To: T13 Technical Committee
From: Rob Elliott, HP (elliott@hp.com)
Date: 4 January 2010
Subject: e09127r3 EDD-4 Hybrid MBR boot code annex

Revision history

Revision 0 (24 July 2009) First revision

Revision 1 (3 November 2009) Added informative annex material about the GPT changes that the UEFI WG does not want to include in UEFI itself and recommended T13 consider for inclusion in EDD-4 instead. Two methods are described: 1) hybrid MBR partition records (already deployed in the industry, but discouraged by the UEFI WG because it violates the existing Protective MBR definition) and 2) hybrid MBR boot code (viewed as a safe approach by UEFI WG because it does not violate the Protective MBR definition).

Revision 2 (6 November 2009) Based on feedback from a conference call, split annex B into a separate proposal e09150 and merged the two remaining annexes. Removed remnants of earlier proposals where the hybrid MBR boot code was expected to support both MBR partition records and GPT.

Revision 4 (4 January 2010) Incorporated feedback from December T13 meeting - changed “understands” to “processes” and “x86” to “IA-32 compatible” or “IA-32”.

Related documents

d2132r1 - *Enhanced Disk Drive - 4 (EDD-4)* revision 1

Unified Extensible Firmware Interface (UEFI) SpecificationVersion 2.3 (May 2009). Chapter 5 defines the GPT disk layout and also defines the Protective MBR format. See http://www.uefi.org.

e09150r0 EDD-4 Hybrid MBR partition records annex (Rob Elliott, HP)

Overview

Individual 2 TB (2 x 10^{12} bytes) disk drives, with a maximum LBA of E8D4A510h based on 512 byte logical blocks, started shipping in 2009. The Master Boot Record (MBR) disk layout's 32-bit LBA addressing limit of 2 TiB (2 x 2^{40} bytes, maximum LBA of FFFFFFFFh) will soon be exceeded. However, there are many operating systems still in use that do not understand GUID Partition Table (GPT) disk layouts (which fix the problem by supporting 64-bit LBAs), and there are many systems still in use with legacy BIOSes (including systems supporting hybrid UEFI/legacy BIOS operation). Allowing both legacy BIOSes and legacy operating systems to support > 2 TiB disk drives and share disks using the GPT disk layout is desired.

The Unified EFI Forum has added a new Attribute value in the GUID Partition Table (GPT) disk layout to mark a GPT partition as containing a legacy BIOS bootable partition. This is documented as an errata on UEFI-2.3.

New GPT-cognizant “hybrid” MBR boot code is responsible for searching through the GPT to find the bootable partition, rather than selecting one of the partitions in the MBR partition table.

The UEFI specification is not a good place to define the responsibilities of legacy BIOS compatible MBR boot code, however. The EDD-4 standard is the home for legacy BIOS INT 13h function definitions, so seems like a good place to describe the responsibilities of MBR boot code and VBR boot code (which invoke the INT 13h calls already defined in EDD-4).

Suggested changes

2.4 Other references

These standards and specifications are also referenced.

BIOS Boot Specification (Compaq, Phoenix and Intel)~~,~~

For the BIOS Boot Specification published by Phoenix Technologies, contact them at www.phoenix.com

El Torito CD-ROM Boot Specification

For the El Torito CD-ROM Boot Specification published by Phoenix Technologies, contact them at www.phoenix.com

ATAPI Removable Media BIOS Specification

For the ATAPI Removable Media BIOS Specification published by Phoenix Technologies, contact them at www.phoenix.com

Universal Serial Bus Revision 1.1

For the Universal Serial Bus Revision 1.1 specification, contact the USB Implementors Forum at www.usb.org

Mass Storage Overview

For the Mass Storage Overview specification, contact the USB Implementors Forum at www.usb.org, www.usb.org/developers

Unified Extensible Firmware Interface Specification, Version 2.3 (UEFI-2.3) plus errata

For the UEFI-2.3 specification, contact the Unified EFI Forum at www.uefi.org

3.1 Definitions and abbreviations

3.1.3 BIOS: The Basic Input/Output System (BIOS) is the firmware stored in non-volatile (NV) memory located on the computer's main board. The BIOS executes POST (see 3.2.21~~xxx~~) to test and initialize the system components and then loads the ~~O/S~~OS. The BIOS also handles the low-level Input/Output to the various peripheral devices connected to the computer.

3.1.16 IPL Device: An Initial Program Load Device is any device in the system that may boot and load an ~~O/S~~OS. In standard AT machines, this is normally the floppy drive or hard drive.

3.1.20 ~~O/S~~OS: An Operating System is a software abstraction layer that provides services that give applications access to system hardware, in a hardware independent fashion. Examples of these services include memory management, multi-threaded task management, file system management, printer management, and screen management.

3.1.22 POST: The Power-On Self-Test is the part of the BIOS that takes control immediately after power is applied to the computer. POST initializes the computer hardware so that an ~~O/S~~OS may be loaded.

Editor's Note 1: Changing O/S to OS since "OS" is already used but O/S is not used anywhere but in the glossary. This proposal adds more uses of the term OS.

Editor's Note 2: Additional global change in EDD-4: change "INT 13" to "INT 13h". Both are used; the latter is correct. These changes are NOT noted in this proposal.

3.1.100 GPT disk layout: The disk layout defined by the Unified EFI specification (see UEFI-2.3).

3.1.101 MBR disk layout: The disk layout traditionally used by BIOS based systems.

3.1.102 Master Boot Record (MBR): The first LBA on a disk.

3.1.103 Volume Boot Record (VBR): The first LBA in a bootable partition.

3.1.104 Hybrid MBR boot code: IA-32 compatible code located in the MBR that processes the GPT disk layout.

3.1.105 Hybrid VBR boot code: IA-32 compatible code located in the VBR that processes the GPT disk layout.

3.2 Symbols and abbreviations

GPT	GUID Partition Table (see UEFI-2.3)
MBR	Master Boot Record (see 3.1.102)
UEFI	Unified Extensible Firmware Interface (see UEFI-2.3)
VBR	Volume Boot Record (see 3.1.103)

Editor’s Note 3: All remaining material is new, so is not blue underlined

Annex A
(informative)

Hybrid MBR boot code

A.1 Hybrid MBR boot code overview

This annex describes how MBR boot code called hybrid MBR boot code may be constructed to support a GPT disk layout (see UEFI-2.3) in a legacy BIOS system.

A.2 GPT support for hybrid MBR boot code

When using hybrid MBR boot code, the GPT disk layout includes one GPT partition with the Legacy BIOS Bootable bit set to one in the Attributes field (see UEFI-2.3).

The Protective MBR defined in UEFI-2.3 is modified as shown in table A.1 to include hybrid MBR boot code.

Table A.1 — Protective MBR with hybrid MBR boot code

Mnemonic	Byte offset	Byte length	Contents
Boot Code	0	440	Hybrid MBR boot code (see A.3)
Unique MBR Disk Signature	440	4	See UEFI-2.3
Unknown	444	2	See UEFI-2.3
Partition Record	446	16 * 4	See UEFI-2.3
Signature	510	2	See UEFI-2.3
Reserved	512	Logical Block Size - 512	See UEFI-2.3

Figure A.1 shows an example of a GPT disk layout with hybrid MBR boot code.

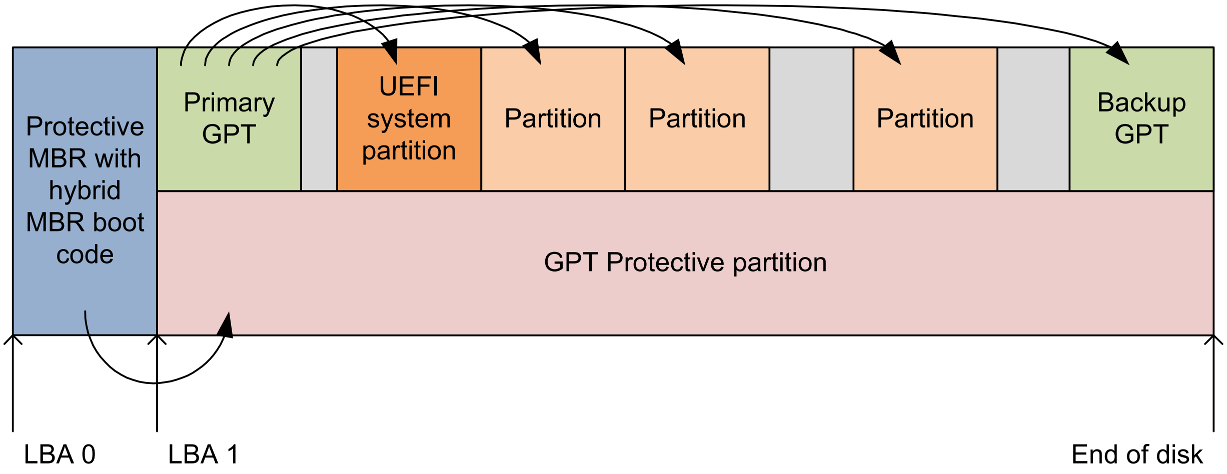

Figure A.1 — GPT disk layout with hybrid MBR boot code example

Figure A.2 shows an example of a GPT disk layout with hybrid MBR boot code on a disk with an ending LBA greater than FFFFFFFFh. The MBR partition record is not able to describe the LBAs above FFFFFFFFh.

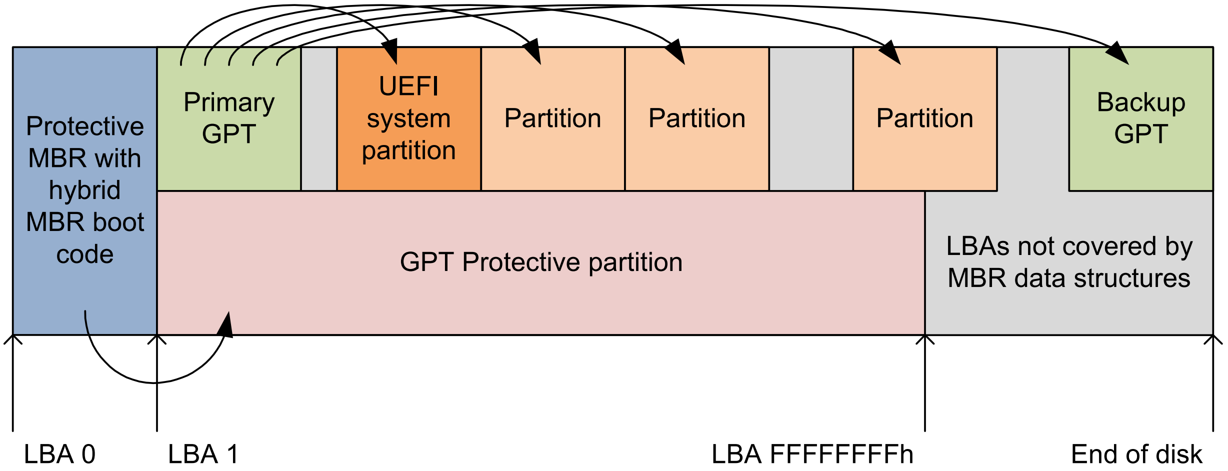

Figure A.2 — GPT disk layout with hybrid MBR boot code truncation example

A.3 Hybrid MBR boot code

Hybrid MBR boot code performs the following steps:

1) use the Extended Read function (i.e, INT 13h FN 42h)(see 8.15) to load the Primary GPT Header. The Device Address Packet (see table 9 in 6.2) includes:
   A) the LBA field set to 00000000 00000001h; and
   B) the Number of Blocks field set to 01h;
2) calculate the CRC32 of the GPT Header and verify that it equals the Header CRC32 field;
3) if the Header CRC32 field is incorrect, use the Extended Read function to load the backup GPT Header. The Device Address Packet includes:
   A) the LBA field set to the last LBA of the disk; and
   B) the Number of Blocks field set to 01h;

NOTE 1 - Due to the limited size (i.e., 440 bytes) of the MBR boot code, the hybrid MBR boot code may not perform all the checks that EFI system firmware would perform.

4) use the Extended Read function to load the GPT Partition Entry Array pointed to by the selected GPT Header. The Device Address Packet includes:
   A) the LBA field set to the value specified in the Partition Entry LBA field; and
   B) the Number of Blocks field set to (Number Of Partition Entries * Size Of Partition Entry) / Logical Block Size;
5) calculate the CRC32 of the GPT Partition Entry Array and verify that it equals the Partition Entry Array CRC32 field in the GPT Header;
6) if the Partition Entry Array CRC32 field is incorrect and the Primary GPT Partition Entry Array was selected, use the Extended Read function to load the backup GPT Header from the last LBA of the disk (see step 3), load the backup GPT Partition Entry Array (see step 4), and calculate the CRC32 (see step 5). If the CRC32 is incorrect again, report an error and stop;
7) search the GPT Partition Entry Array for a partition with the Legacy BIOS Bootable bit set to one in the Attributes field. The code may also check for a special value in the Partition Type GUID field when selecting the partition;
8) use the Extended Read function to load the first logical block of the selected partition into memory address 7C00h; and
9) set the IA-32 registers according to the hybrid MBR boot code handover procedure (see A.4) and jump to 7C00h.

NOTE 2 - In contrast, legacy MBR boot code searches the MBR Partition Records for a partition with the Boot Indicator field set to 80h, then loads the VBR based on the Starting LBA field and passes the Size In LBA field to the VBR.

A.4 Hybrid MBR boot code handover procedure

Hybrid MBR boot code fills in the IA-32 registers according to table A.2 before jumping to 7C00h.

Table A.2 — Hybrid MBR handover IA-32 register values

Register	Description	Differences from legacy MBR handover
DL	Disk number	No change
ES:DI	Pointer to $PnP	No change
EAX	54504721h (i.e., "!GPT"). Indicates that the hybrid MBR handover structure is being passed with DS:SI rather than the legacy MBR Partition Record	New
DS:SI	Pointer to the hybrid MBR handover structure (see table A.3)	New

Table A.3 defines the Hybrid MBR boot code handover structure.

NOTE 3 - With legacy MBR boot code, this structure included fields from the MBR Partition Record of the boot partition. Since the GPT partition may be located at an LBA beyond the 32-bit LBA addressing boundary, additional fields are added to communicate the full information to the VBR.

Table A.3 — Hybrid MBR boot code handover structure

Mnemonic	Byte Offset	Byte Length	Description	Differences from legacy MBR handover structure
Boot Indicator	0	1	Set to 80h (i.e., bootable).	Fixed value
Starting CHS	1	3	Set to FFFFFFh. The VBR boot code should ignore this field.	Fixed value
OS Type	4	1	Set to the OS Type of the bootable partition (i.e., that would have been assigned had the partition been installed in an MBR disk layout).	No change
Ending CHS	5	3	Set to FFFFFFh. The VBR boot code should ignore this field.	Fixed value
Starting LBA	8	4	Set to FFFFFFFFh.	Fixed value
Size In LBA	12	4	Set to FFFFFFFFh.	Fixed value
Size Of Partition Entry	16	4	Set to the Size Of Partition Entry field of the GPT Partition Header.	New field
GPT Partition Entry	20	Partition Entry Size	Set to the GPT Partition Entry of the boot partition.	New field

A.5 Hybrid VBR boot code

Hybrid VBR boot code is responsible for booting the OS, and is usually OS-specific.

If EAX is set to "!GPT", then hybrid VBR boot code uses the GPT Partition Entry field to identify its partition.

NOTE 4 - In contrast, legacy VBR boot code only recognizes the MBR disk layout and uses the Starting LBA field and Size In LBA field to identify its partition.