CTAN002: Multi-Word DMA Compatibility for Cactus Industrial-Grade Flash-Storage Products

Covered Products: CF/PC Cards, SSD, DOM

1. Introduction

The standard Cactus industrial-grade flash-storage products supports Multi-Word DMA Mode 0, 1 and 2. DMA modes alleviates the host processor from managing data transfers by allowing direct data transfers between ATA bus devices and the host memory via a DMA controller.

However, there are some customers who have trouble with DMA mode transfers. This document serves as a troubleshooting guide for DMA mode usage in our products.

2. DMA Mode Prerequisites

There are four prerequisites for DMA support in a computer or embedded system with Cactus flash-storage devices.

1. The Cactus device must report DMA support through IDENTIFY DEVICE.
2. For PC Cards or -201/3 CF Cards, the host hardware must use them in True IDE mode (-301/3 CF Cards support DMA operation in PC Card modes also).
3. Under True IDE mode, the host must connect DMARQ and _DMACK signals to the Cactus device.
4. The host firmware must enable DMA mode transfers at device initialization phase, and software must perform data transfers with ATA DMA read/write commands.

DMA will not operate with any prerequisite missing, and may also cause problems ranging from slow performance to complete functional failure on the target system.

3. Troubleshooting Guide

The following steps serve as a troubleshooting guide for DMA-related operation and performance issues encountered with Cactus flash storage devices. These are based on real world cases encountered and solved by our technical support team.

1. The device is slow and processor usage during data transfer is almost 100%.
2. The BIOS/firmware cannot recognize the device, and indicates it is busy.
3. The device is recognized by the BIOS or the firmware, but read/write always fail.
4. It is impossible to enable DMA under Microsoft Windows.
5. Microsoft Windows disables DMA even after enabling it under Device Manager.

The standard Cactus product configuration with model number suffix -201/3 or -301/3 supports Multi-word DMA modes. If the model number of the product has the suffix -202/4 or -302/4, then it does not support any advanced data transfer modes including Multi-word DMA. Contact your local Cactus distributor to obtain Multi-word DMA-compatible versions of Cactus devices.

For PC Card and -201/3 CF cards, DMA support is available only when the device is under True IDE mode. To enable True IDE mode operation, the CF/PC Card must be connected directly to the host ATA bus (via an adapter), and the device must have _OE pin grounded when powered on.

The first step to troubleshoot DMA issues is to check whether the host hardware supports DMA.

For CF and PC Card ATA devices this is the most important step. Multi-word DMA support is a late addition to the CF and PC Card ATA specifications, and older devices and hosts may not support Multi-word DMA modes due to their hardware designs.

To check for hardware DMA compatibility issues, take the following steps:

1. For CF to IDE adapter, use a continuity meter to check pin 43 (DMARQ) of the CF connector is connected to pin 21 (DMARQ) of the IDE connector.

   Also check pin 44 (_DMACK) of the CF connector is connected to pin 29 (_DMACK) of the IDE connector.

2. For PC Card to IDE adapter, use a continuity meter to check pin 60 (_INPACK) of the PCMCIA connector is connected to pin 21 (DMARQ) of the IDE connector.

   Also check pin 61 (_REG) of the PCMCIA connector is connected to pin 29 (_DMACK) of the IDE connector.

DMARQ and _DMACK on the host interface must not be hardwired to GND and Vcc respectively. These signals must be connected between CF connector and the ATA connector.

The second step to troubleshoot DMA-related operation and performance issues is to check whether BIOS or system firmware supports Multi-word DMA and enables it during device initialization.

1. If the application platform is an x86-based PC or PC-104 device, it is possible to check whether the BIOS has options to enable DMA transfers at on-board ATA channels. You can safely assume that very old systems with ISA-based IDE adapters will not support DMA transfers. Also if the BIOS does not show any configurable DMA transfer options, it is safe to assume that DMA is unsupported by the BIOS.

2. If the BIOS has options for selecting the fastest DMA transfer modes, always select Multi-word DMA Mode 2 for Cactus products connected to the ATA channel. Sometimes leaving the setting to “Auto” may cause performance problems due to the BIOS incorrectly detecting the highest available DMA mode for the Cactus device.

The third troubleshooting step involves checking whether the operating system supports DMA mode transfers.

1. For Microsoft operating systems, MS-DOS itself supports DMA mode transfers only if the BIOS also supports DMA transfers.

   DMA transfers are progressively disabled if Microsoft Windows 2000 and later encounters more than 6 CRC errors or data transfer timeouts during data transfers. The gradual degradation will eventually fall back to PIO mode data transfer.

   CRC errors usually occur when poor quality or damaged cables corrupt signals between device and host. Replacing any suspicious ATA cables between the device and the host usually enables DMA transfers to run correctly.

   For 2.5” SSD with 2.5” to 3.5” adapter board, check the adapter board connections to ensure DMARQ and _DMACK signals are actually connected between the 44 and 40-pin connectors.

2. For Linux operating systems, make sure DMA transfer support is enabled when building the kernel IDE disk driver. Also the following options must be enabled for custom-built kernels for DMA mode disk transfer support:

```
Generic PCI bus-master DMA support:
CONFIG_BLK_DEV_IDEDMA
```

```
Use DMA by default when available:
CONFIG_IDEDMA_AUTO
```

3. If a ready-built kernel is used and the vendor has confirmed the kernel supports DMA, check whether the kernel is using DMA transfer by entering the following

command when logged onto the system as "root":

```
# /sbin/hdparm /dev/hda
```

```
/dev/hda:
multcount    =  0 (off)
I/O support  =  0 (default 16-bit)
unmaskirq    =  0 (off)
using_dma    =  1 (on)
keepsettings =  0 (off)
nowerr       =  0 (off)
readonly     =  0 (off)
readahead    =  8 (on)
geometry     = 523/255/63, sectors =
8406720, start = 0
```

where /dev/hda points to the Cactus device on the system. If the kernel has enabled DMA on the Cactus device, then the using_dma option should be 1.

4. To enable DMA support on a running Linux system, type the following command under "root" account:

```
# /sbin/hdparm -d1 /dev/hda
```

```
/dev/hda:
        setting using_dma to 1 (on)
        using_dma    =  1 (on)
```

This command must be repeated when the system is started. We recommend performing read/write tests on a development system to ensure DMA transfers are reliable.

5. The Linux kernel can be configured for DMA by passing boot options ide0=dma or ide1=dma . For information on passing boot options, consult boot loader documentation.

4. Verifying DMA Operations

To verify whether the target system is running the Cactus device under DMA mode, the designer can always check the following on the target system:

1. Processor utilization during ATA access. If DMA is used, processor utilization should be significantly lower than PIO transfers.
2. Using benchmarking programs or timing disk read/write operation with fixed-size data can also indicate whether DMA mode is used for data transfer.
3. If a logic analyzer is available, then it is a good idea to intercept data transfer between ATA host controller and Cactus device to verify whether DMA read/write commands are actually used for data transfer. There are situations where the electrical connections for DMA transfers were correct yet the system software or drivers do not actually use DMA.

We have observed that some adapters have left out DMA connections, since some manufacturers assumed older CF/PC Cards do not support DMA. It is possible to manually modify these adapters by connecting the DMA pins by soldering wires to both ends for connection. Consult adapter documentation for further information.

In the case where both Cactus device and system firmware/software supports DMA, yet the hardware connections for DMA operation are missing, an intermediate solution to enable normal operation without DMA mode is to completely disable DMA transfer modes in the BIOS and force the Cactus device to run in PIO mode.

5. Version History

Version	***Date***	***Change***
1.00	September 7, 2006	Initial Version
1.01	June 3, 2008	updated for -300 series products