



## **Using SanDisk Flash ATA Components with an 80C51 Microcontroller**

---

### **Application Note**

Version 1.0

Document No. 80-13-00106

January 1998

#### **SanDisk Corporation**

Corporate Headquarters • 140 Caspian Court • Sunnyvale, CA 94089

Phone (408) 542-0500 • Fax (408) 542-0503

[www.sandisk.com](http://www.sandisk.com)

SanDisk® Corporation general policy does not recommend the use of its products in life support applications where in a failure or malfunction of the product may directly threaten life or injury. Per SanDisk Terms and Conditions of Sale, the user of SanDisk products in life support applications assumes all risk of such use and indemnifies SanDisk against all damages.

The information in this document is subject to change without notice.

SanDisk Corporation shall not be liable for technical or editorial errors or omissions contained herein; nor for incidental or consequential damages resulting from the furnishing, performance, or use of this material.

All parts of SanDisk documentation are protected by copyright law and all rights are reserved. This documentation may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from SanDisk Corporation.

SanDisk and the SanDisk logo are registered trademarks of SanDisk Corporation. CompactFlash and CF are trademarks of SanDisk Corporation.

Product names mentioned herein are for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.

© 1998 SanDisk Corporation. All rights reserved.

SanDisk products are covered or licensed under one or more of the following U.S. Patent Nos. 5,070,032; 5,095,344; 5,168,465; 5,172,338; 5,198,380; 5,200,959; 5,268,318; 5,268,870; 5,272,669. Other U.S. and foreign patents awarded and pending.

Lit. No. 80-13-00106 Rev. 1 1/98

Printed in U.S.A.

## ***1.0 Introduction***

SanDisk provides Flash Memory products in several form factors: CompactFlash™, PC Card, IDE FlashDrive, Flash ChipSet, and MultiMediaCard (MMC). With the exception of MMC, all of these physical forms share the same basic components and the same software interface to the host, namely the ATA instruction set. This

application note describes a very simple implementation of ATA data storage using SanDisk components and a microcontroller of the 80C51 family. With minor variations, the interface is equally adaptable to other popular 8 bit controllers, such as the Motorola 68HC11.

## **2.0 Discussion of Operating Modes**

The controller in the SanDisk Flash ChipSet was designed to be compatible with several operating modes as specified in the ANSI ATA and PCMCIA standards.

The ANSI ATA specification, commonly known as IDE, is the most widely used interface for hard disks on personal computers and is also often used in embedded applications. It has the advantage of direct BIOS support on most PCs and industrial single board computers. It is not the best choice for this example, because the default data path width for the IDE mode is 16 bits. A command must be issued at power-up to change the path to 8 bits.

Within the PCMCIA specification, both I/O and memory mapped modes are supported. The reasons for the different modes are largely historical, reflecting differences in host processor architecture. For example, the Intel X86 instruction set includes specific I/O instructions,

and lends itself to the I/O mode, while most other processors lack these instructions, and are more appropriately used with the memory mapped mode. It should be emphasized that regardless of the operating mode, the interface to the SanDisk controller is always through a set of registers, called the ATA registers, or the Task File, and not directly to a memory location. Data transfer between host and device is always accomplished a block or sector at a time. Even though the memory mode provides an artificial memory window which allows memory to memory block moves, random access to a given byte within a sector is not possible.

The memory mode was chosen for this note because it is very simple and requires no device configuration since it is the default operating mode for the SanDisk controller, and because 8 bit operation is directly supported.

## 3.0 Hardware Notes

### 3.1 General Information

Please refer to the schematic in Appendix B for reference designations.

The example shown is a very simple data logging system using an 80C31, which is the ROM-less version of the 80C51 processor. A single serial port is provided for data input and output. Of course other forms of data input could be used, such as a D/A converter from a transducer, or a parallel port.

The 80C31 (U1) requires an external EPROM for program storage. An 80C32, which contains 256 instead of 128 bytes of internal RAM, may be substituted for U1. An 11.0592 MHz crystal is used for the microcontroller's clock. This value is chosen to generate standard frequencies for the internal UART. The UART is connected via U6, an RS-232 transceiver, to J1, the serial port connector. The 80C31 provides an 8 bit multiplexed address/data bus on port 0. U2 is a latch which provides the lower 8 bits of address to the memory devices and to the SanDisk ChipSet. The upper 8 bits of address are provided by port 2 of U1. U3 is an address decoder which assigns address spaces to the other devices. The EPROM (U5) is addressed from 0000H to 1FFFH; the RAM, U4, is addressed from 2000H to 3FFFH, while the SanDisk ChipSet is assigned an address space from E000H to FFFFH. Note that these addresses are arbitrary, except of course, that the EPROM must start its address at 0. Also, the SanDisk ChipSet only requires a 16 byte address space, so any 16 byte block within the 8 Kbyte address space provided by the address decoder may be used.

### 3.2 I/O Signals to the SanDisk ChipSet

The signal names in parentheses are the net names used in the schematic.

The following signals are used for communication with the 80C31.

**HD0-HD7 (AD0-AD7) Host Data Bus** – The 8 bit bidirectional bus between the host and the SanDisk controller.

**HA0-HA3 (A0-A3) Host Address Bus** – The 4 bit binary address issued by the host to select one of the registers in the ATA register set.

**-OE (-RD) Read Strobe** – The falling edge of -OE enables 8 bit data from a register of the controller onto the host data bus. The rising edge of -OE latches data into the host.

**-WE (-WR) Write Strobe** – The rising edge of -WE clocks 8 bit data from the host data bus into a register on the controller.

**-CE1 (-SDCE) Chip Enable** – The enable signal which activates the read and write strobes.

The following input signals to the controller are not used and are tied to ground:

**HA4-HA10, RESET, -CSEL, WPROTCT**

The following input signals to the controller are not used and connected to the +5 volt line:

**-CE2, -IORD, -IOWR, -REG**

The following output or I/O signals are not used and are left open:

**HD8-HD15, RDY, -WAIT, -INPACK, BVD1, BVD2, -IO16**

For detailed information on individual signals, please refer to the SanDisk Flash ChipSet Product Manual.

## 4.0 Software Notes

### 4.1 Development Environment

In order to develop and debug code for the application, some form of hardware and software development system is needed. This could take the form of an in-circuit emulator, or a general purpose development board which has an on-board monitor and a cross assembler or compiler which is run on a PC and down-loaded to the development board. Once the code is developed, an EPROM is programmed for stand-alone operation.

### 4.2 Register Definitions

For a detailed description of the ATA registers, please refer to the SanDisk Flash ChipSet Product Manual.

For the purposes of this application note, only the following registers are needed.

Offset	-OE=0	-WE=0
0	Read Data	Write Data
1	Error	Features
2	Sector Count	Sector Count
3	Sector No. or bits 7-0 of Logical Block Address (LBA)	Sector No. or bits 7-0 of Logical Block Address (LBA)
4	Cylinder Low or LBA 15-8	Cylinder Low or LBA 15-8
5	Cylinder High or LBA 23-16	Cylinder High or LBA 23-16
6	Drive/Head or Drive/LBA 27-24	Drive/Head or Drive/LBA 27-24
7	Status	Command

---

### 4.3 Code Example

Note that the following subroutines use the LBA mode of addressing the flash memory. These are rudimentary routines; any error and timeout code will depend on the specific application.

```

; ATA DRIVE REGISTER ADDRESSES
;
DATA_REG      EQU 0F000H      ; DATA REGISTER (R/W)
ERR_REG       EQU 0F001H      ; ERROR REGISTER (READ)
FEATURE_REG   EQU 0F001H      ; FEATURE REGISTER (WRITE)
SEC_COUNT_REG EQU 0F002H      ; SECTOR COUNT REGISTER (R/W)
SEC_NUM_REG   EQU 0F003H      ; SECTOR NUMBER REGISTER (R/W)
CYL_LOW_REG   EQU 0F004H      ; CYLINDER LOW REGISTER (R/W)
CYL_HI_REG    EQU 0F005H      ; CYLINDER HIGH REGISTER (R/W)
DRV_HD_REG    EQU 0F006H      ; DRIVE/HEAD REGISTER (R/W)
STATUS_REG    EQU 0F007H      ; STATUS REGISTER (READ)
COMMAND_REG   EQU 0F007H      ; COMMAND REGISTER (WRITE)

BUF_ADDR      EQU 2800H       ; START ADDR OF DATA BUFFER

; SUBROUTINE TO READ 512 BYTES FROM LBA 300H

READ_SECTOR:
    CALL WAIT_READY
;
    MOV DPTR,#SEC_COUNT_REG
    MOV A,#1                    ; 1 SECTOR
    MOVX @DPTR,A
    INC DPTR                    ; LBA 7-0
    CLR A
    MOVX @DPTR,A
    INC DPTR                    ; LBA 15-8
    MOV A,#3
    MOVX @DPTR,A
    INC DPTR                    ; LBA 23-16
    CLR A
    MOVX @DPTR,A
    INC DPTR
    MOV A,#0E0H                ; SET THE LBA BIT
    MOVX @DPTR,A
    INC DPTR                    ; COMMAND REGISTER
    MOV A,#20H                 ; READ SECTOR
    MOVX @DPTR,A
    CALL WAIT_DRQ              ; WAIT FOR DRQ
    CALL GET_DATA              ; GET 512 BYTES
;
    RET                        ; END READ_SECTOR

WAIT_READY:
    MOV DPTR,#STATUS_REG
    MOVX A,@DPTR
    JB ACC.1, ABORT1          ; IF ERROR, ABORT

WAIT1:

```

## SanDisk Application Note

```
    MOVX A,@DPTR
    ANL A,#0F0H
    CJNE A,#50H,WAIT1
    RET

ABORT1:    ; USER ABORT ROUTINE
           .
           .
           .

WAIT_DRQ:
    MOV DPTR,#STATUS_REG
    MOVX A,@DPTR
    JB ACC.1,ABORT2    ; IF ERROR, ABORT
WAIT2:
    MOVX A,@DPTR
    ANL A,#0F8H
    CJNE A,#58H,WAIT2
    RET

ABORT2:    ;USER ABORT ROUTINE
           .
           .
           .

GET_DATA:
    MOV DPTR,#BUF_ADDR
    CALL GET_256    ; GET 256 BYTES
    CALL GET_256
    RET
;
GET_256:
    MOV R5,#0    ; COUNTER, 256 BYTES
GET_LOOP:
    PUSH DPH
    PUSH DPL
    MOV DPTR,#DATA_REG
    MOVX A,@DPTR    ; GET A BYTE FROM FLASH MEMORY
    POP DPL
    POP DPH
    MOVX @DPTR,A    ; PUT THE BYTE IN USER BUFFER
    INC DPTR
    DJNZ R5,GET_LOOP
    RET

; SUBROUTINE TO WRITE 512 BYTES TO LBA 302H

WRITE_SECTOR:
    CALL WAIT_READY
;
    MOV DPTR,#SEC_COUNT_REG
    MOV A,#1    ; 1 SECTOR
    MOVX @DPTR,A
    INC DPTR    ; LBA 7-0
```

```
MOV A,#2
MOVX @DPTR,A
INC DPTR                                ; LBA 15-8
MOV A,#3
MOVX @DPTR,A
INC DPTR                                ; LBA 23-16
CLR A
MOVX @DPTR,A
INC DPTR
MOV A,#0E0H                             ; SET THE LBA BIT
MOVX @DPTR,A
INC DPTR                                ; COMMAND REGISTER
MOV A,#30H
MOVX @DPTR,A
CALL WAIT_DRQ                            ; WAIT FOR DRQ
CALL PUT_DATA                            ; WRITE 512 BYTES
;
RET                                       ; END WRITE_SECTOR

PUT_DATA:
MOV DPTR,#DATA_REG
CALL PUT_256                             ; WRITE 256 BYTES
CALL PUT_256
RET
;
PUT_256:
MOV R5,#0                                ; COUNTER, 256 BYTES
PUT_LOOP:
PUSH DPH
PUSH DPL
MOV DPTR,#BUF_ADDR
MOVX A,@DPTR                             ; GET A BYTE FROM USER BUFFER
POP DPL
POP DPH
MOVX @DPTR,A                             ; WRITE THE BYTE TO FLASH MEMORY
INC DPTR
DJNZ R5,PUT_LOOP
RET
```

## ***Appendix A References***

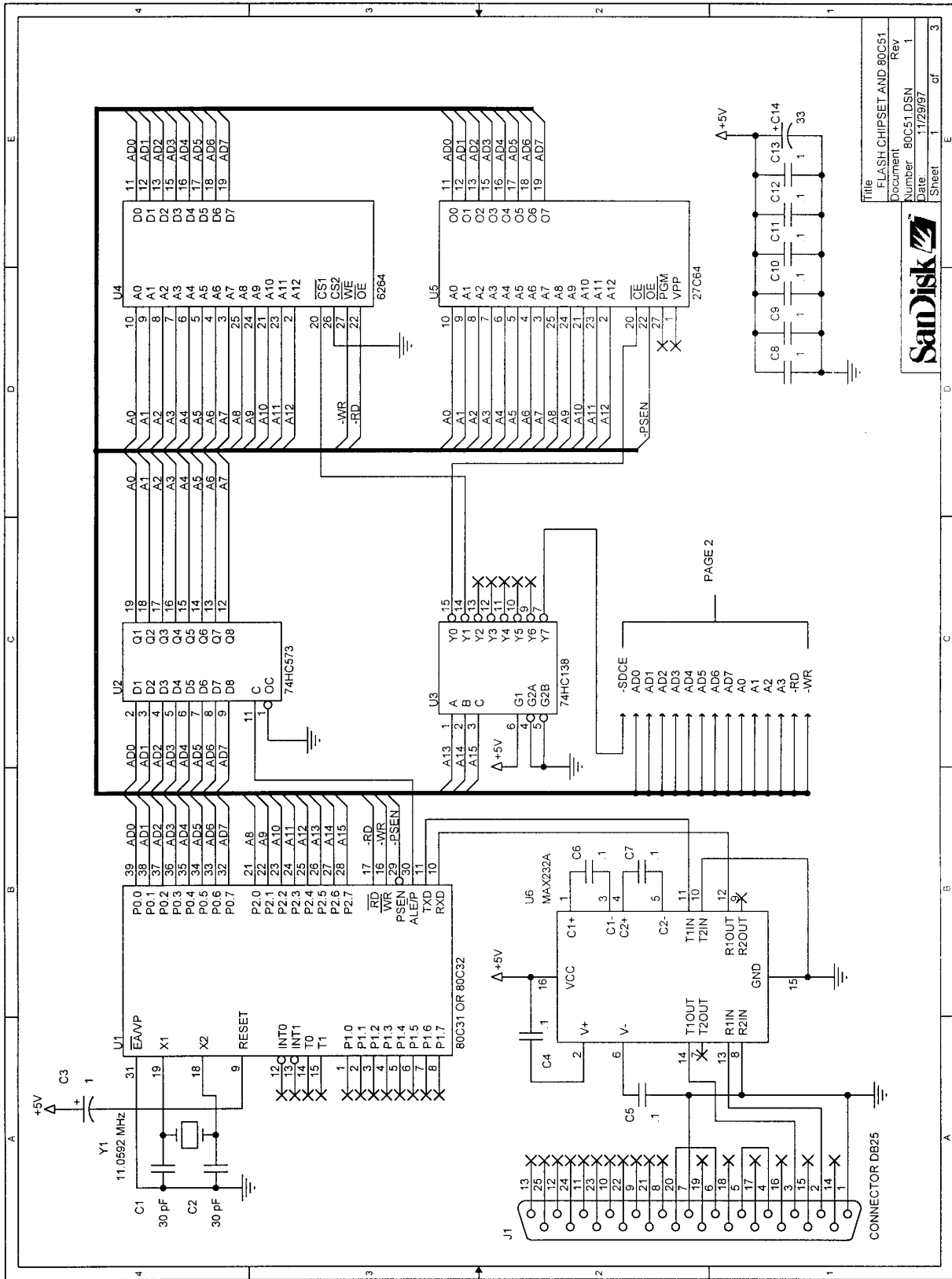
1. PC Card Standard - May 1996. Published by:  
PCMCIA (Personal Computer Memory Card International Association)  
2635 North First Street  
San Jose, CA 95134 USA  
Phone: +1-408-433-2273  
Fax: +1-408-433-9558  
E-Mail: office@pcmcia.org

and

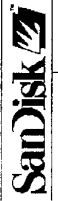
- JEIDA (Japan Electronic Industry Development Association)  
Kikai Shinko Kaikan, 3-5-8, Shibakoen  
Minato-ku, Tokyo 105, Japan  
Phone: +81-3-3433-1923  
Fax: +81-3-3433-6350
2. Flash ChipSet Product Manual  
SanDisk Corporation  
140 Caspian Court  
Sunnyvale CA 94089-0503 USA  
Main phone: +1-408-542-0500  
Applications: +1-408-542-0400  
Main Fax: +1-408-542-0503  
Marketing Fax: +1-408-542-0403

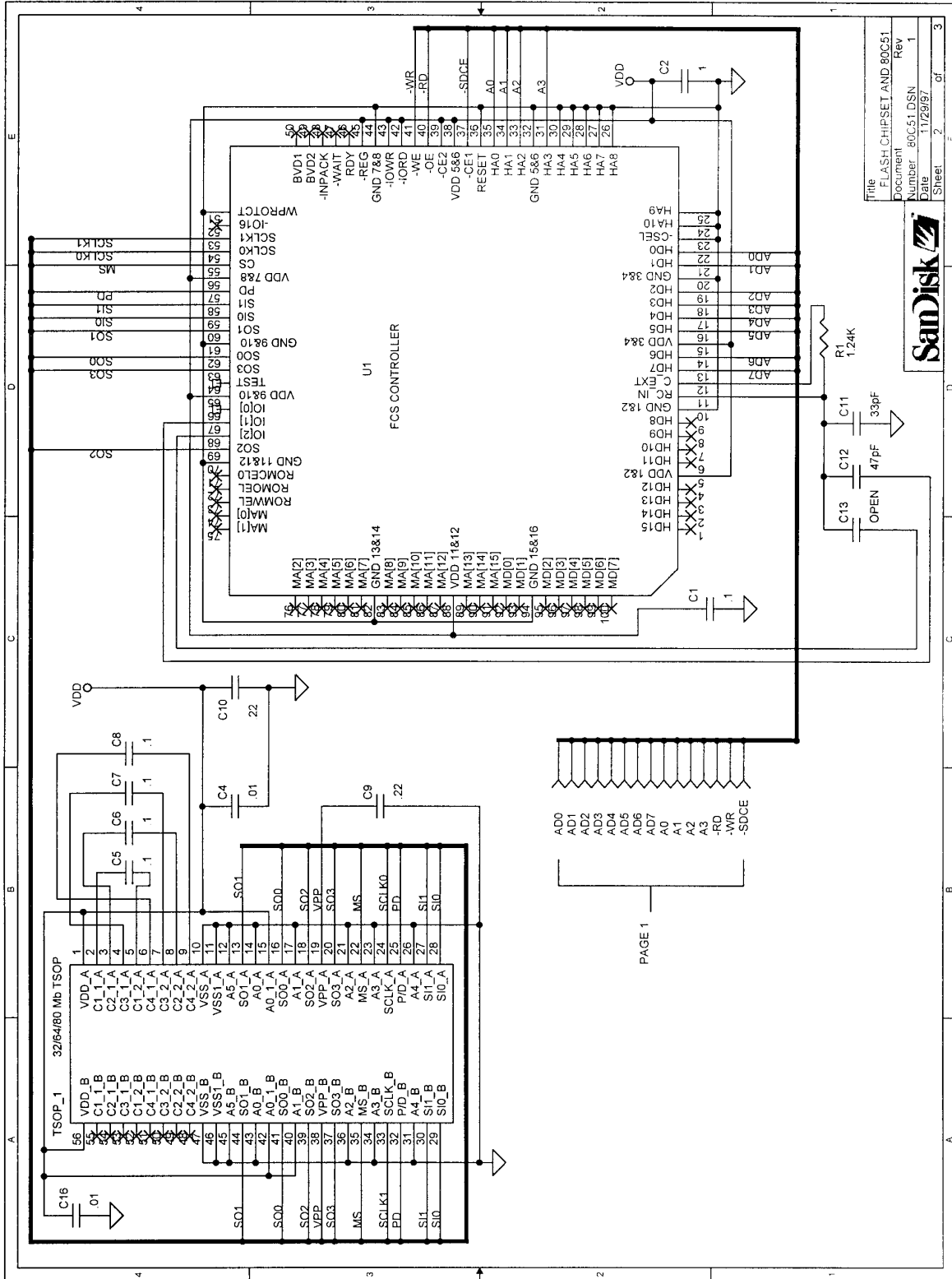
## ***Appendix B Schematic***

# SanDisk Application Note



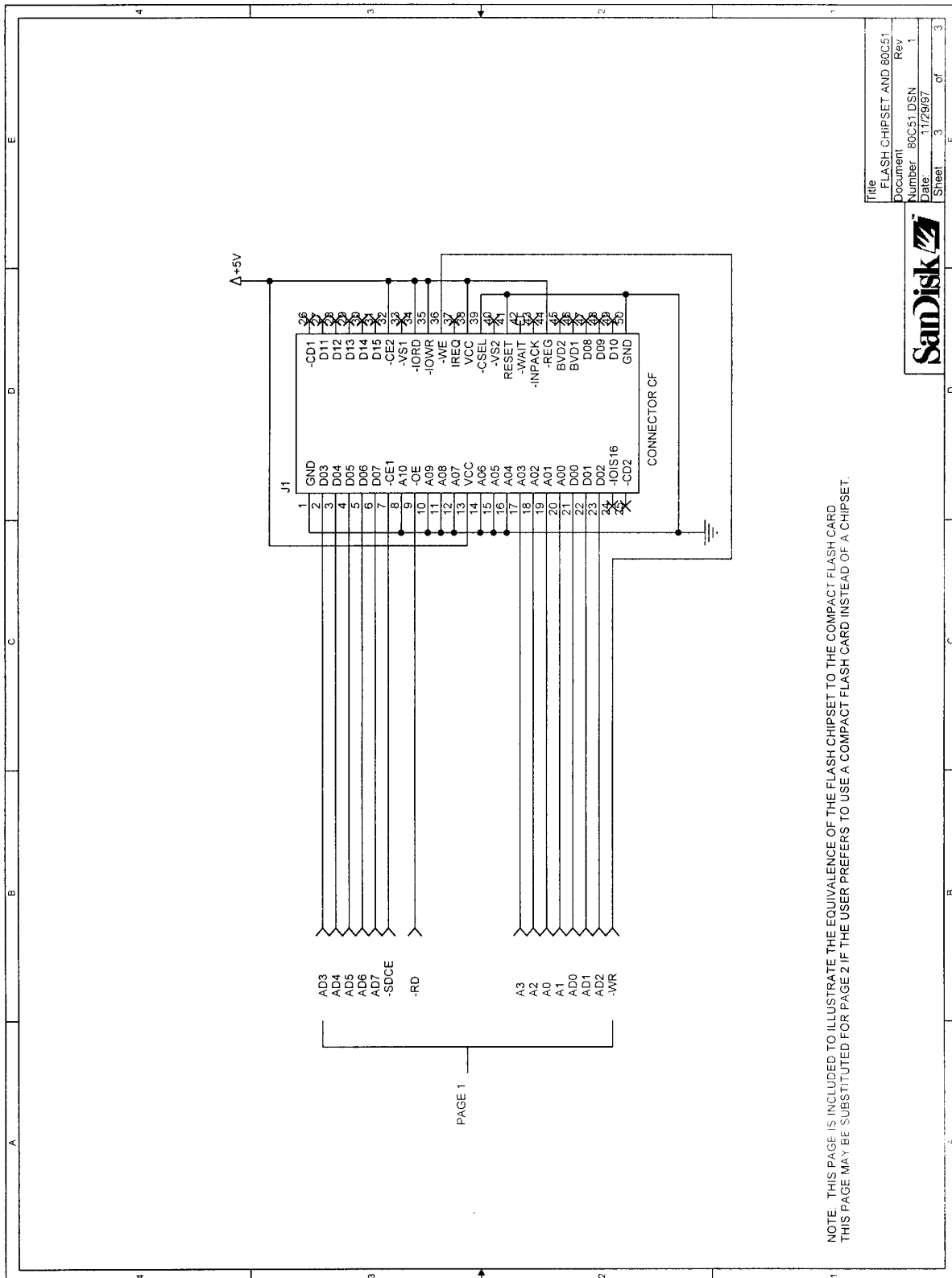
Title	FLASH CHIPSET AND 80C51
Document Number	80C51.DSN
Rev	1
Date	11/29/97
Sheet	1 of 3





File: FLASH.CHIPSET.AND.80C51  
 Document Number: 80C51.DSN  
 Date: 11/29/97  
 Sheet: 2 of 3





NOTE: THIS PAGE IS INCLUDED TO ILLUSTRATE THE EQUIVALENCE OF THE FLASH CHIPSET TO THE COMPACT FLASH CARD.  
THIS PAGE MAY BE SUBSTITUTED FOR PAGE 2 IF THE USER PREFERS TO USE A COMPACT FLASH CARD INSTEAD OF A CHIPSET.

Title	FLASH CHIPSET AND 80C51	Sheet	3	of	3
Document Number	80C51.DSN	Rev	1		
Date	11/29/97				

