

US00RE42397E

(19) **United States**
(12) **Reissued Patent**
Ban et al.

(10) **Patent Number:** **US RE42,397 E**
(45) **Date of Reissued Patent:** **May 24, 2011**

(54) **ARCHITECTURE FOR A UNIVERSAL SERIAL BUS-BASED PC FLASH DISK**

(75) Inventors: **Amir Ban**, Ramat-Hasharon (IL); **Dov Moran**, Kfar-Saba (IL); **Oron Ogdan**, Jerusalem (IL)

(73) Assignee: **Sandisk IL Ltd.**, Kfar Saba (IL)

(*) Notice: This patent is subject to a terminal disclaimer.

(21) Appl. No.: **10/293,986**

(22) Filed: **Nov. 14, 2002**

Related U.S. Patent Documents

Reissue of:

(64) Patent No.: **6,148,354**
Issued: **Nov. 14, 2000**
Appl. No.: **09/285,706**
Filed: **Apr. 5, 1999**

U.S. Applications:

(63) Continuation of application No. 10/292,868, filed on Nov. 13, 2002.

(51) **Int. Cl.**
G06F 12/00 (2006.01)

(52) **U.S. Cl.** **710/301**; 711/115

(58) **Field of Classification Search** 711/115;
710/104, 301, 302, 313, 2, 8-10
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,067,105 A * 11/1991 Borkenhagen et al. 711/202
5,226,168 A 7/1993 Kobayashi et al. 395/800
5,291,584 A 3/1994 Challa et al.

5,297,148 A 3/1994 Harari et al.
5,341,330 A 8/1994 Wells et al. 365/185
5,375,243 A 12/1994 Parzych et al.
5,388,083 A * 2/1995 Assar et al. 365/185.33
5,404,485 A * 4/1995 Ban 711/202
5,412,798 A 5/1995 Garney 395/500
5,420,412 A 5/1995 Kowalski
5,459,850 A 10/1995 Clay et al. 395/497.02
5,491,774 A 2/1996 Norris et al.
5,509,134 A 4/1996 Fandrich et al. 395/430
5,519,843 A 5/1996 Moran et al.

(Continued)

FOREIGN PATENT DOCUMENTS

CN 1201235 A 12/1998

(Continued)

OTHER PUBLICATIONS

“Generic Device Driver for Personal Computer Removable Devices”, IBM Technical Disclosure Bulletin, Jan. 1994, vol. 37, issue 1, pp. 487-490.*

(Continued)

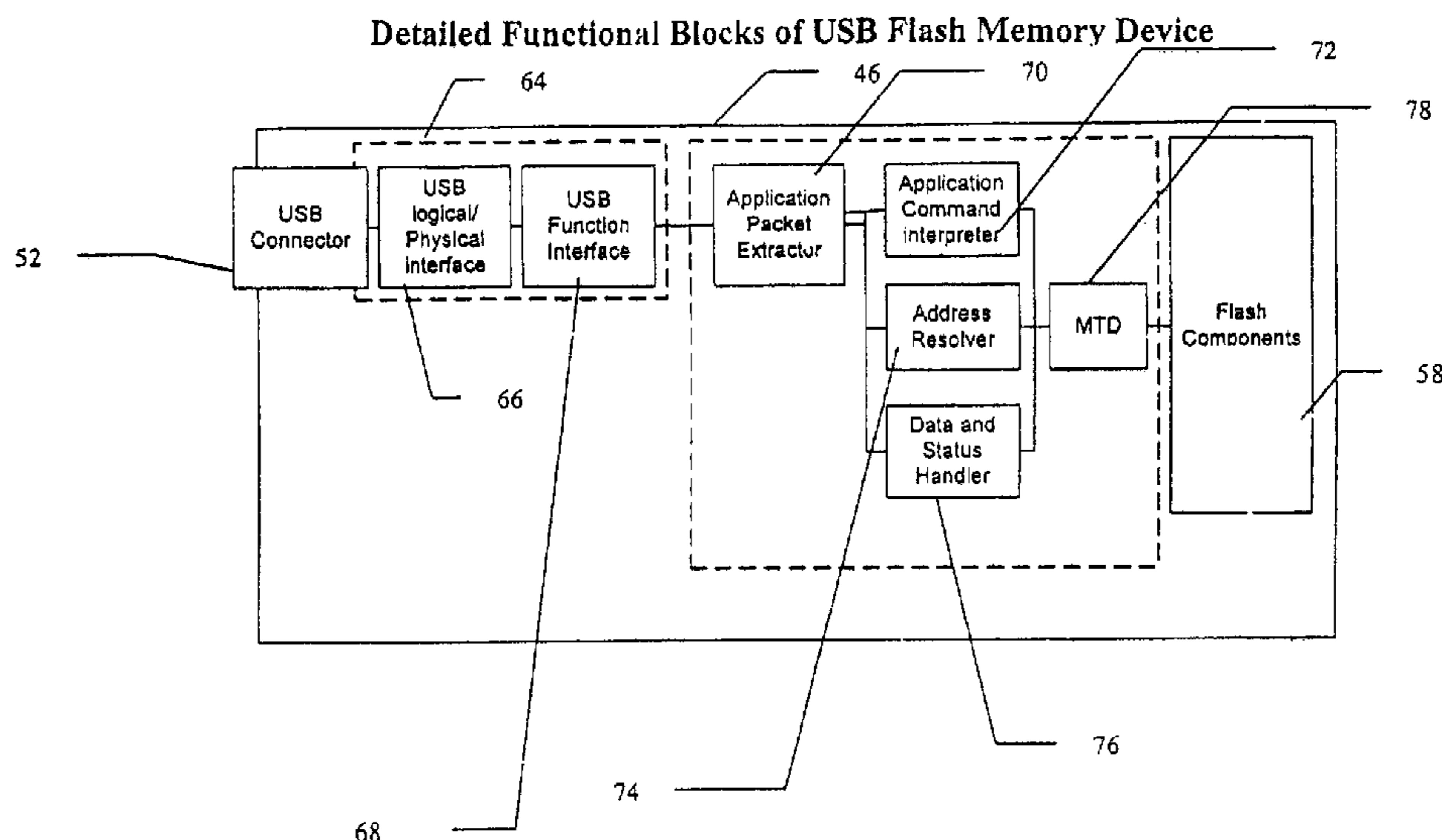
Primary Examiner — Glenn A Auve

(74) *Attorney, Agent, or Firm* — Morgan, Lewis & Bockius LLP

(57) **ABSTRACT**

A storage unit made of flash array and a USB controller[,] is implemented to be compatible with [then] the USB specification. The unit includes memory modules which can accept write commands and read commands and are erasable and non-volatile herein referred to as flash modules. The USB/flash controller is configured to provide USB functionality and compatibility [alone] along with common flash operations such as programming reading and erasing the above mentioned components.

69 Claims, 7 Drawing Sheets



US RE42,397 E

Page 2

U.S. PATENT DOCUMENTS

5,524,230 A 6/1996 Sakaue et al. 395/430
 5,532,945 A * 7/1996 Robinson 713/321
 5,535,357 A 7/1996 Moran et al.
 5,544,356 A 8/1996 Robinson et al. 395/600
 5,546,402 A 8/1996 Niiijima et al. 371/10.2
 5,581,723 A 12/1996 Hasbun et al. 395/430
 5,588,146 A 12/1996 Leroux
 5,602,987 A 2/1997 Harari et al.
 5,630,093 A 5/1997 Holzhammer et al. 395/442
 5,659,705 A 8/1997 McNutt et al.
 5,661,677 A 8/1997 Rondeau, II et al.
 5,663,901 A 9/1997 Wallace et al.
 5,684,742 A 11/1997 Bublitz et al.
 5,719,808 A 2/1998 Harari et al. 365/185.33
 5,724,285 A 3/1998 Shinohara 365/185.25
 5,732,092 A 3/1998 Shinohara 371/40.2
 5,745,418 A 4/1998 Ma et al. 365/185.33
 5,760,986 A 6/1998 Morehouse et al.
 5,774,744 A 6/1998 Story et al. 395/847
 5,778,418 A 7/1998 Auclair et al.
 5,781,028 A 7/1998 Decuir 326/30
 5,784,581 A 7/1998 Hannah
 RE35,881 E 8/1998 Barrett et al. 707/1
 5,799,168 A * 8/1998 Ban 711/103
 5,815,426 A 9/1998 Jigour et al.
 5,822,251 A 10/1998 Bruce et al.
 5,841,424 A 11/1998 Kikinis
 5,845,151 A 12/1998 Story et al. 395/847
 5,845,313 A 12/1998 Estakhri et al.
 5,845,332 A 12/1998 Inoue et al.
 5,847,997 A 12/1998 Harada et al.
 5,860,124 A 1/1999 Matthews et al. 711/165
 5,867,417 A 2/1999 Wallace et al. 365/52
 5,878,142 A 3/1999 Caputo et al.
 5,890,016 A 3/1999 Tso
 5,928,347 A 7/1999 Jones
 5,928,370 A 7/1999 Asnaashari 714/48
 5,935,244 A 8/1999 Swamy et al.
 5,937,423 A 8/1999 Robinson 711/103
 5,937,425 A 8/1999 Ban
 5,938,750 A 8/1999 Shaberman
 5,943,692 A 8/1999 Marberg et al. 711/203
 5,963,983 A 10/1999 Sakakura et al. 711/202
 5,974,486 A 10/1999 Siddappa 710/53
 5,991,194 A 11/1999 Jigour et al. 365/185.04
 5,991,546 A 11/1999 Chan et al. 395/882
 6,003,135 A 12/1999 Bialick et al.
 6,009,480 A * 12/1999 Pleso 710/8
 6,011,486 A 1/2000 Casey
 6,011,741 A 1/2000 Wallace et al.
 6,012,103 A * 1/2000 Sartore et al. 710/8
 6,016,530 A 1/2000 Auclair et al.
 6,016,553 A 1/2000 Schneider et al.
 6,028,807 A * 2/2000 Awsienko 365/230.02
 6,038,320 A 3/2000 Miller
 6,038,640 A 3/2000 Terme
 6,044,428 A * 3/2000 Rayabhari 710/129
 6,058,441 A 5/2000 Shu
 6,067,625 A 5/2000 Ryu
 6,069,827 A * 5/2000 Sinclair 365/185.29
 6,081,850 A * 6/2000 Garney 710/15
 6,088,755 A * 7/2000 Kobayashi et al. 710/300
 6,102,103 A 8/2000 Zobel et al. 165/10
 6,109,939 A 8/2000 Kondo et al.
 6,131,141 A 10/2000 Ravid
 6,137,710 A 10/2000 Iwasaki et al.
 6,145,045 A 11/2000 Falik et al.
 6,145,046 A 11/2000 Jones
 6,151,657 A 11/2000 Sun et al.
 6,163,816 A 12/2000 Anderson et al.
 6,170,743 B1 1/2001 Okaue et al.
 6,182,162 B1 * 1/2001 Estakhri et al. 710/11
 6,199,122 B1 3/2001 Kobayashi
 6,216,230 B1 4/2001 Rallis et al.
 6,226,202 B1 5/2001 Kikuchi
 6,253,300 B1 6/2001 Lawrence et al.
 6,279,069 B1 8/2001 Robinson et al.
 6,279,114 B1 8/2001 Toombs et al.

6,286,087 B1 9/2001 Ito et al.
 6,292,863 B1 * 9/2001 Terasaki et al. 710/313
 6,330,624 B1 12/2001 Cromer et al.
 6,330,648 B1 12/2001 Wambach et al. 711/163
 6,361,369 B1 3/2002 Kondo et al.
 6,370,603 B1 4/2002 Silverman et al.
 6,385,667 B1 5/2002 Estakhri et al.
 6,418,501 B1 * 7/2002 Gama et al. 710/305
 6,424,524 B2 * 7/2002 Bovio et al. 361/686
 6,425,084 B1 * 7/2002 Rallis et al. 713/185
 6,434,648 B1 8/2002 Assour et al.
 6,453,414 B1 9/2002 Ryu 713/2
 6,457,099 B1 9/2002 Gilbert
 6,488,542 B2 12/2002 Laity
 6,493,770 B1 * 12/2002 Sartore et al. 710/8
 6,671,808 B1 * 12/2003 Abbott et al. 713/200
 6,763,399 B2 7/2004 Margalit et al.
 6,920,553 B1 7/2005 Poisner 713/2
 2003/0057285 A1 3/2003 Okaue et al.
 2004/0039854 A1 2/2004 Estakhri et al.
 2006/0230202 A1 10/2006 Lee 710/62

FOREIGN PATENT DOCUMENTS

DE 195 36 206 4/1996
 DE 196 36 087 9/1997
 DE 196 31 050 2/1998
 EP 0 392 895 A2 10/1990
 EP 0 152 024 6/1991
 EP 0 703 544 3/1996
 EP 0 712 067 A2 5/1996
 EP 0 859 325 8/1998
 EP 0 883 083 12/1998
 EP 0 883 083 A1 12/1998
 EP 0 883 084 12/1998
 EP 0 890 905 A2 1/1999
 EP 0 929 043 7/1999
 EP 1 001 329 5/2000
 EP 0 912 939 9/2001
 EP 0 775 956 5/2002
 FR 2 719 939 11/1995
 GB 2 291 990 A 2/1996
 GB 2 304 428 A 3/1997
 GB 2 325 997 A 12/1998
 JP 1 115928 A 5/1989
 JP 5 016746 A 1/1993
 JP 8 171623 A 7/1996
 JP 09-069067 11/1997
 JP 10 063442 A 3/1998
 JP 10 063804 A 3/1998
 JP 10 105296 A 4/1998
 JP 10 261774 A 9/1998
 JP 11 015928 A 1/1999
 JP 11 025681 A 1/1999
 JP 11-53485 2/1999
 JP 11 053485 A 2/1999
 JP 11 086578 A 3/1999
 WO WO 87/07063 11/1987
 WO WO 93/19419 A1 9/1993
 WO WO 96/13004 5/1996
 WO WO 98/03915 1/1998
 WO WO 98/03915 A2 1/1998
 WO WO 98/07255 2/1998
 WO WO 98/29830 7/1998
 WO WO 98/55912 12/1998
 WO WO 99/01820 A1 1/1999
 WO WO 99/12101 3/1999
 WO WO 99/45460 9/1999
 WO WO 99/45460 A2 9/1999
 WO WO 99/49470 A1 9/1999
 WO WO 00/07088 2/2000
 WO WO 00/42491 7/2000
 WO WO 00/49488 A1 8/2000
 WO WO 03/030180 A1 4/2003

OTHER PUBLICATIONS

Universal Serial Bus Specification—Compaq, Intel, Microsoft, NEC; Revision 1.1; Sep. 23, 1998; pp. i-xvi (table of content), 1-311, + 4 pages (Engineering Change Notice—USB Cable Parameters, approved on Nov. 23, 1999 by the USB Core Team).

- Universal Serial Bus Mass Storage Class Specification Overview; V1.0 Revision; Oct. 22, 1998; Technical Editor—Williams, M.; pp. 1-8.
- Universal Serial Bus Mass Storage Class—Control/Bulk/Interrupt (CBI) Transport; Revision 1.0; Dec. 14, 1998; Technical Editors—Stevens, C. E. at al.; pp. 1-26.
- Universal Serial Bus Mass Storage Class—Bulk-Only Transport; Revision 1.0; Sep. 31, 1999; pp. 3-22.
- Universal Serial Bus Mass Storage Class—UFI Command Specification; Revision 1.0; Dec. 14, 1998; pp. 1-12.
- Advertising—“MacHASP USB—Protects Your Software;” Aladdin—The Professional Choice; 2 pages.
- Sharp Model OZ-730/OZ-750 Electronic Organizer—Operation Manual; pp. 1-71 + 11 pages Quick Reference Guide + 10 pages Quick Start Guide.
- Sharp Wizard, Model OZ-570 Electronic Organizer—Operation Manual; pp. 1-101 + Quick Reference Guide 17 pages.
- Sharp Wizard, IntelliSync for Sharp Wizard—User’s Guide; Day Time Organizer for Sharp Electronic Organizer with IntelliSync Translator; pp. 1-33 + 2 pages Installation Instructions + pp. 1-36 Getting Started Guide, Copyright 1997.
- Sharp PC Interface Hardware (Docking Station), Model CE-619—Operation Manual; 5 pages, Copyright 1197.
- “Digital Eyes News—Q3 1998;” Scanner and Digital Camera News, Q3 1998—Digital Eyes; Retrieved from the Internet at <<http://www.image-acquire.com/news/98/q3.shtml>> pp. 1-9.
- “Imaging Press Release—Hagiwara ‘Flash Gate’ for SmartMedia up to 130X Faster than Serial Devices;” 2 pages; Digital Eyes News; Retrieved from the Internet at <<http://www.image-acquire.com/news/49/q2/hagi9921.html>>, Apr. 19, 1999.
- “Product Review—Hagiwara Flash Gate USB;” Hagiwara Flash Gate USB Review from Digital Eyes; 2 pages; Retrieved from the Internet at <<http://www.image-acquire.com/articles/storage/hagiwarausb.html>>.
- “Product Review—Action Tec CameraConnect Pro;” Action Tec Camera Connect Review from Digital Eyes; 3 pages; Retrieved from the Internet at <<http://www.image-acquire.com/articles/storage/ateamconnect.shtml>>.
- “Product Review—SanDisk ImageMate USB;” SanDisk ImageMate USB Review from Digital Eyes; 2 pages; Retrieved from the Internet at <<http://www.image-acquire.com/articles/storage/sdimagemateusb.shtml>>.
- “FRCN Digital Imaging—SanDisk ImageMate USB CF Card Reader;” Updated Jul. 12, 1999; Retrieved from the Internet at <<http://www.quicknet.com/~frcn/SanDiskUSB.html>>.
- “SanDisk Introduces New USB ImageMate; SanDisk ImageMate is a Low-cost, Easy-To-Use External CompactFlash Drive for Desktop PCs and the New Apple iMac;” Business Wire; Oct. 27, 1998; Document ID FC19981027080000045; 3 pages; Retrieved from the Internet at <<http://library.northernlight.com/FC19981027...JqDmQPWAVRU0UIGRRedQx5HQ%3D%3D&cbx=0%3B1004>>.
- ActionTec CameraConnect Pro, Parallel Port Flash Card Reader—User’s Manual; pp. 1-29.
- USB Drive: Products; 2 pages; Retrieved from the Internet at <<http://web.archive.org/web/20020328072516/http://www.usbdive.com/products/index.html>>.
- “Steve’s DigiCams, Hardware Reviews—Camera Connect Pro, External PC Card Read/Write Drive;” 4 pages; Retrieved from the Internet at <http://www.steves-digicams.com/camera_connect_pro.html>, Copyright 1998.
- Actiontec—Broadband Technology Made Easy; “Support Info—CameraConnect Pro;” 2 pages; Retrieved from the Internet at <<http://www.actiontec.com/support/readers/cameraconpro.html>>.
- Brochure—“USB Special, Aladdin—the Key to Software and Security;” Aladdin Knowledge Systems Inc.; 8 pages.
- HP Jornada PC Companions; Software & Hardware Solutions—Microsoft Windows CE 2.11; 2 pages; Retrieved from Internet at <http://web.archive.org/web/19991127141058/www.hp.com/jornada/sol/.../othsoft_rom.htm>.
- HP Jornada PC Companions; Jornada 430se—the Palm-Size PC that Meets Your Needs at Work and Play; 3 pages; Retrieved from the Internet at <<http://web.archive.org/web/19991128125415/http://www.hp.com/jornada/pr.../overview.htm>>.
- HP Jornada Pc Companions; Product Specifications—HP Jornada 430se; 5 pages; Retrieved from the Internet at <http://web.archive.org/web/19991128142035/www.hp.com/jornada/produ.../prod_spec.htm>.
- FujiFilm—Image Memory Card Reader SM-R1 for SmartMedia; Owner’s Manual; 16 pages.
- (BW) (NY-Fuji-Photo-Film) Introducing the Fujifilm MX-2700: The World’s Smallest 2.3 Million-Pixel Digital Camera; Feb. 3, 1999; 2 pages; Retrieved from the Internet at <<http://www.businesswire.com/webbox/bw.020399/1096161.htm>>.
- Fuji film SM-R1 Memory Card Reader; TwoMobile Home of the Investigative Mobile Reviewers; 3 pages; Retrieved from the Internet at <http://web.archive.org/web/20001206024200/http://www.twomobile.com/rev_smr1reader.html>.
- Windows CW Handheld PC Hardware Features; 1 page; Retrieved from the Internet at <<http://web.archive.org/web/19991012094107/microsoft.com/windowsce/Products/hpc/hwspecs.asp>>.
- FujiFilm—Image Memory Card Reader SM-R1 for Windows 98 / Macintosh; Owner’s Manual; 21 pages in English; 21 pages in French; 21 pages in German.
- Universal Serial Bus Mass Storage Class—UFI Command Specification; Revision 1.0; Dec. 14, 1998; pp. 1-53.
- “What is Memory Stick—Shape;” 1 page; Retrieved from the Internet at <http://www.memorystick.org/topics/eng/aboutms/feat_outline.html>.
- Snapshots of Windows ’98 Computer Screen, with “Plug and Play PCMCIA Card”; 4 pages.
- USB-SCSI & ATA/ATAPI Intelligent Cables; 1 page; Retrieved from the Internet at <http://www.multiwave.co.kr/product/scm_product3.html>.
- “USB ATAPI Intelligent Cables;” SCM Microsystems; 2 pages; Retrieved from the Internet at <<http://www.scmmicro.com/connectivity/usbata.html>>, Copyright 1999.
- “USB XpressSCSI—USB to SCSI Converter Cable;” User’s Manual for Models USB-SCSI-DB25 and USB-SCSI-HD50; Version 1.0; Microtech; 37 pages; Copyright 1999.
- “Availability of Universal Serial Bus Support in Windows ’95;” 2 pages; Retrieved from the Internet at <<http://support.microsoft.com/default.aspx?scid=kb;en-us;253756>>.
- Paripatyadar, R.; “1394 Overview;” 19 pages; Nov. 10, 1998; Retrieved from the Internet at <http://grouper.ieee.org/groups/802/802_tutorials/nov98/139411_1198.pdf>.
- Product Brief—AlphaPC 164LX Motherboard; Alpha Processor; Jan. 1999; 2 pages; Retrieved from the Internet at <<http://www.alphalinux.org/firmware/api/LX/164LX.PDF>>.
- SOYO SY-6IEB Mainboard—Quick Start Guide; Version 1.0; Sep. 1998; 14 pages; Retrieved from the Internet at <<http://www.soyola.com/dl/manuals/quick-sg/q6ieb10.pdf>>, Copyright 1995.
- PCMCIA SRAM & Linear Flash Cards, by C1-Tech, Simple Technology, Centennial, Intel and KingMax; 5 pages; Retrieved from the Internet at <<http://www.psism.com/sram.htm>>.
- “Interfacing the Intel Flash 28F001BX-T to Your 186 Based System;” Intel; Mar. 29, 1998; 14 pages; Retrieved from the Internet at <<http://developer.intel.com/design/intarch/applnots/2150.htm>>.
- CY7C63000, CY7C63001, CY7C63100, CY7C63101, CY7C63200, CY7C63201—Universal Serial Bus Microcontroller; Cypress Semiconductor Corp.; 27 pages; Jun. 26, 1997; Retrieved from the Internet at <<http://www.physik.uni-osnabrueck.de/kbetzler/win32/usb/7c63000.pdf>>.
- Wong, T. K. P.; “An Embedded Chip for USB Application: From the Architecture to Implementation;” Apr. 1999; International IC ’99—Conference Proceedings; 10 pages; Retrieved from the Internet at <http://www.eetasia.com/ARTICLES/1999APR/1999APRO8_ICD_NTEK_TAC.PDF>.
- Gupta, R. K. et al.; “Introducing Core-Based System Design;” Oct.-Dec. 1997; IEEE; pp. 15-25; Retrieved from the Internet <<http://www.ics.uci.edu/~rgupta/publications/cores.pdf>>.
- Intel AP-686—Application Note; “Flash File System Selection Guide;” Dec. 1998; 16 pages; Retrieved from the Internet at <<http://www.intel.com/design/flcomp/applnots/29766505.pdf>>.

- Intel AP-684 —Application Note; “Understanding the Flash Translation Layer (FTL) Specification;” Dec. 1998; 20 pages; Retrieved from the Internet at <<http://www.intel.com/design/flcomp/applnots/297816.htm>>.
- VxWorks 5.4; WindRiver Systems; 3 pages; Retrieved from the Internet at <<http://www.vxworks.ru/products/pdf/vxworks-ds.pdf>>, Copyright 1999.
- Apple USB FAQ; 1999; 3 pages; Retrieved from the Internet at <http://www.apple.com/usb/pdf/usb_FAQ-c.pdf>, Copyright 1999.
- Multimedia Workshop Industry News—Oct. 22, 1998; 3 pages; Retrieved from the Internet at <<http://www.m2w.net/News/981019.html>>.
- Davis, J.; “Apple Licensing FireWire for a Fee;” CNET News.Com; Jan. 16, 1999; 4 pages; Retrieved from the Internet at <<http://news.com.com/2100-1040-220209.html>>.
- “Will IEEE 1394 Ultimately Replace USB?;” White Paper Comparing Universal Serial Bus (USB) and IEEE 1394 (Fire Wire) Technologies; American Megatrends, Inc.; 7 pages; Retrieved from the Internet at <http://www.ami.com/support/doc/1394_USB.pdf>.
- WindRiver Systems—WindWord Newsletter; “News and Information for Wind River Systems Customers;” Winter 1998; 32 pages; Retrieved from the Internet at <<http://www.windriver.com>>.
- PC/104 PCMCIA Module—User’s Manual; for models PCM-3110, PCM-3111, PCM-3115B, PCM-3110B, PCM-3113, PCM-3114; 1st Edition; Printed Jul. 1997; 78 pages; Retrieved from the Internet at <<http://www.tempustech.com/pdf/3110manual.pdf>>.
- 8X930Ax Universal Serial Bus Microcontroller User’s Manual; Intel; Jul. 1996; 507 pages; Retrieved from the Internet at <<http://www.mit.edu/afs/sipb/contrib/doc/specs/ic/bridge/i8x930ax.pdf>>.
- “Reviews—Pretec PCMCIA Type I SRAM 2048KB;” EverythingLinux; 1 page; Retrieved from the Internet at <<http://www.everythinglinux.com.au/item/SRAM2M?elinux=e0c72135e544a6ea13772f5c23ba3foe>>.
- “U2 Only Disk User’s Manual;” Netae Technology Co.; Retrieved from the Internet at <<http://www.bestusbdrive.com/downloads/u2english.pdf>>; 1 Page.
- Shmueli, S.; Description of “SanDisk ImageMate USB CompactFlash Reader”; 3 Pages; Date of Introduction of device unknown.
- Shmueli, S.; Description of “Lour Jumpshot Cable”; 2 Pages; Date of introduction of device unknown.
- Shmueli, S.; Description of “Aladdin eToken and Rainbow iKey”; 1 Page; Date of introduction of device unknown.
- Lee Kong Pheng; Description of “SanDisk ImageMate USB CF Card Reader”; 1 Page; Date of Introduction of device unknown.
- Lee Kong Pheong+; Comparison of the block diagrams of the SanDisk ImageMate USB CF Card Reader, the Patent and the SlimDisk; 6 Pages; Date of introduction of device unknown.
- “8+930Hx Universal Serial Bus Microcontroller—Product Review;” Intel; Oct. 1996; 34 Pages.
- “IBM Smart Card Solution Elements—Overview;” Jul. 1997; 26 Pages.
- “MacHasp USB—Software Protection via the USB;” Aladdin—The Professional Choke; 10 Pages, Brochure.
- Roberts, R.; “NCITS-330 T10/1240-D Working Draft—Information Technology—Reduced Block Commands”; Revision 10a; Aug. 18, 1999; 46 Pages.
- Shmueli, S. Testimony relevant to Sandisk in Singapore proceedings; Apr. 30, 2004; line 9 in p. 135 through line 23 in p. 137 and line 8 in p. 139 through Line 18 in p. 140.
- “SL11R USB Controller / 16-Bit RISC Processor Data Sheet;” Cypress Semiconductor Corporation; Revised Dec. 3, 2001; pp. 1-85.
- “What is Memory Stick?;” Retrieved from the Internet at <<http://www.memorystick.org/topics/eng/aboutms>>; 15 Pages.
- “Memory Stick Format;” Retrieved from the Internet at <<http://www.memorystick.org/mseasy/eng/application/>>; 19 Pages.
- U.S. Appl. No. 60/116,006 filed Jan. 15, 1999; Abbott, S. D. et al.; “USB-Compliant Personal Key”.
- “SanDisk ImageMate CompactFlash External Drive;” Product Backgrounder, SanDisk Corporation; Jan. 1999; retrieved from the Internet at <<http://web.archive.org/web/20000229014945/www.sandisk.com/cons/image-USB.htm>>; 2 pages.
- “The New WIBU-KEY Copy Protection System v2.50;” Press Release; Griffin Technologies; copyright 1998; retrieved from the Internet at <http://web.archive.org/web/19990128201628/www.griftech.com/html/wibu25pr.htm><; 3 pages.
- Anderson, D.; “Universal Serial Bus System Architecture”; 1997; MindShare, Inc./Addison-Wesley, 6th Printing, pp. 95, 98-104 and 109-111.
- IC Card Expo; Conference Proceedings; Jul. 25-27, 1994; Santa Clara Convention Center, Santa Clara, CA, USA; 54 pages.
- Mendelsohn, A.; “Solid State Disk Puts Flash to Work;” Feb. 1994; Computer Design; 4 pages.
- “Word-Wide FlashFile™ Memory Family 28F160S5, 28F320S5;” “Advance information”, Order No. 290609-001, of Intel, entitled; Jun. 1997.
- “Aladdin’s eToken cracked;” May 5, 2000; retrieved from the internet at <<http://www.securiteam.com>>.
- Burskey, D.; “Flash and EEPROM technologies combine on feature-rich MCUs;” Electronic Design; May 26, 1997; retrieved from the internet at <http://cma.zdnet.com>; 11 pages.
- “Flash memory;” on-line encyclopedia Wikipedia; retrieved from the internet at <http://en.wikipedia.org/wiki/Flash_memory>; 5 pages.
- Schmidt, D.; “AP-DOC-011 Application Note—Write Protecting the DiskOnChip®2000;” M-Systems Flash Disk Pioneers; Aug. 1997; 4 pages.
- “User Manual—DiskOnChip®2000 PIK (Programmer and Integrator Kit);” M-Systems Flash Disk Pioneers; Jul. 1997; 7 pages.
- “DiskOnChip®2000 Quick Installation Guide;” M-Systems Flash Disk Pioneers; 1 page.
- “User Manual—DiskOnChip®2000 Utilities;” M-Systems Flash Disk Pioneers; Jul. 1997; 8 pages.
- “User Manual—DiskOnChip®-EVB ISA Adaptor for the DiskOnChip;” M-Systems Flash Disk Pioneers; Nov. 1996; 6 pages.
- “M-Systems’ DiskOnChip®2000 Evaluation Board;” Jul. 1997; 1 page.
- “Questions and Answers about DiskOnChip2000;” M-Systems Flash Disk Pioneers; Jul. 1997; 4 pages.
- Ben-Zeev, Y.; “AP-DOC-010 Application Note—Designing with the DiskOnChip®2000;” M-Systems Flash Disk Pioneers; Jul. 1997; 4 pages.
- “Flash PC Card 2000 Data Sheet;” M-Systems Flash Disk Pioneers; Jun. 1997; retrieved from the internet at <<http://www/m-sys.com/pcmcia.htm>> on or before Oct. 13, 1997; 15 pages.
- “Information Technology—Reduced Block Commands;” Working Draft; T10/1228D Revision 6; Feb. 26, 1999.
- “DiskOnChip 2000 MD-2200 Data Sheet;” M-Systems Flash Disk Pioneers; Jul. 1997; 15 pages.
- “A TrueFFS and FLite Technical Overview of M-Systems’ Flash File Systems;” Technology Brief V1.01; M-Systems Flash Disk Pioneers; Oct. 1996; retrieved from the internet at <<http://www.m-sys.com/tech1.htm>> on or before Oct. 13, 1997; 9 pages.
- “PC Card Standard Release 7;” PCMCIA; vols. 1 through 11; Copyright 1999, update Mar. 2000; retrieved from the internet at <<http://www.pcmcia.org/pccard.htm>>.
- “Universal Serial Bus Mass Storage Class—Bulk-Only Transport;” Revision 1.0 [RC3]; Mar. 29, 1999; 25 pages.
- “DiskOnChip2000—The Smallest Flash Disk in the World;” M-Systems Flash Disk Pioneers; retrieved from the internet at <www.m-sys.com/doc2000.htm> on or before Oct. 13, 1997; 5 pages.
- “8x930Hx Universal Serial Bus Microcontroller;” Intel Corporation; Oct. 1996; retrieved from the Internet at <<http://140.114.28.92/course/33249/FILES/FILES/USING%20MCS51/DATASHTS/intel/ds930hx.pdf>>; 34 pages.
- “Sony Announces Memory Stick™ for Audio Application “Memory Stick™ Walkman” and “MagicGate Memmory Stick™” To Be Launched;” Sep. 22, 1999; retrieved from the internet at <http://www.sony.net/sonyinfo/news/press_archive/199909/99-072/> on Mar. 32, 2006; 3 pages.
- “Report on the Japanese Nikon 900 Firmware Upgrade;” Digital Camera News; Jan. 30, 1999; 34 pages.
- International Standard, ISO/IEC 7816-1 through 7816-8; editions dated Sep. 1, 1995 through Oct. 1, 1999.

- “PCMCIA Flash Disk—Removable Flash Disk for Embedded & Mobile Computers”; Oct. 13, 1997; retrieved from the Internet at <www.m-sys.com/pcmcia.htm>.
- Grant, J.; “A Historical Look At Hardware Token Compromises”; presented in the Black Hat USA 2004 briefing, Wednesday Jul. 28, 2004.
- English translation of FR 2 719 939; published on Nov. 17, 1995.
- “Aladdin Announces eToken—Next Generation Security Key Based On Universal Serial Bus Port”; Jan. 19, 1999; retrieved from the internet at <<http://web.archive.org/web/20040414101246/www.ealaddin.com/news/1999/etoken/etoken.asp>> on Feb. 13, 2006; 3 pages.
- “Hardlock USB”; Aladdin; retrieved from the Internet at <http://aladdin.de/produkte/hardlock/hardlock_usb.html> on Feb. 13, 2006; 2 pages.
- “Reduced Block Commands (RBC)”; Draft Proposal (T10/97-260r2); Revision 1; Jan. 16, 1998.
- “Information Technology—Reduced Block Commands”; Working Draft; T10/1228D Revision 7; Mar. 8, 1999.
- “Flash Memory”; Wikipedia, the free encyclopedia; downloaded from the Internet at <http://en.wikipedia.org/wiki/Flash_Memory> on Feb. 20, 2006; 5 pages.
- European Patent Office Summons To Attend Oral Proceedings in EP 1092193.
- “Application Note (AP-684) Understanding the Flash Translation Layer (FTL) Specification,” Intel Corporation, Dec. 1998, 20 pages.
- “PC Card Standard, vol. 1, Overview and Glossary,” PCMCIA/ JEIDA, first printing Feb. 1999, 40 pages.
- “PC Card Standard, vol. 7, Media Storage Formats Specification,” PCMCIA/ JEIDA, first printing Feb. 1999, 53 pages.
- “PC Card Standard, vol. 8, PC Card ATA Specification,” PCMCIA/ JEIDA, first printing Feb. 1999, 39 pages.
- M-Systems, “FlashLite 100 PC Card,” M-Systems Flash Disk Pioneers, Product Specifications, Jul. 1998.
- Office Action dated Nov. 2, 2007.
- European Patent Office Interlocutory decision in Opposition proceedings dated Jan. 21, 2009.
- Auclair, D., “Optimal Solid State Disk Architecture for Portable Computers,” Presented at the Silicon Valley PC Design Conference, Jul. 9, 1991, 5 pages.
- Dipert et al., “Designing with Flash Memory,” Annabooks © 1993, 1994, San Diego, CA, 431 pages.
- Leibson, S., “Nonvolatile, in-circuit-reprogrammable memories,” EDN Magazine, Jan. 3, 1991, 12 pages.
- Cypress Semiconductor, “Universal Serial Bus Microcontroller,” Cypress Semiconductor Corporation, San Jose, CA, Oct. 1996—Revised Jun. 26, 1997, 35 pages.
- Diamond, “Rio 500 Getting Started Guide,” Diamond Multimedia Systems © 1999, 2 pages.
- Fantom Drives, “Fantom Drives Embraces USB & iMac by Announcing USB Hard Disk Drive and CD-ReWritable Technology,” Fantom USB Highlights, Jan. 5, 1999, 3 pages, http://web.archive.org/web/20000526201655/fantomdrives.com/main_si.
- Fantom Drives, “Fantom Drives Ships USB Hard Disk Drives,” Fantom Drives, Apr. 5, 1999, 1 page, <http://www.fantomdrives.com/press/articles/usbrelease.html>.
- IEEE, “IEEE Standard for a High Performance Serial Bus,” IEEE Std 1394-1995, © 1996 by Institute of Electrical and Electronics Engineers, Inc., New York, NY, 393 pages.
- MacHASP USB, “Software Protection via the USB,” MacHASP USB Datasheet, Aladdin Knowledge Systems, Inc., 10 pages.
- M-Systems, “DiskOnChip®2000 MD2200, MD2201 Data Sheet,” M-Systems Inc., Jun. 1998, 15 pages.
- Philips, “PDIUSB12 USB interface device with parallel bus,” Philips Semiconductors—Asia Product Innovation Centre Data Sheet, Jan. 8, 1999, 45 pages.
- Toshiba, MOS Memory (Non-Volatile) Data Book, Toshiba Corporation, Sep. 1996, 166 pages.
- Toshiba, “Toshiba MOS Digital Integrated Circuit Silicon Gate CMOS,” Toshiba Corporation, Jul. 7, 1997, 34 pages.
- Request for Inter Partes Reexamination of U.S. Patent No. 6,829,672, dated Jul. 8, 2008, with transmittal form, 109 pages.
- Inter Partes Reexamination Communication / Office Action dated Nov. 3, 2008, 22 pages.
- Decision Granting-in-Part Petition for Extension of Time dated Dec. 12, 2008, 4 pages.
- Patent Owner’s Response to Office Action, Feb. 3, 2009, 162 pages.
- Transmittal of Communication to Third Party Requester Inter Partes Reexamination / Decision Refusing Entry of Papers, re paper entitled “Expedited Petition Under 37 C.F.R. §§ 1.182 and 1.183” dated Apr. 6, 2009, 7 pages.
- Transmittal of Communication to Third Party Requester Inter Partes Reexamination / Decision Refusing Entry of Papers re paper entitled “Opposition to Patent Owner’s Petition Under 37 C.F.R. § 1.956,” dated Apr. 6, 2009, 7 pages.
- Transmittal of Communication to Third Party Requester Inter Partes Reexamination / Decision Granting-in-Part Petition to Extend Page Limit for Response to Office Action, dated Apr. 6, 2009, 7 pages.
- Written Comments by the Third Party Requester in an Inter Partes Reexamination, Patent Reexamination Control No. 95/000,384, May 20, 2009, 75 pages (Redacted).
- Claims as issued in U.S. Patent No. 6,829,672, Reexamination Control No. 95/000,384.
- IDS with references, filed Jul. 9, 2008, 73 pages.
- European Patent Office Extended Search Report, issued in co-pending European Patent Application 09013951.0 on Apr. 6, 2010, 8 pages.
- “Written Comments by the Third Party Requester in an Inter Partes Reexamination,” filed in Reexamination Control No. 95/000,384, dated Apr. 13, 2010, 49 pages.
- Austrian Patent Office Search Report and Written Opinion issued in related Singapore Patent Application 200604735-1 on Nov. 16, 2009, 8 pages.
- European Patent Office Extended Search Report issued in co-pending European Patent Application 09010734.3 on Apr. 8, 2010, 9 pages.
- Japanese Patent Office Action issued in co-pending Japanese Patent Application No. 2005-241989, dated Apr. 9, 2010, 8 pages.
- Japanese Patent Office Action issued in co-pending Japanese Patent Application No. 2007-085679, dated Apr. 9, 2010, 6 pages.
- Intel Corp., “Common Flash Memory Interface (CFI) Specification,” Release 1.1, May 30, 1997, 16 pages.

* cited by examiner

Figure 1 -- Background Art USB Token Packet Structure

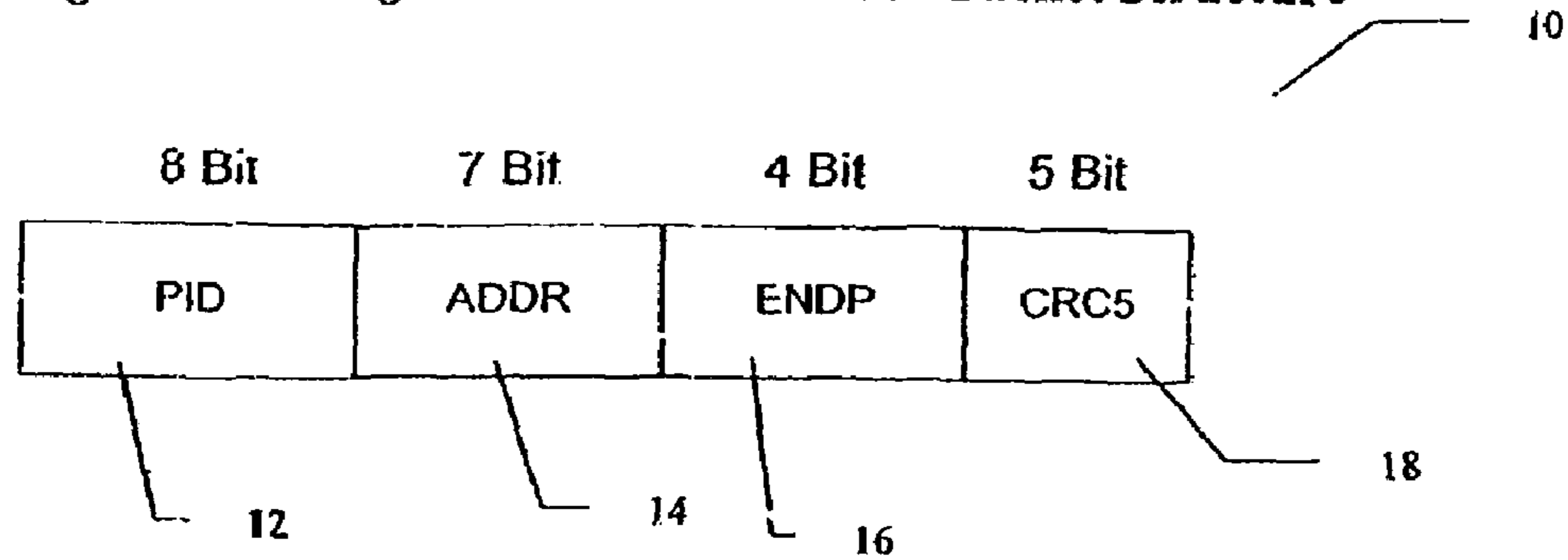


Figure 2 -- Background Art USB Data Packet Structure

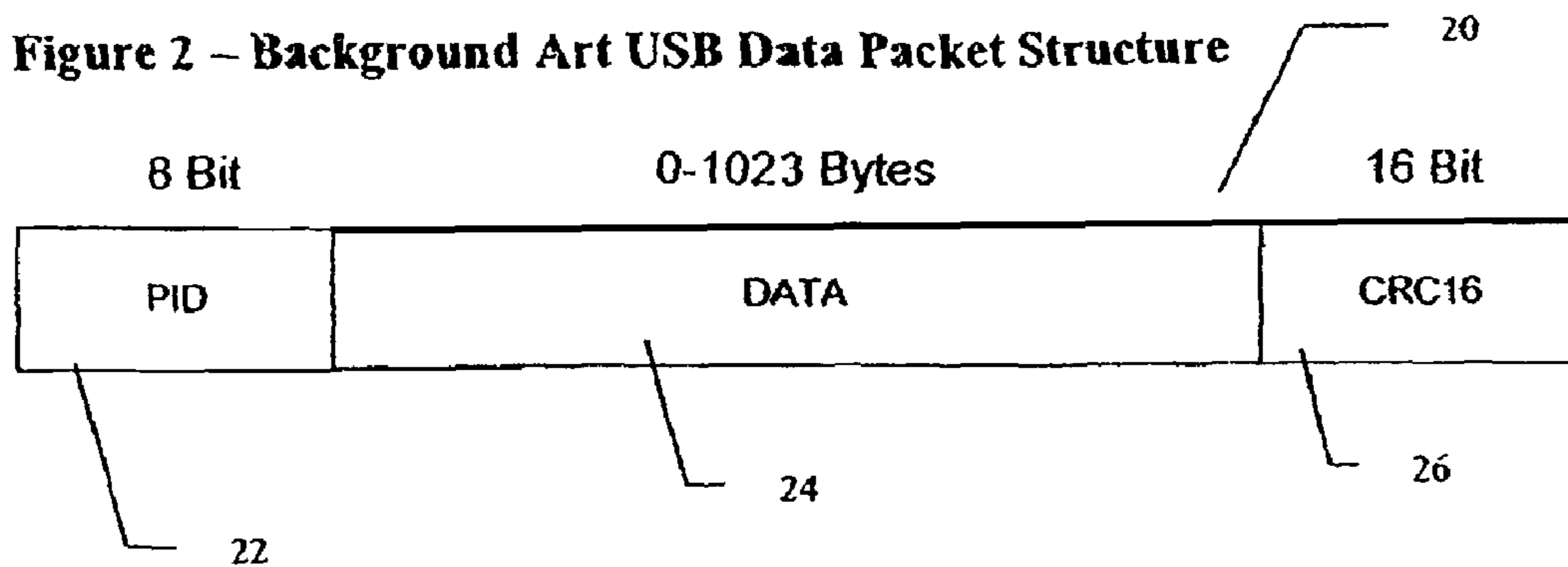


Figure 3 -- Background Art USB Handshake Packet

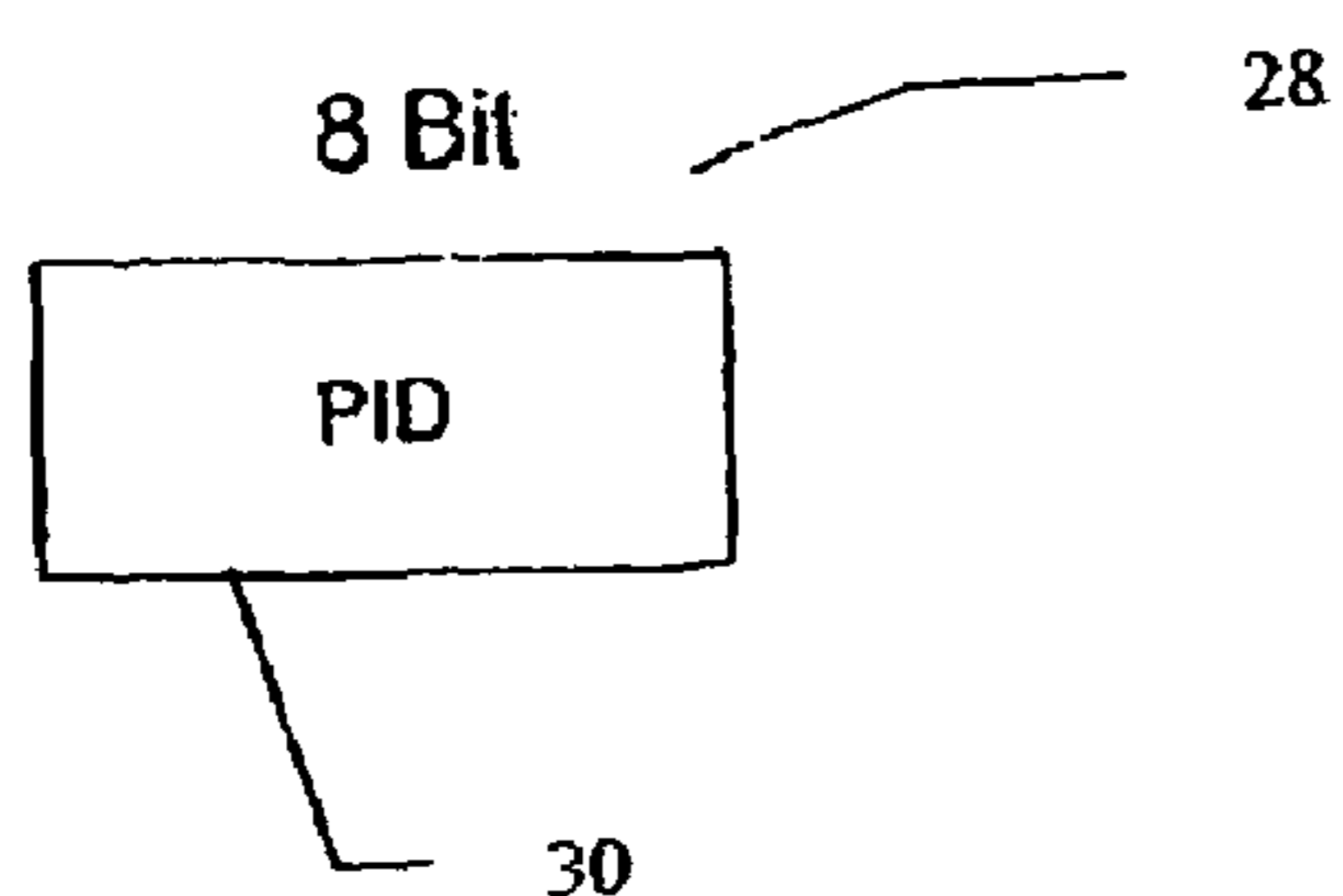
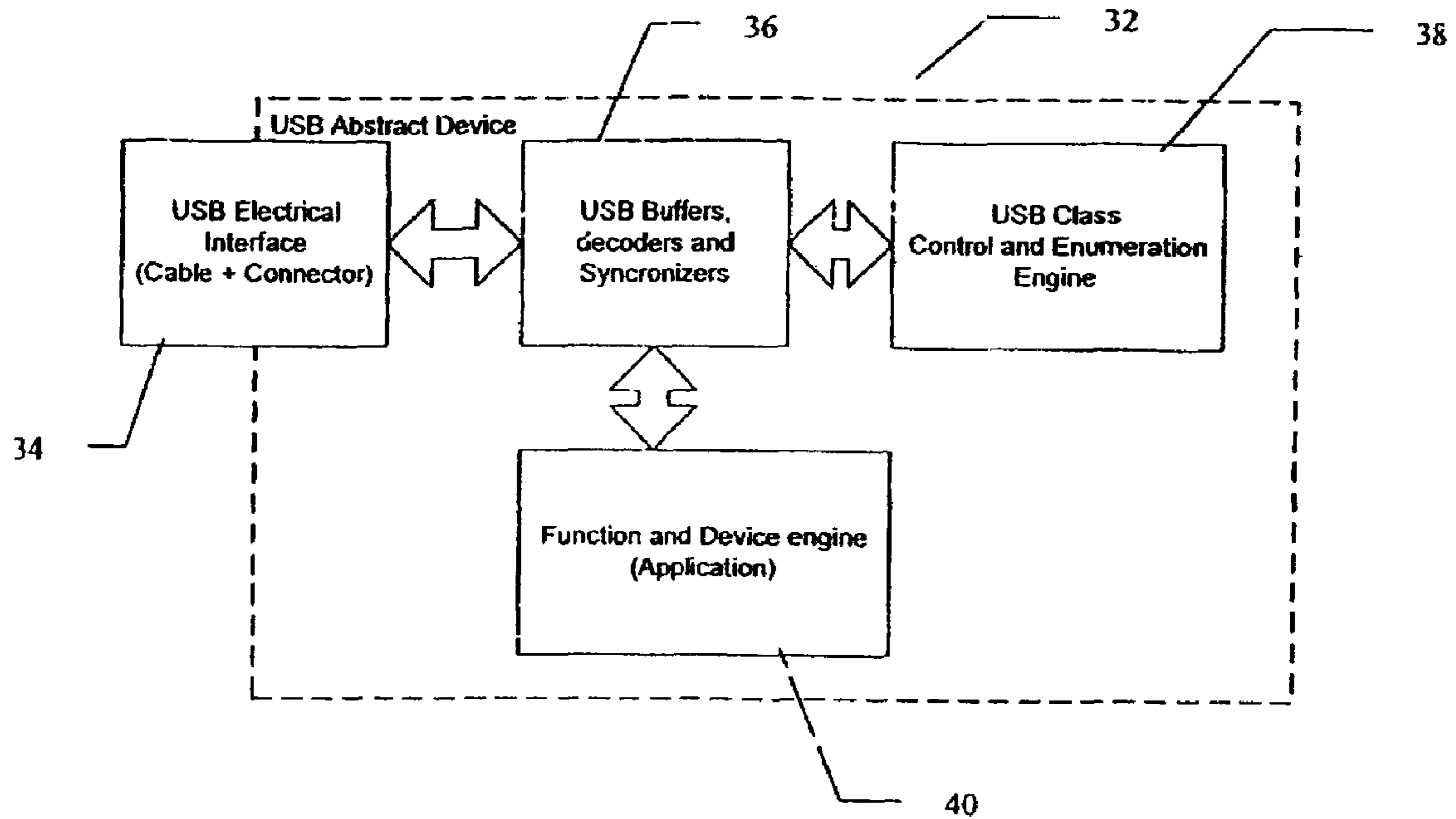
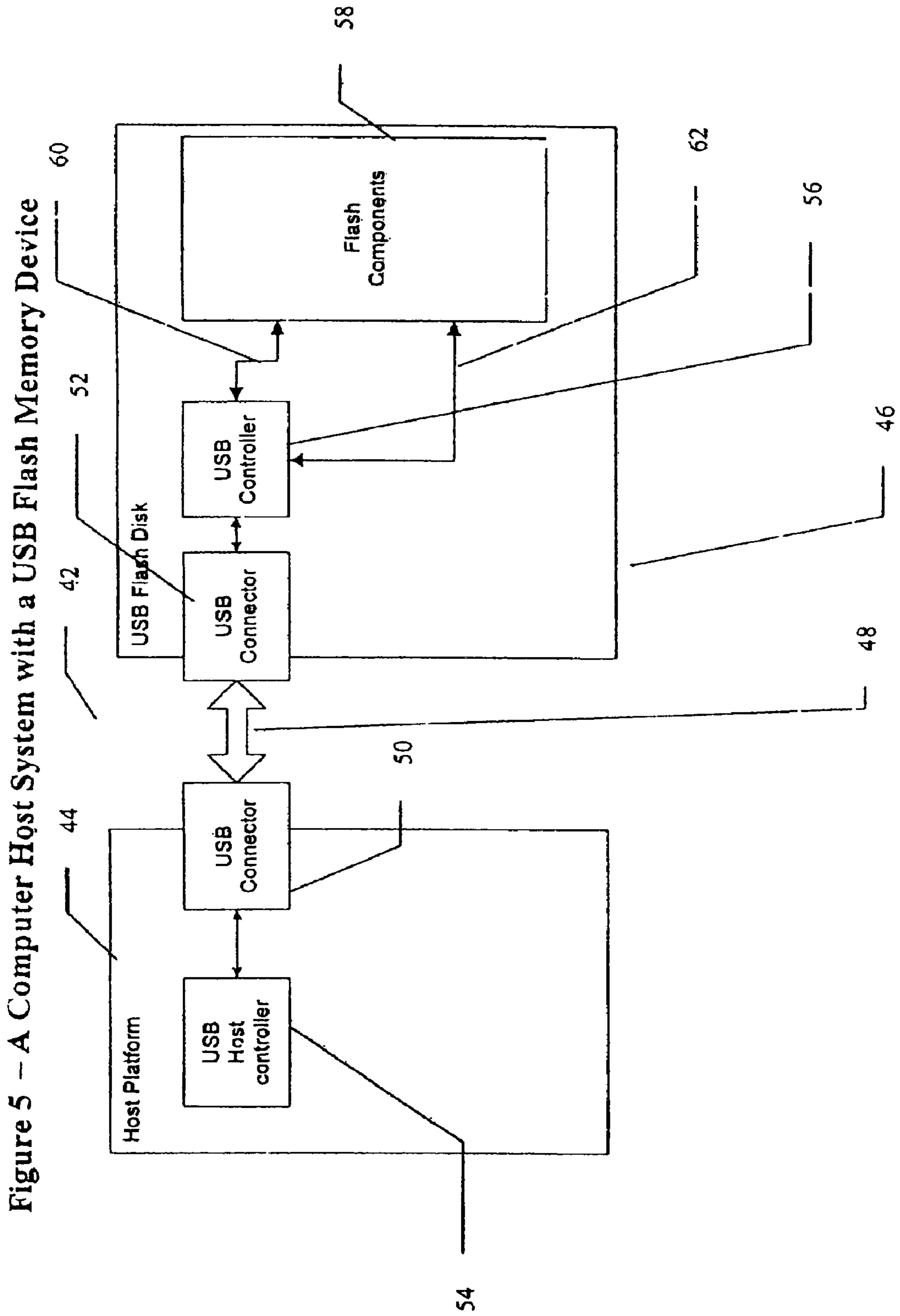


Figure 4 – Functional Blocks of a Background Art USB device





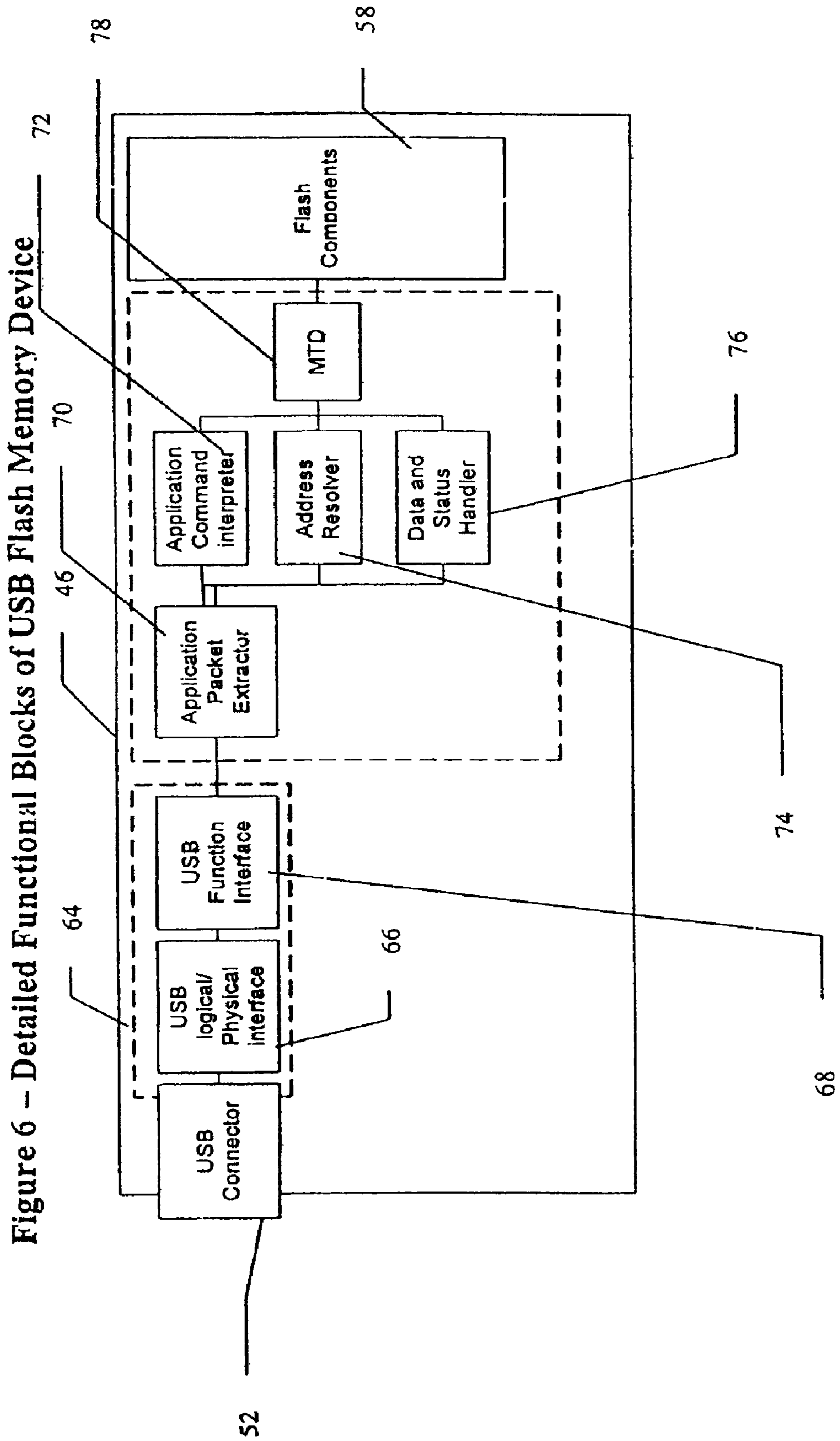


Fig 7 – Flash Identification Request Packet

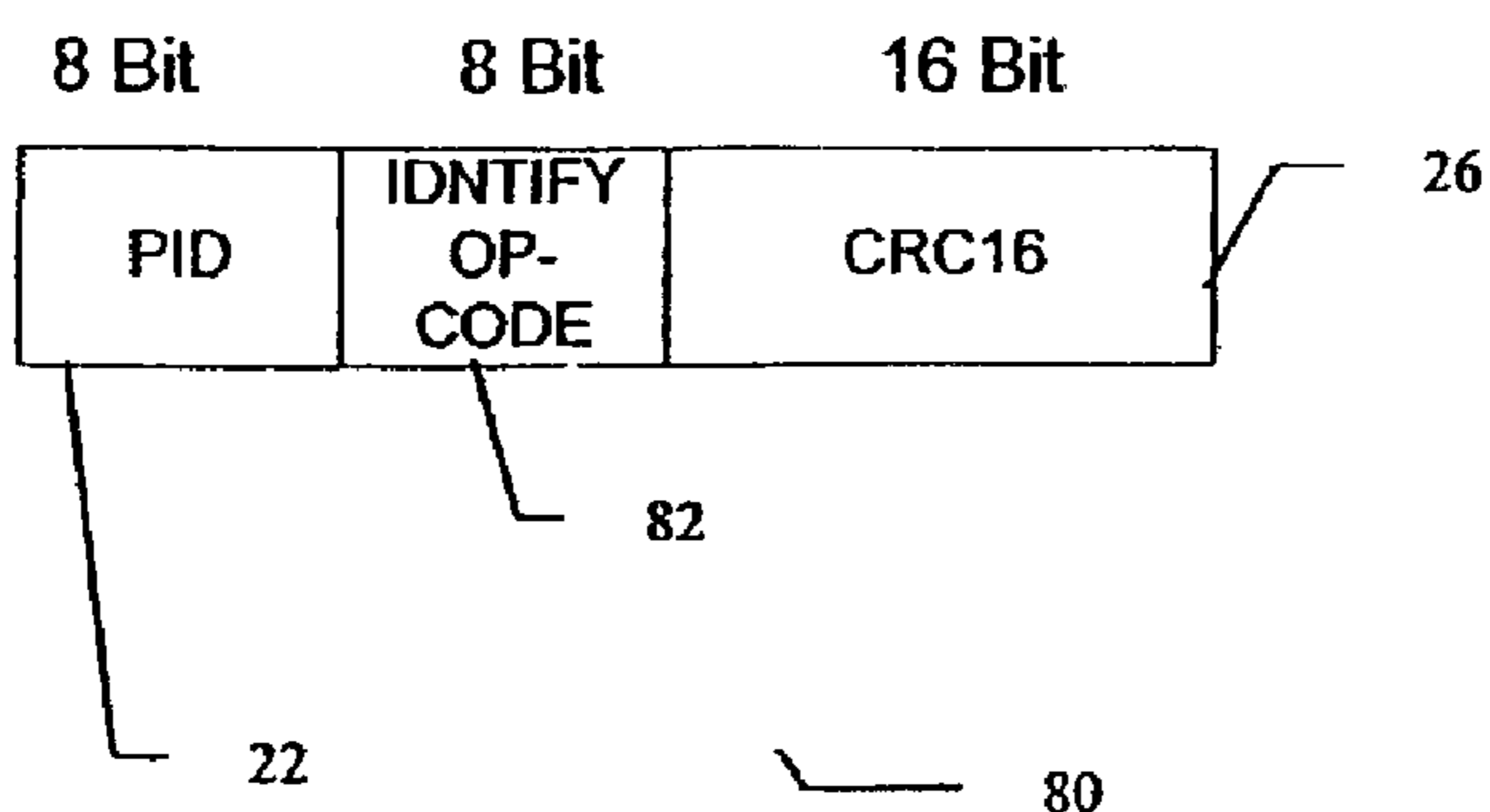


Fig 8 – Flash Identification Replay Packet

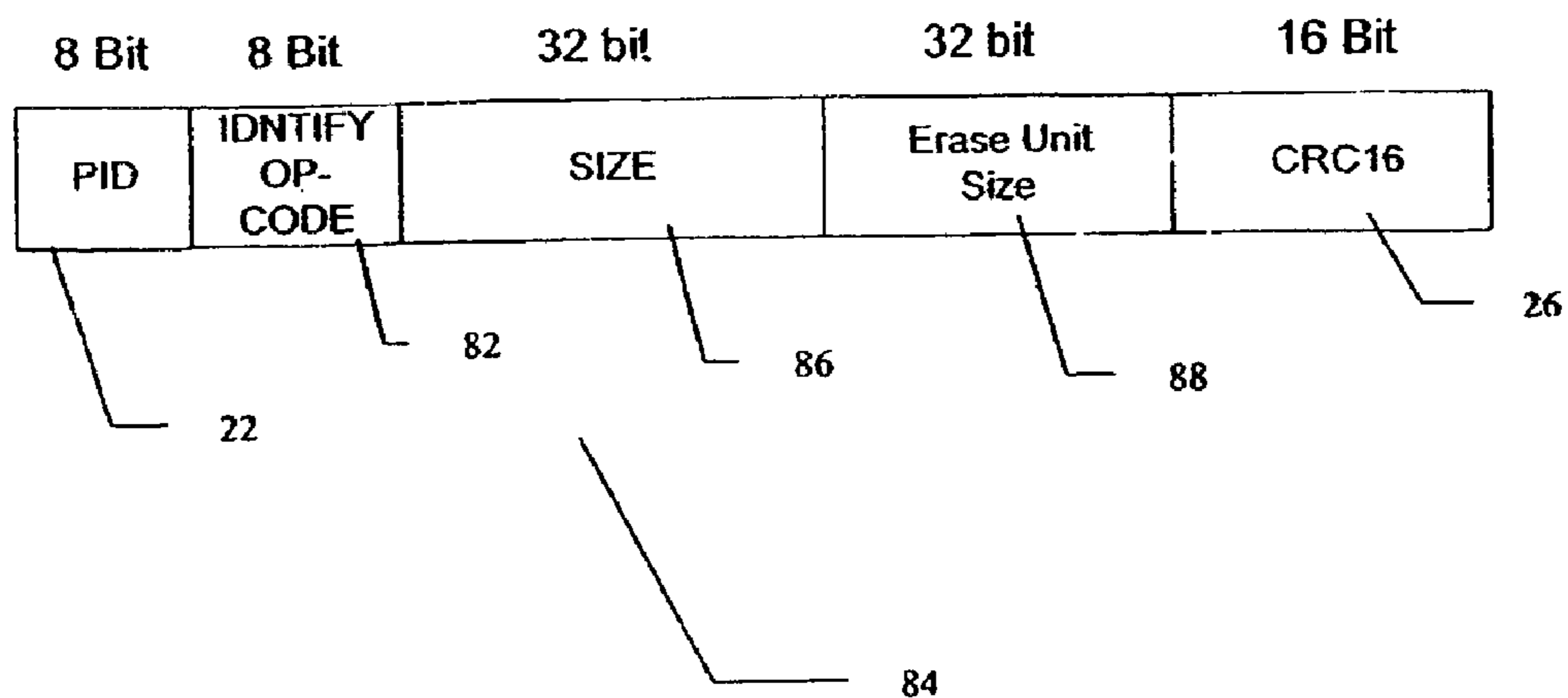
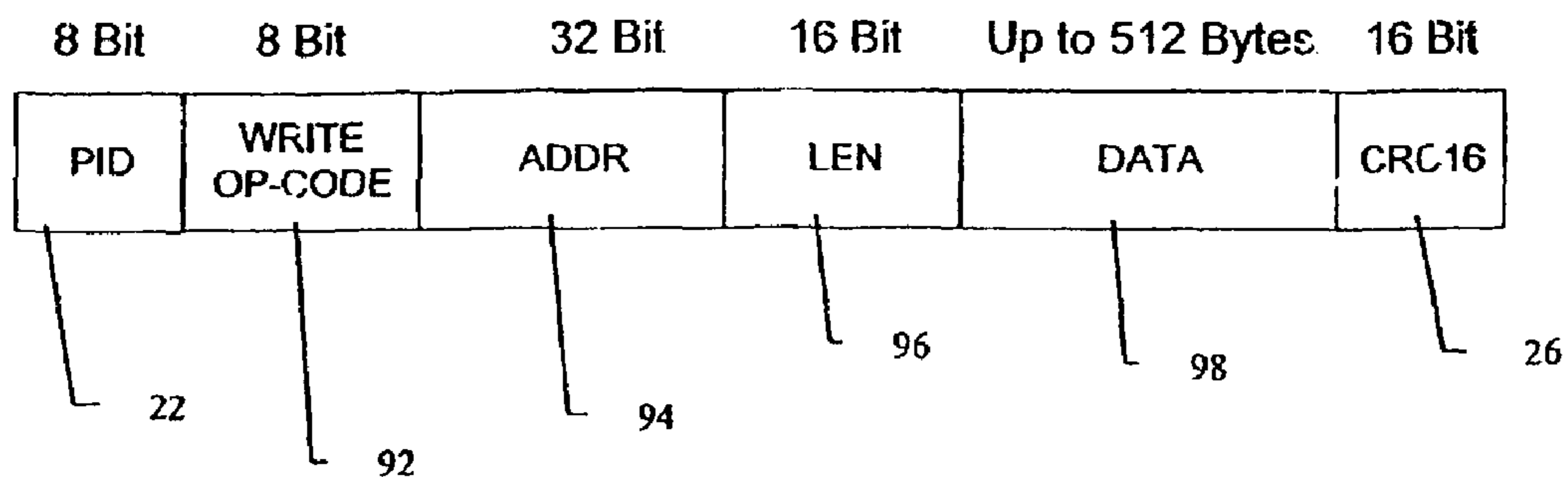


Fig 9 – Write Request Packet



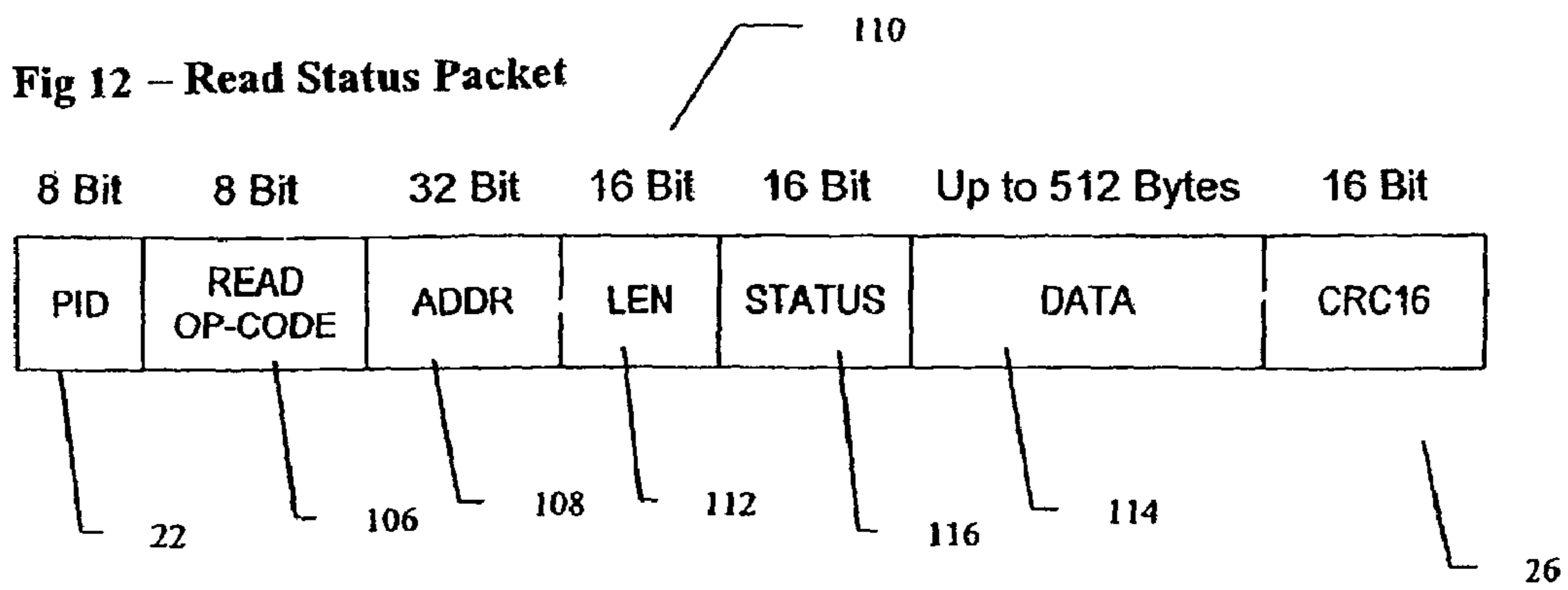
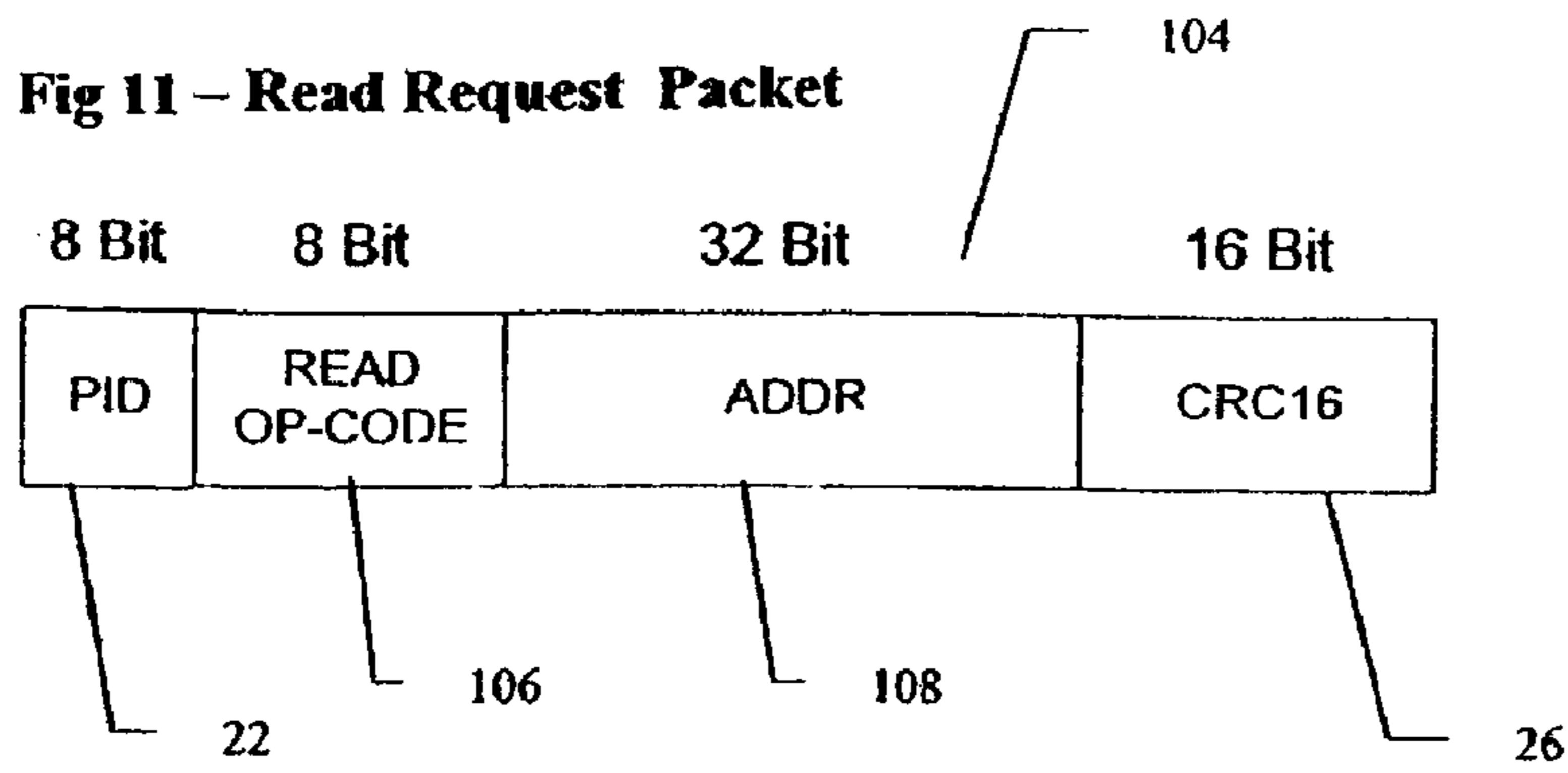
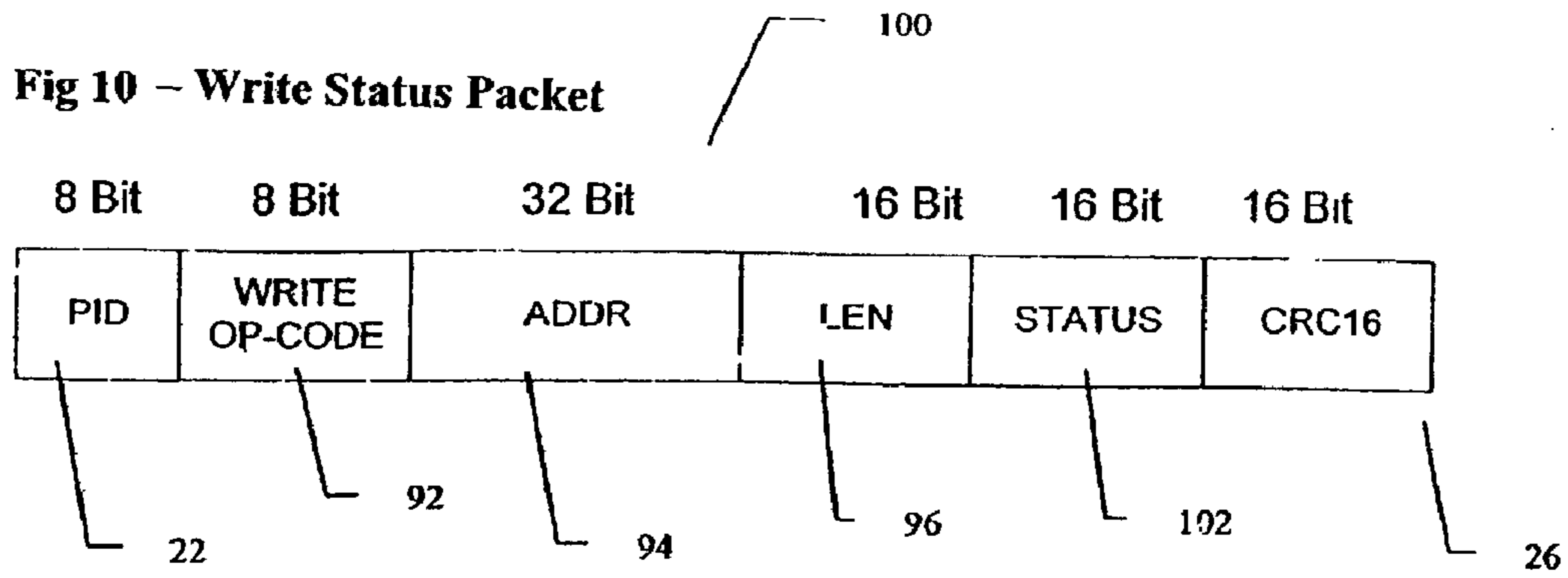


Fig 13 –Erase Request Packet

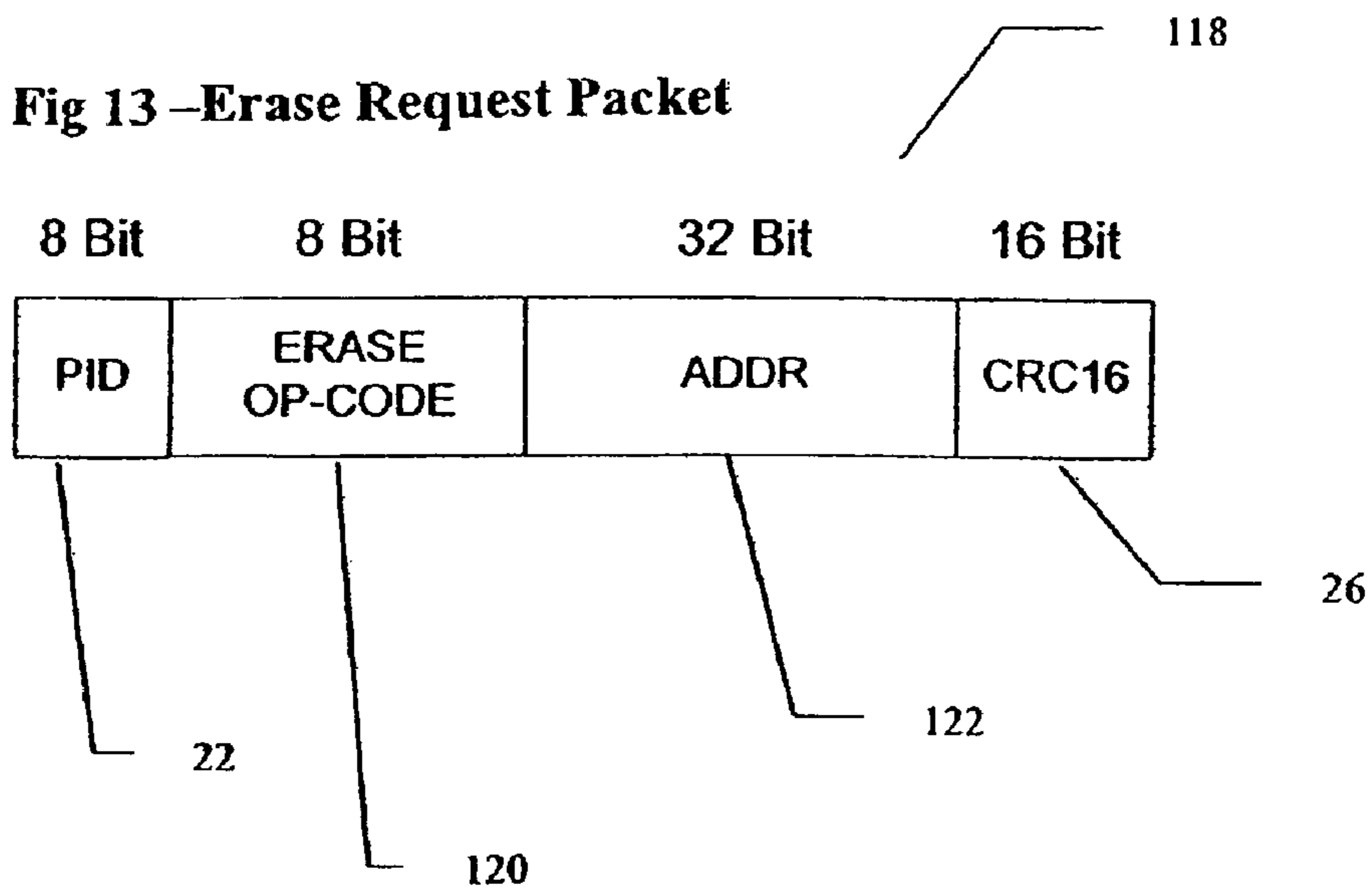
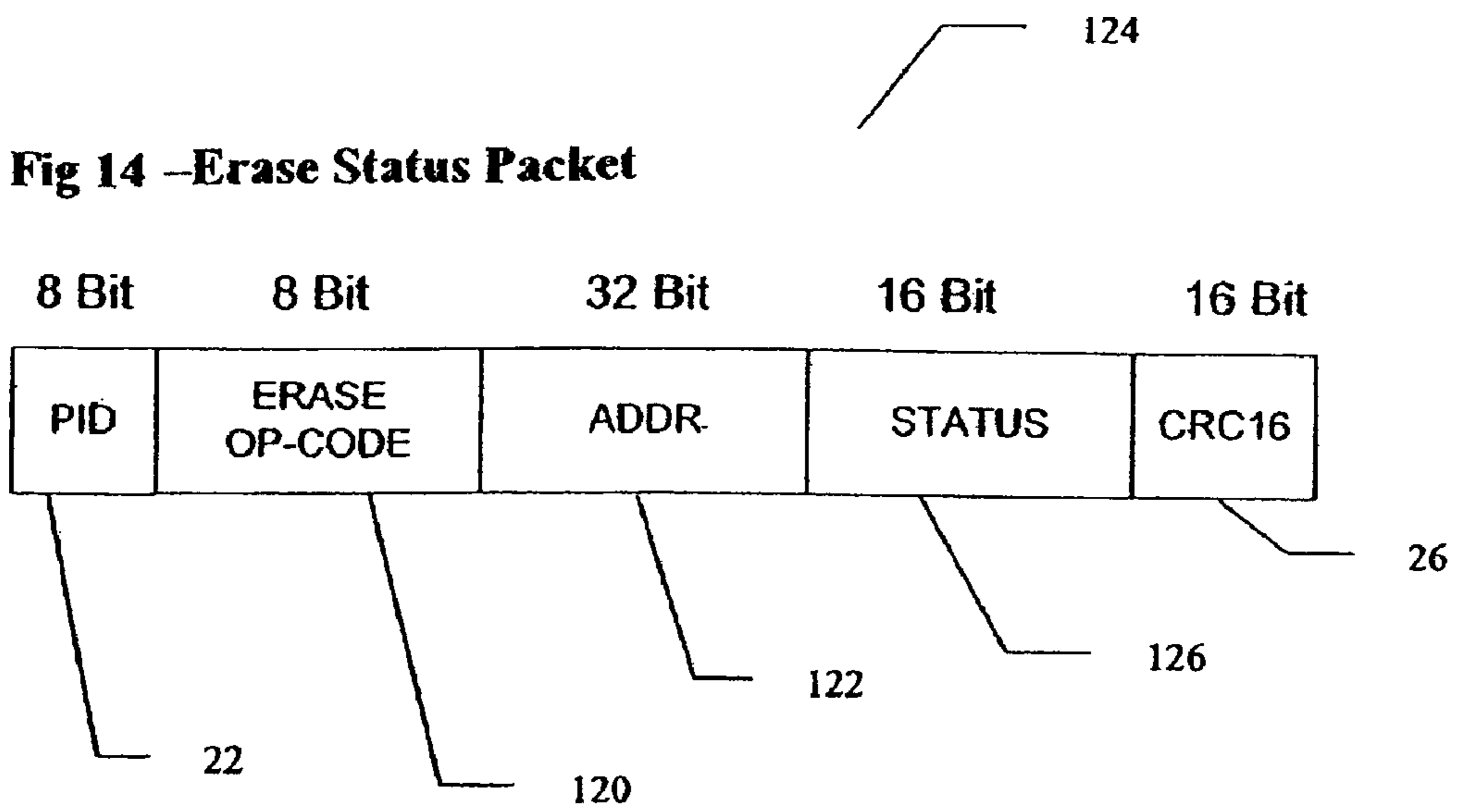


Fig 14 –Erase Status Packet



ARCHITECTURE FOR A UNIVERSAL SERIAL BUS-BASED PC FLASH DISK

Matter enclosed in heavy brackets [] appears in the original patent but forms no part of this reissue specification; matter printed in italics indicates the additions made by reissue.

RELATED APPLICATIONS

This application is a reissue application of U.S. Ser. No. 09/285,706 filed on Apr. 5, 1999, now U.S. Pat. No. 6,148,354 issued on Nov. 14, 2000. More than one reissue application has been filed for the reissue of U.S. Pat. No. 6,148,354. The reissue applications are application Ser. No. 10/293,986 (the present application), which is a continuation of application Ser. No. 10/292,868 filed Nov. 13, 2002.

FIELD AND BACKGROUND OF THE INVENTION

The present invention is related to semiconductor memory devices, and in particular to erasable and programmable non-volatile memory modules which are connected to a host platform using the USB PC Bus.

Erasable and programmable non-volatile memory modules, hereinafter referred to as flash memory or flash devices, are known in the art for storage of information. Flash devices include electrically erasable and programmable read-only memories (EEPROMs) made of flash-type, floating-gate transistors and are non-volatile memories similar in functionality and performance to EPROM memories, with an additional functionality that allows an in-circuit, programmable, operation to erase pages of the memory. One example of an implementation of such a flash device is given in U.S. Pat. No. 5,799,168, incorporated by reference as if fully set forth herein.

Flash devices have the advantage of being relatively inexpensive and requiring relatively little power as compared to traditional magnetic storage disks. However, in a flash device, it is not practical to rewrite a previously written area of the memory without a preceding page erase of the area. This limitation of flash devices causes them to be incompatible with typical existing operating system programs, since data cannot be written to an area of memory within the flash device in which data has previously been written, unless the area is first erased. A software management system, such as that disclosed in U.S. Pat. No. 5,404,485, filed on Mar. 5, 1993, which is incorporated as if fully set forth herein, is required to manage these functions of the flash memory device.

Currently, these flash memory devices have a second limitation, which is that they must be either attached statically to the host platform, or attached and detached dynamically using the PCMCIA [Personal Computer Memory Card International Association] interface. Both implementations have drawbacks, including difficulty of use and high cost.

A more useful implementation would follow the USB standard, as described in the USB Specification Version 1.1 which is incorporated as if fully set forth herein. The USB standard offers a smaller form factor and greater ease of use for the end user, while lowering the cost of the implementation. This standard is specified to be an industry-wide standard promoted by companies such as Compaq Computer Corporation, Microsoft, IBM and Intel to serve as an extension to the PC architecture with a focus on Computer Telephony Integration (CTI), the consumer, and productivity applications.

The criteria which were applied to define the architecture for the USB standard include the ease of PC (personal computer) peripheral expansion, low cost, support of transfer rates up to 12 Mb/second and full support for real-time data, voice, audio, and compressed video. This standard also offers protocol flexibility for mixed-mode isochronous data transfers and asynchronous messaging, integration in commodity device technology and provision of a standard interface for rapid integration into any given host product. In addition, the USB standard represents a single model for cabling and attaching connectors, such that all of the details of the electrical functions, including bus terminations, are isolated from the end user. Through the standard, the peripheral devices are self-identifying, and support automatic mapping of functions to a driver. Furthermore, the standard enables all peripheral devices to be dynamically attachable and re-configurable.

A system constructed according to the USB standard is described by three separate, defined areas: USB interconnection, USB devices and the USB host platform. The USB interconnection is the manner in which USB devices are connected to, and communicate with, the host platform. The associated functions and components include the bus topology, which is the connection model between USB devices and the host platform.

The USB physical interconnection has a tiered star topology. A hub is at the center of each star. Each wire segment is a point-to-point connection between the host platform and a hub or function, or a hub connected to another hub or function.

In terms of a capability stack, the USB tasks which are performed at each layer in the system include a data flow model and a schedule. A data flow model is the manner in which data moves in the system over the USB between data producers and data consumers. A schedule determines access to the interconnection, which is shared. Such scheduling enables isochronous data transfers to be supported and eliminates arbitration overhead.

The USB itself is a polled bus. The host controller on the host platform initiates all data transfers. All bus transactions involve the transmission of up to three packets. Each transaction begins when the host controller, on a scheduled basis, sends a USB packet describing the type and direction of transaction, the USB device address, and endpoint number. This packet is referred to as the "token packet." The USB device, to which the packet is addressed, selects itself by decoding the appropriate address fields. In a given transaction, data is transferred either from the host platform to a device or from a device to the host platform. The direction of data transfer is specified in the token packet. The source of the transaction then sends a data packet or indicates that the source has no data to transfer. The destination, in general, responds with a handshake packet indicating whether the transfer was successful.

The USB data transfer model between a source and destination on the host platform and an endpoint on a device is referred to as a "pipe". There are two types of pipes: stream and message. Stream data has no USB-defined structure, while message data does. Additionally, pipes have associations of data bandwidth, transfer service type, and endpoint characteristics like directionality and buffer sizes. Most pipes come into existence when a USB device is configured. One message pipe, the default control pipe, always exists once a device is powered, in order to provide access to the configuration, status, and control information for the device.

The transaction schedule for the USB standard permits flow control for some stream pipes. At the hardware level, this prevents situations in which buffers experience underrun or

overrun, by using a NAK handshake to throttle the data rate. With the NAK handshake, a transaction is retried when bus time is available. The flow control mechanism permits the construction of flexible schedules which accommodate concurrent servicing of a heterogeneous mix of stream pipes. Thus, multiple stream pipes can be serviced at different intervals with packets of different sizes.

The USB standard, as described, has three main types of packets, including token packets, data packets and handshake packets. An example of each type of packet is shown in background art FIGS. 1-3. Background art FIG. 4 shows an exemplary USB abstract device.

A token packet **10**, as shown in background art FIG. 1, features a PID (packet identification) field **12**, specifying one of three packet types: IN, OUT, or SETUP. If PID field **12** specifies the IN packet type, the data transaction is defined from a function to the host platform. If PID field **12** specifies the OUT or SETUP packet type, the data transaction is defined from the host platform to a function.

An ADDR field **14** specifies the address, while an ENDP field **16** specifies the endpoint for token packet **10**. For OUT and SETUP transactions, in which PID field **12** specifies that token packet **10** is an OUT packet type or a SETUP packet type, ADDR field **14** and ENDP field **16** uniquely identify the endpoint for receiving the subsequent data packet, shown in FIG. 2, which follows after token packet **10**. For IN transactions, in which PID field **12** specifies that token packet **10** is an IN packet type, ADDR field **14** and ENDP field **16** uniquely identify which endpoint should transmit a data packet. A CRC5 field **18** contains the checksum, for determining that token packet **10** has been received without corruption. Only host platform can issue token packets **10**, such that token packets **10** provide control over transmission of the subsequent data packets.

As shown in background art FIG. 2, a background art USB data packet **20** also features a PID (packet identification) field **22** for identifying the type of data packet. Data packet **20** also features a data field **24** for optionally containing data, and a CRC field **26** for containing the checksum as previously described.

Background art FIG. 3 shows a background art USB handshake packet **28**, which features only a PID (packet identification) field **30**. Handshake packets **28** are used to report the status of a data transaction and can return values indicating successful reception of data, command acceptance or rejection, flow control, and halt conditions. Only transaction types which support flow control can return handshake packets **28**. Handshake packets **28** are always returned in the handshake phase of a transaction and may be returned, instead of data packets **20**, in the data phase of a transaction.

These three different types of packets are exchanged during various phases of the transaction which includes a USB device. A schematic block diagram of the functional blocks in a typical USB device **32** is shown in FIG. 4 for an abstract background art USB device. USB device **32** typically includes a USB electrical interface **34**, featuring a cable and a connector, which is a physical interface for receiving and transmitting electrical signals which are compatible with the USB specification as previously described. The signals are then passed to a logical interface **36**, which includes one or more buffers, the device address decoder for decoding the address of the source device for the signals, and a SYNC field synchronizer for synchronizing the signals. Information and structures required for management of USB abstract device **32** as a USB device are stored in a USB class control and enumeration engine **38**. A function and device engine **40**, also termed the "application", controls and manages the specific

functions and properties of USB abstract device **32**. In addition, function and device engine **40** also consumes and produces most of the data over the USB bus.

The USB specification does not define the relationship between different entities in USB abstract device **32**, however. Rather, the USB specification describes only the requirements for the packets, and for the electrical and physical connection between USB abstract device **32** and the bus. Therefore the connections and relationships shown in background art FIG. 4 are only one example of an implementation which fulfills the requirements of the USB specification. Thus, any specific device for fulfilling the USB specification must have a specifically defined and described architecture.

Unfortunately, no such architecture exists for a flash memory device containing one or more flash memory modules, which would enable the flash memory device to connect to a bus defined according to the USB specification and thereby to form part of a USB system on a host platform. For example, U.S. Pat. No. 5,799,168 does not teach or suggest such an implementation for the flash device. As mentioned previously, such an architecture would be particularly useful for a number of reasons, including low cost, ease of use and transparency to the end user.

There is thus a need for, and it would be useful to have, an architecture for defining and describing a flash memory device which is compatible with a USB system and which follows the USB specification, such that the flash memory device could sit on a USB-defined bus and communicate with the host platform through this bus.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a schematic block diagram of a background art USB token packet structure;

FIG. 2 is a schematic block diagram of a background art USB data packet structure;

FIG. 3 is a schematic block diagram of a background art USB handshake data packet structure;

FIG. 4 is a schematic block diagram of an exemplary background art USB device;

FIG. 5 is a schematic block diagram of a system with a flash USB device functionality according to the present invention;

FIG. 6 is a schematic block diagram of the USB Flash disk;

FIG. 7 is a schematic block diagram of a flash identification request packet;

FIG. 8 is a schematic block diagram of a flash identification status packet;

FIG. 9 is a schematic block diagram of a flash write request packet;

FIG. 10 is a schematic block diagram of a flash write status packet;

FIG. 11 is a schematic block diagram of a flash read request packet;

FIG. 12 is a schematic block diagram of a flash read status packet;

FIG. 13 is a schematic block diagram of a flash erase request packet; and

FIG. 14 is a schematic block diagram of a flash erase status packet.

SUMMARY OF THE INVENTION

The present invention is of a flash memory device, containing one or more flash modules, in which the flash memory is mapped to the address space of an ASIC or a controller which has a USB-defined electrical interface and a USB-defined logical interface. This controller/ASIC (hereinafter termed a

5

“controller”) supports the USB functionality according to the USB standard, thereby supporting enumeration onto the USB bus, as well as data reception and transmission over USB pipes to and from USB endpoints. This controller also supports the functionality and control of the flash memory device, as well as the processing of command and data packets from the host controller. The host controller uses one of several possible protocols, either standard or proprietary, to signal the next command to be performed to the USB flash controller. Thus, the entire device acts as a dynamically attachable/detachable non-volatile storage device for the host platform.

According to the present invention, there is provided a USB flash memory device for connecting to a USB-defined bus, the flash memory device comprising: (a) at least one flash memory module for storing data; (b) a USB connector for connecting to the USB-defined bus and for sending packets on, and for receiving packets from, the USB-defined bus; and (c) a USB controller for controlling the at least one flash memory module and for controlling the USB connector according to at least one packet received from the USB-defined bus, such that data is written to and read from the at least one flash memory module.

Hereinafter, the term “computer” includes, but is not limited to, personal computers (PC) having an operating system such as DOS, Windows™, OS/2™ or Linux; Macintosh™ computers; computers having JAVA™-OS as the operating system; and graphical workstations such as the computers of Sun Microsystems™ and Silicon Graphics™, and other computers having some version of the UNIX operating system such as AIX™ or SOLARIS™ of Sun Microsystems™; or any other known and available operating system, including operating systems such as Windows CE™ for embedded systems, including cellular telephones, handheld computational devices and palmtop computational devices, and any other computational device which can be connected to a network. Hereinafter, the term “Windows™” includes but is not limited to Windows95™, Windows 3.X™ in which “x” is an integer such as “1”, Windows NT™, Windows98™, Windows CE™ and any upgraded versions of these operating systems by Microsoft Inc. (Seattle, Wash., USA).

DETAILED DESCRIPTION OF THE INVENTION

The present invention is of a flash memory device, containing one or more flash modules, in which the flash memory is mapped to the address space of an ASIC or a controller which has a USB-defined electrical interface and a USB-defined logical interface. This controller/ASIC (hereinafter termed a “controller”) supports the USB functionality according to the USB standard, thereby supporting enumeration onto the USB bus, as well as data reception and transmission over USB pipes to and from USB endpoints. This controller also supports the functionality and control of the flash memory device, as well as the processing of command and data packets from the host controller. The host controller uses one of several possible protocols, either standard or proprietary, to signal the next command to be performed to the USB flash controller. Thus, the entire device acts as a dynamically attachable/detachable non-volatile storage device for the host platform.

While the invention is susceptible to various modifications and can be implemented using many alternative forms, the embodiment is shown by way of example in the drawings and will be described in details in the following pages. It should be understood that one of ordinary skill in the art appreciates that the present invention could be implemented in various other

6

ways. The intention is to cover all modifications and alternatives falling within the spirit of the current invention.

The principles and operation of a USB flash device and system according to the present invention may be better understood with reference to the drawings and the accompanying description, it being understood that these drawings are given for illustrative purposes only and are not meant to be limiting.

Referring now to the drawings, FIG. 5 is a schematic block diagram of the main components of a flash memory device and system according to the present invention. A flash memory system 42 includes a host platform 44 as shown. Host platform 44 operates USB flash device 46 as a non-volatile storage space.

Host platform 44 is connected to USB flash device 46 according to the present invention through a USB cable 48. Host platform 44 connects to USB cable 48 through a USB host connector 50, while USB flash device 46 connects to USB cable 48 through a USB flash device connector 52. Host platform 44 features a USB host controller 54 for controlling and managing all USB transfers on the USB bus.

USB flash device 46 features a USB flash device controller 56 for controlling the other components of USB flash device 46 and for providing an interface for USB flash device 46 to the USB bus, USB flash device connector 52 and at least one flash memory module 58. Flash memory module 58 is preferably an array of flash memory modules 58 in which the data is stored.

Whenever USB flash device 46 becomes connected to host platform 44, a standard USB enumeration process takes place. In this process host platform 44 configures USB flash device 46 and the mode of communication with USB flash device 46. Although there are many different methods for configuring USB flash device 46, for the purposes of clarity only and without intending to be limiting, the present invention is explained in greater detail below with regard to a method in which host platform 44 issues commands and requests to USB flash device 46 through one endpoint. Host platform 44 queries USB flash device 46 through the other endpoint for status changes, and receives related packets if any such packets are waiting to be received.

Host platform 44 requests services from USB flash device 46 by sending request packets to USB host controller 54. USB host controller 54 transmits packets on USB cable 48. These requests are received by USB flash device controller 56 when USB flash device 46 is the device on the endpoint of the request. USB flash device controller 56 then performs various operations such as reading, writing or erasing data from or to flash memory module(s) 58, or supporting basic USB functionality such as device enumeration and configuration. USB flash device controller 56 controls flash memory module(s) 58 by using a control line 60 to control the power of flash memory module(s) 58, and also through various other signals such as chip enable, and read and write signals for example. Flash memory module(s) 58 are also connected to USB flash device controller 56 by an address/data bus 62. Address/data bus 62 transfers commands for performing read, write or erase commands on flash memory module(s) 58, as well as the addresses and data for these commands as defined by the manufacturer of flash memory module(s) 58.

In order for USB flash device 46 to notify host platform 44 on the result and status for different operations requested by host platform 44, USB flash device 46 transmits status packets using the “status end point”. According to this procedure, host platform 44 checks (polls) for status packets and USB

flash device **46** returns either an empty packet if no packets for new status messages are present, or alternatively returns the status packet itself.

A more detailed structure of the functional components of USB flash device **46** is shown in FIG. **6**. USB flash device **46** includes the physical and electrical interface defined for the USB standard, shown here as USB flash device connector **52** and a connector interface **64**. USB flash device connector **52** receives the electrical signals from USB cable **48** which carries electrical signals from host controller (not shown). These signals are then passed through connector interface **64**. Every millisecond, a USB frame is carried on the USB-defined bus, such that packets could be sent to USB flash device **46**.

Connector interface **64** then receives these packets through a first interface component, which is a combined physical and logical interface **66**. A functional interface **68** is specifically designed to receive token packets as defined in the USB specification and as previously described with regard to FIG. **1**. These token packets are related only to particular functional aspects of USB flash device **46** which are required for the USB standard, and do not have any relation to particular application of USB flash device **46** as a flash disk according to the present invention. These token packets and their respective returned data packets enable USB host controller **54** (not shown) and host platform **44** (not shown) to identify USB flash device **46** and allocate resources for USB flash device **46** on the USB bus. Thus, functional interface **68** only supports USB functionality needed for the identification and registration of USB flash device **46** on the USB bus.

USB flash device **46** also features an application packet extractor **70** which extracts the application data and commands from the USB application packets, such that application packet extractor **70** supports only application related packets. Next, any requests to USB flash device **46** by host platform **44** (not shown), in the form of read, write, identify and erase commands, are interpreted by an application command interpreter **72**. For any commands which involve data or an address, such as read, write and erase commands, an address resolve module **74** translates the address from the logical address space to the physical address space. Host platform **44** (not shown) relates to a linear address space of logical addresses, while USB flash device **46** contains at least one, and preferably a plurality of, flash modules **58**, each of which has a physical address space. Thus, a translation must be performed between the logical address space of host platform **44** (not shown) and physical address space or spaces of USB flash device **46**. There are many ways to implement such a translation which are suitable for the present invention. One example of a suitable implementation of an address translation method is described with regard to U.S. Pat. No. 5,404,485, previously incorporated by reference as if fully set forth herein, which teaches a method for managing a flash memory as a flash disk and which is suitable for operation with the present invention.

A data handler **76** handles data related aspects of any received commands, and conveying the data through functional interface **68** to and from flash module(s) **58**. Optionally and preferably, data handler **76** performs any error correction and detection methods. Application command interpreter **72**, data handler **76** and address resolve module **74** all operate with an underlying Memory Technology Driver (MTD) **78** to write, read or erase a particular flash module **58** and the desired address on that flash module **58**.

Host platform **44** checks for status changes in USB flash device **46** and reads status packets from USB flash device **46** when a new status packet is available. Using these status packets, USB flash device **46** can transmit, to host platform

44, the results of different commands issued by host platform **44** in its requests (not shown). For example, the read command status packet contains one of the available status words such as "success", "error" or "invalid address", which enables host-platform **44** to determine the result of the read command (not shown). Similarly, the erase status packet contains a status word indicating the completion of the erase process. A write status packet is used by USB flash device **46** to notify host platform **44** about the result of the write command, for example whether the command was successful or erroneous, and whether USB flash device **46** is ready for additional write requests from host platform **44**.

A Memory Technology Driver, or MTD **78** typically contains routines to read, write and erase the flash memory device controlled by the controller operating MTD **78**. In addition, MTD **78** optionally contains an identification routine for recognizing the proper type of flash memory device for which MTD **78** was designed, so that the controller can determine which MTD should be activated upon interacting with a particular flash memory device array. In addition, an identification routine should be able to detect the size of the array of flash memory devices, including the number of flash memory devices within the array, and various features of the flash array geometry, such as interleaving and bus width. This information later enables host platform **44** platform to determine the address space and size of the storage media. U.S. Pat. No. 5,799,168, previously incorporated by reference, discloses an example of such an MTD for a flash device.

Using the above described protocol and architecture, host platform **44** can optionally implement any application which is implementable with any regular memory mapped or I/O mapped flash memory device. For example, host platform **44** can give a standard block device interface to each application, such as a magnetic storage medium "hard disk" drive, as disclosed in the previously described U.S. Pat. No. 5,404,485.

As an example of a preferred embodiment of the present invention, the operation of a host system connected to a USB flash device according to the present invention is described with regard to the processes of identifying, programming, reading and erasing the flash device. For the purposes of illustration only and without intending to be limiting in any way, the exemplary USB flash device has an array of two flash memory modules, each of which is 64 Mbit in size. The address translation table is within the flash device so that host platform operates with logical addresses. All commands and return codes between the flash device and the host platform are carried on USB data packets, and are transferred through USB data pipes. The exact structure of the packets, pipes and timings are described in the USB specification.

The operation of the exemplary device and system according to the present invention is as follows. When the USB flash device is first connected to the host platform, the USB host controller assigns an address to the USB flash device on the USB bus, and also assigns resources as described in the USB specification. The USB flash device actually asks the host platform to assign these resources, and must inform the host platform how much of these resources are needed. Thus, the USB flash disk can optionally support slower device speeds if the USB host platform has already allocated resources to other devices.

The USB controller also negotiates with the flash modules and determines the size and manufacturing type of these modules. The controller then builds an identification structure holding this information, as well as the translation table and logical address space.

After the USB host controller identifies the USB flash device, the host platform often uploads a USB client driver.

The driver issues an identification request command to the USB host controller, causing the controller to transmit an identification data packet **80**, shown in FIG. 7. Identification packet **80** contains PID field **22** and checksum field **26**, as described previously for background art FIG. 2. Identification packet **80** also contains an “identify” operation code in an operation code field **82**. The packet extractor of the USB flash device receives identification data packet **80** and transfers the operating code of the “identify” command to the application command interpreter.

In response to the “identify” command, the flash device then sends an identification data packet **84**, shown in FIG. 8. In addition to the fields shown in FIG. 7, identification data packet **84** also contains information about the size of the flash device in a flash device size field **86**, as well as information about the size of the minimal erase unit for erasing the flash memory in an erase unit size field **88**.

All of the packets described in this example are only data packets which are sent on the USB bus. Before each data packet is sent, a USB token packet is transmitted, instructing the USB controller as to the identity of the device end point to which the data packet should be transmitted. Upon successful reception of the packet, the USB controller issues a USBACK packet as described in the USB specification.

Once the device drivers in the host platform receive this status packet, the drivers can start issuing read and write commands to the USB flash device with the application commands. When a write request is sent, a USB data packet with the operation code for the “write” command, and the buffer containing the data, is transferred to the USB flash device. A write data packet **90** is shown in FIG. 9, which again includes the fields shown previously in FIG. 8, except that write data packet **90** also includes a write field **92** with the “write” operational code; an ADDR field **94** with the logical address to be written; a LEN field **96** with the length to be written; and a DATA field **98** which contains the actual data to write. The packet extractor extracts the operational code from write data packet **90** and transfers this code to the application command interpreter. The logical address is transferred to the address resolve module which translates this logical address to a physical address on one of the flash modules. The data handler optionally calculates error correction and detection mechanisms if employed by the USB flash device. Once all of the flash memory modules are ready, a “write” command is sent to the flash module or modules containing the physical address, which may optionally span across more than one flash module to the MTD block. The MTD block then issues a “write” command on the data/address bus which connects the flash modules to the USB device controller. Once the operation is complete and a status packet is returned to the MTD, the result of the operation is transmitted to the host controller and passed to the device driver in the host platform.

When the flash controller finishes the writing process, the controller signals to the host platform that the status of the USB flash memory device has changed, by sending a “write status” packet **100**, as shown in FIG. 10. In place of data field **98**, write status packet **100** contains a status field **102**. The host platform reads the status packets from the flash memory device, and from write status packet **100**, the host platform retrieves information on the completion status of the write command by reading status field **102**. In this example, the flash memory device repeats ADDR field **94** and LEN field **96** in order for the host platform to have a reference to the specific command related to status packet **100**.

As shown in FIG. 11, a “read request” packet **104** contains the operation code for the “read” command in a read field **106**, and the logical address of the desired location from which the

flash controller should read in an ADDR field **108**. Upon receiving this command, the flash controller issues a read command to the MTD block, after the address resolve module has translated the address contained in ADDR field **108** to a specific physical address in one of the flash components.

When the flash controller receives the data from the flash device, either after the read command was issued, or if an error occurred, the flash controller sends a signal to the host platform to indicate that a new status packet must be read. The host platform issues a read request and receives a “read status” packet **110** as shown in FIG. 12. Read status packet **110** contains the address of the read data in ADDR field **108**, as well as the length of the read data in a LEN field **112** and the data itself in a data field **114**. Read status packet **110** also features the status word, according to which the operation was completed, in a status field **116**. The read operation can be completed with many different status situations such as success, fail, error detected, invalid address, invalid length and so forth.

When the host platform needs to erase an erase unit in the flash device, the host platform issues an “erase request” packet **118**, shown in FIG. 13. This packet contains the “erase” operation code in an erase field **120**, and the logical address of the erase unit in an ADDR field **122**. Upon receiving such a request, the flash controller translates the logical address to a physical erase unit address on one of the physical address spaces of the flash modules, and issues an erase command to the MTD block.

The erase process generally takes more time than a read or write process. When this erase process is finished, the controller notifies the host platform a new status packet is ready to transmit. The controller then transmits an “erase status” packet **124**, as shown in FIG. 14. Erase status packet **124** contains the address of the erased unit in ADDR field **122**, thereby providing the host platform with a reference to the erase requests. The status according to which the operation was completed is provided in a status field **126**.

It will be appreciated that the above descriptions are intended only to serve as examples, and that many other embodiments are possible within the spirit and the scope of the present invention.

What is claimed is:

[1. A USB flash memory device for connecting to a USB-defined bus, the flash memory device comprising:

- (a) at least one flash memory module for storing data;
- (b) a USB connector for connecting to the USB-defined bus and for sending packets on, and for receiving packets from, the USB-defined bus;
- (c) a USB controller for controlling said at least one flash memory module and for controlling said USB connector according to at least one packet received from the USB-defined bus, such that data is written to and read from said at least one flash memory module;
- (d) an electrical interface for connecting to said USB connector and for receiving said packets from said USB connector as a plurality of electrical signals;
- (e) a logical interface for connecting to said electrical interface and for translating said plurality of electrical signals to logic signals, said logic signals being passed to said at least one flash memory module;
- (f) a functional interface for receiving said logic signals such that if said logic signals represent a USB functional packet, said functional interface sends a USB command to said USB controller according to said USB functional packet;
- (g) an application packet extractor for connecting to said logical interface and for receiving said logic signals, said

11

application packet extractor extracting at least one packet from said logic signals; and

- (h) an application command interpreter for receiving said at least one packet and for determining a command according to said at least one packet, said command being passed to said USB controller.]

[2. The flash memory device of claim 1, further comprising:

- (i) an address resolver module for receiving said at least one packet and for resolving an address contained in said at least one packet, said address being sent to said USB controller, such that said command is performed according to said address.]

[3. The flash memory device of claim 2, wherein said command is a write command for writing data to said at least one flash memory module and said address is a logical address for writing said data, such that said address resolver module resolves said logical address to a physical address of said at least one flash memory module.]

[4. The flash memory device of claim 2, wherein said command is a read command for reading data from said at least one flash memory module and said address is a logical address for reading said data, such that said address resolver module resolves said logical address to a physical address of said at least one flash memory module.]

[5. The flash memory device of claim 2, further comprising:

- (j) a data handler for performing an error detection and correction routine for said at least one flash memory module.]

[6. The flash memory device of claim 5, further comprising:

- (k) a status handler for receiving said USB functional packet from said functional interface, and for sending a status packet concerning a status of said at least one flash memory module according to said USB functional packet.]

[7. The flash memory device of claim 6, further comprising:

- (l) a MTD (memory technology driver) for receiving a write command and physical address of said at least one flash memory module, and for performing said write command to said physical address.]

8. A USB flash memory device for connecting to a USB-defined bus, the flash memory device comprising:

- (a) at least one flash memory module;
 (b) a USB connector adapted for connection to a USB-defined bus and for conveying USB-defined packets sent to or received from a host via said USB-defined bus; and
 (c) a USB controller adapted to communicate with the host via said USB-defined bus and to carry out at least one of reads and writes in said at least one flash memory module in accordance with said USB-defined packets;

wherein said USB controller comprises:

- a USB-defined interface for sending and receiving the USB-defined packets via said USB-defined bus;
 a packet extractor for receiving USB-defined data packets via the USB-defined interface and extracting information from the USB-defined data packets;
 a command interpreter which is adapted to interpret commands obtained as operation codes extracted from the USB-defined data packets into corresponding actions addressed to said at least one flash memory module; and
 an address resolver module which is adapted to translate logical addresses extracted from said USB-de-

12

defined data packets into physical addresses in one or more of said at least one flash memory module; and wherein the USB controller is configured to activate a memory technology driver within the USB flash memory device to issue the commands to the at least one flash memory module, wherein the issued commands are addressed to the corresponding physical addresses.

9. The USB flash memory device according to claim 8, wherein said USB controller further comprises an identification structure for holding memory size and manufacturing type information of said at least one flash memory module, wherein said memory size and manufacturing type information is used by said USB controller to build an address translation table for use by said address resolver module.

10. The USB flash memory device according to claim 8, wherein said address resolver module is adapted for resolving addresses contained in said USB-defined data packets, such that said commands are performed according to said addresses.

11. The USB flash memory device according to claim 8, wherein, if said commands include a write command for writing data to said at least one flash memory module and said address is a logical address for writing said data, said address resolver module is configured to resolve said logical address to a physical address of said at least one flash memory module.

12. The USB flash memory device according to claim 8, wherein, if said commands include a read command for reading data from said at least one flash memory module and said address is a logical address for reading said data, said address resolver module is configured to resolve said logical address to a physical address of said at least one flash memory module.

13. The USB flash memory device according to claim 8, further comprising:

a data handler for performing an error detection and correction routine for said at least one flash memory module.

14. The USB flash memory device according to claim 8, further comprising:

a status handler for sending status packets concerning a status of said at least one flash memory module according to said USB-defined data packets.

15. The USB flash memory device according to claim 8, wherein the USB flash memory device is being provided as an integral unit with the USB connector.

16. The USB flash memory device according to claim 8, wherein said device is configured to act as a dynamically attachable/detachable non-volatile storage device for said host.

17. The USB flash memory device according to claim 8, wherein said USB controller is implemented as a single integrated circuit.

18. The USB flash memory device according to claim 8, wherein the extracted information includes a write command and said USB controller is configured to interpret the write command and extract corresponding operation codes from said USB-defined data packets.

19. The USB flash memory device according to claim 8, wherein said USB-defined bus is connected to said host and wherein said host is configured to provide commands to said USB controller using a standard protocol.

20. The USB flash memory device according to claim 8, comprising an address/data bus for interconnecting said USB controller and said at least one flash memory module.

21. The USB flash memory device according to claim 8, wherein said USB controller is configured to negotiate with

said at least one flash memory module to determine at least one feature of the at least one flash memory module.

22. The USB flash memory device according to claim 21, wherein said at least one feature comprises a size.

23. The USB flash memory device according to claim 22, wherein said at least one feature comprises a manufacturing type.

24. The USB flash memory device according to claim 23, wherein said USB controller is configured to use said manufacturing type to determine a memory technology driver to activate for said at least one flash memory module.

25. The USB flash memory device according to claim 22, wherein said USB controller is configured to notify said host that it is ready after said negotiation.

26. The USB flash memory device according to claim 22, wherein said at least one feature comprises a size and a manufacturing type.

27. The USB flash memory device according to claim 26, wherein said size and manufacturing type are used to determine a physical address space of said at least one flash memory module and generate a translation table for translating logical addresses into addresses of said physical address space.

28. The USB flash memory device according to claim 27, wherein said USB controller is configured to determine said physical address space of said at least one flash memory module and generate said translation table for translating logical addresses into addresses of said physical address space.

29. The USB flash memory device according to claim 8, wherein said USB controller includes a plurality of chip enable signal lines for attaching to a plurality of flash memory modules.

30. The USB flash memory device according to claim 8, wherein said USB connector is attached to said USB controller by a combined physical/logical interface.

31. The USB flash memory device according to claim 30, wherein said combined physical/logical interface is part of said USB controller.

32. The USB flash memory device according to claim 8, wherein said at least one flash memory module comprises a plurality of flash memory modules.

33. The USB flash memory device according to claim 8, wherein said USB controller further comprises:

a functional interface adapted to receive said USB-defined packets, such that if one of said USB-defined packets is a USB token packet, said functional interface acts on said token packet;

wherein the packet extractor is serially connected after said functional interface.

34. The USB Flash memory device of claim 8, wherein the commands include one or more of read, write and erase commands and the corresponding actions include one or more of read, write and erase.

35. A data-processing method at a USB flash memory device, wherein the USB flash memory device includes at least one flash memory module and a USB connector and a USB controller, wherein the USB connector is configured to be coupled to a host through a USB-defined bus and the USB controller includes a USB-defined interface for communicating with the host via the USB-defined bus, the method comprising:

receiving, at the USB-defined interface, USB-defined packets from the host through the USB-defined bus and the USB connector, wherein the USB-defined packets include USB-defined data packets;

under the control of the USB controller:

extracting from a respective USB-defined data packet a command and a logical address within a logical address space associated with the USB flash memory device;

converting the logical address into a physical address, which physical address is in the at least one flash memory module;

activating a memory technology driver within the USB flash memory device; and

issuing the command to the activated memory technology driver; and

performing, at the at least one flash memory module, an operation corresponding to the command on data associated with the physical address.

36. The data-processing method according to claim 35, further comprising:

under the control of the USB controller,

identifying memory size and manufacturing type information of the at least one flash memory module; and

generating an address translation table using the memory size and manufacturing type information, wherein the address translation table is configured for converting logical addresses in the logical address space into physical addresses, which physical addresses are in the at least one flash memory module.

37. The data-processing method according to claim 36, further comprising storing the memory size and manufacturing type information in an identification structure within the USB flash memory device.

38. The data-processing method according to claim 35, including:

under the control of the USB controller,

extracting a write command and a predefined amount of data from the at least one of the USB-defined data packets; and

writing the predefined amount of data into locations including the physical address in accordance with the write command.

39. The data-processing method according to claim 35, including:

under the control of the USB controller,

extracting a read command and a length of data from the at least one of the USB-defined data packets;

retrieving data from locations including the physical address in accordance with the read command and the length of data; and

transmitting the retrieved data to the host through the USB connector and the USB-defined bus.

40. The data-processing method according to claim 35, wherein the USB controller is implemented as a single integrated circuit.

41. The data-processing method according to claim 35, further comprising:

under the control of the USB controller, negotiating with the at least one flash memory module to determine at least one feature of the flash memory module.

42. The data-processing method according to claim 41, wherein the at least one feature comprises a memory size.

43. The data-processing method according to claim 41, wherein said at least one feature comprises a manufacturing type.

44. The data-processing method according to claim 43, further comprising using the manufacturing type to determine the memory technology driver for the at least one flash memory module.

45. The data-processing method according to claim 41, further comprising:

under the control of the USB controller, notifying the host after its negotiation with the at least one flash memory module.

46. The data-processing method according to claim 35, including:

receiving electrical signals from the host through the USB-defined bus and the USB connector, wherein the electrical signals are USB-compatible; and extracting the USB-defined packets from the electrical signals.

47. A USB flash memory device for connecting to a USB-defined bus, the flash memory device comprising:

(a) at least one flash memory module;

(b) a USB connector adapted for connection to a USB-defined bus and for conveying USB-defined packets sent to or received from a host via the USB-defined bus; and

(c) a USB controller adapted to communicate with the host via the USB connector and to carry out at least one of reads and writes on the at least one flash memory module in accordance with the USB-defined packets, wherein the USB controller comprises a packet extractor for receiving USB-defined data packets via the USB connector and extracting information from the USB-defined data packets, and a command interpreter which is adapted to interpret commands extracted from the USB-defined data packets over the USB connector into actions for the at least one flash memory module; and

(d) memory technology drivers, each adapted for execution by the USB controller to perform the actions on a respective type of flash memory module;

wherein the USB controller is configured to activate a respective memory technology driver within the flash memory device based on an identification of a respective type of the at least one flash memory module and the activated memory technology driver is configured to perform the actions on the identified flash memory module.

48. The USB flash memory device according to claim 47, wherein the device is provided as an integral unit with the USB connector.

49. The USB flash memory device according to claim 47, wherein the USB controller further includes:

an address resolver module which is adapted to translate a logical address extracted from the USB-defined data packets into a physical address in one or more of the at least one flash memory module.

50. The USB flash memory device according to claim 49, wherein, if one of the commands is a write command for writing data on the at least one flash memory module and the logical address is a logical address for writing the data, the address resolver module is configured to resolve the logical address to a physical address of the at least one flash memory module and the determined memory technology driver is configured to write the data to the physical address of the at least one flash memory module.

51. The USB flash memory device according to claim 49, wherein, if one of the commands is a read command for reading data on the at least one flash memory module and the logical address is a logical address for reading the data, the address resolver module is configured to resolve the logical address to a physical address of the at least one flash memory module and the determined memory technology driver is configured to read the data from the physical address of the at least one flash memory module.

52. The USB flash memory device according to claim 49, further comprising:

a data handler for performing an error detection and correction action on the at least one flash memory module.

53. The USB flash memory device according to claim 52, further comprising:

a status handler for sending status packets concerning a status of the at least one flash memory module according to the USB-defined data packets.

54. The USB flash memory device according to claim 47, wherein the device is configured to act as a dynamically attachable/detachable non-volatile storage device for the host.

55. The USB flash memory device according to claim 47, wherein the USB controller is implemented as a single integrated circuit.

56. The USB flash memory device according to claim 47, wherein the USB controller is configured to interpret write commands after extracting corresponding operation codes from the USB-defined data packets.

57. The USB flash memory device according to claim 47, wherein the USB-defined bus is connected to the host and wherein the host is configured to provide commands to the USB controller using a standard protocol.

58. The USB flash memory device according to claim 47, further comprising an address/data bus for interconnecting the USB controller and the at least one flash memory module, wherein the address/data bus is configured to transfer addresses and data associated with the actions to or from the at least one flash memory module.

59. The USB flash memory device according to claim 47, further comprising a control line for interconnecting the USB controller and the at least one flash memory module, wherein the USB controller is configured to use the control line to control the power of the at least one flash memory module.

60. The USB flash memory device according to claim 47, wherein the USB controller is configured to negotiate with the at least one flash memory module to determine at least one feature of the at least one flash memory module.

61. The USB flash memory device according to claim 60, wherein the USB controller is configured to build an identification structure for holding the at least one feature.

62. The USB flash memory device according to claim 60, wherein the at least one feature comprises a size.

63. The USB flash memory device according to claim 60, wherein the at least one feature comprises a manufacturing type.

64. The USB flash memory device according to claim 60, wherein the USB controller is configured to notify the host that it is ready after the negotiation.

65. The USB flash memory device according to claim 64, wherein the notification includes at least one USB-defined data packet.

66. The USB flash memory device according to claim 65, wherein the at least one USB-defined data packet contains information of the at least one feature of the at least one flash memory module.

67. The USB flash memory device according to claim 60, wherein the at least one feature comprises a size and a manufacturing type.

68. The USB flash memory device according to claim 60, wherein the USB controller is configured to use the determined size and manufacturing type to generate a translation table and an address space for the at least one flash memory module.

17

69. The USB flash memory device according to claim 47, wherein the USB controller includes a plurality of chip enable signal lines for attaching to a plurality of flash memory modules.

70. The USB flash memory device according to claim 47, wherein the USB connector is attached to the USB controller by a combined physical/logical interface.

71. The USB flash memory device according to claim 70, wherein the combined physical/logical interface is part of the USB controller.

72. The USB flash memory device according to claim 47, wherein the at least one flash memory module comprises a plurality of flash memory modules.

73. The USB flash memory device according to claim 47, wherein the USB controller further comprises:

a functional interface adapted to receive the USB-defined packets, such that if one of the USB-defined packets is a USB token packet, the functional interface acts on the token packet;

wherein the packet extractor is serially connected after the functional interface.

74. The USB Flash memory device of claim 47, wherein the commands include one or more of read, write and erase commands and the actions correspondingly include one or more of read, write and erase.

75. A data-processing method at a USB flash memory device, wherein the USB flash memory device includes at least one flash memory module and a USB connector and a USB controller, wherein the USB connector is configured to

18

be coupled to a host through a USB-defined bus and the USB controller includes a USB-defined interface for communicating with the host via the USB-defined bus, the method comprising:

receiving, at the USB-defined interface, USB-defined packets from the host through the USB-defined bus and the USB connector, wherein the USB-defined packets include USB-defined data packets;

under the control of the USB controller:

communicating with the host via the USB connector; and

extracting information from the USB-defined data packets;

interpreting commands extracted from the USB-defined data packets into actions for the at least one flash memory module;

activating a respective memory technology driver within the flash memory device based on an identification of a respective type of the at least one flash memory module; and

under the control of the activated memory technology driver, performing the actions on the identified flash memory module.

76. The data processing method of claim 73, wherein the commands include one or more of read, write and erase commands and the actions correspondingly include one or more of read, write and erase.

* * * * *