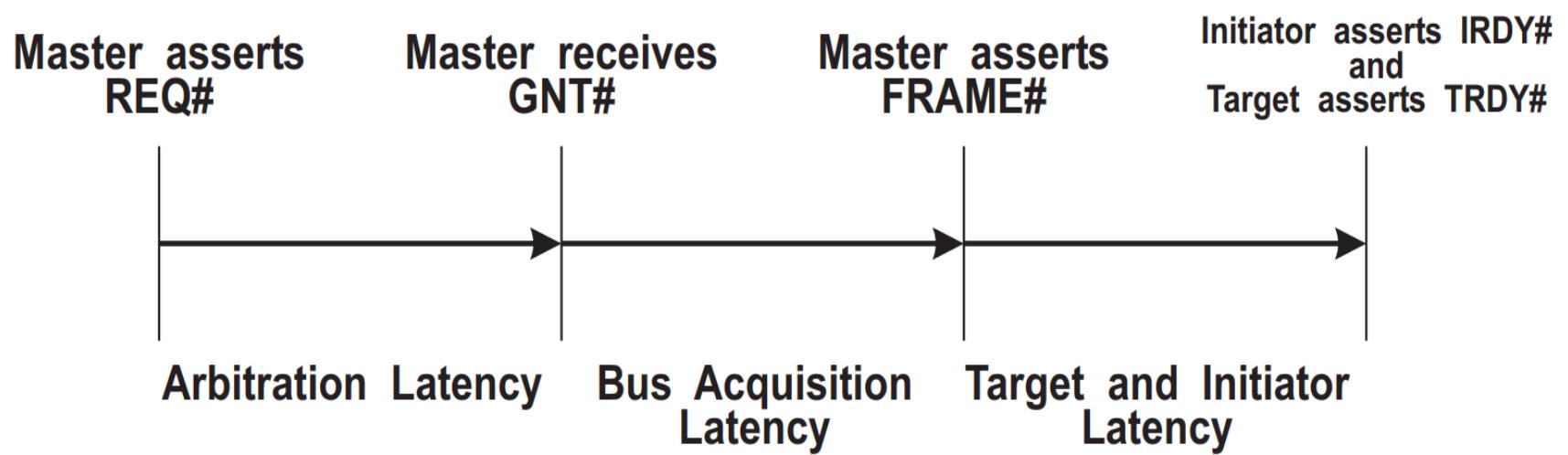


Master and Target Latency

CPE186 Computer Hardware Design
Handout

Bus Access Latency



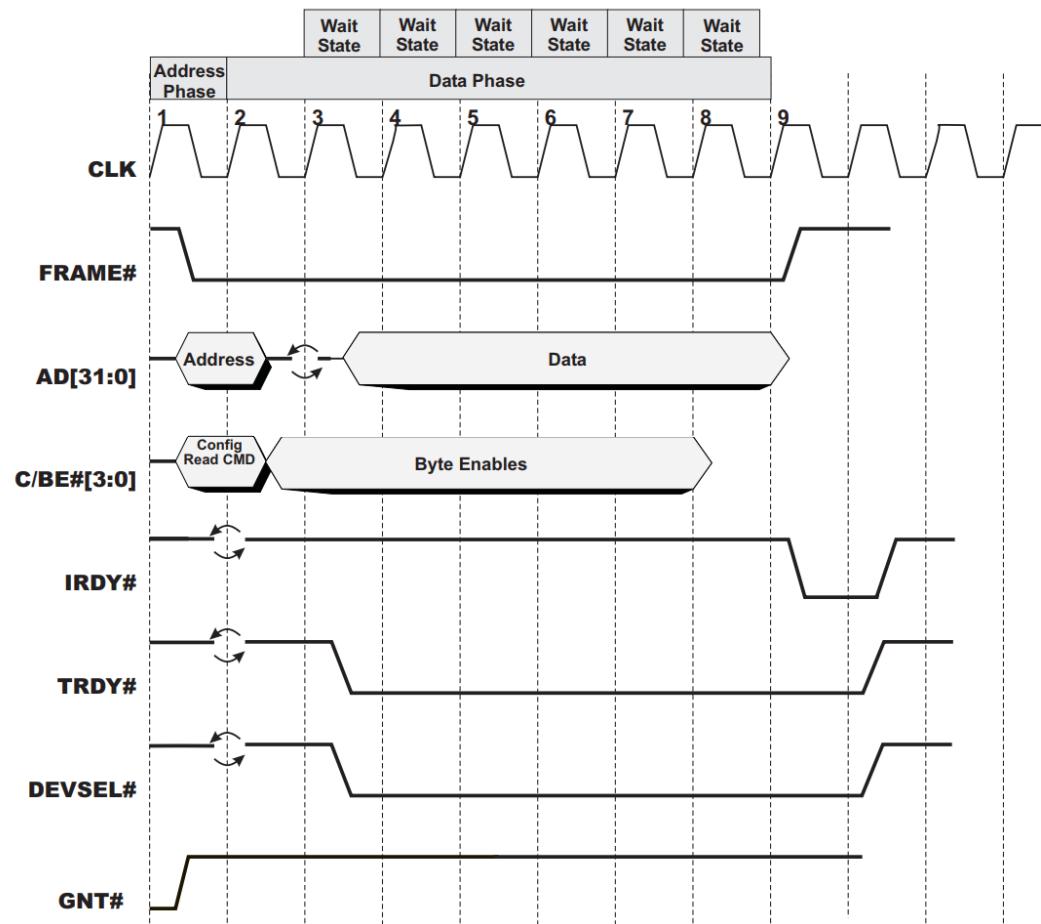
Mechanisms

Master Latency Timer

Target-Initiated Termination

Master Must Transfer Data Within 8 Clock Cycles

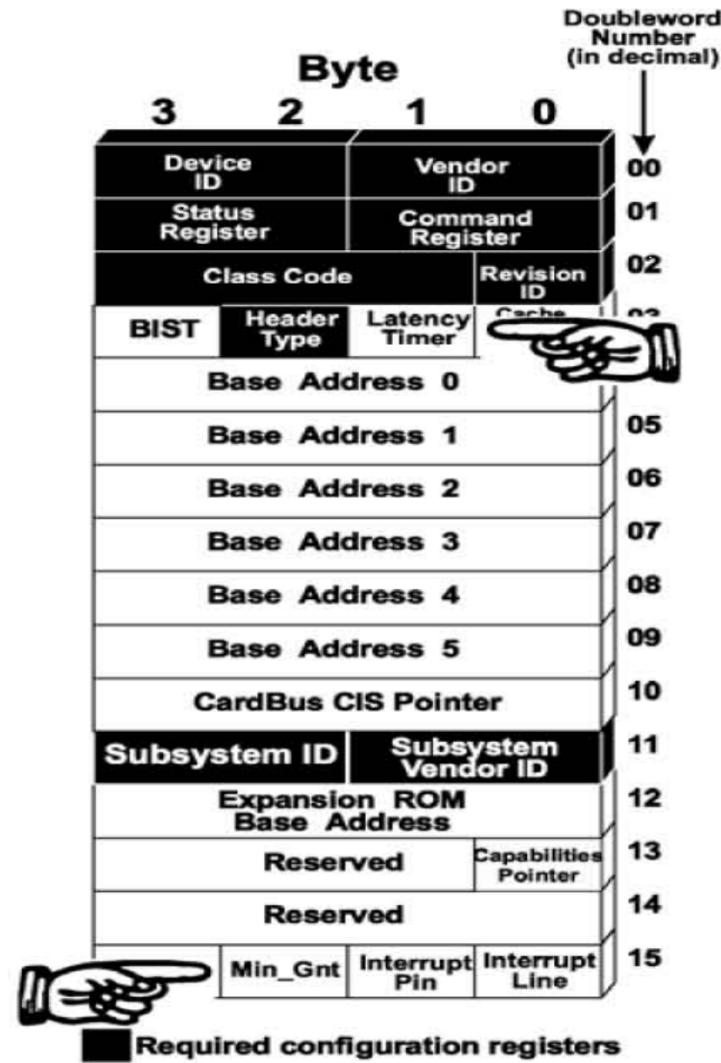
Longest Legal Deassertion of IRDY# In Any Data Phase Is 8 Clocks



Master Must Transfer Data Within 8 Clock Cycles

It is a rule that the initiator must not keep IRDY# deasserted for more than seven PCI clocks during any data phase. In other words, it must be prepared to transfer a data item within eight clocks after entry into any data phase.

Master Latency Timer



Master Latency Timer (LT)

It is either initialized by the configuration software at startup time, or contains a hardwired value.

- LT defines the minimum amount of time (in PCI clock periods) that the bus master is permitted to retain ownership of the bus each time that it acquires bus ownership and initiates a transaction.

Master Latency Timer (LT)

If the current master's LT has not yet been exhausted (decremented all the way down), it has not yet used up its timeslice and may retain ownership of the bus until either:

- it completes its burst transaction or
- its LT expires,

whichever comes first.

Master Latency Timer (LT)

If it is able to complete its burst before expiration of its LT, the other master that has its GNT# may assume bus ownership when it detects that the current master has returned the bus to the idle state. If the current master is not able to complete its burst transfer before expiration of its LT, it is permitted to complete one more data transfer and must then yield the bus.

Master Latency Timer (LT)

If the current master has exhausted its LT, still has its GNT# and has not yet completed its burst transfer, it may retain ownership of the bus and continue to burst data until either:

- it completes its overall burst transfer or
- its GNT# is removed by the arbiter.

In the latter case, the current master is permitted to complete one more data transfer and must then yield the bus.

Master Latency Timer (LT)

Optional

Can be hard wired to zero.

Minimum Grant Register

Minimum Grant Register is implemented by PCI bus master devices only. The value hardwired into this register indicates how long the master would like to retain ownership of the PCI bus (in order to attain good performance) whenever it initiates the transaction.

The value indicates the length of a burst period the device needs, which is in increments of 250ns for 33 MHz and 125ns for 66MHz. A value of 0 indicates the device has no stringent requirement in this area.

Memory Write and Invalidate Command

Any master performing a Memory Write and Invalidate command should not terminate its transfer until it reaches a cache line boundary (even if its LT has expired and it has been preempted) unless STOP# is asserted by the target.

Preventing Master from Monopolizing Bus

1. The inclusion of the LT associated with each master.
2. The rule that requires the initiator to keep IRDY# deasserted for no longer than eight PCI clocks during any data phase.

Preventing Target from Monopolizing Bus

CASE 1. The target cannot transfer the first data item within 16 clocks from the assertion of FRAME#.

CASE 2. Although the target can transfer the first data item within 16 clocks from the assertion of FRAME#, it cannot transfer one of the subsequent data items within eight clocks from the start of the respective data phase. Any data phase other than the first one is referred to as a *subsequent data phase*.

Preventing Target from Monopolizing Bus

Case 1: Target Response

- asserts DEVSEL# to claim the transaction.
- does not assert TRDY#, thereby indicating its unwillingness to transfer the first data item.
- asserts STOP# to indicate that it wants to terminate the transaction with no data transferred in the first data phase and *therefore no data transferred in the transaction*.

Master's Response

terminate the transaction with no data transferred, thus freeing up the bus for other masters to use. After two PCI clocks have elapsed (recommended), the master can reassert its request and, when it receives its GNT#, reinitiate its transaction again.

Two Exceptions of First Data Phase Rule

- **During system INITIALIZATION TIME** (defined as 2^{25} CLKs after RST# removed), targets do not have to adhere to the 16 clock rule. After initialization time has elapsed, all targets must obey the 16 clock rule for first data phase completion.
- A host/PCI bridge that is snooping is permitted to exceed the 16 clock limit, but may never exceed 32 clocks.

Subsequent Data Phases

General. If it will take more than eight PCI clocks to complete a data phase other than the first (referred to as the **subsequent latency timeout**) and it is not the final data phase (FRAME# is still asserted), the target must force the master to terminate the transaction.

Subsequent Data Phases

In Data Phase and Cannot Transfer Data Within 8 Clocks. The target can't transfer the data item in the current data phase within eight clocks after entry into the data phase. In this case, the target had deasserted TRDY# upon entry into the data phase (indicating its unreadiness to transfer the data). The target can handle this scenario in one of two ways:

METHOD 1. Assert STOP# as soon as possible after entry into the data phase (and not later than the eighth clock of the data phase). DEVSEL# remains asserted. This signal combination instructs the master to disconnect from the target with no data transferred in the current data phase. This is referred to as a **disconnect without data transferred**. In the author's opinion, this termination method is greatly preferred over the alternative cited in item 2.

METHOD 2. Alternately, the target could signal a **Target Abort** to the master by keeping TRDY# deasserted, asserting STOP# and deasserting DEVSEL#. Target abort is intended to indicate a fatal error or that the target will never be able to complete the request.

Subsequent Data Phases (continue)

Master's Response

the master must terminate the transaction

After two PCI clocks have elapsed, the master that received the disconnect can reassert its request and, when it receives its GNT#, reinitiate its transaction again at the next data item.

Subsequent Latency (Target) & Latency Timer (LT)

Target's Subsequent Latency and Master's Latency Timer. The subsequent latency timeout is completely independent of the master's LT. The target has no visibility to the master's LT (and vice versa) and therefore cannot tell whether it has timed out or not. This means that slow access targets always (before or after LT expiration) disconnect from the master, thereby fragmenting the overall burst transaction into a series of single data phase transactions.

Target Latency during Initialization Time

Initialization Time vs. Run Time

Initialization time is the period of time during which the POST (Power-On Self-Test) code executes and the PCI devices are discovered and configured. Run time begins after initialization time completes. The 2.1 spec provided a very fuzzy definition of these two time periods and the 2.2 spec is much more definitive. During initialization time, PCI devices are not required to adhere to the 16 clock rule for first data phase completion.

PCI Delayed Transaction

With **PCI Delayed Transaction** enabled, the target device can independently continue the read transaction. So, when the master device successfully gains control of the bus and reissues the read command, the target device will have the data ready for immediate delivery. This ensures that the retried read transaction can be completed within the stipulated latency period.

If the delayed transaction is a write, the master device will rearbitrate for bus access while the target device completes writing the data. When the master device regains control of the bus, it reissues the same write request. This time, the target device just sends the completion status to the master device to complete the transaction.

PCI Delayed Transaction

One advantage of using **PCI Delayed Transaction** is that it allows other PCI masters to use the bus while the transaction is being carried out on the target device. Otherwise, the bus will be left idling while the target device completes the transaction.

Commands That Can Use Delayed Transactions

All transactions that must complete on the destination bus before they can complete on the originating bus may be handled as delayed transactions. This would include:

- Interrupt Acknowledge.
- I/O read.
- I/O write.
- Memory read.
- Memory read line.
- Memory read multiple.
- Configuration read.
- Configuration write.

In other words, everything but the memory write commands.

Request Not Completed and Targeted Again

If a different master attempts to access the target and the target can only deal with one latched request at a time, it must issue a retry to the master without latching its transaction information.

Discard of Delayed Requests

A device that has “memorized” a delayed request and issued a retry to the initiator is permitted to discard the request any time prior to its initiation of the request on the destination bus. This is permitted because the master will retry the transaction again, thereby giving the device the opportunity to “re-memorize” the transaction.

Master Tardy In Repeating Transaction. A bridge must discard data (if a read) and the completion status of a transaction if the master has not retried the transaction within 2^{10} PCI clocks (about 30 μ s at 33 MHz). However, it is recommended that the bridge not discard the transaction until 2^{15} PCI clocks (about 983 μ s at 33 MHz) after it acquired the data or status.

Handling Multiple Data Phases

When the master is successful in completing the first data phase (in other words, it doesn't receive a retry), it may proceed with more data phases.

- If this is a **burst memory read** transaction, the target (assuming that it's a bridge) may have prefetched additional data into a read-ahead buffer and can therefore start streaming data to the master. The bridge could still be reading from the target on the destination bus and a steady data stream could then flow through the bridge until the transfer is complete, the target disconnects, or the master or bridge are preempted on their respective buses.
- If this is a **burst memory write** transaction, the earlier retry may have been issued because the target (a bridge) had a temporarily-full posted memory write buffer. Since then, the bridge could have partially or fully emptied the posted memory write buffer and can therefore start accepting multiple dwords into the buffer rapidly.

The target may issue a disconnect on any data phase after the first. In this case, the master is not required to resume the transaction later. Both the master and the target consider the original request fulfilled.

Master or Target Abort Handling

the transaction on the destination bus ended in a Master Abort

- because no target responded
- or in a Target Abort because the target is broken
- or the target does not support the byte enable combination