# 2.2.8.9 Optimized Buffer Flush/Fill (OBFF) Message

The OBFF Message is optionally used to report platform central resource states to Endpoints. This mechanism is described in detail in Section 6.19 .

The following rules apply to the formation of the OBFF Message:

• Table 2-33 defines the OBFF Message.

• The OBFF Message does not include a data payload (TLP Type is Msg).

• The Length field is Reserved.

• The Requester ID must be set to the Transmitting Port's ID.

• The OBFF Message must use the default Traffic Class designator (TC0). Receivers that implement OBFF support must check for violations of this rule. If a Receiver determines that a TLP violates this rule, it must handle the TLP as a Malformed TLP.

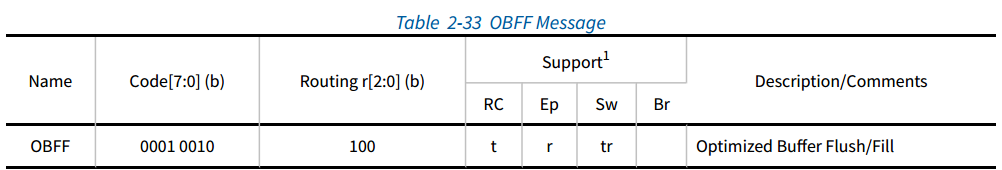
This is a reported error associated with the Receiving Port (see Section 6.2 ).

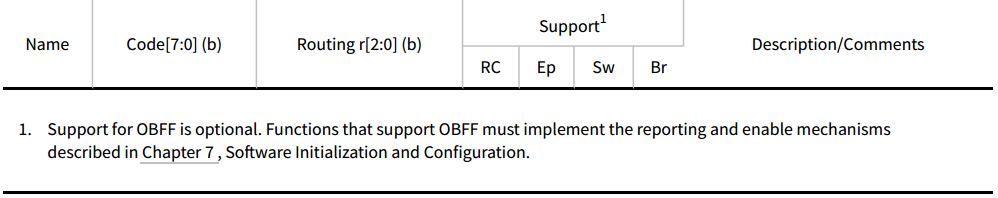
OBFF消息可选地用于向EP报告平台中心资源状态。该机制在第6.19节中有详细描述。

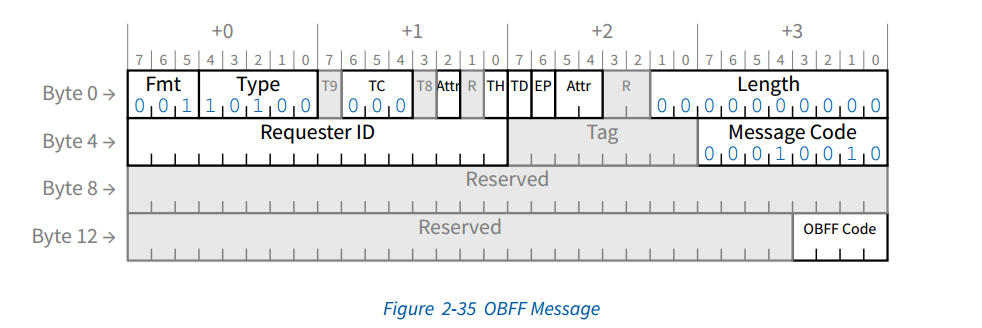
以下规则适用于OBFF消息的形成：

* 表2-33定义了OBFF消息。
* OBFF消息不包括数据有效载荷（TLP类型为Msg）
* “长度”字段为“保留”。
* 请求方ID必须设置为传输端口的ID。
* OBFF消息必须使用默认的交通等级指示符（TC0）。实施OBFF支持的接收者必须检查是否违反了此规则。如果接收方确定TLP违反了此规则，则必须将该TLP作为格式错误的TLP进行处理。

这是一个与接收端口相关的报告错误（见第6.2节）。







# 6.19 Optimized Buffer Flush/Fill (OBFF) Mechanism

The Optimized Buffer Flush/Fill (OBFF) Mechanism enables a Root Complex to report to Endpoints (throughout a hierarchy) time windows when the incremental platform power cost for Endpoint bus mastering and/or interrupt activity is relatively low. Typically this will correspond to time that the host CPU(s), memory, and other central resources associated with the Root Complex are active to service some other activity, for example the operating system timer tick.

The nature and determination of such windows is platform/implementation specific.

An OBFF indication is a hint - Functions are still permitted to initiate bus mastering and/or interrupt traffic whenever enabled to do so, although this will not be optimal for platform power and should be avoided as much as possible.

当EP总线主控和/或中断活动的增量平台功率消耗相对较低时，优化缓冲区刷新/填充（OBFF）机制使RC能够向EP（整个层次结构）报告时间窗口。通常，这将对应于主机CPU、存储器和与RC相关联的其他中央资源活动以服务于一些其他活动（例如操作系统计时器滴答）的时间。

此类窗口的性质和确定是特定于平台/实现的。

OBFF指示是一种提示-无论何时启用，仍允许Function启动总线主控和/或中断业务，尽管这对平台功耗来说不是最佳的，应尽可能避免。

OBFF is indicated using either of the WAKE# signal or a message (see Section 2.2.8.9 ). The message is to be used exclusively on interconnects where the WAKE# signal is not available. WAKE# signaling of OBFF or CPU Active must only be initiated by a Root Port when the system is in an operational state, which in an ACPI compliant system corresponds to the S0 state. Functions that are in a non-D0 state must not respond to OBFF or CPU Active signaling.

The OBFF message routing is defined as 100b, for point-to-point, and is only permitted to be transmitted in the Downstream direction. There are multiple OBFF events distinguished. When using the OBFF Message, the OBFF Code field is used to distinguish between different OBFF cases:

OBFF使用WAKE#信号或消息来指示（见第2.2.8.9节）。该消息仅用于WAKE#信号不可用的互连上。只有当系统处于操作状态时，OBFF或CPU Active的WAKE#信令才能由RP发起，在符合ACPI的系统中，该操作状态对应于S0状态。处于非D0状态的Function不得响应OBFF或CPU活动信令。

OBFF消息路由被定义为100b，用于点对点，并且仅被允许在下行方向上传输。有多个OBFF事件可区分。使用OBFF消息时，OBFF代码字段用于区分不同的OBFF情况：

1111b “CPU Active” - System fully active for all Device actions including bus mastering and interrupts

0001b “OBFF” - System memory path available for Device memory read/write bus master activities

0000b “Idle” - System in an idle, low power state

All other codes are Reserved.

1111b“CPU激活”-所有设备操作（包括总线主控和中断）的系统完全激活

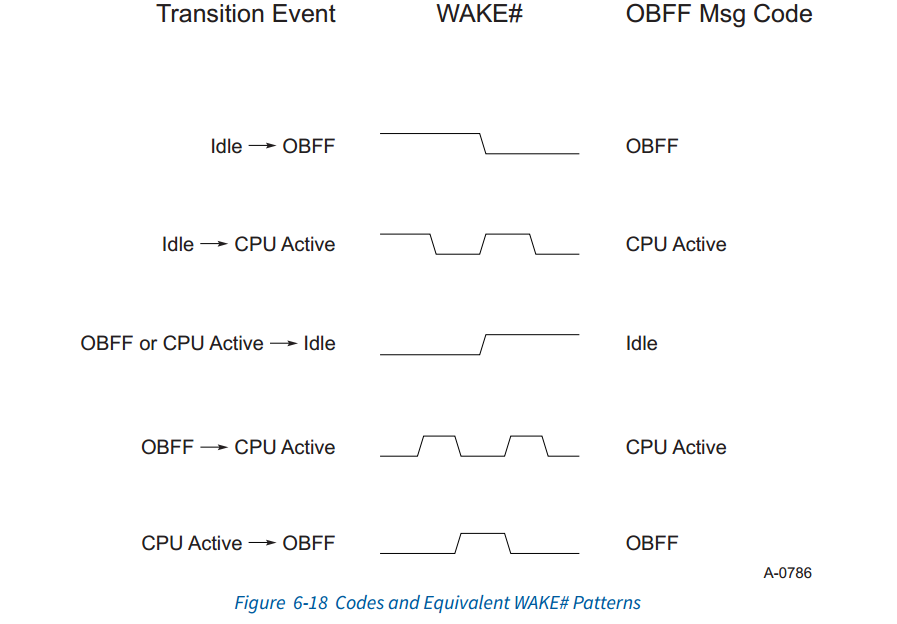
0001b“OBFF”-可用于设备存储器读/写总线主控活动的系统存储器路径

0000b“空闲”-系统处于空闲、低功率状态。

其他Code为保留值。

These codes correspond to various assertion patterns of WAKE# when using WAKE# signaling, as shown in Figure 6-18 .There is one negative-going transition when signaling OBFF and two negative going transitions each time CPU Active is signaled. The electrical parameters required when using WAKE# are defined in the WAKE# Signaling section of [CEM-2.0] (or later).

当使用WAKE#信令时，这些Code对应于WAKE#的各种断言模式，如图6-18所示。当发信号通知OBFF时有一个负向转变，每次发信号通知CPU Active时有两个负向转换。在[CEM-2.0]（或更高版本）的WAKE#信令部分中定义了使用WAKE#时所需的电气参数。



When an OBFF Message is received that indicates a Reserved code, the Receiver, if OBFF is enabled, must treat the indication as a “CPU Active” indication.

An OBFF Message received at a Port that does not implement OBFF or when OBFF is not enabled must be handled as an Unsupported Request (UR). This is a reported error associated with the receiving Port (see Section 6.2 ). If a Port has OBFF enabled using WAKE# signaling, and that Port receives an OBFF Message, the behavior is undefined.

当接收到指示为保留Code的OBFF消息时，如果启用了OBFF，则接收器必须将该指示视为“CPU活动”指示。

在未实现OBFF或未启用OBFF的端口接收到的OBFF消息必须作为不支持的请求（UR）处理。这是一个报告的与接收端口相关的错误（见第6.2节）。如果某个端口使用WAKE#信令启用了OBFF，并且该端口接收到OBFF消息，则该行为未定义。

OBFF indications reflect central resource power management state transitions, and are signaled using WAKE# when this is supported by the platform topology, or using a Message when WAKE# is not available. OBFF support is discovered and enabled through reporting and control registers described in Chapter 7 . Software must not enable OBFF in an Endpoint unless the platform supports delivering OBFF indications to that Endpoint.

When the platform indicates the start of a CPU Active or OBFF window, it is recommended that the platform not return to the Idle state in less than 10 μs. It is permitted to indicate a return to Idle in advance of actually entering platform idle, but it is strongly recommended that this only be done to prevent late Endpoint activity from causing an immediate exit from the idle state, and that the advance time be as short as possible.

It is recommended that Endpoints not assume CPU Active or OBFF windows will remain open for any particular length of time.

OBFF指示反映了CPU功耗管理状态的变化，并且当平台拓扑支持WAKE#时，使用WAKE#来发信号，或者当WAKE#不可用时使用消息来发信号。OBFF支持是通过第7章中描述的报告和控制寄存器发现和启用的。软件不得在EP中启用OBFF，除非平台支持向该EP传递OBFF指示。

当平台指示启动CPU活动或OBFF窗口时，建议平台不要在10μs内返回空闲状态。允许在实际进入平台空闲状态之前指示返回空闲状态，但强烈建议这样做只是为了防止后期EP活动导致立即退出空闲状态，并且提前时间尽可能短。

建议EP不要假设CPU活动或OBFF窗口将在任何特定的时间长度内保持打开。

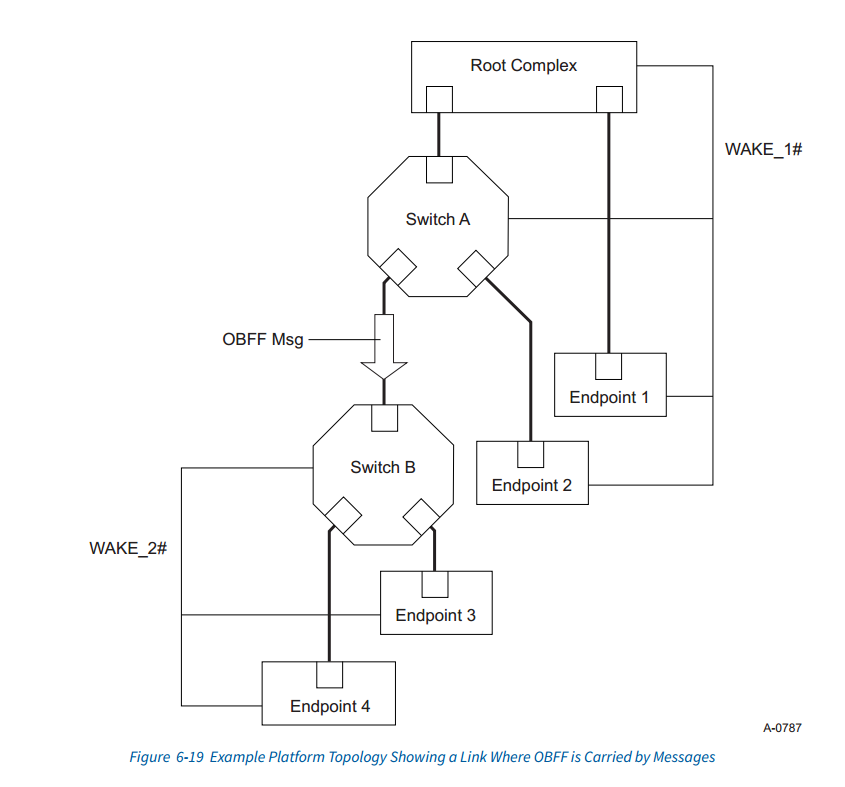


Figure 6-19 shows an example system where it is necessary for a Switch (A) to translate OBFF indications received using WAKE# into OBFF Messages, which in this case are received by another Switch (B) and translated back to using WAKE# signaling. A HwInit configuration mechanism (set by hardware or firmware) is used to identify cases such as shown in this example (where the link between Switch A and Switch B requires the use of OBFF Messages), and system firmware/ software must configure OBFF accordingly.

When a Switch is configured to use OBFF Message signaling at its Upstream Port and WAKE# at one or more Downstream Ports, or vice-versa, when enabled for OBFF, the Switch is required to convert all OBFF indications received at the Upstream Port into the appropriate form at the Downstream Port(s).

图6-19显示了一个示例系统，其中Switch（A）有必要将使用WAKE#接收的OBFF指示转换为OBFF消息，在这种情况下，OBFF消息由另一个Switch（B）接收并转换为使用WAKE#signaling。HwInit配置机制（由硬件或固件设置）用于识别如本例所示的情况（其中Switch A和Switch B之间的链路需要使用OBFF消息），并且系统固件/软件必须相应地配置OBFF。

当Switch配置为在其上行端口使用OBFF消息信令，在一个或多个下行端口使用WAKE#时，反之亦然，当启用OBFF时，Switch需要将在上行端口接收的所有OBFF指示转换为在下行端口的适当形式。

When using WAKE#, the enable for any specific Root Port enables the global use of WAKE# unless there are multiple WAKE# signals, in which case only the associated WAKE# signals are affected. When using Message signaling for OBFF, the enable for a particular Root Port enables transmission of OBFF messages from that Root Port only. To ensure OBFF is fully enabled in a platform, all Root Ports indicating OBFF support must be enabled for OBFF. It is permitted for system firmware/software to selectively enable OBFF, but such enabling is beyond the scope of this specification.

当使用WAKE#时，任何特定RP都会启用WAKE#的全局使用，除非有多个WAKE#信号，在这种情况下，只有相关的WAKE#信号受到影响。当对OBFF使用消息信令时，特定RP的启用仅允许从该RP传输OBFF消息。为了确保OBFF在平台中完全启用，必须为OBFF启用所有指示OBFF支持的RP。允许系统固件/软件选择性地启用OBFF，但这种启用超出了本规范的范围。

To minimize power consumption, system firmware/software is strongly recommended to enable Message signaling of OBFF only when WAKE# signaling is not available for a given link.

OBFF signaling using WAKE# must only be reported as supported by all components connected to a Switch if it is a shared WAKE# signal. In these topologies it is permitted for software to enable OBFF for components connected to the Switch even if the Switch itself does not support OBFF.

为了最大限度地减少功耗，强烈建议系统固件/软件仅在WAKE#信令对给定链路不可用时才启用OBFF的消息信令。

如果使用WAKE#的OBFF信号是共享WAKE#信号，则必须仅报告为连接到Switch的所有组件支持。在这些拓扑结构中，允许软件为连接到Switch的组件启用OBFF，即使Switch本身不支持OBFF。

It is permitted, although not encouraged, to indicate the same OBFF event more than once in succession.

When a Switch is propagating OBFF indications Downstream, it is strongly encouraged to propagate all OBFF indications. However, especially when using Messages, it may be necessary for the Switch to discard or collapse OBFF indications. It is permitted to discard and replace an earlier indication of a given type when an indication of the same or a different type is received.

允许（尽管不鼓励）连续多次指示同一OBFF事件。

当Switch向下游传播OBFF指示时，强烈建议传播所有OBFF指示。但是，尤其是在使用消息时，Switch可能需要丢弃或折叠OBFF指示。当接收到相同或不同类型的指示时，允许丢弃和替换给定类型的较早指示。

Downstream Ports can be configured to transmit OBFF Messages in two ways, which are referred to as Variation A and Variation B. For Variation A, the Port must transmit the OBFF Message if the Link is in the L0 state, but discard the Message when the Link is in the Tx\_L0s or L1 state. This variation is preferred when the Downstream Port leads to Devices that are expected to have communication requirements that are not time-critical, and where Devices are expected to signal a non-urgent need for attention by returning the Link state to L0. For Variation B, the Port must transmit the OBFF Message if the Link is in the L0 state, or, if the Link is in the Tx\_L0s or L1 state, it must direct the Link to the L0 state and then transmit the OBFF Message. This variation is preferred when the Downstream Port leads to devices that can benefit from timely notification of the platform state.

下行端口可以配置为以两种方式传输OBFF消息，这两种方式被称为变体A和变体B。对于变体A，如果链路处于L0状态，则端口必须传输OBFF信息，但当链路处于Tx\_L0s或L1状态时，则丢弃该信息。当下行端口预期设备有非时间关键的通信要求，并且预期设备通过将链路状态返回到L0来发出非紧急关注需求的信号时，变体A是优选的。对于变体B，如果链路处于L0状态，则端口必须发送OBFF消息，或者如果链路处于Tx\_L0s或L1状态，则必须将链路引导到L0状态然后发送OBFF信息。当下行端口导致设备可以从平台状态的及时通知中受益时，这种变化是优选的。

When initially configured for OBFF operation, the initial assumed indication must be the CPU Active state, regardless of the logical value of the WAKE# signal, until the first transition is observed.

When enabling Ports for OBFF, it is recommended that all Upstream Ports be enabled before Downstream Ports, and Root Ports must be enabled after all other Ports have been enabled. For hot pluggable Ports this sequence will not generally be possible, and it is permissible to enable OBFF using WAKE# to an unconnected hot pluggable Downstream Port. It is recommended that unconnected hot pluggable Downstream Ports not be enabled for OBFF message transmission.

当最初配置为OBFF操作时，无论WAKE#信号的逻辑值如何，初始假设指示必须是CPU活动状态，直到观察到第一次转换。

为OBFF启用端口时，建议在启用下游端口之前启用所有上游端口，并且必须在启用所有其他端口之后启用RP。对于热插拔端口，通常不可能执行此顺序，并且允许使用WAKE#对未连接的热插拔下游端口启用OBFF。建议不要为OBFF消息传输启用未连接的热插拔下游端口。

IMPLEMENTATION NOTE

OBFF Considerations for Endpoints

It is possible that during normal circumstances, events could legally occur that could cause an Endpoint to misinterpret transitions from an Idle window to a CPU Active window or OBFF window. For example, a non-OBFF Endpoint could assert WAKE# as a wakeup mechanism, masking the system’s transitions of the signal. This could cause the Endpoint to behave in a manner that would be less than optimal for power or performance reasons, but should not be unrecoverable for the Endpoint or the host system.

实施说明

在正常情况下，可能会合法发生事件，导致EP误解从空闲窗口到CPU活动窗口或OBFF窗口的转换。例如，非OBFF EP可以断言WAKE#作为唤醒机制，从而屏蔽系统的信号转换。这可能会导致EP的行为方式因电源或性能原因而不太理想，但对于EP或主机系统来说不应是不可恢复的。

In order to allow an Endpoint to maintain the most accurate possible view of the host state, it is recommended that the Endpoint place its internal state tracking logic in the CPU Active state when it receives a request that it determines to be host-initiated, and at any point where the Endpoint has a pending interrupt serviced by host software.

为了使EP能够得到和维护主机的最准确状态，建议EP在接收到其确定由主机启动的请求时，以及在EP具有由主机软件服务的挂起中断的任何时刻，将其内部状态跟踪逻辑置于CPU活动状态。