

РАБОТА С IDE HDD ЧЕРЕЗ ПОРТЫ КОНТРОЛЛЕРА

Маргулис М.Б., Ершов А.А.
Кафедра ЭВМ ДГТУ
margulis@cs.dgtu.donetsk.ua

Abstract

Margulis M.B., Yershov A.A. work with ide HDD through controller's ports. Work with HDD is basic at the controller's ports interface. This article contents work with HDD as C++ programs. The program modules used addressing ports.

Введение

Важной частью анализа аппаратно-программного окружения компьютера является работа с жесткими дисками (HDD). В этой статье исследован метод работы с HDD через порты контроллера, который реализован в виде подпрограммы на языке C (компилятор Borland C++ 3.1)

Размеры носителей на жестких дисках достигли таких критических размеров, когда работать с полным объемом данных через прерывание 13h не представляется возможным - в силу ограничений которые положены в основу интерфейса 13-го прерывания. Для того, чтобы решить эту проблему надо обращаться непосредственно, напрямую к контроллеру жесткого диска.

1. Определение реальных параметров HDD (до LBA-трансляции)

Первым этапом реализации исследования доступа к пространству жесткого диска любого размера является реализация задачи идентификации HDD. Это значит, что нужно определить реальные физические параметры жесткого диска. Эти реальные физические параметры получили название IDE-характеристик - они не зависят от BIOS.

При реализации задачи идентификации жесткого диска необходимо обеспечить преобразование сквозного абсолютного физического адреса сектора на диске - Linear Basic Address (LBA-адрес) в специальный адрес сектора, который определяется его реальным физическим расположением в определенном цилиндре и на определенной стороне (головке) - Cylinder, Head, Sector (CHS-адрес).

В соответствии с данными о работе IDE-интерфейса [2], реальные физические параметры жесткого диска могут быть определены путем анализа реакции IDE HDD на обращение к его портам (для Primary Master -1f6, 1f7). При этом будет заполняться специальная структура данных. Поля этой структуры определяют всевозможные характеристики диска - в том числе и размеры (геометрию) исследуемого жесткого диска.

Фрагмент программы, который реализует идентификацию HDD, имеет вид:

```
union {  
int infs[256];
```

```

struct
{
  int Config;           //General configuration bit-inf.
                       //IDE inf: Cyl, Rsrv, Head, Byte/trk,
                       //Byte/sec, Sec/trk
  int Cyls, Rsrv1, Heads, BPT, BPS, SPT;
  int Vendor[3];       //Vendor
  char Serial[20];     //Serial number. 0=not.
  int BufType;         //Buffer type
  int BufSize;         //Buffer size in 512 byte increm. 0=not.
  int ECC;             //# of ECC bytes for read/write long
                       //cmds.0=not.
  char Revision[8];   //Firmware revision. 0=not.
  char Model[40];     //Model number. 0=not.
  int Features;       //Features inf.
  int DwordIO;        //1=Can perform doubleword I/O
  int Capabil;        //Capabilities
  int Rsrv2;          //Reserved
  int PIO, DMA;       //PIO,DMA data transfer inf.
  int ExtValid;       //Extended data validation inf.
                       //Current logical geometric
  int CurrCyls, CurrHeads, CurrSect;
  long Capacity;      //Current capacity in sect
  int BlkMode;        //Multiple sect. transfer inf.
  long LBACapacity;   //Number of user addressable sect.LBA only.
  int SingleDMA;      //Single word DMA transfer inf.
  int MultiDMA;       //Multiword DMA transfer inf.
  int AdvancedPIO;    //Advanced PIO Transfer inf.
                       //Inf. about PIO,DMA TransferCycleTime in nsecs:
  int MinDMACycle, RecDMACycle, MinPIOCycle, MinIORDYPIOCycle;
} su;

}um;

void init_hd1(void)
{
  int BasePort=0x1f0;
  long tt,i;

  tt = biostime(0,0);
  outportb(BasePort+6,0xA0);
  outportb(BasePort+7,0xEC);
  while((inportb(BasePort+7)&0x80)!=0)
  if((biostime(0,0)-tt)>5) break;
  for(i=0;i<255;i++)
  um.infs[i]=inport(BasePort);
}

```

2. Перевод из LBA-адреса сектора в физический CHS-адрес

Формула для перевода из физических CHS-координат в линейный LBA-адрес имеет следующий вид:

$$\text{LBAsect} = \text{Cylinder} * \text{HeadPerCyl} * \text{SectPerTrk} + \text{Head} * \text{SectPerTrk} + \text{Sector}.$$

Для перевода LBA-адреса в CHS-адрес, можно использовать несколько формул. Например, для определения номера сектора CHS-стандарта, нужно учесть количество цилиндров и дорожек в линейном адресе текущего сектора:

$$\text{Sect} = \text{LBAsect} \bmod (\text{HeadPerCyl} * \text{SectPerTrk}.)$$

Следует заметить, что нулевой результат здесь означает: $\text{Sect} = \text{SectPerTrk}$. Приведенная формула описывает самую простую часть перевода в стандарт CHS. Другие элементы перевода требуют дополнительных условий.

С другой стороны, тот же результат может быть достигнут простым итерационным циклом. Ниже приведен пример такого перевода.

Итерационный метод имеет некоторые недостатки. Так, для достаточно больших номеров LBA могут возникнуть некоторые задержки по времени. С другой стороны, при использовании современных процессоров этот недостаток не существен. Реализация алгоритма перевода в стандарт CHS:

```
union byte_byte
{ struct { ... } one; unsigned int all; } itrack;
for(i=0, itrack.all=0, isec=0, ihead=0; i<=sector; i++)
{ isec++;
  if(isec>um.su.SPT){isec=1; ihead++;}
  if(ihead>um.su.Heads-1){ihead=0; itrack.all++;}
}
if(itrack.all>um.su.Cyls) return(2);
```

3. Чтение из секторов

Перед первым обращением к процедуре манипулирования секторами HDD следует заполнить объединение (um), описывающее этот HDD. В подпрограмму передается номер LBA-сектора, который внутри самой процедуры переводится в физический адрес CHS. Эти физические параметры заносятся в порты контроллера и из порта данных (для Primary Master- это 1f0h) читаются байты (это и есть содержимое сектора).

```
outportb(0x1f6, ihead);
outportb(0x1f2, Nsec);
outportb(0x1f3, isec);
outportb(0x1f4, itrack.one.lo);
outportb(0x1f5, itrack.one.hi);
outportb(0x1f7, command);
still_going: //Цикл ожидания готовности контроллера
asm {
```

```
    in    al,dx
    test  al,8
    jz    still_going
}
asm {          //Подготовка буфера
    push  ds
    pop   es
    mov   cx,N
    mov   di,offset buffer
    mov   dx,0x1f0
    rep  insw
}
```

4. Запись секторов на диск

Запись секторов на диск производится по тому же принципу, что и их чтение. Здесь единственное отличие в том, что в командный порт посылается команда чтения и данные из порта не читаются (командой `rep insw`), а записываются командой Ассемблера `rep outsw`.

```
outportb(0x1f6,ihed);
outportb(0x1f2,Nsec);
outportb(0x1f3,isec);
outportb(0x1f4,itrack.one.lo);
outportb(0x1f5,itrack.one.hi);
outportb(0x1f7,command);
still_going3:
asm{          //Цикл ожидания готовности контроллера
    mov  dx,0x1f7
    in   al,dx
    test al,8
    jz   still_going3
}
asm{         //Подготовка буфера
    push  ds
    pop   es
    mov   cx,N
    mov   si,offset buffer
    mov   dx,0x1f0
    rep  outsw
}
```

Заключение

Исследуемая тема достаточно актуальна. Использование прерывания `13h` не удовлетворяет современным запросам т.к. это прерывание обеспечивает работу с HDD размером только до 8Гб. Если же реализовать аналогичные операции через порты контроллера, то для режима АТА порты позволяют работать с объемами 31,5Гб, а для

режима ATA2 - до 0,5Тб. Следует отметить, что возможности контроллера IDE HDD не ограничены только чтением и записью. Список команд приведен в таблице 1.

Список команд контроллера

Таблица 1

Код команды	Пояснения
90	Диагностика диска
50	Форматировать дорожку
ЕС	Идентификация диска
91	Инициализировать параметры диска
Е4	Чтение буфера
С8	Чтение DMA (с повтором)
С9	Чтение DMA (без повтора)
20	Чтение сектора (с повтором)
21	Чтение сектора (без повтора)
22	Длинное чтение (с повтором)
23	Длинное чтение (без повтора)
30	Запись сектора (с повтором)
31	Запись сектора (без повтора)
32	Длинная запись (с повтором)
33	Длинная запись (без повтора)
99 Е6	Установить режим sleep
Е8	Запись буфера
СА	Запись DMA (с повтором)
СВ	Запись DMA (без повтора)
С3	Запись с верификацией

Далее в таблице 2 приводится список портов которые работают с Primary HDD.

Список портов для работы с контроллером

Таблица 2

Номер порта	Пояснения
1f0	Порт данных (r/w)
1f1	Порт ошибок (r)
1f2	Номер сектора для обработки (w)
1f3	Число секторов для обработки (w)
1f4	Мл. часть знач. цилиндра (слово)(w)
1f5	Ст. часть знач. цилиндра (слово)(w)
1f6	Диск/головка
1f7	Порт команд (w)
1f7	Порт статуса (r)

Литература

1. Фролов А.В. Фролов Г.В. "MS-DOS для программиста" М: Диалог МИФИ 1992. – 254с.
2. Техническая документация по портам IDE HDD. В электронном виде.
Поступила в редакційну колегію 17.011.2001 р.