To: T10 Technical Committee
From: Rob Elliott, Compaq Computer Corporation (Robert.Elliott@compaq.com)
Carl Zeitler, Compaq Computer Corporation (Carl.Zeitler@compaq.com)
Date: 5 November 2000
Subject: FCP-2 Bidirectional command support annex

Related documents

00-377r1: Bidirectional data transfers in FCP-2 – integrated text in FCP-2 to support bidirectional transfers. The 30 Oct 2000 Joint T10/T11.3 meeting recommended including this material as an informative annex to avoid late changes that might hold up FCP-2 revision 5 letter ballot. The material may be incorporated into FCP-3.

00-309r1: Bidirectional data transfers in SAM-2.

Revision history

Revision 0: Initial revision.

Proposed annex

The proposed annex follows. It is labeled Annex X here.

Annex X Bidirectional command support (informative)

X.1 Introduction

This annex describes how bidirectional commands may be supported. Bidirectional commands transfer data in both read (data-in) and write (data-out) directions. Changes to the FCP_CMND IU and FCP_RSP IU are included. Additional information on FCP_DATA IU handling, error recovery, and some examples are also included.

X.2 FCP_CMND IU changes

X.2.1 FCP_CMND IU payload

For a bidirectional command, the FCP_CMND IU is extended by four bytes to include an FCP_BIDIRECTIONAL_READ_DL field after the FCP_DL field.

[Editor’s note: copy the existing table 24 and add four rows at the bottom. The new fields are in italics – it seems useful to highlight them in some manner.]

Table X.1 – FCP_CMND payload for a bidirectional command

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
	FCP_LUN							
7								(LSB)
8	COMMAND REFERENCE NUMBER							
9	RESERVED					TASK ATTRIBUTE		
10	TASK MANAGEMENT FLAGS							
11	ADDITIONAL FCP CDB LENGTH						RDDATA	WRDATA
12	(MSB)							
	FCP CDB							
27								(LSB)
28	(MSB)							
	ADDITIONAL FCP CDB							
n								(LSB)
n+1	(MSB)							
	FCP DL							
								(LSB)

n+4	
n+5	*(MSB)* *FCP BIDIRECTIONAL READ DL* *(LSB)*
n+8	

X.2.2 Task management flags

If any task management flag is set to 1, the FCP_BIDIRECTIONAL_READ_DL field is not valid and is ignored.

X.2.3 RDDATA and WRDATA

For a bidirectional command, both RDDATA and WRDATA are set to 1. This indicates that an FCP_BIDIRECTIONAL_READ_DL field is included in the FCP_CMND payload.

X.2.4 FCP_DL

For a bidirectional command, the FCP_DL field contains a count of the greatest number of data bytes expected to be transferred from the application client data buffer by the SCSI command. The parameter is the data-out buffer size defined by SAM-2.

X.2.5 FCP_BIDIRECTIONAL_READ_DL

If both RDDATA and WRDATA are set to 1, a FCP_BIDIRECTIONAL_READ_DL field follows the FCP DL field. The FCP_BIDIRECTIONAL_READ_DL field contains a count of the greatest number of data bytes expected to be transferred to the application client data buffer by the SCSI command. The parameter is the data-in buffer size defined by SAM-2. An FCP_BIDIRECTIONAL_READ_DL value of 0 indicates that no read data transfer is expected regardless of the state of the READ DATA bit and that no FCP_DATA IUs are transferred for read data.

If either RDDATA or WRDATA is set to 0, the FCP_BIDIRECTIONAL_READ_DL field is not included in the FCP_CMND_IU data payload.

X.3 FCP_DATA IU changes

During any write data transfer for a bidirectional command (an operation that uses Data Out actions, IUs T6 or T7), the initiator always has available a buffer of length FCP_DL containing data to be transferred to the target.

During any read data transfer for a bidirectional command (an operation that uses the Data In action, IU I3), the initiator always has available a buffer of length FCP_BIDIRECTIONAL_READ_DL that receives the data.

If a command requests that data beyond FCP_DL be transferred, the FCP_RESID_OVER bit is set to 1 in the FC_RSP IU. The command is completed normally except that data beyond the FCP_DL count is not transferred and the appropriate overrun condition is presented. See 9.4.4.

If a command requests that data beyond FCP_BIDIRECTIONAL_READ_DL be transferred, the FCP_BIDIRECTIONAL_READ_RESID_OVER bit is set to 1 in the FCP_RSP IU. The command is completed normally except that data beyond the FCP_BIDIRECTIONAL_READ_DL count is not transferred and the appropriate overrun condition is presented. See X.4.3.
[points to FCP_BIDIRECTIONAL_READ_RESID_OVER]

If the amount of data transferred does not match FCP_DL for the write data transfer of a bidirectional command or FCP_BIDIRECTIONAL_READ_DL for the read data transfer of a bidirectional command, the error detection and recovery procedure described in clause 12 may be invoked or the FCP I/O operation may be terminated with a recovery abort or other failure indication.

X.4 FCP_RSP IU changes

X.4.1 FCP_RSP IU payload

For a bidirectional command, the FCP_RSP IU contains two new bits and one new field.

[Editor’s note: just copy the existing table 26 and add byte 10 bits 6 and 5, and bytes 24+m+n to 27+m+n. 8 point small caps seems to fit all the names in byte 10.]

Table X.2 – FCP_RSP payload for bidirectional commands

Bit **Byte**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0-7	Reserved							
8	Reserved							
9	Reserved							
10	RSVD	*FCP BIDIRECTIONAL READ RESID UNDER*	*FCP BIDIRECTIONAL READ RESID OVER*	FCP CONF REQ	FCP RESID UNDER	FCP RESID OVER	FCP SNS_LEN VALID	FCP RSP_LEN VALID
11	SCSI status code							
12-15	FCP_RESID							
16-19	FCP_SNS_LEN (= n)							
20-23	FCP_RSP_LEN (= m)							
24	FCP_RSP_INFO (m bytes long)							
23+m								
24+m	FCP_SNS_INFO (n bytes long)							
23+m+n								
24+m+n	*FCP_BIDIRECTIONAL_READ_RESID*							
27+m+n								

X.4.2 FCP_BIDIRECTIONAL_READ_RESID_UNDER

An FCP_BIDIRECTIONAL_READ_RESID_UNDER bit of 1 indicates that the FCP_BIDIRECTIONAL_READ_RESID field is valid and contains the count of bytes that were expected to be transferred, but were not transferred. The application client should examine the FCP_BIDIRECTIONAL_READ_RESID field in the context of the command to determine whether or not an error condition occurred.

X.4.3 FCP_BIDIRECTIONAL_READ_RESID_OVER

An FCP_BIDIRECTIONAL_READ_RESID_OVER bit of 1 indicates that the FCP_BIDIRECTIONAL_READ_RESID field is valid and contains the count of bytes that could not be transferred because the FCP_BIDIRECTIONAL_READ_DL was not sufficient. The application client should examine the FCP_BIDIRECTONAL_READ_RESID field in the context of the command to determine whether or not an error condition occurred.

X.4.4 FCP_BIDIRECTIONAL_READ_RESID

The FCP_BIDIRECTIONAL_READ_RESID field is included in the FCP_RSP IU for all bidirectional commands. If either the FCP_BIDIRECTIONAL_READ_RESID_UNDER bit or the FCP_BIDIRECTIONAL_READ_RESID_OVER bit is 1, the FCP_BIDIRECTIONAL_READ_RESID field contains a count of the number of residual data bytes that were not transferred in the FCP_DATA IUs for this bidirectional SCSI command. Upon successful completion of a FCP I/O operation, the residual value is normally 0 and the FCP_BIDIRECTIONAL_READ_RESID value is not valid. FCP devices having indeterminate data lengths may have a nonzero residual byte count after completing valid operations. Targets are not required to verify that the data length implied by the contents of the CDB cause an overrun or underrun before beginning execution of a SCSI command.

If the FCP_BIDIRECTIONAL_READ_RESID_UNDER bit is set to 1, a transfer that did not fill the buffer to the expected displacement FCP_BIDIRECTIONAL_READ_DL was performed and the value of FCP_BIDIRECTIONAL_READ_RESID is defined as follows:
FCP_BIDIRECTIONAL_READ_RESID = FCP_BIDIRECTIONAL_READ_DL - highest offset of any byte written - 1

A condition of FCP_BIDIRECTIONAL_READ_RESID_UNDER may not be an error for some FCP devices and some commands.

If the FCP_BIDIRECTIONAL_READ_RESID_OVER bit is set to 1, the transfer was truncated because the data transfer required by the SCSI command extended beyond the displacement value of FCP_BIDIRECTIONAL_READ_DL. Those bytes that could be transferred without violating the FCP_DL value may be transferred. The FCP_BIDIRECTIONAL_READ_RESID is defined as follows:
FCP_BIDIRECTIONAL_RESID = (Read transfer length required by command) - FCP_BIDIRECTIONAL_READ_DL

If a condition of FCP_BIDIRECTIONAL_READ_RESID_OVER is detected, the termination state of the FCP I/O operation is not certain. Data may or may not have been transferred and the SCSI status byte may or may not provide correct command completion information.

If both the FCP_BIDIRECTIONAL_READ_RESID_UNDER and the FCP_BIDIRECTIONAL_READ_RESID_OVER bits are 0, the FCP_BIDIRECTIONAL_READ_RESID field is not meaningful and may have any value.

X.5 Error recovery changes

X.5.1 Sequence level error recover

Sequence level recovery is not used for bidirectional commands.

X.5.2 FCP-2 Error Detection using protocol errors for all classes of service

The Exchange originator (initiator) detects the following errors for bidirectional commands.

a) a bidirectional command completed with the write data count smaller than FCP_DL and FCP_RESID_UNDER is set to 0;
b) a bidirectional command completed with the read data count smaller than FCP_BIDIRECTIONAL_READ_DL and FCP_BIDIRECTIONAL_READ_RESID_UNDER is set to 0;
c) a bidirectional command completed with the write data count smaller than FCP_DL, FCP_RESID_UNDER is set to 1, and the write data count plus FCP_RESID is not equal to FCP_DL; and,
d) a bidirectional command completed with the read data count smaller than FCP_BIDIRECTIONAL_READ_DL, FCP_BIDIRECTIONAL_READ_RESID_UNDER is set to 1, and the read data count plus FCP_BIDIRECTIONAL_READ_RESID is not equal to FCP_BIDIRECTIONAL_READ_DL.

X.6 FCP Examples

X.6.1 SCSI FCP bidirectional command with write before read

A typical SCSI FCP bidirectional command with a single data IU transferred in each direction is shown in table X.3. The example command accepts write data before returning read data.

Table X.3. FCP bidirectional command with write before read, example

Initiator function	IU	Target function

Command request	T1, FCP_CMND ->	
		[Prepare data out transfer buffer]
	<- I1, FCP_XFER_RDY	Data out delivery request
Data out action	T6, FCP_DATA ->	
		[Prepare data in transfer]
	<- I3, FCP_DATA	Data in action
		[Prepare response message]
	<- I4, FCP_RSP	Response
[Indicate command completion]		

X.6.2 SCSI FCP bidirectional command with read before write

A typical SCSI FCP bidirectional command with a single data IU transferred in each direction is shown in table X.4. The example command returns read data before accepting write data.

Table X.4. FCP bidirectional command with read before write, example

Initiator function	IU	Target function
Command request	T1, FCP_CMND ->	
		[Prepare data in transfer]
	<- I3, FCP_DATA	Data in action
		[Prepare data out transfer buffer]
	<- I1, FCP_XFER_RDY	Data out delivery request
Data out action	T6, FCP_DATA ->	
		[Prepare response message]
	<- I4, FCP_RSP	Response
[Indicate command completion]		

X.6.3 SCSI FCP bidirectional command with write before read and write FCP_XFER_RDY disabled

A SCSI FCP bidirectional command with two write data IUs and one read data IU is shown in table X.5. The example command accepts write data before returning read data. The initial write FCP_XFER_RDY IU has been disabled during process login.

Table X.5. FCP bidirectional command with write before read and FCP_XFER_RDY disabled, example

Initiator function	IU	Target function
Command request	T1, FCP_CMND ->	
Data out action	FCP_DATA ->	First Data out
	<- I1, FCP_XFER_RDY	Second Data out delivery request
Data out action	T6, FCP_DATA ->	
	<- I1, FCP_XFER_RDY	Last Data out delivery request
Data out action	T6, FCP_DATA ->	
		[Prepare data in transfer]
	<- I3, FCP_DATA	Data in action
		[Prepare response message]
	<- I4, FCP_RSP	Response
[Indicate command completion]		

X.6.4 SCSI FCP bidirectional command with intermixed writes and reads

A SCSI FCP bidirectional command with three data IUs transferred in each direction is shown in table X.6. The example command accepts some write data before returning read data, but intermixes writes and reads thereafter.

Table X.6. FCP bidirectional command with intermixed writes and reads, example

Initiator function	IU	Target function
Command request	T1, FCP_CMND ->	
		[Prepare data out buffer]
	<- I1, FCP_XFER_RDY	First Data out delivery request
First Data out action	T6, FCP_DATA ->	
		[Prepare data in transfer]
	<- I3, FCP_DATA	First Data in action
	<- I1, FCP_XFER_RDY	Second Data out delivery request
Second Data out action	T6, FCP_DATA ->	
	<- I1, FCP_XFER_RDY	Last Data out delivery request
Third Data out action	T6, FCP_DATA ->	
	<- I3, FCP_DATA	Second Data in action
	<- I3, FCP_DATA	Last Data in action
		[Prepare response message]
	<- I4, FCP_RSP	Response
[Indicate command completion]		