X3T9.2 92-141 rev 7

Date: Jul 25, 1993 X3T9.2/92-141 Rev 7

To: X3T9.2 Committee (SCSI)

From: George Penokie (IBM)

Subject: SCSI-3 Queuing Model

| Indicates a change from Rev 5 to Rev 6
+ Indicates a change from Rev 6 to Rev 7

0. Queue

A target resident list of tasks which have not completed.

0.1 Suspended Information

Information stored within a device which is not available to any pending Tasks. Task Management Functions shall not be suspended.

0.2 SCSI Device

Anything which is connected to a SCSI physical transport system and supports the SCSI protocol.

0.30 Completion

The response that a task has ended. The actual events which cause a completion are protocol specific.

0.4 Task Set

One or more Tasks which are grouped by the target so that their interaction is dependent on the queuing and Auto Contingent Allegiance rules.

1. Task Management

1.1 Task Set Management

A target that can accept more than one Task is capable of queuing. The number of Tasks that may exist in a task set depends on the design of the target.

If an initiator requests the creation of more Tasks than can be managed by the Target, each such request is rejected with a status of Queue Full (see x.x).

Several commands may be linked together to form a single Task.

1.2 Untagged Queueing

Untagged queuing allows a target to accept one untagged Task at a time per task set from each Initiator. The target shall allow tagged and untagged Tasks to be contained within a

single task set.

1.3 Untagged Tasks

The target shall handle untagged Tasks in a way which allows tagged and untagged Tasks to be contained within a single task set. Untagged Tasks shall be treated by the target as simple Tasks.

1.4 Tagged Queueing

An initiator may have more than one Tagged Task within each task set under control of the target. The target has the freedom to optimize the sequence of Task completion within a task set unless the initiator chooses to override the optimization.

1.5 Tagged Tasks

The target shall handle tagged Tasks in a way which allows tagged and untagged Tasks to be contained within a single task set.

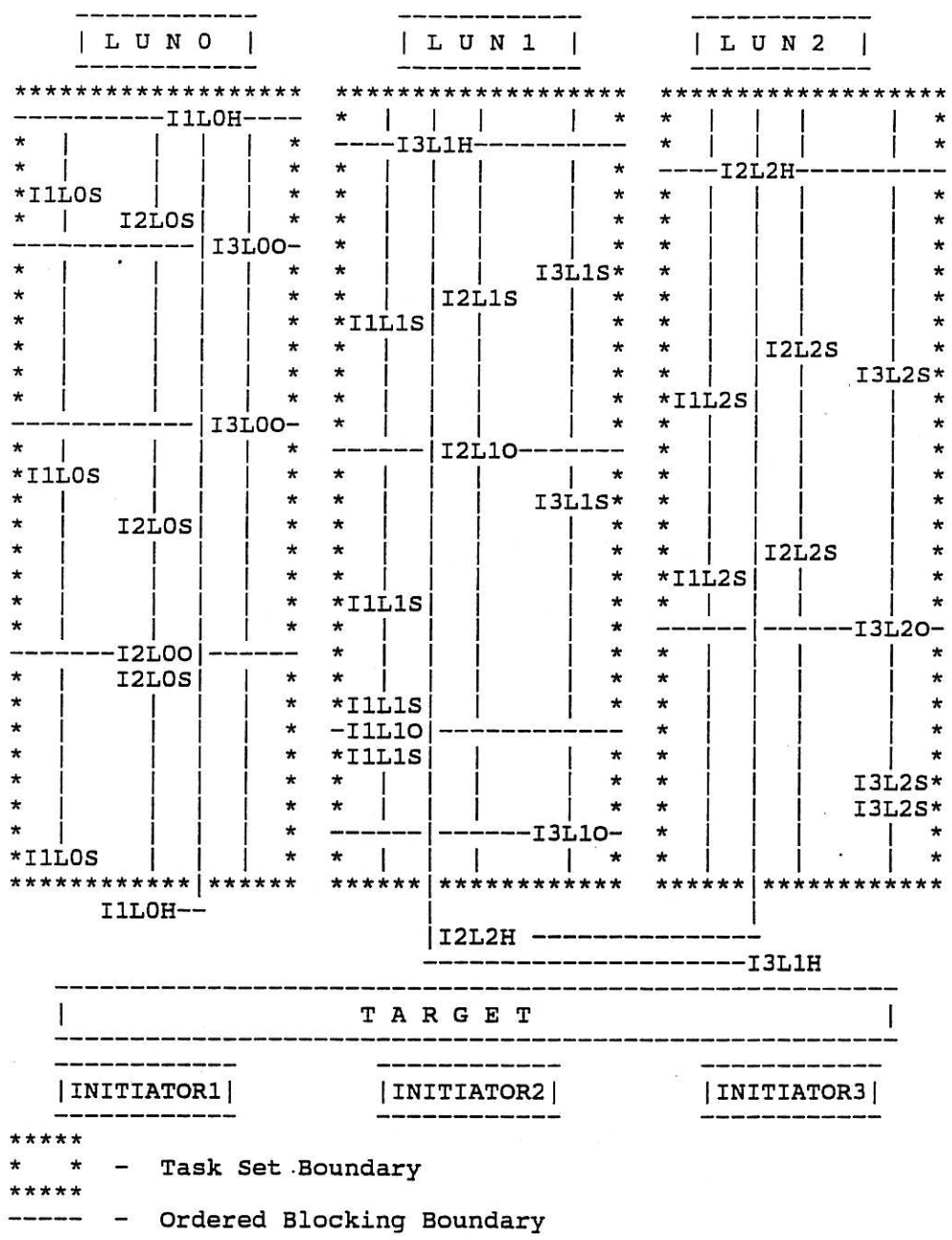
2. Task Ordering Boundaries

A Task ordering boundary defines the scope over which a group of simple Tasks within a task set may be reordered.

The target shall manage all the Tasks from all initiators on all ports accepted for each logical unit as a single task set. Each task set shall be managed independent of all other task sets.

Table 1 shows how task sets are managed in a target with three logical units (i.e. task sets) attached to three initiators.

Table 1 - Per logical unit ordering boundaries



3. Exception Handling

3.1 Auto Contingent Allegiance Condition (ACA)

3.1.1 Auto Contingent Allegiance Condition (ACA)

+ If Auto Contingent Allegiance Condition is not enabled the
+ handling of error conditions is protocol specific.

+ The Auto Contingent Allegiance Condition, if enabled, shall exist within the task set when the target reports an error condition to the initiator through the return of Check Condition or Command Terminated status (see x.x). The Auto Contingent Allegiance Condition shall not cross task set boundaries and shall be preserved within the task set until it is cleared as described in x.x.x.x.

If a condition which would create an Auto Contingent Allegiance Condition occurs during an Auto Contingent Allegiance Condition, the original Auto Contingent Allegiance Condition shall remain. Sense data shall be available for the new Auto Contingent Allegiance Condition.

Implememtors Note: There are issues outside the scope of this standard which occur in multi-initiator systems which should be considered when error recovery is needed.

+ Note: The SCSI-2 Contingent Allegiance condition and Extended
+ Contingent Allegiance condition have been replaced in SCSI-3
+ with the Automatic Contingent Allegiance.

3.1.1.1 Reporting Auto Contingent Allegiance Condition (ACA)

The Auto Contingent Allegiance Condition is reported by the successful transfer of a Check Condition status or a Command Terminated status.

3.1.2 Response to Auto Contingent Allegiance Condition

While the Auto Contingent Allegiance exists the target shall only allow one ACA Task to become a Pending Task per task set. The target shall respond to any other Task which attempts to create a task for the task set with an ACA Active status. The target shall not allow any Pending non-ACA Tasks within the task set to become current Tasks until the Auto Contingent Allegiance is cleared.

| If a Task becomes a current task because of a previous
| request for information that information shall be suspended
| until the ACA is cleared.

3.1.3 Auto Contingent Allegiance Processing

Only ACA Tasks within a task set which has an Auto Contingent Allegiance Condition shall become pending Tasks.

All SCSI operations are permitted while processing an ACA Task

3.1.3.1 Acquiring Sense Data

The sense data for a task set shall be cleared after completion of the first command of an ACA Task for the task set which contains the Auto Contingent Allegiance Condition.

3.1.4 Clearing Auto Contingent Allegiance Condition (ACA)

The Auto Contingent Allegiance Condition shall be cleared after:

-a power on condition,
-performing a hard reset,
-a Clear Auto Contingent Allegiance Task Management request issued to the logical unit by the initiator receiving the Check Condition or Command Terminated status error.
-a Target Reset Task Management request

+ The Auto Contingent Allegiance when enabled shall not be cleared for any reason other than those listed above.

3.2 Duplicate Tag Handling

When issuing a tagged task (see x.x) the initiator shall not reuse the tag until:

| -A service response of Command Complete is received with a status other than INTERMEDIATE or INTERMEDIATE-CONDITION MET or

| -A service response of Service Delivery or Target Failure is received. In this case, system implementations shall guarantee that the task associated with that command has been terminated.

+ If an Task has a duplicate tag an error condition shall be created for the task set with a key of Illegal Request, an ASC of Tagged Overlapped Commands, and an ASCQ which contains the value of the queue tag of the duplicate tag. If the tag value is greater than FFh then a key of Overlapped Commands Attempted shall be used.

The target shall abort all Tasks within the task set for the Initiator which sent the duplicate nexus. The target shall
+ create a single error condition for the task set in which the duplicate nexus error occurred regardless of how many Tasks are aborted.

4. Current Task

4.1 Current Task

A Current Task is a Task which has information being sent or accepted on a physical transport system.

- More than one Current Task may exist at a time on a physical transport system. (e.g. multiport systems)

- A SCSI device on a physical transport system may have a Current Task which is not the same Current Task as another

device on the same physical transport system.

- A SCSI device may simultaneously send information for one or more Tasks and accept information for one or more Tasks.

5. Pending Task

A Task which is not a Current Task and has not yet completed.

6. Simple Task (Simple Tag)

A Simple Task is one that is tagged with a Simple Queue Tag. If the task set only contains Simple Tasks, then any Simple Task may become a Current Task at any time, depending on the setting of queue algorithm mode as specified in the control mode page.

7. Ordered Task (Ordered Tag)

When an Task is tagged as an Ordered Task any information the target has or accepts for the Ordered Task shall be suspended and the Ordered Task shall not complete until all Simple and Ordered Tasks which were accepted into the task set before the Ordered Task have completed.

If Simple or Ordered Tasks are accepted into the task set after an Ordered Task the target shall suspend any information accepted for the new Task and shall not complete any of the new Tasks before the Ordered Task completes.

8. Head Of Queue Task (Head of Queue Tag)

When an Task is tagged as a Head Of Queue Task the target shall suspend any information accepted for any simple or ordered tasks accepted into the task set after the Head Of Queue Task. The target shall not complete any of the new simple or ordered tasks before the most recent Head Of Queue Task completes.

Any Task which was accepted into the task set before a Head Of Queue Task may complete before the Head Of Queue Task completes.

9. ACA Task (ACA Queue Tag)

An ACA Task is tagged with an ACA tag.

An ACA Task is not associated with the Task which caused an Auto Contingent Allegiance Condition.

The ACA Task shall complete before any other Tasks within the task set.

If an ACA Task is accepted for a task set which does not have an Auto Contingent Allegiance Condition the target shall return a Check Condition status with a sense key of Illegal Request and an additional sense code of Invalid Message

Error.

While the Auto Contingent Allegiance exists the target shall only allow one ACA Task per task set to become a Pending Task. The target shall respond to any other Task which attempts to become the current task for that ACAed task set with an ACA Active status.

The target shall create a duplicate tag for an ACA Task if the tag value is not unique for the task set. The target shall respond to the duplicate tag as described in the duplicate tag handling section (see 3.2).

11 Clear Auto Contingent Allegiance

The initiator invokes Clear Auto Contingent Allegiance to clear an auto contingent condition from the specified task set.

The target shall clear the Auto Contingent Allegiance and complete the current Task on acceptance of this task management function.

If the target accepts a Clear Auto Contingent Allegiance and no Auto Contingent Allegiance Condition is in effect for that initiator on that task set, then the target shall complete the current Task.

After the target accepts the Clear Auto Contingent Allegiance any Pending Task within the task set may become a Current Task, subject to the ordering rules.

12. New ASC/ASCQ code

BYTE 12	13	DTLPWRSOMC	DESCRIPTION
4D	NN	DTLPWRSOMC	Tagged Overlapped Commands (NN = Queue Tag)

13. New Status code

Code	Description
30h	Auto Contingent Allegiance Active

AUTO CONTINGENT ALLEGIANCE ACTIVE. This status shall be returned when an Auto Contingent Allegiance exists and an initiator issues a command while one or more or the following conditions are true:

-There is another Auto Contingent Allegiance task in the task set or
-The initiator issuing the command did not cause the Auto Contingent Allegiance condition.

Appendix A - Ordering Boundaries

A.1. Ordering Boundary Disclaimer

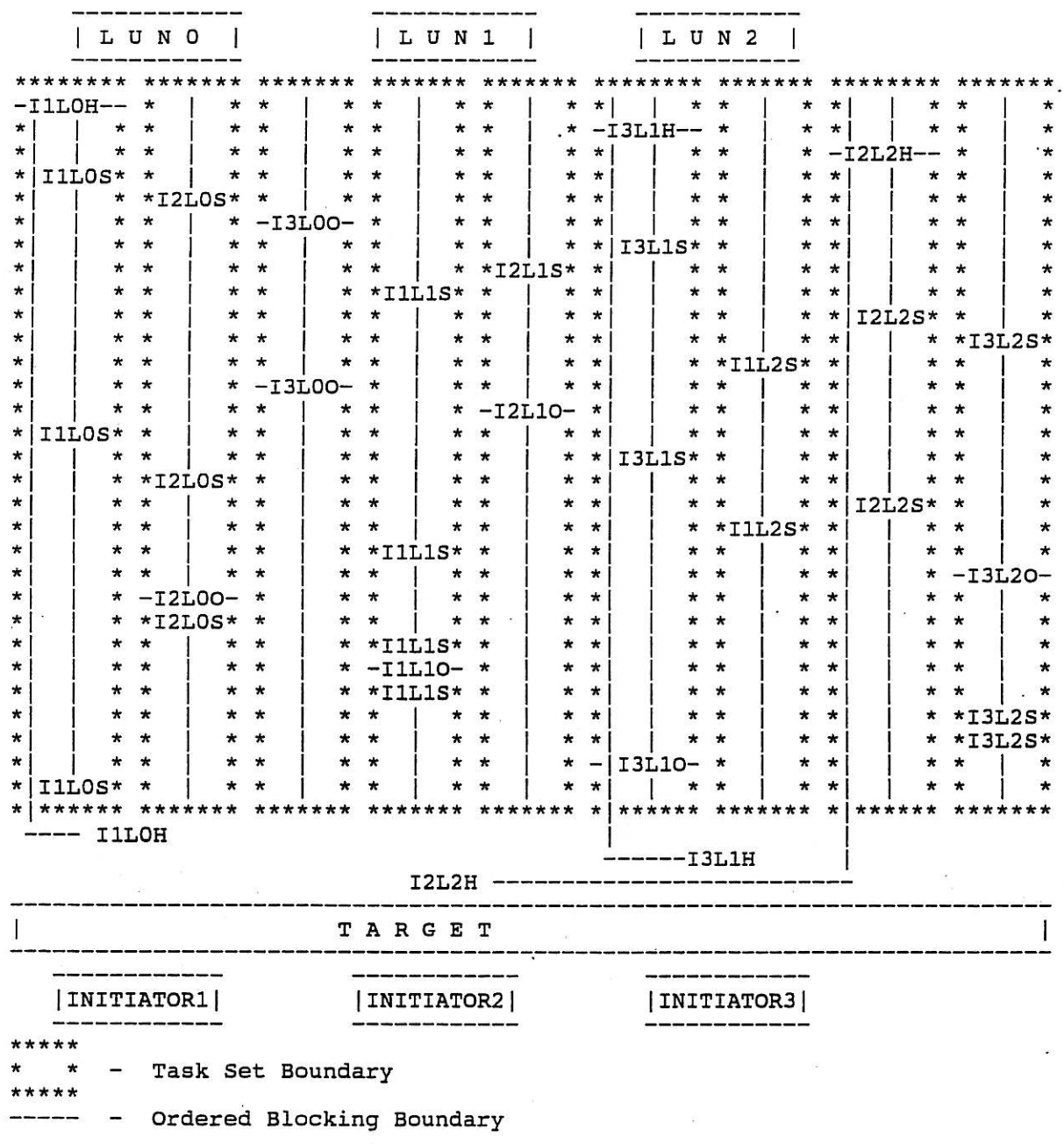
This appendix recognizes that there are other ways to order boundaries than the one chosen by the Standards Committee.

Described below are four different ordering boundaries. Any of these ordering boundaries could be implemented by a SCSI device using the rules for task sets as defined throughout this standard. The standard, however, requires that SCSI devices use the per logical unit ordering boundary.

A.2 Per Initiator/Logical Unit Ordering Boundaries

When the ordering boundary is per initiator/logical unit the target manages all the Tasks accepted from a initiator/logical unit combination on all ports as a single task set. Each task set of Tasks shall be independent of all other task sets.

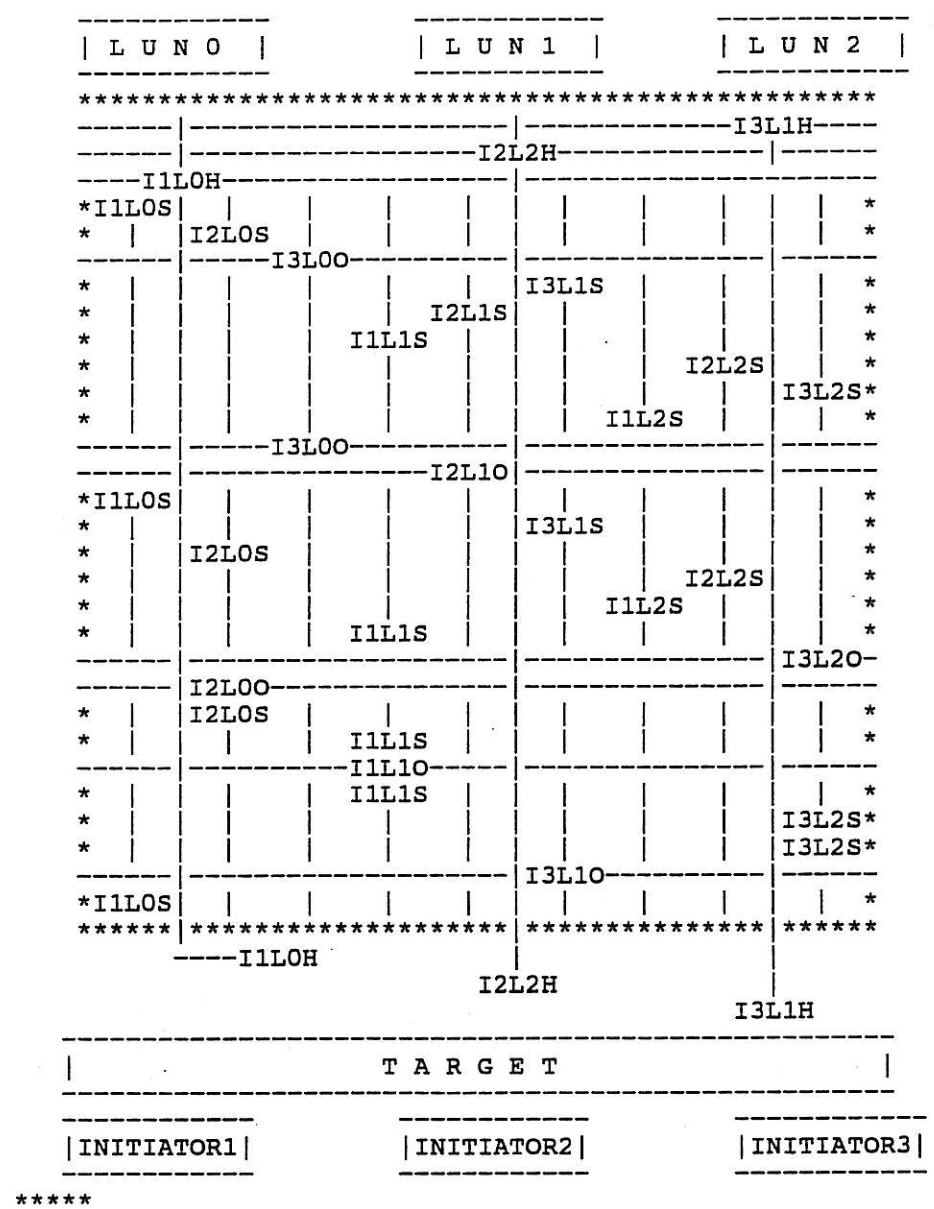
Table 2 - Per initiator/logical unit ordering boundaries



A.3 Per Target Ordering Boundaries

When the ordering boundary is per target the target manages all the Tasks from all initiators on all ports as a single task set.

Table 3 - Per target ordering boundaries



A.4 Per Initiator Ordering Boundaries

The target manages all the Tasks from each initiator, on all ports, as a single task set. Each task set of Tasks shall be independent of all other task sets.

Table 4 - Per initiator ordering boundaries