To: T13 Technical Committee
From: Rob Elliott, HP (elliott@hp.com)
Date: 3 November 2009
Subject: e09127r1 EDD-4 Hybrid MBR support

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).

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.

Overview

Individual 2 TB (2 x 10^{12}) disk drives, with a maximum LBA of 0xE8D4A510 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}, maximum LBA of 0xFFFFFFFF) 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 an ECR on UEFI-2.3.

New GPT-cognizant hybrid MBR boot code would be 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).

Editor's Note 1: Some UEFI formatting conventions are leftover like using red italics forUEFI field names; the EDD-4 editor is welcome toapply any overriding T13 conventions.

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)

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

3.1 Definitions and abbreviations

3.1.xx OS: Operating system

Editor's Note 2: All remaining material is new, so is not blue underlined

---

Annex A
(informative)

Hybrid MBR boot code

A.1 Definitions and abbreviations

A.1.1 GPT disk layout: The disk layout defined by the Unified EFI specification (see UEFI-2.3).

A.1.2 MBR disk layout: The disk layout traditionally used by BIOS based systems.

A.1.3 Master Boot Record (MBR): The first LBA on a disk.

A.1.4 Volume Boot Record (VBR): The first LBA in a bootable partition.

A.1.5 Hybrid MBR boot code: x86 code located in the MBR that understands both the MBR disk layout and the GPT disk layout.

A.2 Symbols and abbreviations

GPT	GUID Partition Table (see UEFI-2.3)
MBR	Master Boot Record (see A.1.3)
UEFI	Unified Extensible Firmware Interface (see UEFI-2.3)
VBR	Volume Boot Record (see A.1.4)

A.3 Legacy MBR boot code

Legacy MBR boot code searches the MBR Partition Records for a partition with the *Boot Indicator* field set to 0x80, 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

Hybrid MBR boot code performs the following steps:

1) use INT 13h to load the Primary GPT Header (LBA 0x00000001, Size: 0x00000001);
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 INT 13h to load the backup GPT Header (LBA: Last LBA of the disk, Size: 0x00000001);

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

4) use INT 13h to load the GPT Partition Entry Array pointed to by the selected GPT Header (LBA: specified in the *Partition Entry LBA* field, Size: (*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 INT 13h to load the backup GPT Header from the last LBA of the disk (see step 3) and 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 INT 13h to load the first logical block of the selected partition into memory address 0x7C00; and
9) set the x86 registers according to the Hybrid MBR Handover Procedure (see A.5) and jump to 0x7C00.

A.5 Hybrid MBR boot code handover procedure

Hybrid MBR boot code fills in the x86 registers according to Table A.1 before jumping to 0x7C00 if it selected a partition from GPT rather than MBR.

Table A.1 — Hybrid MBR handover x86 register values

Register	Description	Differences from legacy MBR handover
DL	Disk number	No change
ES:DI	Pointer to $PnP	No change
EAX	0x54504721 (i.e., "!GPT"). Indicates the Hybrid MBR Handover Structure is being passed with DS:SI rather than just the Legacy MBR Partition Record.	New
DS:SI	Pointer to the Hybrid MBR Handover Structure (see Table 2)	New

Table A.2 defines the Hybrid MBR boot code handover structure.

NOTE 2 - Historically this structure included only 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.2 — Hybrid MBR boot code handover procedure

Mnemonic	Byte Offset	Byte Length	Description	Differences from legacy MBR handover
Boot Indicator	0	1	Set to the *Boot Indicator* field of the Partition Record of the bootable partition (i.e., set to 0x80 (i.e., Bootable)). Set to 0x80 if the bootable partition is from GPT.	No change
Starting CHS	1	3	Set to the *Starting CHS* field of the Partition Record of the bootable partition. Set to 0xFFFFFF if the bootable partition is from GPT. The VBR should ignore this field.	No change
OS Type	4	1	Set to the *OS Type* field of the Partition Record of the bootable partition. Set to 0xED if the bootable partition is from GPT.	No change
Ending CHS	5	3	Set to the *Ending CHS* field of the Partition Record of the bootable partition. Set to 0xFFFFFF if the bootable partition is from GPT. The VBR should ignore this field.	No change
Starting LBA	8	4	Set to the *Starting LBA* field of the Partition Record of the bootable partition. Set to 0xFFFFFFFF if the bootable partition is from GPT.	No change
Size In LBA	12	4	Set to the *Size In LBA* field of the Partition Record of the bootable partition. Set to 0xFFFFFFFF if the bootable partition is from GPT.	No change
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.6 VBR boot code

The VBR boot code is responsible for booting the operating system, and is generally operating system specific.

Legacy VBR code only recognizes the MBR disk layout, although the OS that is booted may recognize the GPT disk layout.

If EAX is set to "!GPT", then hybrid MBR cognizant VBR code uses the *GPT Partition Entry* field and the GPT tables rather than the *Starting LBA* field and the *Size In LBA* field.

Annex B
(informative)

GPT support for hybrid MBR partition records

B.1 Overview

Hybrid MBR partition records are a technique that has been used to enable booting legacy BIOS compatible OSes on systems supporting both UEFI and legacy BIOS system firmware. With this technique, the GPT disk layout is used, but the MBR does not comply with the protective MBR requirements (see UEFI-2.3).

The MBR contains:

a) legacy MBR boot code; and
b) one partition record protecting the Primary GPT and the UEFI system partition;
c) one partition record pointing to a hybrid partition (pointed to by both MBR and GPT partition tables) containing legacy VBR boot code;
d) additional partition records pointing to additional hybrid partitions, or protecting the remaining capacity of the disk.

Figure B.1 shows an example of hybrid MBR partition records.

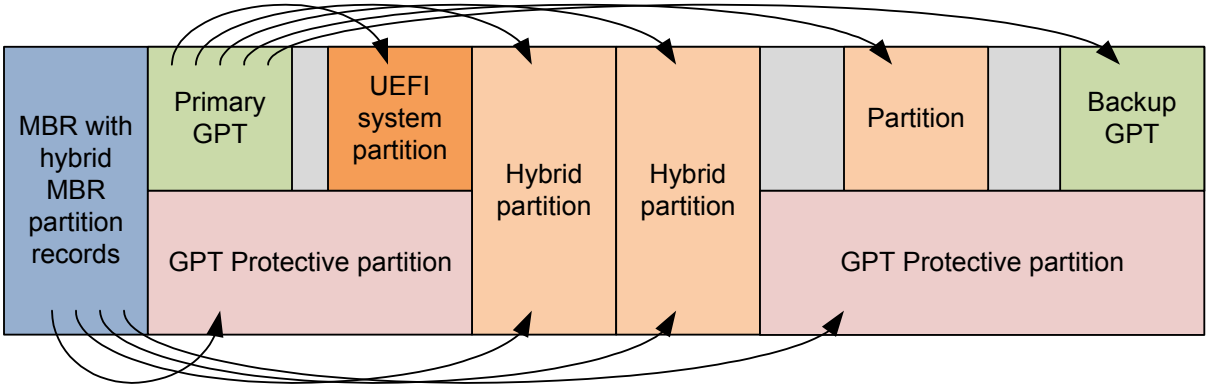

Figure B.1 — GPT disk layout with hybrid MBR partition records example

Because this technique does not comply with the protective MBR requirements, it might interfere with the proper behavior of UEFI OSes and UEFI system firmware. This technique requires partitioning tools to keep the MBR and GPT synchronized; running a tool that does not understand the technique might result in overlapping partitions and lost data.

Annex C
(informative)

GPT support for hybrid MBR boot code

C.1 Overview

Annex A describes how MBR boot code may be constructed to support a GPT disk layout in a legacy BIOS system; this is called hybrid MBR boot code. This annex describes the changes to the Protective MBR format defined in UEFI-2.3 to implement this approach.

The GPT disk layout includes one GPT partition with the Legacy BIOS Bootable bit set to one in the Attributes field.

The Protective MBR is modified as shown in table C.1 to include hybrid MBR boot code.

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

Mnemonic	Byte offset	Byte length	Contents
Boot Code	0	440	Hybrid MBR boot code (see A.4)
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 C.1 shows an example of a GPT disk layout with hybrid MBR boot code.

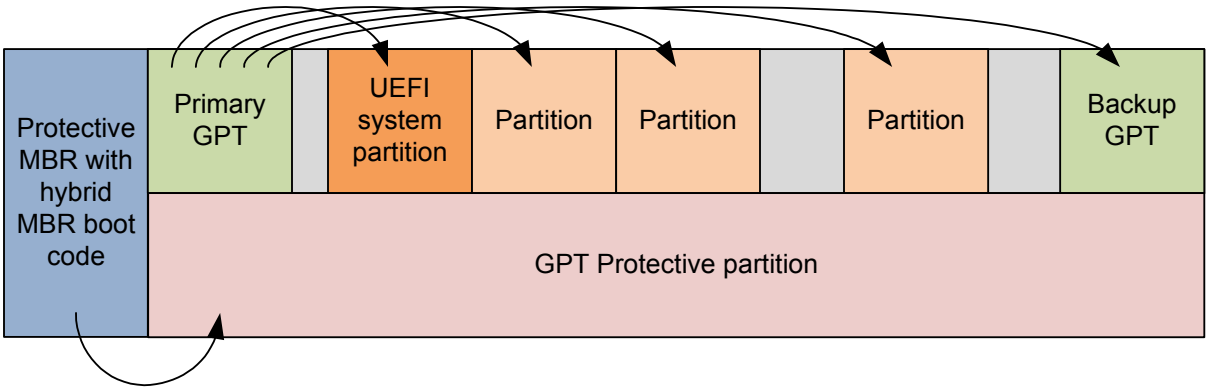

Figure C.1 — GPT disk layout with hybrid MBR boot code example