

# Sizing of the Replay Buffer in PCI Express Devices



*MindShare, Inc.*

Joe Winkles  
[joe@mindshare.com](mailto:joe@mindshare.com)

October 2003

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designators appear in this document, and MindShare was aware of the trademark claim, the designations have been printed in initial capital letters or all capital letters.

The authors and publishers have taken care in preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

Copyright ©2003 by MindShare, Inc.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher.

Printed in the United States of America.

First Printing, October 2003

Find MindShare on the World-Wide Web at:  
<http://www.mindshare.com/>

Introduction.....	4
Function of the Replay Buffer .....	4
Size Does Matter.....	5
Too Small.....	5
Too Big .....	5
Sizing Considerations .....	6
Wire Time .....	6
<i>AckLatency</i> .....	7
<i>TxL0sAdjustment</i> .....	8
<i>RxL0sAdjustment</i> .....	9
<i>SafetyFactor</i> .....	9
<i>LinkUtilizationFactor</i> .....	9
The Equation.....	10
Example Calculation.....	11
L1 ASPM Consideration.....	12
The Table .....	12

# Replay Buffer Sizing in PCI Express

---

## Introduction

The Replay Buffer (also known as the Retry Buffer) is an integral part of every PCI Express device. This buffer holds each Transaction Layer Packet (TLP) that is transmitted from a device until that TLP is implicitly or explicitly acknowledged by the receiving (target) device on the other end of the link. If the Replay Buffer is not sized correctly based on the device's capabilities and characteristics, then it could have adverse effects on that device's performance or cost effectiveness.

This paper will briefly discuss the role of the Replay Buffer in a PCI Express system and point out the effects of not sizing this buffer correctly. It then will show the terms that should be taken into consideration when determining the size of a device's Replay Buffer, as well as give an example equation of what the resulting calculation should look like.

---

## Function of the Replay Buffer

One of the most attractive features of PCI Express is the notion of Quality of Service (QoS). A significant piece in QoS is data integrity, ensuring that a transaction arrives at its destination fully intact without error. A major component implemented by PCI Express to address this issue is the Replay Buffer. The Replay Buffer's purpose is to keep a copy of all transmitted TLPs until the receiving device at the other end of the link notifies the transmitter that it received the TLPs successfully. This process is known as the ACK/NAK protocol (a detailed description of this protocol can be found in Chapter 5 of MindShare's book entitled "*PCI Express System Architecture*").

The function that the Replay Buffer provides can be described without going into too much detail about how the ACK/NAK protocol works. Before a TLP is transmitted on the line, it is placed in the Replay Buffer and remains there until that TLP has been positively acknowledged by the receiving device on the other end of the link. Once an acknowledge packet arrives from the receiver which acknowledges that TLP, it can be removed from the Replay Buffer providing additional room for new TLPs. However, if no acknowledge packet was received or the acknowledge packet that was received indicated a negative acknowledge, then that TLP and any TLPs transmitted after it must be retransmitted or "replayed" out of the Replay Buffer. Even after being retransmitted, those TLPs must remain in the Replay Buffer until the receiver positively acknowledges it received them in order and without error. To ensure that the device on the other side of the link receives the transmitted packets correctly at least once, unacknowledged TLPs in the Replay Buffer cannot be removed or overwritten with new TLPs.

---

## Size Does Matter

As with any data transfer technology, there are always components that can become bottlenecks in terms of performance. The Replay Buffer in PCI Express is one of those components which is why sizing this buffer correctly is critical for devices which desire high performance, low latency transfers, efficiency, and cost effectiveness. From a performance viewpoint, a large Replay Buffer looks attractive, but from a cost and area perspective, a smaller Replay Buffer is more enticing. Both of these extremes and their effects on the device are discussed in the following two sections.

### Too Small

As with all physically implemented buffers, there is a limit to its size, and since unacknowledged TLPs in the Replay Buffer cannot be removed or overwritten, new requests may have to be stalled until there is room in the Replay Buffer for more packets. This situation throttles the performance of a device and can result in a large amount of “dead-time” on the PCI Express link. Thus the device is not utilizing the significant amount of bandwidth that is available with a high-speed, serialized technology such as PCI Express. The scenario described above results from a Replay Buffer that is too small. In other words, the Replay Buffer fills up long before acknowledge packets arrive from the receiver allowing TLPs to be removed from the buffer. The good news is that the spec defines the maximum amount of time that a receiver can wait before sending an acknowledge packet for a received TLP. Using this value and some other terms discussed later, the appropriate size of a device’s Replay Buffer can be calculated and this scenario can be avoided.

### Too Big

All manufactured semiconductor devices have some cost and silicon area limitations. Implementing a Replay Buffer that is larger than necessary will not hinder the performance of the device at all, but will increase the cost to manufacture and the die area needed for the buffer. The same performance can be achieved with a smaller and less expensive solution by correctly calculating the size of the Replay Buffer based on the device’s characteristics instead of selecting some unnecessarily large buffer without justification. This paper describes the important characteristics of a device for the Replay Buffer sizing calculation.

## Sizing Considerations

In order to size the Replay Buffer appropriately, we need to be able to determine the amount of time between when a TLP is placed in the Replay Buffer and when that packet is positively acknowledged by the receiver and removed from the buffer (referred to as the round-trip time of the packet). There are several characteristics of the device that contribute to this round-trip time and must be factored in when calculating the size of its Replay Buffer. These characteristics, or terms, that will be used in the sizing equation will be discussed in detail in the following sections. Each of these terms is going to be a part of the round-trip time and is measured in symbol times - the amount of time needed to transmit a symbol or 10 bits. The following sections define these terms as well as describe their relationship to the round-trip time of a packet.

### Wire Time

This is the amount of time it takes to transmit a packet on the wire. The wire time is needed twice in the sizing equation: once to determine how long it takes to transmit the TLP, and once to account for the amount of time it takes the target device to transmit the ACK packet. The number of symbol times each of these packets takes on the wire is based on the size of the packet and the width of the link. This is because each lane of a link transmits one symbol during a symbol time. For example, a x1 link can transmit one symbol during a single symbol time. However a x2 link can transmit two symbols during a single symbol time, and an x8 link can transmit 8 symbols during a single symbol time, et cetera.

For the TLP transmission, the *MaxPayloadSize* parameter must be used to account for the largest sized packet. This only accounts for the actual payload of the packet, in addition there is the header and framing information (*TLPOverhead*) that is transmitted as part of the packet as well. In order to get the total number of symbol times it takes to transmit this TLP, the *MaxPayloadSize* and *TLPOverhead* terms are added together and divided by the *LinkWidth*, as shown below.

$$\frac{MaxPayloadSize + TLPOverhead}{LinkWidth}$$

The size of each acknowledge packet is always the same at 8 symbols (STP(1) + DLLP(4) + CRC(2) + END(1)). However, the number of symbol times it takes to transmit an acknowledge packet could vary based on the width of the link. Therefore the term used in the equation to describe this amount of time is referred to as *AckPacket* and its value is shown below. (*AckPacket* should always be an

---

# Replay Buffer Sizing in PCI Express

---

integer, so if the calculation shown below results in a fraction, that value should be rounded up.)

$$AckPacket = \frac{8}{LinkWidth}$$

## **AckLatency**

This is the amount of time the target device (receiving device) is allowed to wait after receiving the last symbol of a TLP before the first symbol of the acknowledge packet must be transmitted back to the originating device. The PCI Express specification defines what this time is, based on the *LinkWidth* and *MaxPayloadSize* of the target device. (While the spec never explicitly says so, it must be true that the enabled *MaxPayloadSize* for each device on both ends of a link must be set to the same value.) The equation in the spec for calculating the *AckLatency* value is shown below (not including the *L0sAdjustment* term, this term is discussed later in this paper).

$$AckLatency = \frac{(MaxPayloadSize + TLPOverhead) * AckFactor}{LinkWidth} + InternalDelay$$

The *AckFactor* term in this equation is defined by the spec to be “the number of maximum size TLPs which can be received before an Ack is sent.” Reasonable values (between 1.0 and 3.0 based on *MaxPayloadSize* and *LinkWidth*) were selected for this term in order to balance the Replay Buffer size versus the bandwidth efficiency of the link. If the *AckFactor* was very large, then the Replay Buffer would also have to be very large, but the efficiency of the link would be high (in terms of transmitting TLPs instead of many overhead acknowledge packets). If the *AckFactor* term was very small, then the Replay Buffer would not need to be very big, but more acknowledge packets would be traversing the link, decreasing its efficiency.

The *InternalDelay* term in this equation is supplied in the spec as a constant of 19 symbol times. It indicates the amount of time it takes to process received TLPs and DLLPs. This process time includes verifying the packet is error free, determining what type of packet it is, updating any internal registers or flags based on the contents of the packet, et cetera.

The *AckLatency* values calculated in the spec for each *MaxPayloadSize* and *LinkWidth* are shown in Table 1. (Symbol times are the units for each value in the cells, and the AF term indicates the *AckFactor* used for that calculation.)

# Replay Buffer Sizing in PCI Express

MaxPayloadSize	x1 Link	x2 Link	x4 Link	x8 Link	x12 Link	x16 Link	x32 Link
128 Bytes	237 (AF=1.4)	128 (AF=1.4)	73 (AF=1.4)	67 (AF=2.5)	58 (AF=3.0)	48 (AF=3.0)	33 (AF=3.0)
256 Bytes	416 (AF=1.4)	217 (AF=1.4)	118 (AF=1.4)	107 (AF=2.5)	90 (AF=3.0)	72 (AF=3.0)	45 (AF=3.0)
512 Bytes	559 (AF=1.0)	289 (AF=1.0)	154 (AF=1.0)	86 (AF=1.0)	109 (AF=2.0)	86 (AF=2.0)	52 (AF=2.0)
1024 Bytes	1071 (AF=1.0)	545 (AF=1.0)	282 (AF=1.0)	150 (AF=1.0)	194 (AF=2.0)	150 (AF=2.0)	84 (AF=2.0)
2048 Bytes	2095 (AF=1.0)	1057 (AF=1.0)	538 (AF=1.0)	278 (AF=1.0)	365 (AF=2.0)	278 (AF=2.0)	148 (AF=2.0)
4096 Bytes	4143 (AF=1.0)	2081 (AF=1.0)	1050 (AF=1.0)	534 (AF=1.0)	706 (AF=2.0)	534 (AF=2.0)	276 (AF=2.0)

**Table 1.** *AckLatency values defined in the PCI Express specification.*

## ***TxL0sAdjustment***

This is the amount of time it takes before the originating device's transmitter can start transmitting TLPs or DLLPs upon initiating an exit from the L0s state. This time can be divided into two stages: (1) the amount of time it takes the originating device's transmitter in the Physical Layer (PHY) to start sending Fast Training Sequence (FTS) ordered-sets once stimulated to do so, and (2) the amount of time it takes the target device's receiver to re-establish bit and symbol lock once it starts receiving the FTS ordered-sets. The first stage in this value is known as the P0s to P0 transition time as defined in the Intel document entitled "*PHY Interface for the PCI Express Architecture*" (aka the PIPE document). The second stage of this value is directly proportional to the number of FTS ordered-sets that the receiver on the target device needs to re-acquire bit and symbol lock. The amount of time needed in each of these stages is simply added together to calculate the value to be used for the *TxL0sAdjustment* term.

This term must be taken into consideration because a device can start depositing TLPs into its Replay Buffer while that device's Physical Layer (PHY) has placed its transmission lines into the L0s Active State Power Management (ASPM) state. In this situation, the PHY may not know to initiate an exit from L0s until the TLP actually hits the PHY, which means that the TLP (or at least a portion of it) has already been deposited into the Replay Buffer. Based on information from the industry, this value is expected to be around 16-32 symbol times. (For calculations in this paper, 20 symbol times is used for this value. This incorporates 4 symbol times needed for the P0s to P0 transition time + 12 symbol times needed to transmit 3 FTS ordered-sets + 4 symbol times to send a single SKP ordered-set).



---

# Replay Buffer Sizing in PCI Express

---

## ***RxL0sAdjustment***

This is the amount of time it takes before the target device's transmitter can start transmitting TLPs or DLLPs upon initiating an exit from the L0s state. (The same two stages apply to this term as described previously for the *TxL0sAdjustment* term.) The *RxL0sAdjustment* term is necessary because of the situation where a target device has received a TLP for which it needs to send an acknowledge packet, but its transmission line is in the L0s ASPM state. Then, when its *AckLatency* timer expires, it must transition the transmitting line back to the L0 state before it is able to send the acknowledge packet. The PCI Express specification refers to this value in its discussion of the *AckLatency* timer, and even includes it in the *AckLatency* equation, but the values calculated in the table (as shown in Table 1) do not include this adjustment. (The spec refers to this value as the *TxL0sAdjustment*, but for this application we are calling it the *RxL0sAdjustment* because we are referring to it from the originating device's perspective, and the spec was looking at it from the target device's perspective.) Based on information from the industry, this value is expected to be around 16-32 symbol times. (For calculations in this paper, 20 symbol times is used for this value. This incorporates 4 symbol times needed for the P0s to P0 transition time + 12 symbol times needed to transmit 3 FTS ordered-sets + 4 symbol times to send a single SKP ordered-set).

## ***SafetyFactor***

This is a customization term in the sizing equation which allows the Replay Buffer to tolerate some abnormal conditions on the link or in the target device. For example, if the device on the other end of the link is not 100% spec compliant, and its *AckLatency* timeout value exceeds the spec defined timeout value, then this could cause the Replay Buffer to fill-up, and thus throttle the transmission of new TLPs because the target device is not sending acknowledge packets back at the rate expected. This situation can be guarded against by using a *SafetyFactor* greater than 1. Another example would be the case in which a designer would like the Replay Buffer to be able to tolerate receiving a corrupt acknowledge packet or missing one or more transmitted acknowledge packets. In this case, the *SafetyFactor* could also be adjusted to ensure that the transmitting portion of the link stays saturated with new TLPs instead of creating dead-time on the link. The *SafetyFactor* term and its usefulness will be discussed in more detail later in this paper.

## ***LinkUtilizationFactor***

This is another customization term in the sizing equation which allows the Replay Buffer to be sized based on the device's capabilities. For instance, if a device does not have the ability to transmit enough packets to keep its transmission lines 100% saturated, then its Replay Buffer should be sized accordingly. A value of 1 in this term indicates that this device could consume 100% of its transmission line's bandwidth. A value of  $\frac{3}{4}$  indicates 75% utilization of the transmit portion of

# Replay Buffer Sizing in PCI Express

---

the link, et cetera. This term should only take into consideration the transmit (or “sourcing”) capabilities of the device and not the receive (or “sinking”) device capabilities.

---

## The Equation

Note that the equation shown in this section is not mandated by the PCI Express specification. This equation was derived by MindShare after analyzing what terms should be taken into consideration when calculating the size of the Replay Buffer. This equation is simply a reference to be used by PCI Express device designers.

The Replay Buffer sizing equation for PCI Express devices is shown below. The basis of this equation calculates the worst case round-trip time of a TLP in symbol times and then converts that time measurement into bytes for sizing the Replay Buffer.

$$\left[ TxL0sAdjustment + \frac{MaxPayloadSize + TLPOverhead}{LinkWidth} + (AckLatency + RxL0sAdjustment + AckPacket) * SafetyFactor + InternalDelay \right] * LinkUtilizationFactor * LinkWidth$$

The first two terms in the equation (*TxL0sAdjustment* and the wire time of the TLP) describe how long it takes the packet to actually reach the target device once it started being placed in the originating device’s Replay Buffer. The next three terms (*AckLatency*, *RxL0sAdjustment*, and *AckPacket*) describe the maximum amount of time allowed between acknowledge packets sent by the target device. The next term, *SafetyFactor*, allows the designer to enable the Replay Buffer to tolerate longer than expected intervals between acknowledge packets.

The *SafetyFactor* should not be set to less than 1 or more than 3. It should not be less than 1 because this would indicate that the Replay Buffer could overflow before the target device (under normal operation) sends an acknowledge packet. The Replay Buffer should always be sized large enough to handle normal intervals between acknowledge packets without throttling the transmission of new TLPs. The *SafetyFactor* should not be more than 3 because this would indicate that the Replay Buffer is large enough to tolerate receiving more than 2 consecutive corrupt acknowledge packets. At this point, the *Replay\_Timer* in the originating device is going to timeout and begin a replay of all the TLPs in the Replay Buffer.

The *InternalDelay* term in the equation indicates the amount of time it takes the originating device to receive the acknowledge packet and process it. The result of

# Replay Buffer Sizing in PCI Express

---

processing the acknowledge packet is to flush an entry (or entries) out of the Replay Buffer which completes the round-trip time of a TLP. This entire round-trip time of the TLP is then multiplied by the *LinkUtilizationFactor* which allows the designer to size the Replay Buffer based on the amount of transmission bandwidth the device is expecting to use. If a device is not expecting (or does not have the capability) to saturate the transmit lines of a link with TLPs, then that device's Replay Buffer can be smaller than a device which will utilize 100% of the link's transmission bandwidth.

Up to this point in the equation, everything has been measured in symbol times. However because we are looking for the size of the Replay Buffer, we need to convert the measured worst case round-trip time of a TLP into the number of bytes the Replay Buffer should be. This can easily be accomplished by multiplying the round-trip time by the width of the link. The round-trip time is measured in symbol times, and each symbol time represents the time needed to transmit one symbol. A symbol is simply a byte that has been turned into a 10-bit value through 8B/10B Encoding. Therefore, for a x1 link, 10 symbol times corresponds to transmitting 10 bytes out of the Replay Buffer. For a x2 link, 10 symbol times corresponds to transmitting 20 bytes out of the Replay Buffer because there is a symbol (equivalent to a byte of data) being transmitted on each lane of the link during one symbol time. Multiplying the *LinkWidth* by the round-trip time indicates the max number of bytes that the originating device is capable of transmitting before the acknowledge packet is received and processed. This max number of bytes that can be transmitted indicates the number of bytes the Replay Buffer should be able to hold.

## Example Calculation

Assuming a *MaxPayloadSize* of 2048, a *LinkWidth* of 4, a *SafetyFactor* and *LinkUtilizationFactor* of 1.0, the Replay Buffer sizing calculation would be:

$$\left[ TxL0sAdjustment + \frac{MaxPayloadSize + TLPOverhead}{LinkWidth} + (AckLatency + RxL0sAdjustment + AckPacket) * SafetyFactor + InternalDelay \right] * LinkUtilizationFactor * LinkWidth$$

$$\left[ 20 + \frac{2048 + 28}{4} + (538 + 20 + 2) * 1.0 + 19 \right] * 1.0 * 4 = 4472bytes$$

# Replay Buffer Sizing in PCI Express

## L1 ASPM Consideration

The equation described above only takes into consideration transitions out of the L0s state and not the L1 ASPM state. This is due to the fact that the author feels the Replay Buffer does not need to be sized to handle transitions out of the L1 ASPM state. This is based on the facts that the recovery time to transition out of L1 back into L0 is significantly larger than the transition time from L0s to L0 (up to 2 or 3 times in magnitude larger), and the L1 ASPM state is going to be entered significantly less often than the L0s state is entered. These two factors indicate that the typical case is going to be the L0s transitions and that the effect on the size of the Replay Buffer would be drastic if incorporating L1 ASPM transitions. Due to these facts, it seems that the reasonable design point for the size of the Replay Buffer should be based on L0s and not L1 ASPM. However, if sizing the Replay Buffer based on the L1 ASPM state is of interest, then this can be calculated by simply replacing the *TxL0sAdjustment* term with the transition time to transfer out of L1 ASPM into L0 (*L1RecoveryTime* measured in symbol times).

## The Table

Using the following values, the sizing values for the Replay Buffer is shown in Table 2 (the units are bytes).

- L0sAdjustment – 20 symbol times
- TLPOverhead – 28 symbol times
- InternalDelay – 19 symbol times
- SafetyFactor – 1.0
- LinkUtilizationFactor-1.0

MaxPayloadSize	x1 Link	x2 Link	x4 Link	x8 Link	x12 Link	x16 Link	x32 Link
128 Bytes	460	538	692	1172	1568	1876	3108
256 Bytes	767	844	1000	1620	2080	2388	3620
512 Bytes	1166	1244	1400	1708	2564	2868	4100
1024 Bytes	2190	2268	2424	2732	4096	4404	5636
2048 Bytes	4238	4316	4472	4780	7172	7476	8708
4096 Bytes	8334	8412	8568	8876	13312	13620	14852

Table 2. Replay Buffer Sizing Table.