Changes to FCP, Revision 4

- Reformatted to meet editorial requirements.
- Added Foreword and Introduction
- Removed reference to P1275 Open Boot Standard
- Examples moved to Annex A
- Technical Changes:
  - Link Management improved
  - Optional Data Transfer Ready provided
  - Combined Command/Data Sequence provided
  - FCP Entity Address, Data Descriptor, Status matched to FC-PH
  - Informative Annex for Entity Address included
  - Default Entity Address Defined
Link Management

- Link Application Frames for FCP

- FCP login
  Command + Data Allowed bit
  Transfer Ready Disabled bit

- FCP logout
  Terminates FCP login

- FCP reject
  Rejects Link Application Frame

- FCP accept
  Accepts Link Application Frame
Sequence Formats

- **Optionally disable Data Transfer Ready**
  
  Controlled by FCP Login
  
  Default value is Data Transfer Ready required

- **Combined Command/Data Sequence**
  
  Controlled by FCP Login
  
  Default value is separated Command/Data Sequence
  
  Combined Command/Data Sequence allowed only if Data Transfer Ready is disabled.
  
  Data/Status Sequence cannot be combined.
## Sequence Formats

- FCP Sequences sent to Targets

<table>
<thead>
<tr>
<th>SN</th>
<th>PHASE</th>
<th>DATA BLOCK</th>
<th>F/M/L</th>
<th>SI</th>
<th>SC</th>
<th>RO</th>
<th>M/O</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CAT CONTENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T1</td>
<td>CMD</td>
<td>6 COMMAND</td>
<td>F</td>
<td>T</td>
<td>0</td>
<td>0</td>
<td>M</td>
</tr>
<tr>
<td>T2</td>
<td>CMD</td>
<td>6 COMMAND</td>
<td>F</td>
<td>H</td>
<td>0</td>
<td>0</td>
<td>O</td>
</tr>
<tr>
<td>T3</td>
<td>DATA</td>
<td>1 DATA</td>
<td>M</td>
<td>T</td>
<td>0</td>
<td>disp</td>
<td>M</td>
</tr>
<tr>
<td>T4</td>
<td>DATA</td>
<td>1 DATA</td>
<td>M</td>
<td>H</td>
<td>0</td>
<td>disp</td>
<td>O</td>
</tr>
<tr>
<td>T5</td>
<td>CMD/DATA</td>
<td>6/1 COMMAND + DATA</td>
<td>F</td>
<td>T</td>
<td>0</td>
<td>0</td>
<td>O</td>
</tr>
<tr>
<td>T6</td>
<td>CMD/DATA</td>
<td>6/1 COMMAND + DATA</td>
<td>F</td>
<td>H</td>
<td>0</td>
<td>0</td>
<td>O</td>
</tr>
</tbody>
</table>
Sequence Formats

- FCP Sequences sent to Initiators

<table>
<thead>
<tr>
<th>SN</th>
<th>PHASE</th>
<th>DATA BLOCK</th>
<th>FML</th>
<th>SI</th>
<th>SC</th>
<th>RO</th>
<th>M/O</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CONTENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>DATA</td>
<td>5 WRITE XFER RDY</td>
<td>M</td>
<td>T</td>
<td>0</td>
<td>0</td>
<td>M</td>
</tr>
<tr>
<td>12</td>
<td>DATA</td>
<td>5 READ XFER RDY</td>
<td>M</td>
<td>H</td>
<td>0</td>
<td>0</td>
<td>M</td>
</tr>
<tr>
<td>13</td>
<td>DATA</td>
<td>1 DATA</td>
<td>M</td>
<td>H</td>
<td>0</td>
<td>disp</td>
<td>M</td>
</tr>
<tr>
<td>14</td>
<td>STATUS</td>
<td>7 STATUS</td>
<td>L</td>
<td>T</td>
<td>0</td>
<td>0</td>
<td>M</td>
</tr>
</tbody>
</table>
### FCP Entity Address, Data Descriptor

#### Entity Address

<table>
<thead>
<tr>
<th>Field</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>BNT_ADDR_0</td>
<td>2 bytes</td>
</tr>
<tr>
<td>BNT_ADDR_1</td>
<td>2 bytes</td>
</tr>
<tr>
<td>BNT_ADDR_2</td>
<td>2 bytes</td>
</tr>
<tr>
<td>BNT_ADDR_3</td>
<td>2 bytes</td>
</tr>
</tbody>
</table>

#### Data Descriptor

At beginning of Read/Write Xfer Ready Sequences (Category 5)

<table>
<thead>
<tr>
<th>Field</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEQ_RO</td>
<td>4 bytes</td>
</tr>
<tr>
<td>BURST_LEN</td>
<td>4 bytes</td>
</tr>
<tr>
<td>Reserved</td>
<td>4 bytes</td>
</tr>
</tbody>
</table>
FCP Generic Status

- Generic Status

Included in first four bytes of FCP Status Sequence (Category 7)

<table>
<thead>
<tr>
<th>Field</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>reserved</td>
<td>4 bytes</td>
</tr>
<tr>
<td>reserved</td>
<td>4 bytes</td>
</tr>
<tr>
<td>FCP_STATUS</td>
<td>4 bytes</td>
</tr>
<tr>
<td>FCP_RESID</td>
<td>4 bytes</td>
</tr>
<tr>
<td>FCP_SNS_LEN</td>
<td>4 bytes</td>
</tr>
<tr>
<td>FCP_RSP_LEN</td>
<td>4 bytes</td>
</tr>
<tr>
<td>FCP_SNS_INFO</td>
<td>n bytes</td>
</tr>
<tr>
<td>FCP_RSP_INFO</td>
<td>m bytes</td>
</tr>
</tbody>
</table>

Bytes are presently reserved = 0, pending future definition
Entity Address

- Annex B defined to describe Entity Address model

- Default Entity Address defined (7.1.1)

  Address of '0000 0000 0000 0000' hexadecimal shall exist and accept INQUIRY command.

  Type, Manufacturer, Model, Driver Requirements, and configuration mechanism can be determined from INQUIRY data.
Discussion Items for FCP

- Relative Offset not available
- Data Descriptor boundary restrictions
- Class 1 Behavior
- ACK_N behavior
- Class 3 Behavior
- Reset and related behavior
- ACA behavior
Discussion Item
Relative Offset not available

- Limitations if RO not supported

  Propose: In order delivery required within a sequence.

  Data Transfer Ready establishes order among sequences, if enabled.

  Data transferred in a single sequence if Transfer Ready not enabled either.

- Data Base operations may require "in order" delivery

  Propose: Use of FCP_CNTL bit = 1 to require "in order" delivery over entire data transfer, even if RO and Data Transfer Ready enabled.

  Use bit 0 in Byte 2.
Discussion Item
Data Descriptor boundary restrictions

- Cache line boundaries easier to handle on Data Transfer

  Present: No boundaries enforced between Data sequences. A Data sequence may be any length and end on any boundary.

  Suggestion: Some have requested that Data sequence length be restricted to a multiple of 4, 16, or 64 bytes.

  Propose: No change from present architecture.

  Reason: Since there are no inherent limitations on where a sequence may begin putting data in memory, hardware should be capable of managing any boundary. If so, then it can manage any boundary for every sequence. Then sequence length does not matter.
Discussion Item
Class 1 Behavior

- **Class 2 Behavior**
  Defined by normal FCP transactions

- **Class 1 Behavior**
  Propose: To maintain parallel operation among many I/O operations, connections will be made for the duration of one sequence. If multiple commands or status are available for the same destination, the connection may carry multiple sequences. Reverse sequences are optional.

- **Hybrid Behavior**
  Propose: For fabrics supporting mixed Class 1 and 2 operation, performance optimization may put Data sequences in Class 1 and the short sequences in Class 2. All have same exchange id.
Discussion Item
ACK N behavior

- No problem for FCP

Present: No problem for FCP because sequences within an exchange are not streamed. If a sequence is not complete, the exchange stops and times out. Parallel exchanges continue normally and with no possibility of ambiguity.
Discussion Item
Class 3 Behavior

- Class 3 Behavior

Propose: Class 3 should not be allowed.

FCP depends on verified receipt of sequences within an exchange to continue the processing through an exchange. Class 3 would require additional sequences to perform the same function, putting the ACK_N function into the ULP behavior and removing it from the hardware assisted behavior typical of Class 1 and Class 2. FCP would then have two independent protocols, one using a ULP ACK_N, the other using an FC-PH ACK_N.
Discussion Item
Reset and related behavior

- **Under Study**

- **Straw Man**
  - SCSI
  - RST
  - Bus Device Reset
  - Abort, Abort Tag
  - Clear Queue
  - Terminate I/O
  - FCP
  - Logout/Login
  - FCP_CNTL Reset (byte 2, bit 5)
  - Abort Exchange
  - n x Abort Exchange
  - NA
ACA Behavior

- Under Study

- Straw Man

Propose:

Request Sense Data returned as part of Response sequence

Normal/default operation ends ACA automatically at end of command.

Extended Contingent Allegiance mode of ACA may be invoked by FCP_LOGI option. If invoked, extended ACA begins automatically at end of command presenting Check Condition and Request Sense Data. All other commands to the Entity remain suspended. All commands with Queue Type ACA_Q from the same source and image are accepted by the Entity. The extended ACA is cleared at the end of the first Queue Type ACA_Q command with the Clear ACA bit set in byte 2 of FCP_CNTL.