December 2000

D12 Mass Storage Kit

(USB to CompactFlash™ or IDE)

D12 Mass Storage Kit (USB-to-CompactFlash/IDE)

Table of Contents

D12 Mass Storage Kit (USB-to-CompactFlash/IDE)

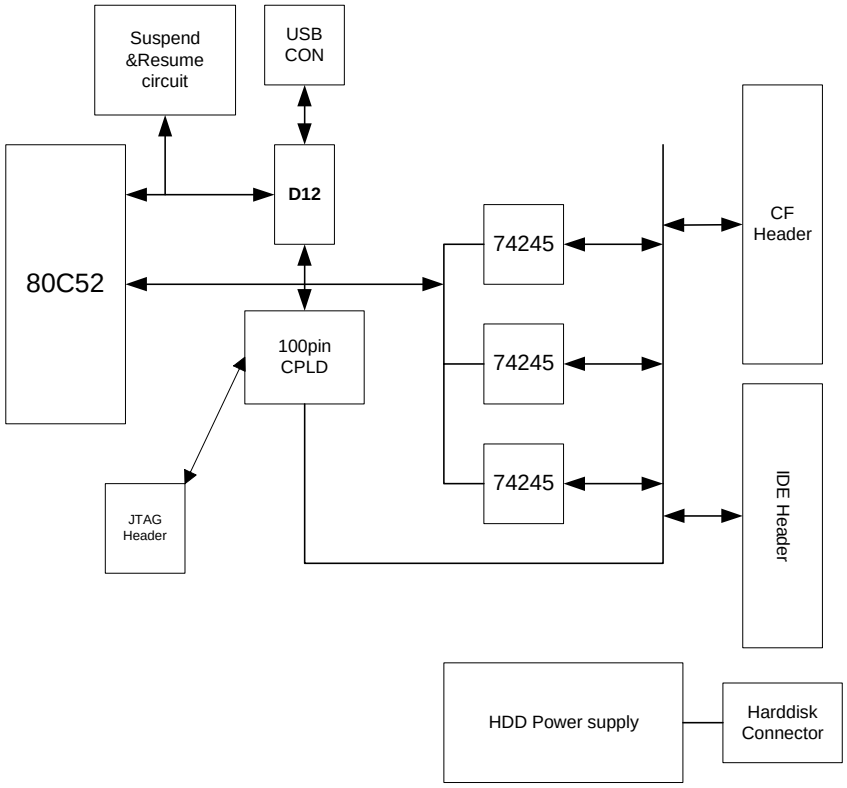
Overview

This figure shows the printed circuit board (PCB) assembly of a USB mass storage kit designed using the Philips USB 1.1 parallel interface device, PDIUSBD12. The current configuration for this mass storage kit supports only one peripheral, that is, either the CompactFlash (CF) card or the IDE 3.5” hard disk can be connected at any one time.

This mass storage kits complies with USB Mass Storage Class with sub class of USB BULK-only transport protocol.

The following block diagram illustrates the mass storage kit:

D12 Mass Storage Kit (USB-to-CompactFlash/IDE)



This architecture shows a generic interface control that consists of two techniques to control IDE peripherals. It consists of an 8052 microcontroller with RAM size of 256 bytes, operating at its maximum frequency of 24 MHz.

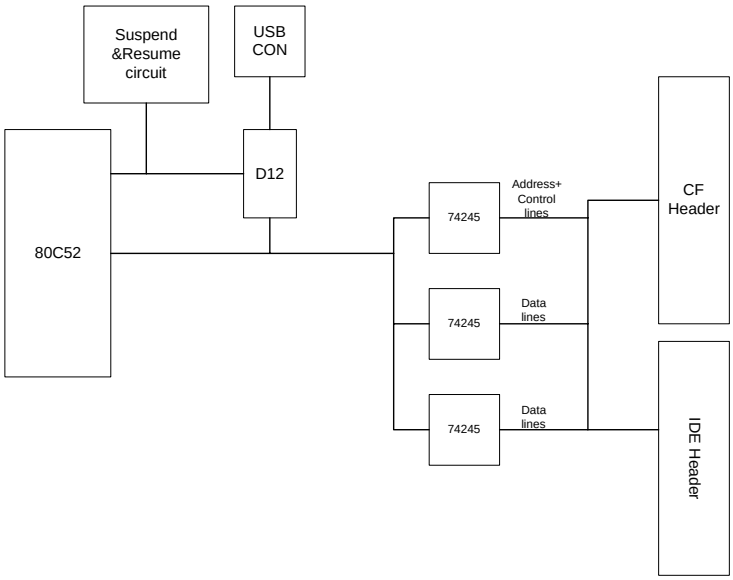
A switching power supply regulator is provided to step down the adapter provided in the Mass Storage kit. It allows 12 V to be stepped down to 5 V without having massive efficiency reduction, unlike that of a linear power supply.

This controller can control the IDE through a simple buffer (3 units of 74LS245), which results in slower throughput. Another technique is to access the IDE through memory mapping using CPLD.

D12 Mass Storage Kit (USB-to-CompactFlash/IDE)

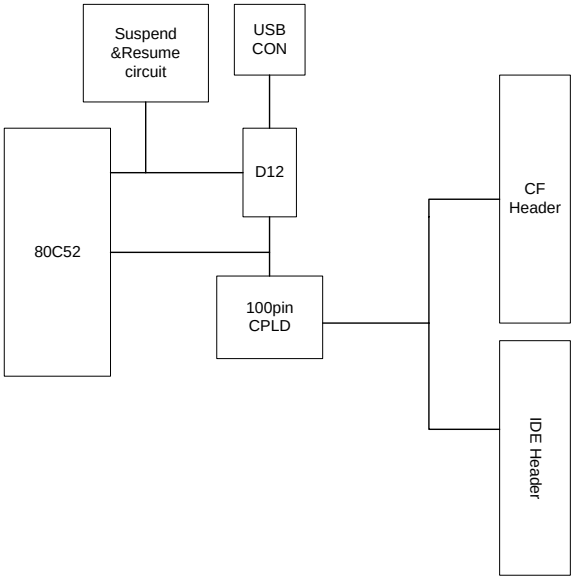
Technique 1

This diagram illustrates a simple method of accessing IDE devices. All registers and data access are emulated by 8051 I/O ports; therefore, the throughput is very slow.



D12 Mass Storage Kit (USB-to-CompactFlash/IDE)

Technique 2



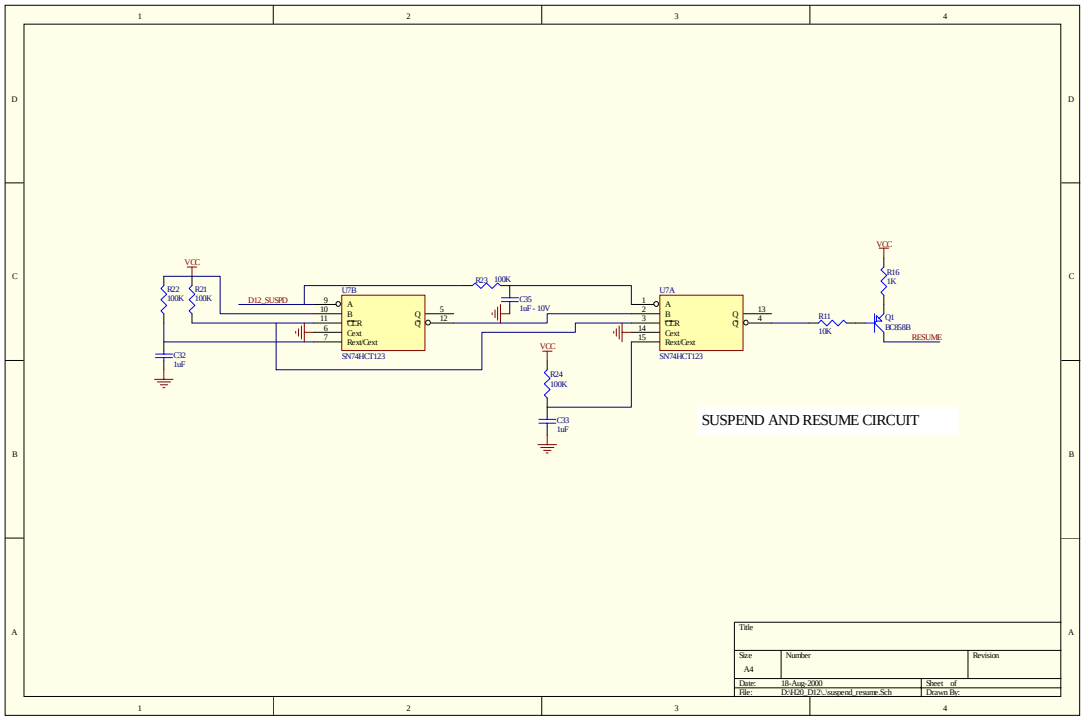
The current board configuration uses this technique, in which the PIO address of the ATA drive is memory mapped to the on-board CPLD:

Memory Location	Address Content
0x4000H	D12_COMMAND
0x4001H	D12_Data
0x8000H	IDE_WR_1F0(Data register write)
0x8002H	IDE_RD_1F0(Data register read)
0x8004H	IDE_1F1(error/feature register)
0x8005H	IDE_1F2(sector count register)
0x8006H	IDE_1F3(sector number register)
0x8007H	IDE_1F4(cylinder low register)
0x8008H	IDE_1F5(cylinder high register)
0x8009H	IDE_1F6(drive/head register)
0x800AH	IDE_1F7(status register)
0x800BH	IDE_3F6(alternate status register)
0x800CH	IDE_3F7(driver address register)

D12 Mass Storage Kit (USB-to-CompactFlash/IDE)

Suspend and Resume Circuit

A suspend and resume circuit has been incorporated to allow the device to enter suspend mode and yet achieve the restricted suspend current stated in the USB Specification Rev. 1.1.



The implementation forces the microcontroller unit (MCU) to enter power-down mode, in which its oscillator will stop working. Only a hard reset will recover the MCU operation. From the circuit, the resume signal is connected to the reset circuit and upon the PC recovering from suspend stage will cause PDIUSBD12 to toggle the suspend signal by which the resume signal will be asserted by the circuit.

Only a board with I/O mode access—in this case technique 1—can achieve the 500 µA suspend current. In the CPLD version, it is not possible to achieve the USB suspend current spec because CPLD cannot be disabled and it draws more than 100mA of current during operation.

During PC suspend mode, the power to the CompactFlash card will also be cut off.

D12 Mass Storage Kit (USB-to-CompactFlash/IDE)

Mass Storage Protocol

The flow of communication for Mass Storage class is as follows:

1. Host sends command wrapped in Mass Storage Class Transport protocol.
2. Within the transport protocol, the operation code is wrapped in SCSI command protocol.
3. When the message reaches the hardware, the controller will decode the message and translate it into ATA interface protocol to communicate with the IDE peripheral.

Technical Overview of USB Mass Storage Class—Bulk-Only Transport

The following is an extract from the specification of Bulk-Only Transport:

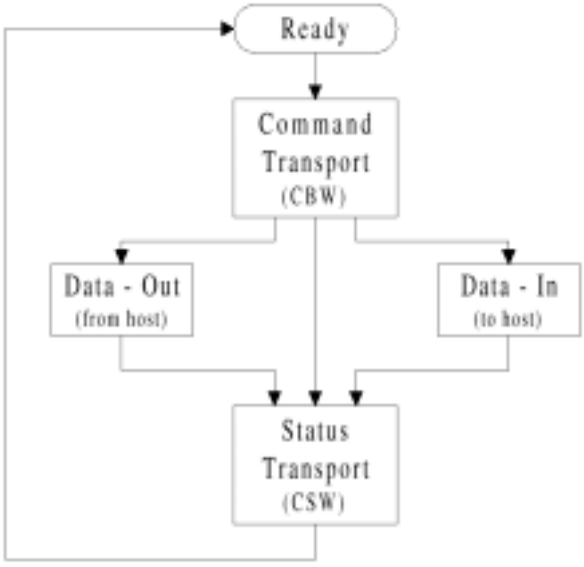


Figure 1 - Command/Data/Status Flow

Command Block Wrapper (CBW) - A packet containing a command block and associated information.

Command Status Wrapper (CSW) - A packet containing the status of a command block.

Data-In - Indicates a transfer of data IN from the device to the host.

Data-Out - Indicates a transfer of data OUT from the host to the device.

D12 Mass Storage Kit (USB-to-CompactFlash/IDE)

Command Block Wrapper (CBW)

bit / Byte	7	6	5	4	3	2	1	0
0-3	*dCBWSignature*							
4-7	*dCBWTag*							
8-11 (08h-0Bh)	*dCBWDataTransferLength*							
12 (0Ch)	*bmCBWFlags*							
13 (0Dh)	Reserved (0)				*bCBWLUN*			
14 (0Eh)	Reserved (0)				*bCBWCBLength*			
15-30 (0Fh-1Eh)	*CBWCB*							

The CBW shall start on a packet boundary and shall end as a short packet with exactly 31 (1FH) bytes transferred. Fields appear aligned to byte offsets equal to a multiple of their byte size. All subsequent data and the CSW shall start at a new packet boundary. All CBW transfers shall be ordered with the LSB (byte 0) first (little endian).

PACKET #	F	SYNC	OUT	ADDR	ENDP	CRC5	EOP	IDLE
0	S	00000001	0x87	3	1	0x07	3.00	2

PACKET #	F	SYNC	DATA1	DATA	CRC16	EOP	IDL
1	S	00000001	0xD2	0000: 55 53 42 43 28 E8 31 FE 0008: 00 02 00 00 80 00 0A 28 0016: 00 00 00 00 00 00 00 01 0024: 00 00 00 00 00 00 00	0xBA5D	3.00	4

PACKET #	F	SYNC	ACK	EOP	IDLE
2	S	00000001	0x4B	2.75	47641

PACKET #	F	SYNC	IN	ADDR	ENDP	CRC5	EOP	IDLE
3	S	00000001	0x96	3	2	0x1E	3.00	3

PACKET #	F	SYNC	DATA1	DATA	CRC16	EOP	IDL
4	S	00000001	0xD2	0000: FA BE 00 7C BF 00 7A B9 0008: 00 01 FC 0E 1F 0E 07 F3 0016: A5 EA 16 7A 00 00 BB BE 0024: 7B 33 C9 80 3F 80 75 06 0032: FE C5 8B F3 EB 07 80 3F 0040: 00 75 02 FE C1 83 C3 10 0048: 81 FB FE 7B 72 E5 83 F9 0056: 04 74 0B 81 F9 03 01 74	0xA765	2.75	7

PACKET #	F	SYNC	ACK	EOP
5	S	00000001	0x4B	3.00

Read command

D12 Mass Storage Kit (USB-to-CompactFlash/IDE)

From this USB traffic with reference to packet 1, one can decode the contents as

- 55 53 42 43 is the CBW signature followed by
- 28 E8 31FE is the CBWTag, which is generated by the host.
- 00 02 00 00 is the CBWDateTransferLength, which for this case is 0000,0200H = 512Bytes.
- 80 is the Flag followed by
- 00 is the CBWLUN followed by
- 0A is the CBWCBLength meaning the command length is 10Bytes long.

For the command block, see the next section on SCSI.

- 28 equal to read command
- 00 equal to reserved
- 00 00 00 00 equal to the logical address
- 00 00 00 01 equal to the transfer length and for this case equal to 1 sector (512Bytes)

Command Status Wrapper (CSW)

bit / Byte	7	6	5	4	3	2	1	0
0-3	*dCSWSignature*							
4-7	*dCSWTag*							
8-11 (8-Bh)	*dCSWDataResidue*							
12 (Ch)	*bCSWStatus*							

The CSW shall start on a packet boundary and shall end as a short packet with exactly 13 (0DH) bytes transferred. Fields appear aligned to byte offsets equal to a multiple of their byte size. All CSW transfers shall be ordered with the LSB (byte 0) first (little endian).

PACKET #	F	SYNC	IN	ADDR	ENDP	CRC5	EOP	IDLE
0	S	00000001	0x96	3	2	0x1E	3.00	3

PACKET #	F	SYNC	DATA1	DATA	CRC16	EOP
1	S	00000001	0xD2	0000: 55 53 42 53 28 E8 31 FE 0008: 00 00 00 00 00	0xB902	2.75

From this USB traffic at packet 1:

- 55 53 42 53 will be the CSWsignature followed by
- 28 E8 31 FE equals the CSWTag generated earlier from the host.
- 00 00 00 00 equals the CSWDataResidue
- 00 indicates the CSWstatus, which for this case is equal to OK.

SCSI protocol

Table 2 — Typical CDB for 10-byte commands

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE							
1	Reserved			SERVICE ACTION (if required)				
2	(MSB)							
3				LOGICAL BLOCK ADDRESS (if required)				
4								
5								(LSB)
6	Reserved							
7	(MSB)			TRANSFER LENGTH (if required) PARAMETER LIST LENGTH (if required) ALLOCATION LENGTH (if required)				
8								(LSB)
9	CONTROL							

The field description table shows the SCSI primary commands structure.

In typical usage, the **OPERATION CODE** field contains the code value identifying the operation being requested by the CDB (command descriptor block).

The **LOGICAL BLOCK ADDRESS** field contains a logical block address. The TRANSFER LENGTH field specifies the number of bytes, logical blocks, or other command-specific units to be transferred.

The **PARAMETER LIST LENGTH** field specifies the number of bytes of command parameter data to be sent from the application client to the device server.

The **ALLOCATION LENGTH** field specifies the number of bytes set aside by the application client to receive command parameter data from the device server. The contents of the CONTROL field are defined in SAM-2.

D12 Mass Storage Kit (USB-to-CompactFlash/IDE)

Table 3 – READ(10) command format

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (28h)							
1	Reserved							
2	(MSB)							
3				LOGICAL BLOCK ADDRESS				
4								
5								(LSB)
6	Reserved							
7	(MSB)							
8				TRANSFER LENGTH				(LSB)
9	CONTROL = 00h							

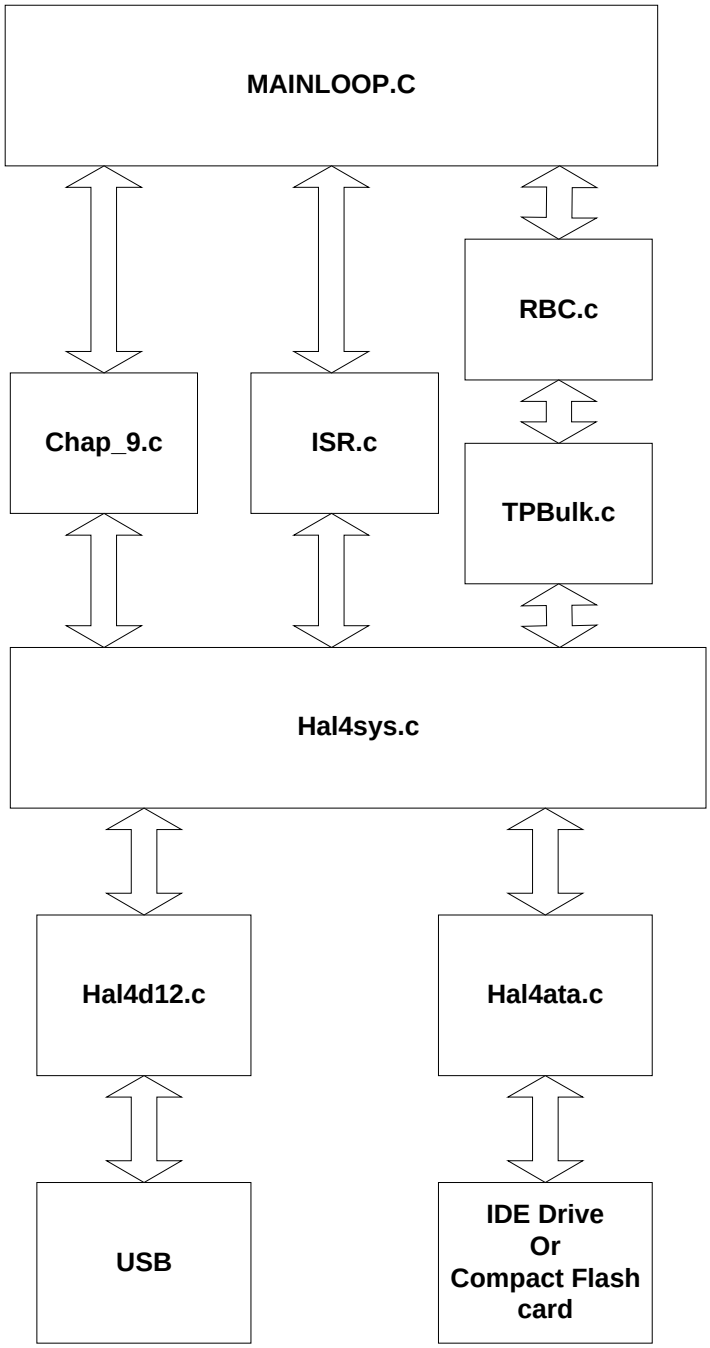
This is an example of one of the SCSI primary read commands.

The command list that is supported in this mass storage kit is:

1. Inquiry
2. Requests sense
3. Test unit ready
4. Mode sense
5. Mode select
6. Prevent allow medium removal
7. Read buffer
8. Write buffer

D12 Mass Storage Kit (USB-to-CompactFlash/IDE)

Firmware Programming Guide



D12 Mass Storage Kit (USB-to-CompactFlash/IDE)

1. Hardware Abstraction Layer for D12 – Hal4d12.C

This is the lowest layer code in the firmware. It performs hardware dependent I/O access to the PDIUSBD12.

2. Hardware Abstraction Layer for ATA – Hal4ata.C

This part of the code provides handles to the IDE protocol interface.

3. Hardware Abstraction Layer for System – Hal4sys.C

This part of the code is the lowest layer for IDE peripherals.

4. Reduced Block Command Layer – RBC.C

The subset of SCSI primary commands. This code decodes SCSI commands from the Host.

5. Interrupt Service Routine - ISR.C

This part of the code handles interrupts generated by the PDIUSBD12. It retrieves data from PDIUSBD12's internal FIFO to MCU memory, and sets up the proper event flags to inform Main Loop program for processing.

6. Main Loop - MAINLOOP.C

The Main Loop checks the event flags and passes to appropriate subroutine for further processing.

7. Protocol Layer - CHAP_9.C

The Protocol layer handles standard USB device requests.

8. Bulk Only Transport Layer - TPBulk.C

This code handles the USB mass storage class protocol.

Jumper Settings

Jumper	Description
JP6	CSEL- low (default on)
JP7	CSEL – high (default off)
JP8	CPLD IO supply – 5V (default on)
JP9	CPLD IO supply – 3.3V (default off)

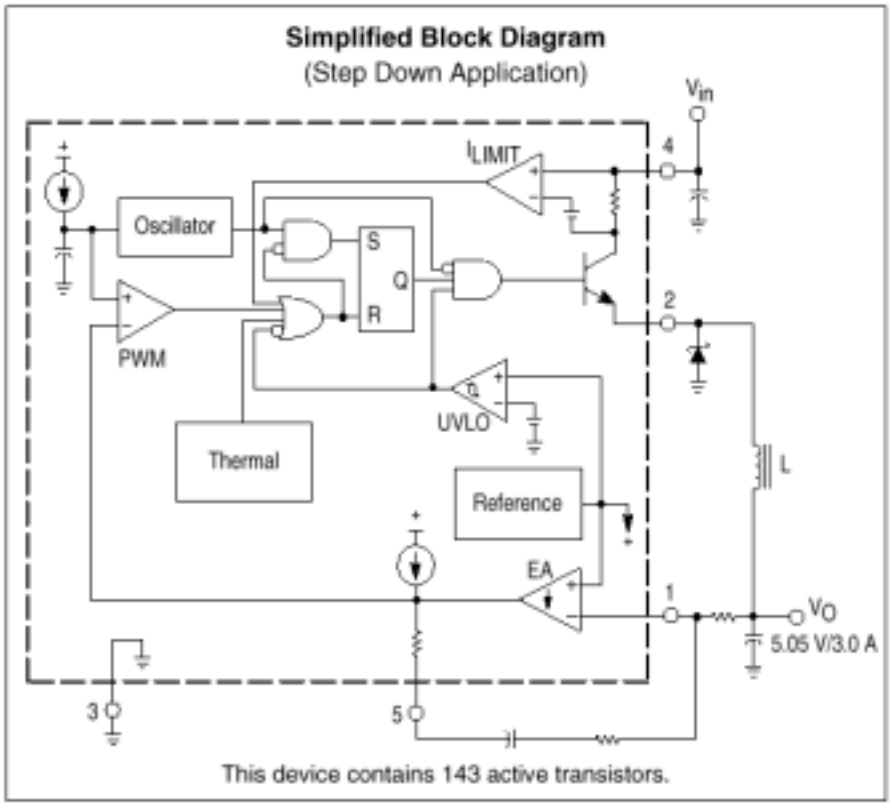
The setting for CSEL on the IDE drive is default low because the current board supports only a single IDE drive. JP7 is installed for you to run both the IDE drive and the CompactFlash card simultaneously.

For the CPLD IO supply, the default is 5V as the current IDE drive and CompactFlash card are 5V tolerant. Only when an attempt to interface a 3.3V IDE device will setting the CPLD IO supply to 3.3V be needed.

D12 Mass Storage Kit (USB-to-CompactFlash/IDE)

HDD Power Supply

A DC-DC power supply is included in the mass storage kit for powering the external HDD. As a typical HDD requires both 5V and 12V supplies, a step-down regulator is included.



The MC33166 series consists of high performance fixed frequency power switching regulators that contain the primary functions required for DC-to-DC converters. It is specifically designed to be incorporated in step-down and voltage-inverting configurations with a minimum number of external components and can also be used cost effectively in step-up applications. Its has output Switch Current in excess of 3.0 A using Fixed Frequency Oscillator (72 kHz) with On–Chip Timing. It provides 5.05 V Output without External Resistor Divider with operation voltage from 7.5 V to 40 V.

D12 Mass Storage Kit (USB-to-CompactFlash/IDE)

BOM

The storage kit comes along with a power adapter (12V output) which has wide input voltage range type. It has also hard-disk cable (40-pin flat cable) and power cable (4 wires). The PCBA has the following components:

S/No	Designator	Part number	Description
1	22 pF	C3 C4 C5 C6 C7 C9	Ceramic capacitor 0805
2	0.1 µF	C10 C11 C12 C13 C14 C15 C16 C17 C18 C19 C20 C21 C22 C23 C24 C25 C26 C27 C28 C29 C30 C31	Ceramic capacitor 0805
3	1.0µF16V	C2	Ceramic capacitor
4	100K	R21 R22 R23 R24	Resistor 0805
5	10K	R? R? R10 R11 R12 R13 R14 R7 R8 R9	Resistor 0805
6	18R	R1 R2	Resistor 0805
7	190µH	L3	inductor
8	1K	R15 R16 R17 R18 R19	Resistor 0805
9	1M	R3 R4 R5	Resistor 0805
10	1N5822	D3	diode
11	1µF	C32 C33 C36	capacitor
12	1µF - 10V	C35	capacitor
13	2200µF	C43	capacitor
14	24MHz	X2	Crystal
15	330	R64	Resistor 0805
16	3mm GREEN LED	D1	LED
17	4.7K	RP1 RP2 RP3 RP4	Resistor 0805
18	4.7µF16V	C1 C8	capacitor
19	470R	R6	Resistor 0805
20	5.6K	R26	Resistor 0805
21	640	R37	Resistor 0805
22	68K	R30 R31	Resistor 0805
23	6MHz	X1	Crystal
24	80C52	U2	MCU
25	BC848	Q3 Q4	transistor
26	BC858B	Q1	transistor
27	BLM11B121SB	FB2 FB3	Ferrite bead
28	BLM31P121SGP	FB1	Ferrite bead
29	BNC	C40	DC jack connector
30	CF_HEADER 25X2	JP3	Compact flash connector
31	HardDisk Power Connector	JP5	HDD power connector
32	IDEHEADER 20X2	JP1	
33	JTAG HEADER 6	JP4	
34	JUMPER	JP8 JP9	
35	LED	D7	
36	LP2985IM5-33	Q5	3.3V regulator
37	MC33166	C41	5V regulator for HDD
38	MMC Connector	JP2	
39	PDIUSBD12	U1	USB interface
40	Si2301DS	Q2	MOSFET
41	SN74HCT123	U7	
42	SN74HCT245	U3 U4 U5	

D12 Mass Storage Kit (USB-to-CompactFlash/IDE)

S/No	Designator	Part number	Description
43	SN74LVT244MTC	U8	
44	USBCON TYPE A	CON1	
45	XC95108-7PQ100C(100)	U6	CPLD
46	Tact Switch	SW1	Reset button

APPENDIX

IDE hard disk setup for USB Mass Storage kit

Setup Notes:

1. Perform FORMAT or FDISK hard disk using a normal PC.
   (set HDD to slave mode, and connect in parallel to original HDD in the PC)
2. Remove HDD from system and ensure that the hard disk is set to master mode.
3. Connect the HDD as shown in the diagram.
4. Power on hard disk for at least 5 seconds (by connecting a 12 V adapter to the DC jack on the PCB), and then connect the USB mass storage kit to host.
5. If GoodLink LED (D1) is not lit, reset the mass storage system by pressing SW1.
   (this may happen because of hard disk is slow in powering up).
   The D7 LED indicates Prevent Media Removal status.
6. During data transfer, the LED is lit to indicate that the CompactFlash card must not be removed from board.