Atari Partition Table (APT) layout
revision S

Table of Contents

General remark: all values discussed throughout the document are unsigned unless stated otherwise.

I. General structure

The APT data consists of two main parts:

1) the mapping slots, providing information about current assignment of the partitions to DOS drive numbers. This is the only part of the entire table which is relevant for programs such as disk interface BIOS-es, which want to read the partition information at start-up or when a RESET interrupt occurs.

2) the partition table entries, providing information on how the entire disk space covered by the APT is divided into physical partitions. This part is only relevant for partitioning programs, partition managers and such utility software.

The first part consists of at least 15 entries, 16 bytes each (240 bytes). The other part immediately follows the first one, and may consist of up to 65536 entries, 16 bytes each (1 MB).

II. Data container

The entire APT is stored in a number of physical sectors chained by a bi-directional linked list. Each sector contains a header which provides the chain links to the next and previous sectors. The header of the first sector occupied by APT contains additional information about the APT structure.

By default, the partition table header sits in LBA 0. This should make a conversion from the older version (KMK/JŻ IDE or IDEa) painless, *but such a partition table may not occupy more sectors than one LBA 0*, so consider this situation rather exceptional than a rule.

Normally, the LBA 0 should contain a protective, PC-like MBR (with a single partition entry of type $7F, embracing the whole disk, or a part of it, if some place is to be provided for native PC partitions) preventing e. g. Windows from destroying the partition table. In this case the partition table header is located in the first sector occupied by the $7F partition.

III. 16-byte partition table header entry

The header is present in each sector of the partition table chain. The first header is located in the first sector allocated for the partition table, at offset 0. The header is followed by 16-byte partition entries, the number of them is indicated in the header at offset 6. Therefore, a 512-byte sector may contain up to 31 partition entries + 1 header entry.

III.1 Structure

Offset	Size	Purpose
0	1	Global flags: * bit 7 = 1 – partition table data stored in even (low) bytes only * bits 6-5 – partition table layout revision number (now %00) * bit 4 – global Metadata flag (as of rev. S) * bits 3-0 – number of mapping slots, less 15 (normally %0000)

Offset	Size	Purpose
		Global Metadata flag indicates whether new partitions (type $00 and type $03) created in this APT should have Metadata leaders (1) or not (0).

Number of mapping slots is the number of extra 16-byte entries reserved at the beginning of the table for partition->DOS drive mapping information. Normally there are 15 slots for 15 drives directly available via SIOV. These slots are sequentially assigned to DOS drive numbers, slot 0 is D1:, slot 1 is D2: etc. A slot contains a copy of the partition entry that was selected by the user to correspond to a DOS drive number. The place for extra slots is provided for future extension. The actual partition entries begin after the mapping slots. In minimal configuration: 1 header entry + 15 mapping slots + up to 16 partitions, the entire partition table occupies 1 sector. The first byte of a mapping slot equal to $00 means that the corresponding mapping slot is unpopulated. |
1	3	Signature: $41, $50, $54 („APT")
4	1	Boot drive number (1-15, or 0, when unspecified; high nybble reserved)
5	1	Number of entries in current table sector (including the header entry). This entry may not be 0 nor it may be greater than 32. This may be used for additional sanity check. This number includes the mapping slots.
6	1	Offset to the header entry in next table sector (counted in 16-byte increments). This value may only be in range 0-31.
7	1	Offset to the header entry in previous table sector (as above). This value may only be in range 0-31.
8	4	Next sector in table chain (absolute LBA sector number, 0 in last header)
12	4	Previous sector in table chain (absolute LBA sector number, 0 in first header)

* The *global flags*, *signature* and *boot drive* fields are only valid in the first header. In the remaining headers the *signature* may be set for sanity-checking; the remaining fields are reserved and set to zero.

* Unused bits which are not explicitly mentioned above, are reserved and should be set to 0.

* In a header with known revision number, any reserved bits set should be ignored. A header with an unknown revision number should be considered read-only by partitioning software.

IV. 16-byte partition entry

Offset	Size	Purpose
0	1	Partition access flags:

* bit 7 = %1 – partition is reserved (skip regardless of other settings)
* bit 6 = %1 – Metadata leader sector is present
* bits 5-4 – reserved
* bits 3-2 – see the explanation given later on
* bits 1-0 – %00 = end table, %01 = 128 BPS, %10 = 256 BPS, %11 = 512 BPS |

Offset	Size	Purpose
		A value of $00 means that the corresponding entry is void and terminates the table. If any of the reserved bits is set, such a partition entry should be considered read-only (not changeable). The space occupied by such a partition should *not* be considered free. *Partition access flags* bits 2 and 3 contain information on how 256-byte sectors are mapped onto 512-byters, if applicable. The information is provided to prevent access, if the storing method used is not compatible with the firmware that is trying to access the disk. The bits thus mean: | Bit 2	Bit 3	Meaning | |---|---|---| | 0 | 0 | Actual data is stored in low order byte of each word in 512-byte sector, high order byte of the word being unused („NUL-padded bytes"). | | 0 | 1 | Actual data occupies the first 128 consecutive words of a 512-byte sector, the other half being unused („NUL-padded sectors"). | | 1 | 0 | Low order bytes of each word are occupied by the lower order logical sector, and the high order bytes of each word are occupied by the higher order („interleaved bytes"). | | 1 | 1 | The first 128 consecutive words of a 512-byte sector are occupied by the lower order logical sector, and the other 128 consecutive words are occupied by the higher order logical sector („interleaved sectors"). | The bits only tell how 256-byters are mapped onto physical 512-byte sectors. A 128-byter should always use the same method, just leaving every other byte of a 256-byter unused; i. e. a 128-byter is a 256-byter with high order (odd) bytes unused. If a physical sector on the disk is (or is available as) a 256-byter, therefore making mapping not necessary, then the partitioning software should set both bits to 0.
1	1	Partition type: * $00 – DOS partition * $01 – firmware config partition (firmware-specific contents) * $02 – „floppy drawer" (partition split into fixed-size chunks) * $03 – external (MBR) DOS partition Other values reserved. *Partition type* is a field which determines how to interpret the last six bytes

Offset	Size	Purpose
		of the partition entry. A partition entry with an unknown value in this field should be considered read-only (not changeable). The space occupied by such a partition should *not* be considered free. The *firmware-config* type of partition is optional, and its contents is specific to the firmware using it. The firmware in question should provide a method of uniquely identifying the contents as valid, and avoid accessing the data stored by different firmware. The *external DOS partition* is a partition which is located outside the APT container. This will be any partition which is to be shared between PC and Atari, such as a FAT partition.
2	4	Starting sector (absolute LBA sector number). If this value is 0, the partition entry should be ignored (considered invalid).
6	4	Sector count (number of physical sectors occupied by partition). If this value is 0, the partition entry should be ignored (considered invalid).
10	2	Partition ID (unique number identifying the partition, the first one is $0000). In actual partition entries (located past the mapping slots) this field has no meaning and may be kept $0000. When a partition entry is transferred to the mapping slot, the mounting software should write here the partition entry number, from where the slot is being filled in. When partition entries change location within the table, they also change this number; therefore in that case the corresponding mapping slots should be updated accordingly.
12	4	Four bytes dependent on the above *Partition type* value. Type $00 (DOS partition): <table><tr><th>Offset</th><th>Size</th><th>Purpose</th></tr><tr><td>+0</td><td>1</td><td>DOS access flags: * bit 7 = %1 – partition is write-protected * bit 6 = %1 – partition active („automount bit") * bits 5-0 – reserved These are partition flags associated with the DOS partition type defined above. The high nibble is reserved for the actual flags. The „automount" bit is named so because from the user perspective it makes the disks to be made automatically available at certain drive numbers (i.e. like automatically mounted). In fact all DOS drives are „mounted", i. e. assigned to DOS drive numbers, but these with the automount bit cleared are not marked active. They are in fact semi-mounted, i. e. they occupy drive numbers on the list of current partitions, but they do not respond to system inquiries (they're „transparent" and</td></tr></table>

Offset	Size	Purpose		
		Offset	**Size**	**Purpose**
				allow e. g. SIO drives occupying the same drive number to respond instead). See section IV below.
		+1	1	Reserved byte.
		+2	2	Offset to sector 1 (normally $0000). This value should be added to *Starting sector* value in order to obtain the LBA address of the data sector 1. *Offset to sector 1* is normally 0, but it may be increased to create a „reserved" area (up to 32 MB) between the Metadata leader and the partition data.

Type $01 (firmware config): 4 reserved bytes.

Type $02 (floppy drawer):

Offset	Size	Purpose
+0	2	Chunk size (in physical sectors)
+2	2	Reserved bytes.

Type $03 (external DOS partition):

Offset	Size	Purpose
+0	1	DOS access flags: * bit 7 = %1 – partition is write-protected * bit 6 = %1 – partition active („automount bit") * bits 5-4 – reserved * bits 3-0 – bits 27-24 of LBA address of Metadata sector The meaning of bits 6-7 is the same as in Type $01.
1	3	Bits 0-23 of LBA address of Metadata sector, little endian order.

* Unused bits which are not explicitly mentioned above, are reserved and should be set to 0.

V. Metadata area

Metadata area is an optional data structure associated with a partition. The first Metadata sector, called *the Metadata leader*, is located at LBA address *Starting sector* less 1 (right before the partition data sectors) for any partition type except $03, where its location is explicitly specified inside the APT partition entry. The Metadata area may consist of more sectors, as defined inside its structure.

The purpose of Metadata is to contain the following information:

1) optional partition name (up to 40 ASCII characters, EOL-terminated)
2) optional extended partition definition fields (if the ordinary partition entry is not sufficient)
3) optional partition tags for partition management program filtering procedures
4) optional pointers to extra Metadata sectors

V.1. Metadata leader structure

Offset	Size	Purpose
0	3	Partition table signature: $41, $50, $54 („APT")
3	1	Metadata revision number (now $00)
4	4	Metadata signature: $4D, $45, $54, $41 („META")
8	8	Reserved bytes
16	40	Partition name: up to 39-char, NUL-terminated ATASCII string
56	456	Reserved, all zeros

The first 16 bytes is the actual Metadata header, which should be present in every Metadata sector.

A Metadata sector (including the Metadata leader) may be empty (all zeros), or not existing at all (*Access flags* bit 6 = %0). In any case, when *Access flags* bit 6 = %1 but the signatures are invalid, such a Metadata sector should be ignored.

VI. DOS drive mounting considerations

When two (or more) physical drives with APT on them meet in one system, it may (and certainly will most of the time) happen that the same DOS drive numbers are assigned in each copy of the APT. In such a case, the following approach may be used while automounting disks:

1) all disks from the primary device should be mounted (in ATA/IDE: the master drive should be mounted first) respecting the drive assignments found within the partition table.

2) after that, while mounting disks located on the secondary and every further device (in ATA/IDE: the slave drive), the order of drive assignments found within its partition table should be respected, but the actual drive assignments must be shifted so that the secondary device partitions appear after the last drive mounted from the primary device.

For example, let us assume that the primary device has four partitions on it: 01, 02, 03, 04 and 05, out of which 01 and 03 are DOS drives to be automounted, 02 and 04 are DOS drives not to be automounted, and 05 is not a DOS drive. The drive numbers assigned are 3, 4, 5, 6.

Partition	Drive no.	Status
01	3	DOS drive, automount
02	4	DOS drive
03	5	DOS drive, automount
04	6	DOS drive

05	-	non-DOS drive

The secondary device has three partitions on it, 01, 02 and 03, out of which 01 is not a DOS drive, 02 is one, but it is not automounted, and 03 is an automounted DOS drive. The assigned numbers are 2 and 3.

Partition	Drive no.	Status
01	-	non-DOS drive
02	2	DOS drive
03	3	DOS drive, automount

In first step, the primary device partitions should be handled as follows (P – primary):

P01 mounted as D3:, marked active P02 mounted as D4:, marked inactive P03 mounted as D5:, marked active P04 mounted as D6:, marked inactive

P05 ignored

In the next step, the secondary device partitions should be mounted starting from the first free drive number found after the highest already assigned drive number. In the example above the last drive number assigned is D6:, therefore secondary partitions should be mounted to drive numbers from D7: onwards (S – secondary):

S01 ignored

S02 mounted as D7:, marked inactive

S03 mounted as D8:, marked active

If after mounting the primary device partitions there is still place in mapping slots, this approach solves conflicting drive assignments, which are sure to occur, when two users want to attach their primary disks to one HDD interface in order to share files.