

Document: T10/02-319r1
To: T10 Committee Membership
From: Edward A. Gardner, Basil Networks
Subject: SAS Relative Offset

Date: September 8, 2002

SAS does not presently allow random access to data buffers. Targets must transfer data sequentially for each command. While this is a reasonable restriction for disk and tape drives, other environments have other needs.

Basil Networks is designing a RAID controller for use with SAS disk drives. RAID controllers commonly implement caching. Many applications require redundant RAID controllers with critical data duplicated in both controllers' caches.

SAS is an attractive way to communicate critical data between redundant controllers. It appears to have the requisite characteristics, and it reuses hardware that we will already have implemented. However, using SAS for this purpose requires random access to data. While we could define a vendor unique way to implement random access, we would prefer not to do so. For this purpose we only require random access for data returned by READ commands.

Other environments that may require random access to data include copy managers that implement the EXTENDED COPY command and similar commands that use the data buffer to hold a large, complex data structure.

This proposal defines random access completely, including both read and write data and a way to enable it for individual commands. I have made this a complete proposal because I assumed it was necessary to get this approved. The only issue we actually care about is the changes to DATA information units sent by the target (read data). That is the changes to clause 9.2.2.4 and the new fields in table 59. These are the changes that would affect our hardware. If it were plausible to approve those while deferring other changes to SAS-2, we would be happy to do so.

Revision 1 of this proposal incorporates changes suggested at the August 21-23 SAS working group meeting. The significant changes from revision 0 are:

1. Rename the bit that enables data offsets and move it to the position that Fibre Channel used for relative offset valid.
2. Move the data offset field in XFER_RDY frames into the header.

The working group also suggested defining an ASC/ASCQ code for reporting invalid data offsets. The value 0Eh/03h would seem appropriate, since 0Eh reports various kinds of invalid information units. However, upon review I question whether an ASC/ASCQ should be defined. The DATA OFFSET field only appears in information units sent from a target to an initiator. It is reserved in all information units sent to targets. The DATA OFFSET field is analogous to the WRITE DATA LENGTH field in XFER_RDY information units. It is impossible for a target to detect or report an error in either. We don't define an ASC/ASCQ for invalid WRITE DATA LENGTH fields, therefore we shouldn't define one for invalid DATA OFFSET fields either.

References are to SAS-r01b.

Clause 9.2.1, SSP frame format, and table 59. Define bit 3 of byte 11 in table 59 to be DOE. Define bytes 20-23 in table 59 to be DATA OFFSET. Add the following definitions to the text following table 59:

The data offset enable (DOE) bit may be set to one in COMMAND information units to indicate that the target may specify data offsets when accessing the command's data buffer(s).

The data offset enable (DOE) bit is set to one in DATA information units sent by a target to indicate that the DATA OFFSET field is valid. If the data offset enable (DOE) bit was set to zero in a COMMAND information unit, it shall be set to zero in all DATA information units sent by the target for that command.

The data offset enable (DOE) bit is set to one in XFER_RDY information units to indicate that the DATA OFFSET field is valid. If the data offset enable (DOE) bit was set to zero in a COMMAND information unit, it shall be set to zero in all XFER_RDY information units sent for that command.

The data offset enable (DOE) bit shall be set to zero in TASK information units, RESPONSE information units and DATA information units sent by an initiator.

The DATA OFFSET field may be used in XFER_RDY and DATA information units sent by a target to specify a data offset into the command's data buffer(s). The DATA OFFSET field contains a data offset when the data offset enable (DOE) bit is set to one in a XFER_RDY or DATA information unit sent by a target. The DATA OFFSET field is reserved when the data offset enable (DOE) bit is set to zero and in other information unit types.

Clause 9.2.2.3, XFER_RDY information unit. Add the following:

If the data offset enable (DOE) bit (see table 59) is set to zero in the first XFER_RDY information unit for a command, the initiator shall transfer data from the beginning of the data-out buffer. If the data offset enable (DOE) bit is set to zero in a subsequent XFER_RDY information unit for a command, the initiator shall transfer data from the data-out buffer immediately following the data transferred by the previous XFER_RDY information unit for that command.

If the data offset enable (DOE) bit (see table 59) is set to one in a XFER_RDY information unit, the initiator shall transfer data from the data-out buffer starting at the byte offset specified by the DATA OFFSET field (see table 59). The value in the DATA OFFSET field shall be a multiple of 4.

If a target sends a XFER_RDY information unit containing a WRITE DATA LENGTH field that is not a multiple of 4, the next XFER_RDY information unit (if any) sent by the target for that command shall contain a data offset enable (DOE) bit set to one. If the data offset enable (DOE) bit was set to zero in the COMMAND information unit for that command, the target shall not send any subsequent XFER_RDY information units for that command.

Clause 9.2.2.4, DATA information unit. Add the following:

If the data offset enable (DOE) bit (see table 59) is set to zero in the first DATA information unit sent by a target for a command, the DATA field shall be transferred to the beginning of the data-in buffer. If the data offset enable (DOE) bit is set to zero in a subsequent DATA information unit sent by a target for a command, the DATA field shall be transferred to the data-in buffer immediately following the data transferred by the previous DATA information unit sent by the target for that command.

If the data offset enable (DOE) bit (see table 59) is set to one in a DATA information unit sent by a target, the DATA field shall be transferred to the data-in buffer at the byte offset specified by the DATA OFFSET field (see table 59). The value in the DATA OFFSET field shall be a multiple of 4.

If a target sends a DATA information unit containing a non-zero value in the NUMBER OF FILL BYTES field, the next DATA information unit (if any) sent by the target for that command shall contain a data offset enable (DOE) bit set to one. If the data offset enable (DOE) bit was set to zero in the COMMAND information unit for that command, the target shall not send any subsequent DATA information units for that command.

An initiator may set the NUMBER OF FILL BYTES field to a non-zero value in the last DATA information unit that it sends in response to a XFER_RDY. An initiator shall set the NUMBER OF FILL BYTES field to zero in all other DATA information units that it sends.