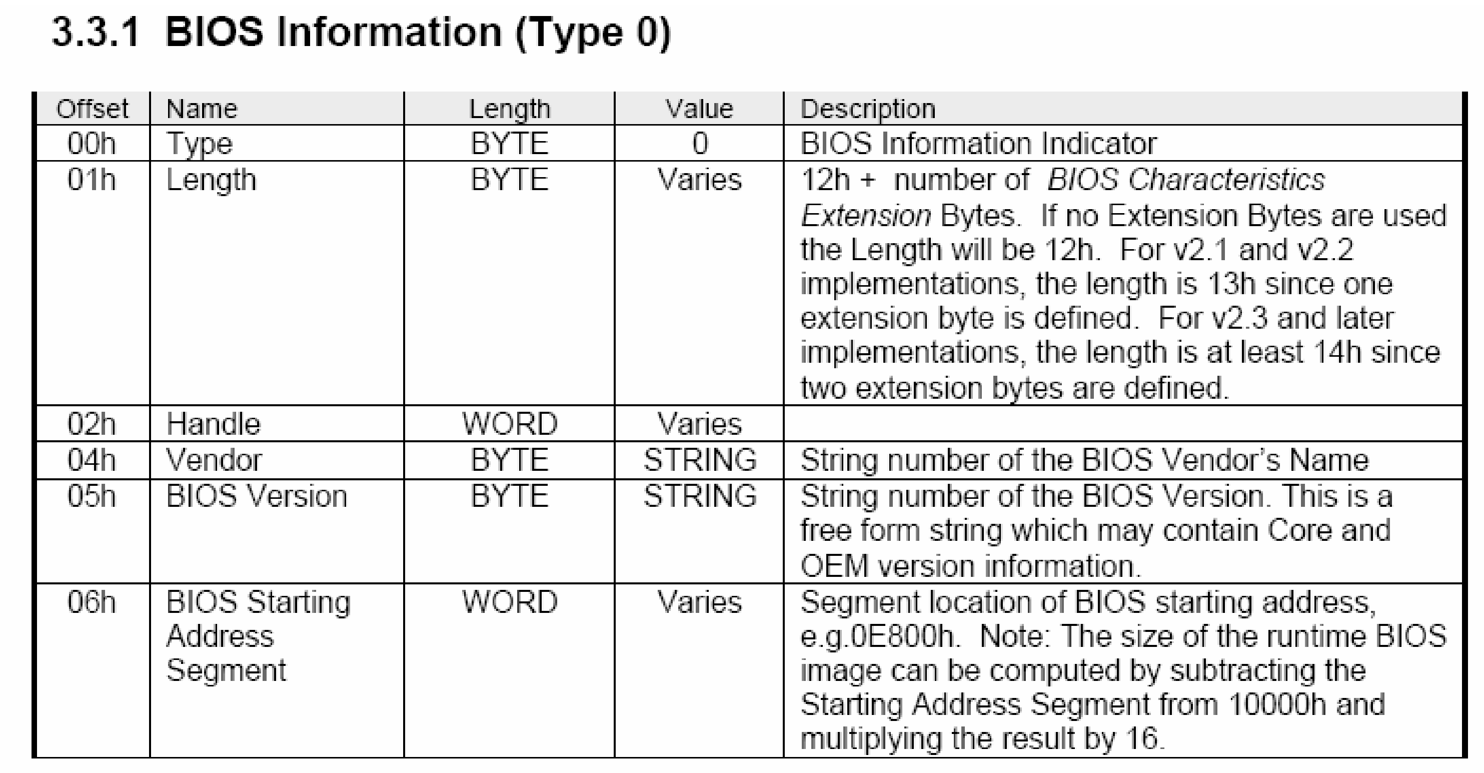
第一个 byte 是类型,表示这个 structure 代表什么信息(在 P27 3.2 Required Structures and Data 有描述);第二个是这个 structure 的长度(随着类型不同具体含义有些差别);第三个是一个标号(我自己的理解就是一个编号).

察看上图,第一个字节是 0,表示



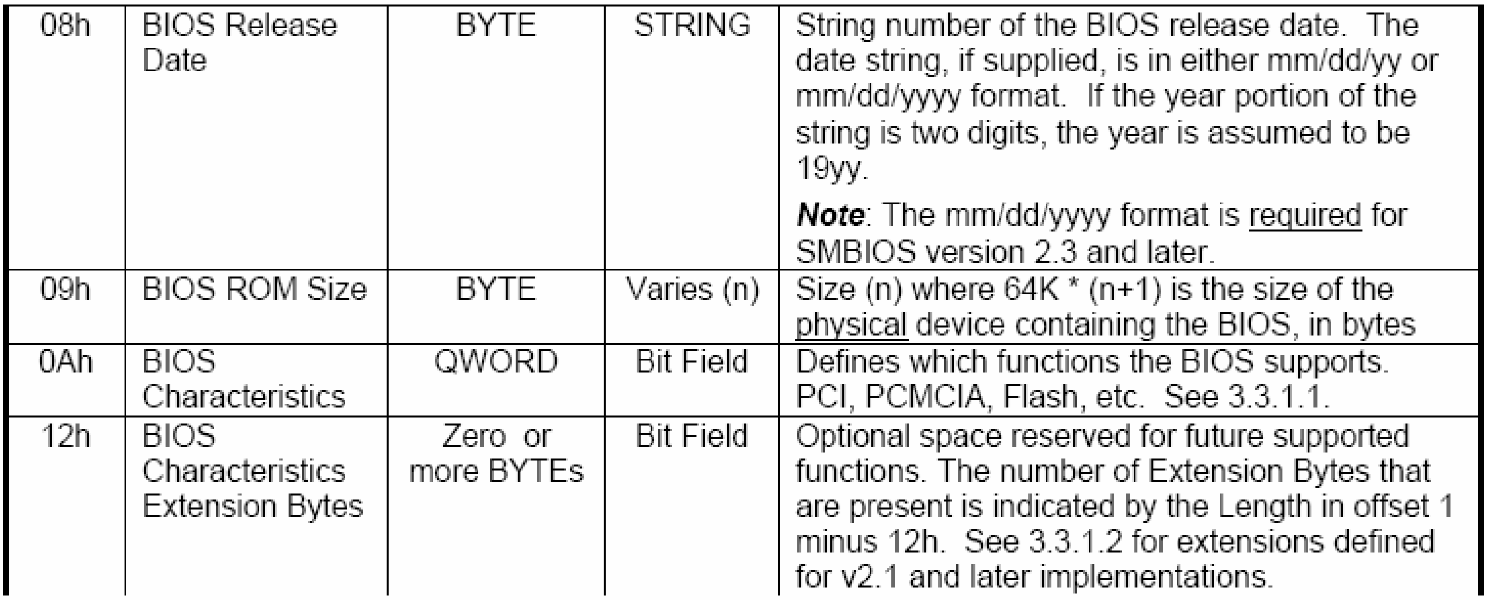
BIOS Information (Type 0)	One and only one structure is present in the structure-table. *BIOS Version* and *BIOS Release Date* strings are non-null;. the date field uses a 4-digit year (e.g. 1999). All other fields reflect full BIOS support information

第二个字节是 14h. 对照



3.3.1 BIOS Information (Type 0)

Offset	Name	Length	Value	Description
00h	Type	BYTE	0	BIOS Information Indicator
01h	Length	BYTE	Varies	12h + number of *BIOS Characteristics Extension* Bytes. If no Extension Bytes are used the Length will be 12h. For v2.1 and v2.2 implementations, the length is 13h since one extension byte is defined. For v2.3 and later implementations, the length is at least 14h since two extension bytes are defined.
02h	Handle	WORD	Varies	
04h	Vendor	BYTE	STRING	String number of the BIOS Vendor's Name
05h	BIOS Version	BYTE	STRING	String number of the BIOS Version. This is a free form string which may contain Core and OEM version information.
06h	BIOS Starting Address Segment	WORD	Varies	Segment location of BIOS starting address, e.g.0E800h. Note: The size of the runtime BIOS image can be computed by subtracting the Starting Address Segment from 10000h and multiplying the result by 16.

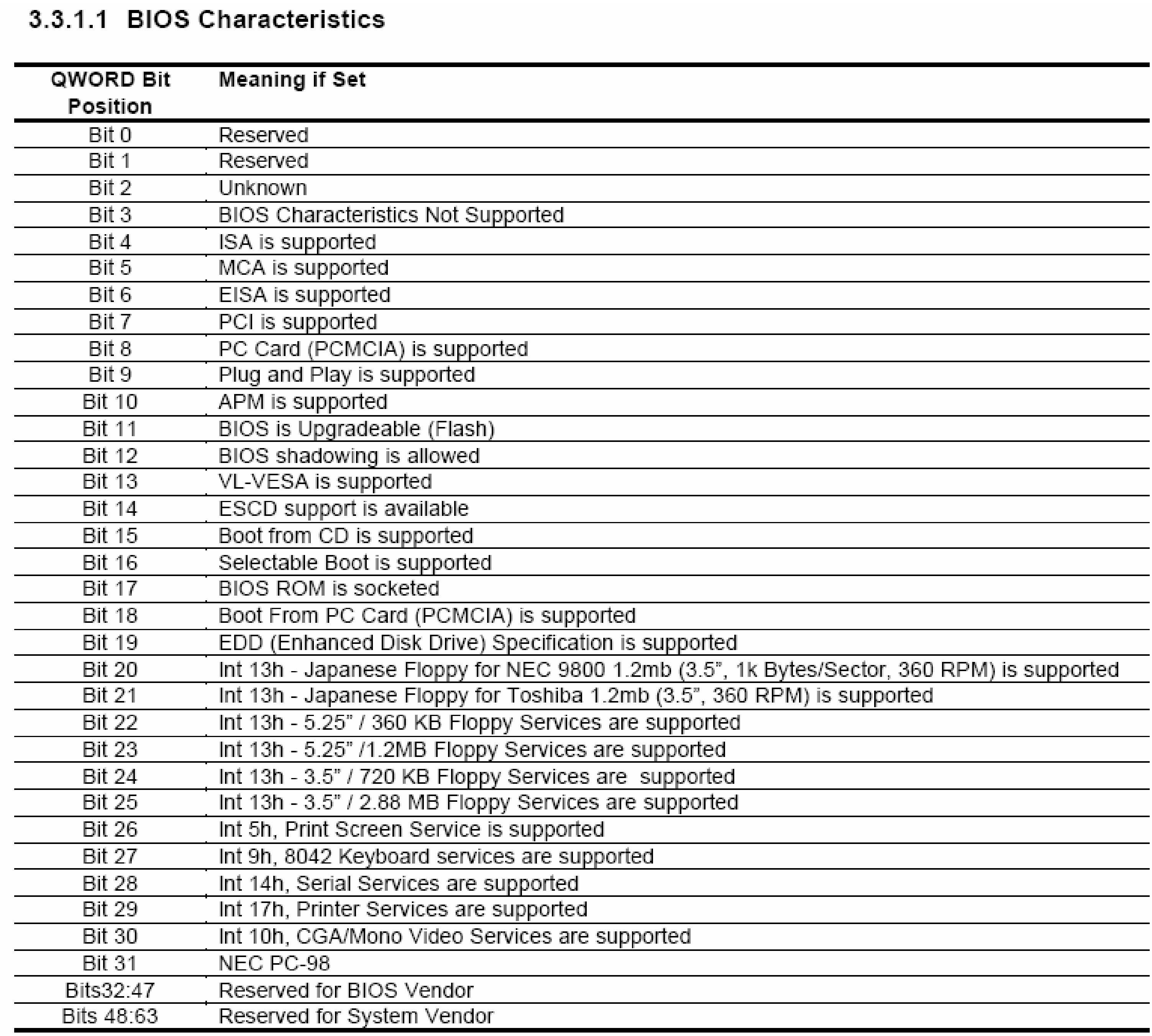


08h	BIOS Release Date	BYTE	STRING	String number of the BIOS release date. The date string, if supplied, is in either mm/dd/yy or mm/dd/yyyy format. If the year portion of the string is two digits, the year is assumed to be 19yy. ***Note***: The mm/dd/yyyy format is required for SMBIOS version 2.3 and later.
09h	BIOS ROM Size	BYTE	Varies (n)	Size (n) where 64K * (n+1) is the size of the physical device containing the BIOS, in bytes
0Ah	BIOS Characteristics	QWORD	Bit Field	Defines which functions the BIOS supports. PCI, PCMCIA, Flash, etc. See 3.3.1.1.
12h	BIOS Characteristics Extension Bytes	Zero or more BYTEs	Bit Field	Optional space reserved for future supported functions. The number of Extension Bytes that are present is indicated by the Length in offset 1 minus 12h. See 3.3.1.2 for extensions defined for v2.1 and later implementations.

14h 的含义应该是 12h+BIOS Characteristics 数量(2 个,2.3 最多也只支持 2 个扩展);handle 是 0000; 制造商是 01h string; BIOS 版本是 02h string;

BIOS 起始段地址是 E000; BIOS 编译日期是 03h string; BIOS 大小是 07h ,意思是这个 rom 是 64k*(7+1)=512KB;BIOS 特性 0000 0000 7FCB DE90;扩展特性为 0037h(长度应该是 14h 指出的).

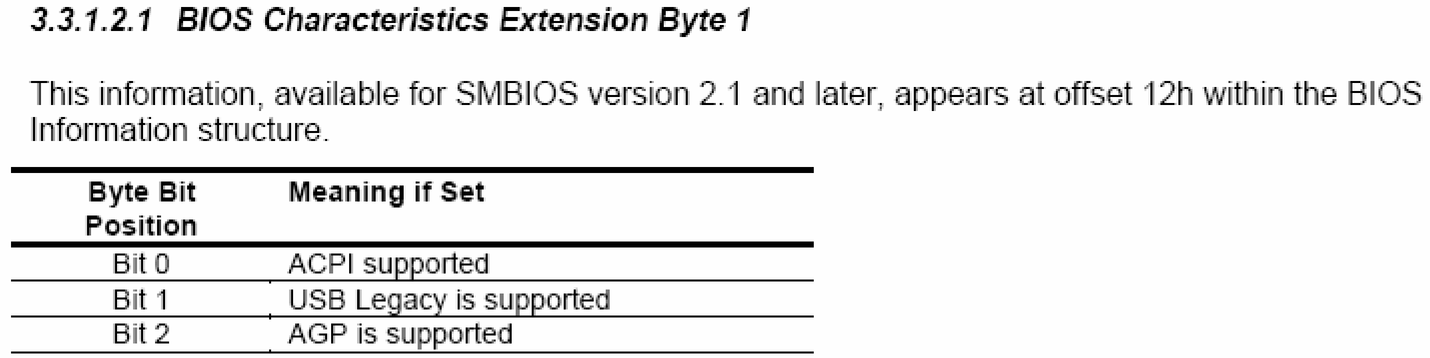
对照 spec 很容易解释 0000 0000 7FCB DE90



3.3.1.1 BIOS Characteristics

QWORD Bit Position	Meaning if Set
Bit 0	Reserved
Bit 1	Reserved
Bit 2	Unknown
Bit 3	BIOS Characteristics Not Supported
Bit 4	ISA is supported
Bit 5	MCA is supported
Bit 6	EISA is supported
Bit 7	PCI is supported
Bit 8	PC Card (PCMCIA) is supported
Bit 9	Plug and Play is supported
Bit 10	APM is supported
Bit 11	BIOS is Upgradeable (Flash)
Bit 12	BIOS shadowing is allowed
Bit 13	VL-VESA is supported
Bit 14	ESCD support is available
Bit 15	Boot from CD is supported
Bit 16	Selectable Boot is supported
Bit 17	BIOS ROM is socketed
Bit 18	Boot From PC Card (PCMCIA) is supported
Bit 19	EDD (Enhanced Disk Drive) Specification is supported
Bit 20	Int 13h - Japanese Floppy for NEC 9800 1.2mb (3.5", 1k Bytes/Sector, 360 RPM) is supported
Bit 21	Int 13h - Japanese Floppy for Toshiba 1.2mb (3.5", 360 RPM) is supported
Bit 22	Int 13h - 5.25" / 360 KB Floppy Services are supported
Bit 23	Int 13h - 5.25" /1.2MB Floppy Services are supported
Bit 24	Int 13h - 3.5" / 720 KB Floppy Services are supported
Bit 25	Int 13h - 3.5" / 2.88 MB Floppy Services are supported
Bit 26	Int 5h, Print Screen Service is supported
Bit 27	Int 9h, 8042 Keyboard services are supported
Bit 28	Int 14h, Serial Services are supported
Bit 29	Int 17h, Printer Services are supported
Bit 30	Int 10h, CGA/Mono Video Services are supported
Bit 31	NEC PC-98
Bits32:47	Reserved for BIOS Vendor
Bits 48:63	Reserved for System Vendor

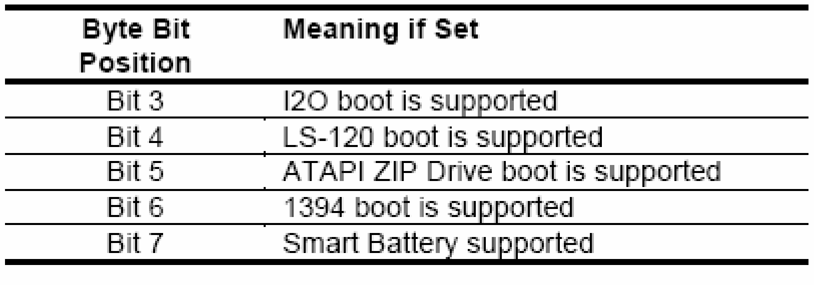
对照 spec 也容易弄清楚扩展特性



3.3.1.2.1 BIOS Characteristics Extension Byte 1

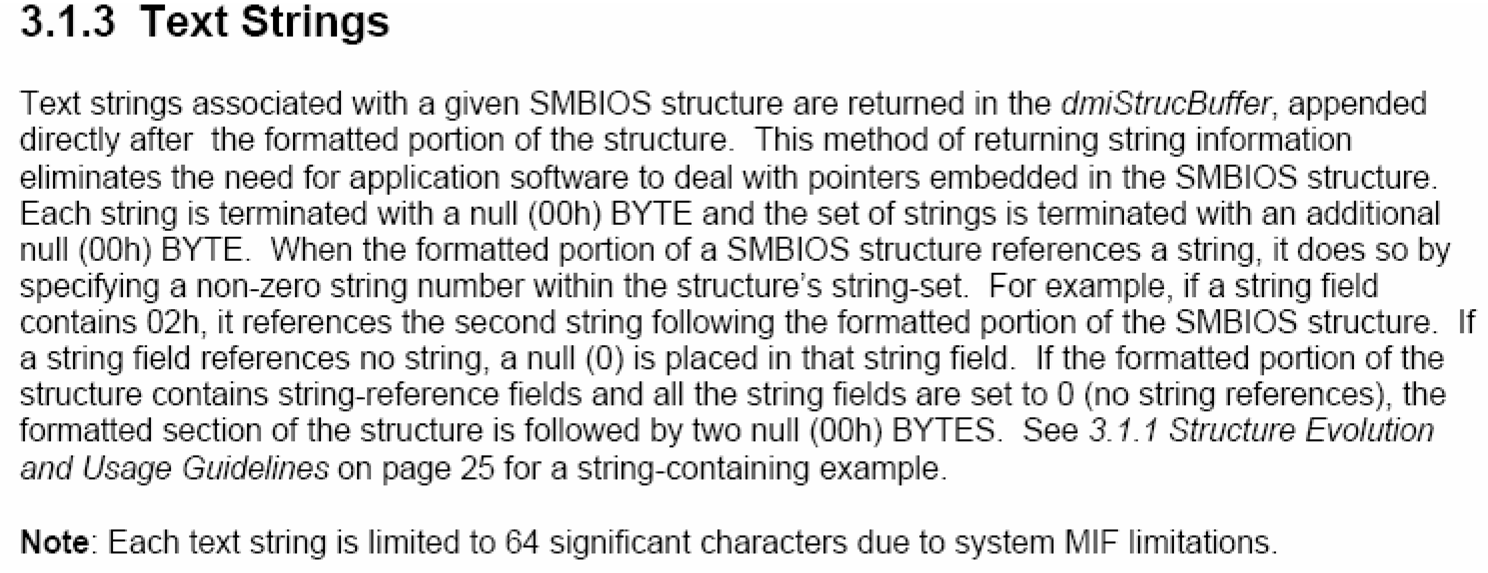
This information, available for SMBIOS version 2.1 and later, appears at offset 12h within the BIOS Information structure.

Byte Bit Position	Meaning if Set
Bit 0	ACPI supported
Bit 1	USB Legacy is supported
Bit 2	AGP is supported



Byte Bit Position	Meaning if Set
Bit 3	I2O boot is supported
Bit 4	LS-120 boot is supported
Bit 5	ATAPI ZIP Drive boot is supported
Bit 6	1394 boot is supported
Bit 7	Smart Battery supported

再后面就是字符串了.关于字符串 spec 有如下描述



3.1.3 Text Strings

Text strings associated with a given SMBIOS structure are returned in the *dmiStrucBuffer*, appended directly after the formatted portion of the structure. This method of returning string information eliminates the need for application software to deal with pointers embedded in the SMBIOS structure. Each string is terminated with a null (00h) BYTE and the set of strings is terminated with an additional null (00h) BYTE. When the formatted portion of a SMBIOS structure references a string, it does so by specifying a non-zero string number within the structure's string-set. For example, if a string field contains 02h, it references the second string following the formatted portion of the SMBIOS structure. If a string field references no string, a null (0) is placed in that string field. If the formatted portion of the structure contains string-reference fields and all the string fields are set to 0 (no string references), the formatted section of the structure is followed by two null (00h) BYTES. See *3.1.1 Structure Evolution and Usage Guidelines* on page 25 for a string-containing example.

Note: Each text string is limited to 64 significant characters due to system MIF limitations.

就是说一个字符串以 0 结尾,并且所有字符串都结束的时候多加一个 0.

```
Db “Phoenix Technologies, LTD”,0 ;第一个字符串
Db ” 6. 00 PG”,0                 ;第二个字符串
Db “01/10/2005”,0                ;第三个字符串
Db 0                             ;所有的字符串结束
```

四．结束语

这篇文章只是简单的介绍ＤＭＩ以及它的读取。也算是我学习的一点笔记。