Document:	X3T10/95-208r0	Date:	April 18, 1998
To:	X3T10 Committee Membership, FC-AL Direct Disk Attach participants		
From:	Edward A. Gardner		
Subject:	FCP or DDA Profile use of Disconnect-Reconnect Mode Page		

At the April 7 Direct Disk Attach ad hoc meeting in Monterey I presented several questions concerning FCP's use of the Disconnect-reconnect page as specified by the DDA Profile. From the ensuing discussion it was clear that different people were interpreting this mode page's parameters in different ways, and several useful parameters were lacking. The following is a first cut at addressing these issues.

One important question is which standards document should contain the detailed definition of the Disconnect-reconnect page parameters. Definition of these parameters is inherently specific to particular protocols. The current wording is "ideally matched to the mechanisms of the SCSI-3 Interlocked Protocol" (to borrow Ralph Weber's wonderful euphemism in SPC), with little attention given to other protocols. In my opinion SPC should describe these parameters only very loosely, perhaps not at all, and the detailed definition included in the various protocol documents. The alternative is to include considerable protocol specific information in SPC, requiring that SPC be amended whenever a new protocol is defined or an existing one changed. I do not recall the committee having addressed this previously.

> **Note: I strongly oppose delaying FCP or any other protocol standard to incorporate this material. This seems a natural item for inclusion FCP-2. I feel strongly that we should first reach agreement on the desired technical content, and only then choose a particular revision of FCP and/or SPC to incorporate this material.**

The current format of the Disconnect-reconnect page from SPC section 8.3.2 section is shown below. Current wording from that section is shown in italics.

1. Protocol Identification

The present page definition does not contain any parameter indicating the protocol which the other parameters are intended for. Curious results might occur if, for example, parameter settings intended for SIP were passed to an FCP device. Current CAM-like driver interfaces (CAM, ASPI, mini-ports, etc.) do not inform command level drivers of the protocol being used, simply because there is only one protocol in SCSI-2. Thus it is quite plausible that parameter settings intended for SIP will be passed to non-SIP devices. To the best of my knowledge there is no mechanism at the CDB level for determining what protocol a device uses. But even if one were defined, no current software will check that before issuing a MODE SELECT command.

I believe the best solution is to add a protocol identifier parameter (non-changeable) to the Disconnect-reconnect page. For MODE SENSE targets would return the protocol they use. For MODE SELECT targets would confirm that the proper protocol is specified before accepting any parameter changes.

The obvious place to put this is byte 13. The three bit reserved field in byte 12 is just a bit too small for comfort; and is arguably better left for bit flags. Note that this is the only parameter that must be defined in SPC. The format and definition of the rest of the page could be delegated to the protocol specifications.

2. Buffer Full/Empty Ratios

According to Dal Allan, these parameters were originally defined for tape drives, and were never intended for use with disks. I can say that, as a disk implementor, their current definition seems

Bit Byte	7	6	5	4	3	2	1	0
0	PS	Reserved		Page code (02h)				
1				Page length (0Eh)				
2				Buffer full ratio				
3				Buffer empty ratio				
4	(MSB)			Bus inactivity limit				
5								(LSB)
6	(MSB)			Disconnect time limit				
7								(LSB)
8	(MSB)			Connect time limit				
9								(LSB)
10	(MSB)			Maximum burst size				
11								(LSB)
12	EMDP		Reserved		DImm		DTDC	
13				Reserved				
14				Reserved				
15				Reserved				

to have no relationship to how our current disks manage buffer memory. If anyone does use these with disks, I request that they provide an example to clarify the current wording. Otherwise I suggest that the current note on SPC page 96 be amended to read:

> *NOTE 55 As an example, consider a device server with ten 512-byte buffers and a specified buffer full ratio of 3Fh. The formula is: INTEGER((ratio/256)*number of buffers). Thus INTEGER((3Fh/256)*10) = 2. On read operations, the device server should attempt to move data on the interconnect whenever two or more buffers are full.* These parameters are primarily intended for device servers that devote themselves to a single continuous transfer stream, such as is typical for stream devices.

3. Time Values

Three parameters specify time limits, all presently expressed in 100 µs units. This granularity is much too coarse for FCP, and arguably too coarse for modern SIP implementations as well. Three alternatives come to mind:

1. Tie the time limit granularity to the protocol identifier in byte 13. Protocol 0 would be archaic SIP, with 100 µs units. Protocol 1 would be modern SIP, with 1 µs units. Protocol 2 would be FCP, also with 1 µs units. Other protocol identifiers and time units would be defined.
2. Define a separate field indicating the time limit units. Perhaps a two bit field, bits 4 and 5 of byte 12, indicating 100 µs units, 1 µs units, and two reserved values.
3. Switch to what amounts to a floating point format for the time limits. For example, the high two bits of each value would encode the units for that time limit, the low 14 bits would encode the actual time limit value.

I have no strong feelings on the choice of alternative. Left to my decision, I would probably choose alternative 1, simply because I think it requires the least time for me to write the proposed wording. I expect the committee to direct me on the appropriate choice.

4. Bus inactivity limit

SPC currently reads:

> *The bus inactivity limit field indicates the maximum time in 100 ms increments that the target is permitted lockout* [sic] *other uses of the interconnect without actually moving data. If the bus inactivity limit is exceeded the device server shall attempt to interrupt the data moving operation, within the restrictions placed on it by the application client. The contents of the DTDC parameter in this mode page shall be one such restriction. This value may be rounded as defined in clause 5.2. A value of zero indicates that there is no bus inactivity limit.*

The concept of bus activity or inactivity does not apply to fabric attached FCP devices, since there are no shared resources controlled by the target. However this does apply to FC-AL devices, where it would limit how long a target may continue a loop tenancy without sending data. I suggest wording such as the following for FCP:

> For FC-AL attached targets, the bus inactivity limit field indicates the maximum time in /tbd/ increments that the target's L_Port may defer closing the loop without sending frames. This is the maximum time from the end of one frame sent by the target to the beginning of the next frame sent by the same L_Port within the same loop tenancy. The target shall close the loop immediately if it has consumed the initiator's available BB_credit or it cannot send the next frame within this time limit. This parameter does not affect frame reception or non-FC-AL attached targets.

5. Disconnect time limit

SPC currently reads:

> *The disconnect time limit field indicates the minimum time in 100 ms increments that the target shall wait between attempts to move data on the interconnect. This value may be rounded as defined in clause 5.2. A value of zero indicates that there is no disconnect time limit.*

I believe this parameter has been used to prevent a fast device from monopolizing the SCSI bus and locking out slower devices. While the literal implementation described could be applied to FCP, I don't believe that provides a useful feature for busses such as FC-AL that ensure non-lockout through fair arbitration. I suggest that this parameter be reserved for FCP.

6. Connect time limit

SPC currently reads:

> *The connect time limit field indicates the maximum time in 100 ms increments that the target is allowed to use the interconnect during a single data moving operation. If the connect time limit is exceeded the device server shall attempt to interrupt the data moving operation, within the restrictions placed on it by the application client. The contents of the DTDC parameter in this mode page shall be one such restriction. This value may be rounded as defined in clause 5.2. A value of zero indicates that there is no connect time limit.*

Again, this has no meaning for fabric attached FCP devices, but is suitable for limiting an FC-AL target's loop tenancy. I suggest wording such as the following for FCP:

> For FC-AL attached targets, the connect time limit field indicates the maximum time in /tbd/ increments that the target's L_Port may keep the loop open after having openned it. The target's L_Port shall send CLS no later than this time limit after sending each OPNy. This parameter does not affect non-FC-AL attached targets or loop tenancies opened by an initiator.

7. Maximum burst size

This parameter has already been defined by FCP to be the maximum sequence length. I suggest retaining the current definition. Note that this applies to both data sent and requested by the target.

8. Enable modify data pointers

SPC currently reads:

> *The enable modify data pointers (EMDP) bit indicates whether or not the initiator allows the Modify Data Pointers message to be sent by the target. (The Modify Data Pointers message is defined by in the SIP.) If the EMDP bit is a zero, the target shall not issue the Modify Data Pointer* [sic] *message. If the EMDP bit is a one, the target is allowed to issue Modify Data Pointer messages.*
>
> *If the EMDP bit is a one...*
>
> *The EMDP bit shall be zero for all devices except those using the SCSI-3 Interlocked Protocol (SIP).*

While the Modify Data Pointers message is unique to SIP, the concept of in order vs. out of order transfers is common to all interfaces. Furthermore the ability to control this is a recurring customer request. I suggest deleting the last sentence from SPC, renaming this bit to something less SIP specific (perhaps EOOT, enable out of order transfers), and wording such as the following for FCP:

> Indicates whether the sequences comprising a data transfer must be performed in ascending offset order. If set, the sequences may be sent in any order. If clear, the sequences shall be sent in ascending offset order. Does not affect the order of frames within a sequence.