Logical block addressing

From Wikipedia, the free encyclopedia

Logical block addressing (LBA) is a common scheme used for specifying the location of blocks of data stored on computer storage devices, generally secondary storage systems such as hard disks.

LBA is a particularly simple linear addressing scheme; blocks are located by an integer index, with the first block being LBA 0, the second LBA 1, and so on.

IDE standard included 22-bit LBA as an option, which was further extended to 28-bit with the release of ATA-1 (1994) and to 48-bit with the release of ATA-6 (2003). Most hard drives released after 1996 implement logical block addressing.

Contents

- 1 Overview
- 2 Enhanced BIOS
  - 2.1 LBA-Assisted translation
  - 2.2 LBA48
- 3 CHS conversion
  - 3.1 OS dependencies
- 4 See also
- 5 References
- 6 External links

Overview

In logical block addressing, only one number is used to address data, and each linear base address describes a single block.

The LBA scheme replaces earlier schemes which exposed the physical details of the storage device to the software of the operating system. Chief among these was the cylinder-head-sector (CHS) scheme, where blocks were addressed by means of a tuple which defined the cylinder, head, and sector at which they appeared on the hard disk. CHS did not map well to devices other than hard disks (such as tapes and networked storage), and was generally not used for them. CHS was used in early MFM and RLL drives, and both it and its successor Extended Cylinder-Head-Sector (ECHS) were used in the first ATA drives. However, current disk drives use zone bit recording, where the number of sectors per track depends on the track number. Even though the disk drive will report some CHS values as sectors per track (SPT) and heads per cylinder (HPC), they have little to do with the disk drive's true geometry.

LBA was first introduced in SCSI as an abstraction. While the drive controller still addresses data blocks by their CHS address, this information is generally not used by the SCSI device driver, the OS, filesystem code, or any applications (such as databases) that access the "raw" disk. System calls requiring block-level I/O pass LBA definitions to the storage device driver; for simple cases (where one volume maps to one physical drive), this LBA is then passed directly to the drive controller.

In RAID devices and SANs and where logical drives (LUNs) are composed via LUN virtualization and aggregation, LBA addressing of individual disk should be translated by a software layer to provide uniform LBA addressing for the entire storage device.

Enhanced BIOS

See also: INT 13H

The earlier IDE standard from Western Digital introduced 22 bit LBA; in 1994, the ATA-1 standard allowed for 28 bit addresses in both LBA and CHS modes. The CHS scheme used 16 bits for cylinder, 4 bits for head and 8 bits for sector, counting sectors from 1 to 255. This means the reported number of heads never exceeds 16 (0–15), the number of sectors can be 255 (1–255; though 63 is often the largest used) and the number of cylinders can be as large as 65,536 (0–65535), limiting disk size to 128 GiB (≈137.4 GB), assuming 512 byte sectors. These values can be accessed by issuing the ATA command "Identify Device" (`ECh`) to the drive.[1]

However IBM BIOS implementation defined in the INT 13H disk access routines used quite a different 24-bit scheme for CHS addressing, with 10 bits for cylinder, 8 bits for head, and 6 bits for sector, or 1024 cylinders, 256 heads, and 63 sectors.[2] This INT 13H implementation had pre-dated the ATA standard, as it was introduced when the IBM PC had only floppy disk storage, and when hard disk drives were introduced on the IBM PC/XT, INT 13H interface could not be practically redesigned due to backward compatibility issues. Overlapping ATA CHS mapping with BIOS CHS mapping produced the lowest common denominator of 10:4:6 bits, or 1024 cylinders, 16 heads, and 63 sectors, which gave the practical limit of 1024×16×63 sectors and 528 Mbytes (504 MiB), assuming 512 byte sectors.

In order for BIOS to overcome this limit and successfully work with large hard drives, a CHS translation scheme had to be implemented in BIOS disk I/O routines which would convert between 24-bit CHS used by INT 13H and 28-bit CHS numbering used by ATA. The translation scheme was called **Large** or **Bit Shift Translation**. This method would remap 16:4:8 bit ATA cylinders and heads to 10:8:6 bit scheme used by INT 13H, generating much more "virtual" drive heads than the physical disk reported. This increased the practical limit to 1024×256×63 sectors, or 8.4 Gbytes (7.8 GiB).

To further overcome this limit, INT 13H Extensions were introduced with **BIOS Enhanced Disk Drive Services** specification, which removed practical limits on disk size for operating systems which are aware of this new interface, such as *DOS 7.0* component in Windows 95. This *Enhanced BIOS* subsystem supports LBA addressing with **LBA** or **LBA-Assist** method, which uses native 28-bit LBA for addressing ATA disks and performs CHS conversion as needed.

The **Normal** or **None** method reverts to the earlier 10:4:6 bit CHS mode which does not support addressing more than 528 Mbytes.

Until the release of ATA-2 standard in 1996, there were a handful of large hard drives which did not support LBA addressing, so only Large or Normal methods could be used. However using the Large method also introduced portability problems, as different BIOSes often used different and incompatible translation methods, and hard drives partitioned on a computer with BIOS from a particular vendor often could not be read on a computer with a different make of BIOS. The solution was to use conversion software such as OnTrack Disk Manager, EZ-Drive, etc., which installed to the disk's OS loader and replaced INT 13H routines at boot time with custom code. This software could also enable LBA and INT 13H Extensions support for older computers with non LBA-compliant BIOSes.

LBA-Assisted translation

When the BIOS is configured to use a disk in LBA-Assisted translation mode, the BIOS access the hardware using LBA mode, but also presents a translated CHS geometry via the INT 13H interface. The number of cylinders, heads, and sectors in the translated geometry depends on the total size of the disk, as shown in the following table.[3]

Disk size	Sectors/track	Heads	Cylinders
1 < X ≤ 504 MiB	63	16	X/(63*16*512)
504 MiB < X ≤ 1008 MiB	63	32	X/(63*32*512)
1008 MiB < X ≤ 2016 MiB	63	64	X/(63*64*512)
2016 MiB < X ≤ 4032 MiB	63	128	X/(63*128*512)
4032 MiB < X ≤ 8032.5 MiB	63	255	X/(63*255*512)

LBA48

The current 48-bit LBA scheme, introduced in 2003 with ATA-6 standard, allows addressing up to 128 PiB. Current PC-Compatible computers support INT 13H Extensions, which use 64-bit structures for LBA addressing and should encompass any future extension of LBA addressing, though modern operating systems implement direct disk access and do not use the BIOS subsystems, except at boot load time. However, the common DOS style Master boot record partition table only supports disk partitions up to 2 TiB in size. For large partitions this needs to be replaced by another scheme, for instance the GUID Partition Table which has the same 64-bit limit as the current INT 13H Extensions.

CHS conversion

CHS (cylinder/head/sector) tuples can be mapped to LBA address with the following formula:

$$LBA = ((C \times HPC) + H) \times SPT + S - 1$$

where,

- C, H and S are the cylinder number, the head number, and the sector number
- LBA is the logical block address
- HPC is the maximum number of heads per cylinder (reported by disk drive, typically 16 for 28-bit LBA)
- SPT is the maximum number of sectors per track (reported by disk drive, typically 63 for 28-bit LBA)

LBA addresses can be mapped to CHS tuples with the following formula:

$$C = LBA \div (SPT \times HPC)$$
$$H = (LBA \div SPT) \bmod HPC$$
$$S = (LBA \bmod SPT) + 1$$

where

- mod is the modulo operation, i.e. the remainder, and

- ÷ is integer division, i.e. the quotient of the division.

According to the ATA specifications, "If the content of words (61:60) is greater than or equal to 16,514,064 then the content of word 1 [the number of logical cylinders] shall be equal to 16,383."[1] Therefore for LBA 16450559, an ATA drive may actually respond with the CHS *tuple* (16319, 15, 63), and the number of cylinders in this scheme must be much larger than 1024 allowed by INT 13H.[4]

LBA and CHS equivalence with 16 heads per cylinder

LBA Value	CHS *Tuple*
0	0, 0, 1
1	0, 0, 2
2	0, 0, 3
62	0, 0, 63
945	0, 15, 1
1007	0, 15, 63
1008	1, 0, 1
1070	1, 0, 63
1071	1, 1, 1
1133	1, 1, 63
1134	1, 2, 1
2015	1, 15, 63
2016	2, 0, 1
16,127	15, 15, 63
16,128	16, 0, 1
32,255	31, 15, 63
32,256	32, 0, 1
16,450,559	16319, 15, 63
16,514,063	16382, 15, 63

OS dependencies

Operating systems that are sensitive to BIOS-reported drive geometry include Solaris, DOS and Windows NT family, where NTLDR (NT, 2000, XP, Server 2003) or WINLOAD (Vista, Server 2008, Windows 7 and Server 2008 R2) use Master boot record which addresses the disk using CHS; x86-64 and Itanium versions of Windows can partition the drive with GUID Partition Table which uses LBA addressing.

Some operating systems do not require any translation because they do not use geometry reported by BIOS in their boot loaders. Among these operating systems are BSD, Linux, Mac OS X, OS/2 and ReactOS.

See also

- Cylinder-head-sector
- Block (data storage)
- Disk storage
- Disk formatting
- Disk partitioning
- SCSI Read Commands

References

1. ^ ***a b*** *Working Draft* of ATA/ATAPI-5 (http://www.t10.org/t13/project/d1321r3-ATA-ATAPI-5.pdf) Sections 6.2.1 and 8.12 of the T13 Technical Committee's, 29 February 2000.
2. ^ "KB224526: Windows NT 4.0 supports maximum of 7.8-GB system partition" (http://support.microsoft.com/kb/q224526/en-us). Support.microsoft.com. 2007-02-23. Retrieved 2013-07-30.
3. ^ Steunebrink, Jan. "The BIOS IDE Harddisk Limitations" (http://archive.is/fg442#LBA). Archived from the original (http://web.inter.nl.net/hcc/J.Steunebrink/bioslim.htm#LBA) on 6 October 2013. Retrieved 6 October 2013.
4. ^ Though CHS addressing definitely uses the mathematical concept of **tuple**, it may also be considered an example of the general scheme called mixed radix by viewing its cylinders, heads and sectors as having different numerical bases; e.g., cylinders counting from 0 to 1023, heads from 0 to 254 and sectors from 1 to 63.

External links

- LBAs explained (http://www.dewassoc.com/kbase/hard_drives/lba.htm)
- LBA and CHS format, LBA mapping (http://www.boot-us.com/gloss11.htm)

- CHS to LBA Translation Tutorial (http://viralpatel.net/taj/tutorial/chs_translation.php)
- CHS/LBA conversion utility (http://homepage2.nifty.com/cars/misc/chs2lba.html)
- Microsoft article on 7.8GB limit on NT 4.0 (http://support.microsoft.com/kb/q224526/)
- Hard Drive Size Limitations and Barriers (http://www.dewassoc.com/kbase/hard_drives/drive_size_barrier_limitations_2.htm)
- Upgrading and Repairing PC's (http://books.google.com/books?id=E1p2FDL7P5QC&pg=RA1-PA527&lpg=RA1-PA527&dq=bit+shifting+lba&source=bl&ots=M1poB75_cv&sig=3q6B9DsqOj08xkvE0iE6iiake44&hl=en&ei=HuABS5u6DpPNngefzYwR&sa=X&oi=book_result&ct=result&resnum=10&ved=0CEAQ6AEwCQ#v=onepage&q=bit%20shifting%20lba&f=false), by Scott Mueller. Pages 524–531.

Retrieved from "http://en.wikipedia.org/w/index.php?title=Logical_block_addressing&oldid=575979831"
Categories: Computer storage devices | SCSI | AT Attachment | BIOS