Proposed Draft

Serial ATA International Organization

Version 0
March 12, 2007

Serial ATA Revision 2.6 Design Guide # 001 Title : Data Transmission Error Reporting Variance

Author Information

Author Name	Company	Email address
Joseph Chen	Samsung	joseph.chen@samsung.com

Workgroup Chair Information

Workgroup (Phy, Digital, etc…)	Chairperson Name	Email address
Digital	Brian Dees	Brian.m.dees@Intel.com

Document History

Version	Date	Comments
0	03/12/2007	Initial release. Re-release of previously ratified design guide 005 to log against SATA Revision 2.6.

1 Introduction

This design guide describes data transfer options when the device encounters an error. Multiple ways of error handling exist in today's implementation. The reason behind multiple implementations is that neither the ATA nor Serial ATA specifications provide requirements or guidance as to the device response expectations in the scenarios described here. These various methods apply to all DMA and First Party DMA commands. Although the device error often occurs on read, write commands may have errors as well. Therefore, the data transfer options described in this document apply to both read and write commands.

The SATA host must handle all of the data transfer options listed. In other words, the host should expect the device to behave in different ways when a read or write error occurs. Although there are variations in error handling today, it is intended for future designs to follow consistent methods.

Read command errors may occur during a data transfer. Typical read command failures are due to ECC error. When the device is not able to retrieve data from the media, the device should report an error for the command. Write command errors may also occur during the host-to-device data transfer. Write errors may happen when the disk is not able to locate the media location upon which to write the data due to abnormal conditions such as servo failure from the media, or external shock events. In either read or write cases, the device should complete the command successfully and should return error status.

2 Supplemental Information

The following sections describe the possible error recovery procedures for read and write commands which host controllers should be made aware of. DMA commands and NCQ commands are discussed in separate sections.

Disclaimer: the following descriptions are not meant to replace or modify in any way the state machines and descriptions in the Serial ATA standard, but instead are meant to provide sufficient information necessary to understand and apply this design guide. Refer to the Serial ATA standard for more details.

2.1 READ DMA Commands (Regular and Extended DMA Commands)

The READ DMA data transfer may be broken down into the following steps:

1. READ DMA command is issued by the host (Register FIS 27h)
2. Data is sent by the device (Data FIS 46h)
    a. NOTE: there may be a loop condition for step 2 above for transfers which require more than 1 Data FIS to transfer the entire set of data.
3. Status is returned by the device at the end of the command (Register FIS 34h)

If there is no error on a READ DMA command, the data transfer length should be exactly the same length as requested by the command. The data transfer may be broken into many Data FISes, until the data is completely sent (the transfer size may be long for extended commands; the total data length may be up to 64K sectors). The size of each Data FIS is determined by the device, and the maximum size may not exceed the maximum Data FIS size of 2048 DWORDs.

When a READ DMA command encounters an error reading data in the middle of the data transfer, the device may terminate the data transfer before sending all data that the host has requested. The device reports error information by transmitting a D2H Register FIS, and terminates the command with error.

2.2 WRITE DMA Commands (Regular and Extended DMA Commands)

The WRITE DMA data transfer may be broken down into the following steps:

1. WRITE DMA command is issued by the host (Register FIS 27h)
2. The device indicates it is ready to accept write data from the host (DMA Activate FIS 39h)
3. Data is sent by the host (Data FIS 46h)
    a. NOTE: there may be a loop condition for steps 2 & 3 above for transfers which require more than 1 Data FIS to transfer the entire set of data.
4. Status is returned by the device at the end of the command (Register FIS 34h)

If there is no error on a WRITE DMA command, the data transfer length should be exactly the same length as requested by the command. An additional step is required for a WRITE DMA command before the data transfer; the device sends the DMA Activate FIS to the host to signal it to start sending write data. The data transfer may be broken into many Data FISes until the data is completely sent (the transfer may be long for the extended commands; the total data length may be up to 64K sectors).

When a WRITE DMA command encounters an error writing the data in the middle of the data transfer, the device may terminate the data transfer before taking all data that the host has

requested. The device reports error information by transmitting a D2H Register FIS, and terminates the command with error.

2.3 READ FPDMA QUEUED Commands

The READ FPDMA QUEUED data transfer may be broken down into the following steps:

1. READ FPDMA QUEUED command is issued by the host (Register FIS 27h)
2. The command is queued and BSY is cleared by the device (Register FIS 34h)
3. The device indicates it is ready to send data to the host. (DMA Setup FIS 41h)
4. Data is sent by the device (Data FIS 46h)
    a. NOTE: there may be a loop condition for step 4 above for transfers which require more than 1 Data FIS to transfer the entire set of data.
    b. NOTE: there may be a loop condition for steps 3 & 4 above for transfers which require more than 1 DMA Setup FIS to transfer the entire set of data.
5. Status is returned by the device at the end of the command (Set Device Bits FIS A1h)

If there is no error on a READ FPDMA QUEUED command, the total data transfer length should be exactly the same length as requested by the command. The data transfer may be sent with many groups of data; each group starts with a DMA Setup FIS. The DMA Setup FIS describes the tag number of this data group, the buffer offset (for out-of-order data transfer), and the transfer count in bytes for this group. The transfer count may be up to 32-bits in length.

In each group of transfer, the data is broken down into one or more Data FISes. The size of each Data FIS is determined by the device, and the maximum size may not exceed the maximum Data FIS size of 2048 DWORDs. The total transfer length may be up to 64K sectors.

After all data has been sent to the host, the device completes this command (and, possibly, other queued commands) by sending the Set Device Bits FIS. If no error has occurred, the device sets the bit in the SActive field which corresponds to the tag number of the completed command. If any error occurs, the device sets the error bit in the Set Device Bits FIS and clears the bit in the SActive field which corresponds to the tag number of the completed command.

When a READ FPDMA QUEUED command encounters an error reading the data in the middle of the data transfer, the device may terminate the data transfer before sending all data that the host has requested. The device reports error information by transmitting a Set Device Bits FIS, and terminates the command with error.

When an error occurs, the host may expect that the transfer count for the command was not fulfilled. In other words, the data transfer to the host may be terminated earlier than the transfer count defined in the DMA Setup FIS. There is no required offset where the data transfer should be ended. In some cases these errors may be encountered before any data has been transferred, while others may occur in the middle of a data transfer. Devices may exhibit various error handling behaviors in these cases, including but not limited to the following:

1. Following the error condition, additional data may be transferred & padded with dummy values, but the transfer length is not equal to the count defined in the DMA Setup FIS.
2. Following the error condition, additional data may be transferred & padded with dummy values, such that the transfer length reaches the length defined in the DMA Setup FIS.
3. Data transfer is completed for the group of data defined by a DMA Setup FIS, but fail to transfer all groups to complete the total transfer length defined by the command.
4. Complete the entire data transfer for all requested data by padding with dummy data.

The host should expect and correctly handle all of the cases listed above at a minimum, while this list may not outline all possible behaviors. The data received by a host should not be used when there is an error in a READ FPDMA QUEUED command. If dummy values are transferred as a

part of the data, the device should also include a corrupt CRC to ensure a host does not incorrectly act on invalid transmitted data.

2.4 WRITE FPDMA QUEUED Commands

The WRITE FPDMA QUEUED data transfer may be broken down into the following steps:

1. WRITE FPDMA QUEUED command is issued by the host (Register FIS 27h)
2. The command is queued and BSY is cleared by the device (Register FIS 34h)
3. The device indicates it is ready to accept the data from the host (DMA Setup FIS 41h). If the device supports the Auto-Activate feature, no DMA Activate is required before the data transfer (skip to step 5).
4. The device indicates it is ready to accept write data from the host (DMA Activate FIS 39h)
5. Data is sent by the host (Data FIS 46h)
    a. NOTE: there may be a loop condition for steps 4 & 5 above for transfers which require more than 1 Data FIS to transfer the entire set of data.
    b. NOTE: there may be a loop condition for steps 3, 4 & 5 above for transfers which require more than 1 DMA Setup FIS to transfer the entire set of data.
6. Status is returned by the device at the end of the command (Set Device Bits FIS A1h)

If there is no error on a WRITE FPDMA QUEUED command, the total data transfer length should be exactly the same length as requested by the command. The data transfer may be sent with many groups of data; each group starts with a DMA Setup FIS. The DMA Setup FIS describes the tag number of this data group, the buffer offset (for out-of-order data transfer), and the transfer count in bytes for this group. The transfer count may be up to 32-bits in length.

In each group of transfer, the data is broken down into one or more Data FISes. The size of each Data FIS is determined by the device, and the maximum size may not exceed the maximum Data FIS size of 2048 DWORDs. The total transfer length may be up to 64K sectors.

After all data has been sent to the device, the device completes this command (and, possibly, other queued commands) by sending the Set Device Bits FIS. If no error has occurred, the device sets the bit in the SActive field which corresponds to the tag number of the completed command. If any error occurs, the device sets the error bit in the Set Device Bits FIS and clears the bit in the SActive field which corresponds to the tag number of the completed command.

When a WRITE FPDMA QUEUED command encounters an error writing the data in the middle of the data transfer, the device may terminate the data transfer before receiving all data that the host intends to write. The device reports error information by transmitting a Set Device Bits FIS, and terminates the command with error.

When an error occurs, the host may expect that the transfer count for the command was not fulfilled. In other words, the data transfer to the device may be terminated earlier than the transfer count defined in the DMA Setup FIS. There is no required offset where the data transfer should be ended.

The host should expect to correctly handle any case where a device terminates execution of a write command at any point during data transfer. The device may not indicate which portion (if any) of the failed write command was successful, and which portion was not. Thus, none of the data for the command may be assumed written to the media when an error has occurred in a WRITE FPDMA QUEUED command.