

# USB 海量存储类

单批量传输

## 1 规范概述

### 1.1 概述

该规范适合于已经熟悉 USB1.0、USB1.1 规范和大致了解 USB 海量存储类 (Mass Storage Class) 规范的读者。

这个规范描述了单批量 (Bulk-Only) 传输方式，该传输方式即是仅使用批量传输端点 (不使用中断、控制传输端点) 对命令、数据、状态进行传输。缺省管道仅用来请求批量端点上的 STALL 停止的状态和执行类特定请求命令。该规范不需要使用中断端点。

该规范支持逻辑单元共享设备特性。尽管这种特性提供了对类似海量存储设备共享 USB 接口描述符的支持，但它并不是为实现桥接设备的功能而设计的。

## 2 术语和缩写

### 2.1 协定

没有特别强调的数为十进制数.....1, 17, 23

十六进制数后紧跟 “h” ..... 1Fh, FCh, 38h

二进制数后紧跟

“b” ..... .011b, 101b, 01110010b

斜体字是 USB 协议或是该规范定义的术语 ..... *bRequese*, *dCSWTag*

### 2.2 定义

**Command Block Wrapper (CBW)**

一个包含一个命令块和一些附加信息的封包。

**Command Status Wrapper (CSW)**

一个包含一个命令块状态的封包。

**Data-In**

代表一个从设备到主机的输入传输。

Data-Out

代表一个从主机到设备的输出传输。

Device Request

主机通过缺省管道对设备的请求。

Phase Error

设备返回的一个错误信息，该信息指明如果不对设备进行复位操作，那么对后面 CBWs 处理的结果将会不确定。

Processed

被设备接收和内在控制的数据不再需要主机干涉。

Relevant

设备发送到主机的重要数据的数量。

Reset Recovery

主机为设备准备的为以后的 CBWs 进行错误恢复的程序。

Thin Diagonal

主机和设备完全认同在哪里进行数据传输的情况。见第六章 — 《主机/设备数据传输》附加的关于错误情况和 “thin diagonal” 信息。

### 3 功能特性

#### 3.1 Bulk-Only Mass Storage 复位（类特定请求命令）

该请求用于复位 Mass Storage 设备及其接口。

该类特定请求命令使设备处于准备就绪状态，以接收下一次主机发出的 CBW。

主机将通过缺省管道发送请求给设备。设备保存该批传输的 toggle 位和端点 STALL 停止状态除非该 Bulk-Only Mass Storage 复位。

该设备将 NAK 设备请求状态阶段直到 Bulk-Only Mass Storage 复位完成。

为了产生 Bulk-Only Mass Storage 复位，主机将在缺省管道上产生设备请求：

- *bmRequestType*:类、接口、主机到设备
- *bRequest* 字段设置为 255 (FFh)
- *wValue* 字段设置为 0
- *wIndex* 字段设置为接口号
- *wLength* 字段设置为 0

表 3.1Bulk-Only Mass Storage 复位

<i>bmRequestType</i>	<i>bRequest</i>	<i>wValue</i>	<i>wIndex</i>	<i>wLength</i>	Data
00100001b	11111111b	0000h	Interface	0000h	none

3.2 Get Max LUN（类特定请求命令）

该设备可能实现若干个逻辑单元共享设备功能特性。主机使用 *bCBWLUN*来指定该设备的哪个逻辑单元来接收主机发出的命令。“取得最大逻辑单元数（Get Max LUN）”设备请求用来确定设备所拥有的逻辑单元的数量。“设备逻辑单元数（Logical Unit Numbers）”应该按从 LUN0 开始到最大 LUN15 (Fh) 来定序。

为了产生 Get Max LUN 设备请求，主机将在缺省管道上产生设备请求：

- *bmRequestType*:类、接口、设备到主机
- *bRequest* 字段设置为 254 (Feh)
- *wIndex* 字段设置为 0
- *wIndex* 字段设置为接口号
- *wLength* 字段设置为 1

表 3.2Get Max LUN

<i>bmRequestType</i>	<i>bRequest</i>	<i>wValue</i>	<i>wIndex</i>	<i>wLength</i>	Data
10100001b	11111110b	0000h	Interface	0001h	1 byte

设备将返回 1 字节用于描述设备所包含逻辑单元数的数据。例如，如果设备包含 4 个逻辑单元，则 LUNs 应该被连续从 0 到 3 进行编号，返回值为 3。若设备没有 LUN，则返回值为 0。主机不能针对不存在的逻辑单元发送“命令块封包(CBW)”。

不支持多 LUNs 的设备将 STALL 该命令。

### 3.3 主机/设备包传输顺序

主机在输出传输 (Data-Out) 前先发出 CBW, 设备在主机发出 CBW 后进行输入传输 (Data-In), 然后发出 CSW。主机可能在发出 CBW 前请求输入 (Data-In) 或 CSW。

### 3.4 命令顺序

主机不会传输 CBW 给设备直到主机接收到任何一个 CBW 的 CSW。如果主机产生两个连续的 CBWs 而没有一个介于其间的 CSW 或者复位, 则设备对第二个 CBW 的响应是不确定的。

### 3.5 Bi-Directional 命令协议

该规范不提供在单一命令中进行 bi-directional 数据传输。

## 4 标准描述符

设备支持以下的标准 USB 描述符:

- **设备 (Device)**. 每个 USB 设备有一个设备描述符。
- **配置 (Configuration)**. 每个 USB 设备有一个默认的配置描述符, 它至少支持一个接口。
- **接口 (Interface)**. 设备支持至少一个接口, 在这里知道诸如 Bulk-Only 数据接口。有的设备可能支持额外的接口, 以此来提供其它的能力。
- **端点 (Endpoint)**. 除了各种 USB 设备都需要的缺省管道外, 设备还支持以下端点:
  - 批量输入 (Bulk-In) 端点
  - 批量输出 (Bulk-Out) 端点

有些设备可能会支持额外的端点, 以此来提供其它的能力。主机对选中的接口使用先前提到的 Bulk-In 和 Bulk-Out 端点。

- **字符串 (String)**. 设备支持一个唯一的序列号。详见 4.1.1—《序列号》。

该规范定义没有 class-specific 描述符。

这部分余下的章节描述了标准的 USB 设备、配置、接口、端点和字符串描述符。如果了解更多的关于标准描述符的信息，参见 USB 规范的章节 9，《USB 设备架构》。

4.1 设备描述符 (Device Descriptor)

每个 USB 设备只有一个设备描述符。设备类码和设备子类码是在接口描述符 (Interface Descriptor) 而不是在设备描述符中给定的。

表 4.1 设备描述符

Offset	Field	Size	Value	Description
0	<i>bLength</i>	字节	12h	该描述符的字节数
1	<i>bDescriptorType</i>	字节	01h	设备描述符类型
2	<i>bcdUSB</i>	字	xxxxh	USB 版本号
4	<i>bDeviceClass</i>	字节	00h	USB 分配的设备类代码
5	<i>bDeviceSubClass</i>	字节	00h	USB 分配的之类代码
6	<i>bDeviceProtocol</i>	字节	00h	USB 分配的设备协议代码
7	<i>bMaxPacketSize0</i>	字节	xxh	端点 0 的最大包的大小
8	<i>idVendor</i>	字	xxxxh	厂商编号
10	<i>idProduct</i>	字	xxxxh	产品编号
12	<i>bcdDevice</i>	字	xxxxh	设备出厂编号
14	<i>iManufacturer</i>	字节	xxh	描述厂商字符串的索引
15	<i>iProduct</i>	字节	xxh	描述产品字符串的索引
16	<i>iSerialNumber</i>	字节	xxh	描述设备序列号字符串的索引
17	<i>bNumConfigurations</i>	字节	xxh	可能的配置数量

注：该表的信息摘自 USB 规范 1.1 的表 9-7。

4.1.1 序列号 (Serial Number)

*iSerialNumber* 字段被设置为包含序列号的字符串描述符 (string descriptor) 的索引值。序列号包含至少 12 个合法阿拉伯数字，它是以 UNICODE 方式编码的。针对每个 USB 设备的 *idVerdor* 和 *idProduct* 对，序列号中最后 12 个阿拉伯数字是唯一的。

通过连接 16 位的 *idVendor*、16 位的 *idProduct* 和由 *iSerial Number* 字段索引的最后 12 个字符所代表的值，主机可能产生一个通用的唯一的标识。

*iSerialNumber* 是字符串描述符的索引，并不包含字符串本身。下面是一个字符串描述符（String descriptor）的示例。

4.3 接口描述符

设备最少支持一个接口，例如 Bulk-Only 数据接口。Bulk-Only 数据接口使用了 3 个端点。

复合的 mass storage 设备可能支持而外的接口，以此来提供诸如音频、视频等其它的功能特性。该规范没有定义这类接口。

接口可能支持多种可供选择使用的设置。主机检查每一个选择设置来寻找 *bInterfaceProtoocol* 和 *bInterfaceSubClass* 以提供最佳的支持。

表 4.5Bulk-Only 数据接口描述符

Offset	Field	Size	Value	Description
0	<i>bLength</i>	字节	09h	接口描述符的字节数大小
1	<i>bDescriptorType</i>	字节	04h	接口描述符类型编号
2	<i>bInterfaceNumber</i>	字节	0xh	接口的编号
3	<i>bAlternateSetting</i>	字节	xxh	备用的接口描述符编号
4	<i>bNumEndpoints</i>	字节	xxh	该接口使用的端点数，不包括端点 0
5	<i>bInterfaceClass</i>	字节	08h	接口类型
6	<i>bInterfaceSubClass</i>	字节	0xh	接口子类类型码
7	<i>bInterfaceProtocol</i>	字节	50h	接口遵循的协议
8	<i>iInterface</i>	字节	xxh	描述该接口的字符串索引值

注：该表的信息摘自 USB 规范 1.1 的表 9-9。

4.4 端点描述符（Endpoint Descriptors）

设备支持至少 3 个端点：控制、Bulk-In、Bulk-Out.

每一个 USB 设备都定义了控制端点（Endpoint 0）。这就是缺省端点，它不需要描述符。

4. 4. 1 Bulk-In 端点

Bulk-In 端点用于从设备传输数据和状态到主机。

表 4. 6Bulk-In 端点描述符

Offset	Field	Size	Value	Description
0	<i>bLength</i>	字节	07h	端点描述符的字节数大小
1	<i>bDescriptorType</i>	字节	05h	端点描述符类型编号
2	<i>bEndpointAdress</i>	字节	8xh	该端点在 USB 设备上的地址。该地址编码如下：  BitDescription  3..0 端点号  6..4 保留，设置为 0  70=In
3	<i>bmAttributes</i>	字节	02h	这是一个 Bulk 端点
4	<i>wMaxPacketSize</i>	字	00xxh	最大封包大小。应该为 8，16，32 或 64 字节（08h, 10h, 20h, 40h）
6	<i>bInterval</i>	字节	00h	在 Bulk 端点中不使用

4. 4. 2 Bulk-Out 端点

表 4. 7Bulk-Out 端点描述符

Offset	Field	Size	Value	Description
0	<i>bLength</i>	字节	07h	端点描述符的字节数大小
1	<i>bDescriptorType</i>	字节	05h	端点描述符类型编号
2	<i>bEndpointAdress</i>	字节	0?h	该端点在 USB 设备上的地址。该地址编码如下：  BitDescription

				3..0 端点号  6..4 保留，设置为 0  70=Out
3	<i>bmAttributes</i>	字节	02h	这是一个 Bulk 端点
4	<i>wMaxPacketSize</i>	字	00xxh	最大封包大小。应该为 8, 16, 32 或 64 字节 (08h, 10h, 20h, 40h)
6	<i>bInterval</i>	字节	00h	在 Bulk 端点中不使用

## 5 命令/数据/状态协议

图 1 展示了命令、Data-In、Data-Out 传输的流程，接下来的部分定义了命令和状态的传输。图 2 展示状态传输的细节。紧接着的部分概括了主机/从机通信的各种条件、可能的错误、恢复机制。

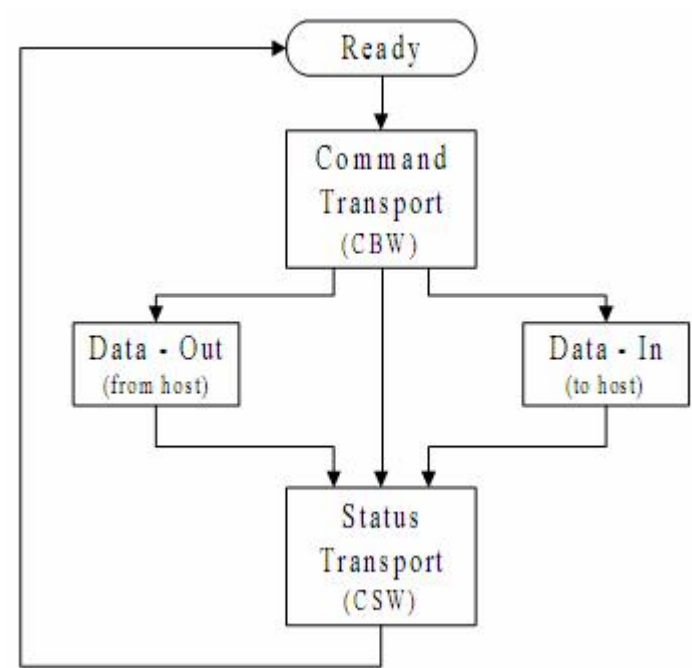


图 1 命令/数据/状态流

### 5.1 命令块封包 (Command Block Wrapper, 简称为 CBW)

表 5.1 命令块封包 (Command Block Wrapper)

bit	7	6	5	4	3	2	1	0
Byte								



0-3	<i>dCBWSignature</i>	
4-7	<i>dCBWTag</i>	
8-11 (08h-0Bh)	<i>dCBWDataTransferLength</i>	
12 (0Ch)	<i>bmCBWFlags</i>	
13 (0Dh)	<i>Reserved(0)</i>	<i>bCBWLUN</i>
14 (0Eh)	<i>Reserved(0)</i>	<i>bCBWCBLength</i>
15-30 (0Fh-1Eh)	<i>CBWCB</i>	

#### ***dCBWSignature:***

该字段是 CBW 的标志。值固定为 43425355h。注意：这个值是按照通常的高位在前、低位在后的二进制数顺序排列的；但是，CBW 在 USB 总线上传输的时候，总是先发送 LSB，再发送 MSB。该字段占用 4 字节。

#### ***dCBWTag:***

该字段是命令块标签（Command Block Tag），由 USB 主机产生并发送给设备。设备会将此值填入 CSW 的 dCSWTag 字段，以此回应主机的命令。该字段占用 4 字节。

#### ***dCBWDataTransferLength:***

该字段在命令执行期间，主机希望在批量输出或批量输入端点上传输的数据大小。该字段占用 4 字节。

#### ***bmCBWFlags:***

该字段的内容为 1 字节位图，位 D7 表示数据的传输方向。当 D7=0 时，表示主机到设备的数据输出；当 D7=1 时，表示设备到主机的数据输入。位 D6 没有用，设置为 0。位 D5~D0 保留为 0。

#### ***bCBWLUN:***

该字段表示接收该命令的设备的逻辑单元号 LUN。如果设备只有 1 个逻辑单元，那么该值设置为 0。该字段占用半个字节。

*bCBWCBLength:*

该字段表示 CBWCB 的字节数，也就是将要发送的特定子类命令的长度。其合法值为 1 到 16（01h~10h）。其它值保留。

*CBWCB:*

该字段中填入的内容就是特定的各种子类命令。详细的该字段值的结构就可以在一个子类的命令协议中有定义。该字段占用 16 字节。如果相应的子类命令长度小于 16，那么该字段中剩余的字节就会被设备忽略。

5.2 命令状态封包（Command Status Wrapper, 简称为 CSW）

命令状态封包总共用了 13 个字节。

表 5.2 命令状态封包

bit	7	6	5	4	3	2	1	0
Byte								
0-3	<i>dCSWSignature</i>							
4-7	<i>dCSWTag</i>							
8-11 (8-Bh)	<i>dCSWDataResidue</i>							
12 (Ch)	<i>bCSWStatus</i>							

*dCSWSignature:*

该字段是 CSW 的标志，固定为 53425355h。注意：这个值是按照通常的高位在前、低位在后的二进制数顺序排列的；但是，CSW 在 USB 总线上传输的时候，总是先发送 LSB，再发送 MSB。该字段占用 4 字节。

*dCSWTag:*

该字段是命令状态标签（Command Status Tag），该值与相应的 *dCBWTag* 字段相同。该字段占用 4 字节。

*dCSWDataResidue:*

该字段表示 *dCBWDataTransferLength* 字段中主机希望的数据长度与实际发送的数据长度之间的差额。该字段占用 4 字节。

*bCSWStatus:*

该字段指示了该命令的执行情况。如果命令执行成功，设备将会设置该字节为 0。如果命令执行失败，将会由下表所示的非零值来指示：

表 5.3 命令块状态值

Value	Description
00h	命令执行成功
01h	命令执行失败
02h	段错误
03h 和 04h	保留（没有用）
05h 到 FFh	保留

5.3 数据传输情况

这个部分描述了主机和设备是如何保持同步的。

主机通过设定 CBW 中的方向位和 *dCBWDataTransferLength* 字段来指示想要进行何种传输。紧接着设备决定实际传输的数据的长度和方向并按照第 6 章《主机/设备数据传输》所阐述的方式响应主机，如果错误则 STALLing 端点，并且返回适当的 CSW。

5.3.1 命令传输

主机可以通过 Bulk-Out 端点向设备发送包含一个命令块的命令块封包（CBW）。发送的 CBW 有 31（1Fh）个字节。

如果传输成功，设备通过 ACK 握手包返回确认信息。如果设备接收到不合法的 CBW，请参考 6.6.1 章节《CBW 不合法》。如果主机在 Bulk-Out 端点检测到 STALL 信息，主机将以复位恢复响应。（参见 5.3.4 章节《复位恢复》）

### 5.3.2 数据传输

所有的数据传输都应该起始于封包的边界。主机将会尝试发送或者接收由 `dCBWDataTransferLength` 字段和方向位所指定的精确个数的字节数据。设备会按第 6 章《主机/设备数据传输》所阐述的方式响应主机。

为了在数据传输完成之前报告错误情况并尽量保证数据的完整性，设备可能通过停止（STALLing）端点的使用来终止命令。

### 5.3.3 状态传输

设备通过 Bulk-In 端点来发送每一个 CSW 到主机。CSW 起始于封包的边界，包括 13 (Dh) 个字节的内容。图 2 定义了主机所使用的任何 CSW 传输的法则。

CSW 向主机描述相应的 CBW 中的命令的执行状况。`dCSWDataResidue` 字段表示 `dCBWDataTransferLength` 字段中主机希望的数据长度与实际发送的数据长度之间的差额。主机忽略收到的多余的数据。

#### 5.3.3.1 段错误 (Phase error)

如果 CSW 返回的是段错误状态，主机将执行复位恢复。

### 5.3.4 复位恢复

为了产生复位恢复，主机将遵循以下顺序：

- Bulk-Only Mass Storage 复位
- 对 Bulk-In 端点进行 *Clear Feature HALT* 动作
- 对 Bulk-Out 端点进行 *Clear Feature HALT* 动作

## 6 主机/设备数据传输

### 6.1 概述

主机尝试通过发送一个 CBW 启动一个数据传输（输入、输出或者无）时，一个批传输过程就开始了。设备接收到这个 CBW 后，对该封包进行检查、译码，尝试满足主机的请求，然后通过 CSW 返回该命令执行的状态。本节更加详细的描述了主机和设备之间正常和非正常的批传输过程。

## 6.2 合法且有意义的 CBW

主机通过发送 CBW 来向设备表示它的目的。设备对收到的每个 CBW 后执行两次确认。首先，设备检测收到的数据是否是一个 CBW。接着，设备检测该 CBW 中的数据是否有意义。

设备只将 *dCBWTag* 字段的内容复制到相应的 CSW 中的 *dCSWTag* 字段中。

### 6.2.1 合法的 CBW (Valid CBW)

如果满足下述情况，设备则认为收到的 CBW 是合法的：

- 接收到的 CBW 是在设备发送 CSW 或复位之后，
- CBW 长度为 31 (1Fh) 字节，
- *dCBWSignature* 的值为 43425355h。

### 6.2.2 有意义的 CBW (Meaningful CBW)

如果满足下述情况，设备则认为收到的 CBW 是有意义的：

- 保留字段没有被设置，
- *bCBWLUN* 字段所指示的逻辑单元数 (LUN) 不大于设备所拥有的逻辑单元数的实际值，
- *bCBWCBLength* 和 *CBWCB* 的内容同 *bInterfaceSubClass* 一致。

## 6.3 合法且有意义的 CSW

设备一般把它尝试满足主机的要求的结果通过 CSW 通告主机。主机在每个 CSW 上执行两个确认过程。主机首先确认所接收到的封包是否是一个 CSW，其次，主机确认收到的 CSW 是否有意义。

如果满足下述情况，主机则认为收到的 CSW 是合法的：

- CSW 的长度为 13 (Dh) 个字节，

- *dCSWSignature* 的值是 53425355h,
- *dCSWTag* 同相应的 CBW 的 *dCBWTag* 一致。

### 6.3.2 有意义的 CSW

如果满足下述情况，主机则认为收到的 CSW 是有意义的：

*bCSWStatus* 字段的值为 00h 或 01h 并且 *dCSWDataResidue* 不大于 *dCBWDataTransferLength*。

或者

*bCSWStatus* 值为 02h。

### 6.4 设备错误处理

设备可能不能完全满足主机的请求。当设备发现它无法完全满足主机的请求时，那可能是因为总线上正在进行一个 Bulk-In 或者 Bulk-Out 传输，主机可能有其它的还没有完成的请求。设备可能使主机通过 STALLing 相应的管道来结束这个传输。

被设备响应的无意义的 CBW 是不被处理的。

STALL 是否真正的出现在总线上决定于当设备准备好 STALL 该端点时总线上是否有传输正在进行。

### 6.5 主机错误处理

当主机收到一个不合法的 CSW 时，主机会执行复位恢复。当主机接收到一个无意义的 CSW 时，主机可能执行复位恢复。

### 6.6 错误类型

在主机和设备间的传输中，总共存在 4 类错误。这些错误通常是相互关联的并且它们可能发生在传输过程中的任何时候。

### 6.6.1 CBW 不合法

如果 CBW 不合法，设备会 STALL Bulk-In 管道。同样，设备会 STALL Bulk-Out 管道或者接收并丢弃所有的 Bulk-Out 数据。设备将维持现存状态直到主机进行复位恢复。

### 6.6.2 内部设备错误 (Internal Device Error)

如果除了复位外没有一个可靠的手段来恢复，设备会进行内部设备错误的检测。如果复合下述条件，设备将会响应：

STALLing 任何正在进行的数据传输并且返回段错误 (Phase Error) 状态 (*bCSWStatus=02h*)。

或者

STALLing 全部的将要在 Bulk-In 和 Bulk-Out 管道上进行的请求命令直到复位恢复。

### 6.6.3 主机/设备冲突

在确认接收到的 CBW 是合法且有意义的之后，当没有内部错误时，设备可能检测为什么它不能满足主机所希望进行的由 CBW 中 *bmCBWFlags* 字段的方向位和 *dCBWDataTransferLength* 字段所指示的数据传输。在有的情况中，设备可能需要复位才能得以恢复。在这些情况中，设备会返回段错误 (Phase Error) 状态 (*bCSWStatus=02h*)。段错误 (Phase Error)、无段错误 (no Phase Error) 的细节情况将会在 6.7 节描述。

### 6.6.4 命令失败

在确认接收到的 CBW 是合法且有意义的之后，设备仍然可能会因为尝试满足主机命令请求而失败。设备会通过返回命令失败状态 (Command Failed) (*bCSWStatus=01h*) 来报告这种情况。

## 6.7 十三种情况

这部分描述了主机希望且设备打算在没有重要错误的情况下可能产生的十三种情况。表 6.1 形象的显示了这十三种情况。

十三种情况的要点。

- 情况（1），（6）和（12）代表大多数的主机和设备之间的传输。它们表明了主机和设备在传输的方向和传输数据的数量上达成一致的情况。这些情况同样涉及诸如“the thin diagonal”。
- 在下个章节中，任何没有被明确描述的主机或设备的行为不属于该规范讨论的范围并且它的作用结果是不确定的。

表 6.1 主机/ 设备数据传输矩阵

		主机		
		$H_n$	$H_i$	$H_o$
设备	$D_n$	(1) $H_n=D_n$	(4) $H_i>D_n$	(9) $H_o>D_n$
	$D_i$	(2) $H_n<D_i$	(5) $H_i>D_i$	(10) $H_o<>D_i$
			(6) $H_i=D_i$	
			(7) $H_i<D_i$	
	$D_o$	(3) $H_n<D_o$	(8) $H_i<>D_o$	(11) $H_o>D_o$
				(12) $H_o=D_o$
				(13) $H_o<D_o$

LEGEND
主机希望
$H_n$ 主机希望没有数据传输
$H_i$ 主机希望从设备获取数据
$H_o$ 主机希望发送数据到设备
设备目的
$D_n$ 设备打算不传输数据
$D_i$ 设备打算传输数据给主机
$D_o$ 设备打算从主机获得数据



### 6.7.1 $H_n$ -主机希望没有数据传输情况：(1) $H_n=D_n$

当  $dCBWDataTransferLength$  字段的值为 0 时这种情况将会发生。(2)  $H_n<D_i$

这表明主机不希望发送数据到设备或者从设备获取数据。(3)  $H_n<D_o$

这种情况的一般需求为：

- 方向位的值不影响这些情况的结果。

指定的主机请求是：

- 主机发送合法且有意义的 CBW。
  - 主机可能设置或清除方向位 (*Direction* bit)。
- 主机尝试接收一个 CSW。
- 在 STALL 在状态下接收 CSW，然后：
- 主机清除 Bulk-In 管道。
- 主机再次尝试接收 CSW。
- 当 CSW 是合法且有意义的，然后：
- [情况(1)]

$bCSWStatus=00h$  或  $01h$ ,  $dCSWDataResidue$  为 0。

- [情况(2)或(3)]

如果  $bCSWStatus=02h$ , 然后：

主机忽略  $dCSWDataResidue$  的值。

主机执行复位恢复。

制定的设备请求是：

- 设备接收 CBW。
- 当 CBW 是合法且有意义的，然后：
  - 设备尝试该命令。
  - [情况(1)]

如果设备没有数据要发送或者接收，然后：

设备将 *bCSWStatus* 设置为 00h 或 01h。

设备将 *dCSWDataResidue* 设置为 02h。

- [情况(2)或(3)]

如果设备确实有数据要接收或发送，那么：

设备将 *CSWStatus* 设置为 02h。

- 设备返回一个合法且有意义的 CSW。
  - 如果 *bCSWStatus* 值为 00h 或 01h，设备将可能 STALL Bulk-In 管道。

### 6.7.2 *H<sub>i</sub>*-主机希望从设备获得数据情况：(4) $H_i > D_n$

当 *dCBWDataTransferLength* 字段为非零值且方向 (5)  $H_i > D_i$

位 (*Direction* bit) 为 1 (Data-In) 时，这些情况会发生。(6)  $H_i = D_i$

(7)  $H_i < D_i$

(8)  $H_i < > D_o$

指定的主机请求是：

- 主机发送一个合法且有意义的 CBW。
- 主机尝试从设备获得数据。
- 在 STALL 状态下接收数据，然后：
  - 主机接受接收到的数据。
  - 主机清除 Bulk-In 管道。
- 主机尝试接收 CSW。
- 在 STALL 状态下接收 CSW，然后：
- 主机清除 Bulk-In 管道。
- 主机再次尝试接收 CSW。
- 当 CSW 合法且有意义时：
- [情况(4), (5)或者(6)]

如果 *bCSWStatus*=00h 或 01h，然后：

主机通过判断 *dCBWDataTransferLength* 和 *dCSWDataResidue* 字段的差值来确定接收到的数据的数量。

- [情况(7)或(8)]

如果 *bCSWStatus*=02h，然后：

主机将忽略 *dCSWDataResidue* 字段的值。

主机执行复位恢复。

指定的设备请求是：

- 设备接收一个 CBW。
- 当该 CBW 合法且有意义时：
  - 设备尝试命令。
  - [情况(6)]

如果设备想要发送 *dCBWDataTransferLength* 时：

设备发送 *dCBWDataTransferLength* 字节的数据。

设备设置 *bCSWStatus* 为 00h 或 01h。

设备设置 *dCSWDataResidue* 为 0。

- [情况(4)或(5)]

如果设备想要发送的数据的数量少于主机指定的数据数量时：

设备发送该数据。

设备可能发送填充数据以符合 *dCBWDataTransferLength* 字段的要求。

如果设备实际传输的数据数量少于主机所指定的数据数量时：

设备可能以一个短封包结束该传输。

设备 STALL 该 Bulk-In 管道。

设备将 *bCSWStatus* 字段设置为 00h 或 01h。

设备设置 *dCSWDataResidue* 字段的值为 *dCBWDataTransferLength* 的值与实际发送数据数量的差值。

- [情况(7)或(8)]

如果设备想要发送或者接收到的数据的数量都大于主机所指定的数量时：

设备可能发送的数据的数量为 *dCBWDataTransferLength* 所能表示的最大值。

如果设备实际传输的数据的数量比主机所指定的数量少，那么：

设备可能以一个短封包结束该传输。

设备 STALLBulk-In 管道。

如果设备实际传输 *dcBWDataTransferLength* 的数量的数据：

设备可能停止（STALL）批量输入（Bulk-In）管道。

设备将 *bCSWStatus* 字段的值设置为 02h。

- 设备将返回一个合法且有意义的 CSW。

### 6.7.3 Ho-主机想要发送数据到设备情况：(9)Ho>Dn

当 *dcBWDataTransferLength* 为非零值并且方向 (10)Ho<>Di

位 (*Direction* bit) 为 0 (Data-Out) 时发生这种 (11)Ho>Do

情况。这表示主机想要发送数据到设备。(12)Ho=Do

(13)Ho<Do

这种情况的一般要求是：

- 主机不能发送 0 长度的封包。

主机的特殊要求是：

- 主机发送合法且有意义的 CBW。
- 主机发送数据给设备。
  - 主机仅需要在在数据传输结束时发送一个短封包。
- 在 STALL 状态下发送数据，那么：
- 主机清除 Bulk-Out 管道。
  - 主机尝试接收 CSW。
  - 在 STALL 状态下接收 CSW，那么：
- 主机清除 Bulk-In 管道。
- 主机再次尝试接收 CSW。
  - 当该 CSW 是合法且有意义的，那么：
- [情况 (9), (11) 或 (12)]

如果 *bCSWStatus*=00h 或 01h，那么：

主机确定 *dcBWDataTransferLength* 与 *dCSWDataResidue* 的差值个数的被处理过的数据

- [情况(10)或(13)]

如果 *bCSWStatus*=02h, 那么:

主机忽略 *dCSWDataResidue* 字段的值。

主机执行复位恢复。

设备的特殊要求是:

- 设备接收一个 CBW。
- 当该 CBW 合法且有意义, 那么:
- 设备尝试该命令。
- [情况(9), (11)或(12)]

如果设备想要处理的数据的数量不大于主机指示的数据的数量, 那么:

设备将接收想要接收的数据。

设备接收 *dCBWDataTransferLength* 所支持的最大数量的数据或者通过 STALLing 提前结束该 Bulk-Out 管道。

设备将 *bCSWDataResidue* 字段设置为 00h 或 01h。

设备设置 *dCSWDataResidue* 为 *dCBWDataTransferLength* 和实际被设备处理的数据的数量的差值。

- [情况(10)或(13)]

如果设备既想处理多于主机指定的数据又想发送数据, 那么:

设备可能接收 *dCBWDataTransferLength* 所支持的最大数量的数据。

设备接收 *dCBWDataTransferLength* 所支持的最大数量的数据或通过 STALLing 提前结束该 Bulk-Out 管道。

设备设置 *bCSWStatus* 为 02h。

- 设备返回一个合法且有意义的 CSW。

如果 *bCSWStatus* 字段的值不是 00h 或者 01h, 设备将停止批量输入 (Bulk-In) 管道。