To: X3T9.2 Committee (SCSI)

From: George Penokie (IBM)

Subject: SCSI-3 Queuing Model

Note: The following list of words are barred from SCSI-3:
- execution -executing -execute
- Active I/O Process -
- Queued I/O Process -

0. Queue

A set of I/O Processes contained within a device which have not completed.

0.1 Media Information

Information stored within a device which is non-volatile (retained through a power cycle) and is accessible over the interface.

0.2 Device

Anything which is connected to SCSI and supports the SCSI protocol.

0.31 Complete (SIP document)

An I/O Process is complete following the next BUS FREE phase after a successful transfer of a COMMAND COMPLETE or a RELEASE RECOVERY message. An I/O Process also completes with the next BUS FREE phase following an ABORT, ABORT TAG, BUS DEVICE RESET, CLEAR QUEUE message, or a hard RESET condition, or an unexpected disconnect.

0.32 Complete (SBP document)

An I/O Process is complete after the successful transfer of an information packet which contains a command complete indication.

0.4 Set

One or more I/O Processes

1. I/O Process Processing

1.1 I/O Process Processing (SAM document)

A target that can accept more than one I/O Process is capable of queueing. The number of I/O Processes that may exist is dependent on the design of the target.

If an initiator attempts to send more I/O Processes to a target than can be queued, they are rejected with a status of QUEUE FULL. I/O Processes that are queued may be Untagged or Tagged.

Several commands may be linked together to form a single I/O Process.

1.2 Untagged

Untagged queuing allows a target to accept one I/O process for a Logical Unit or target routine from each Initiator while there are other Pending I/O Processes. Only one I/O Process for each Initiator/Logical Unit combination shall be queued within a set at a time.

1.3 Untagged I/O Processes

An Untagged I/O Process is one which establishes an I_T_x nexus.

1.4 Tagged

An initiator may have more than one Tagged I/O Process for each Logical Unit under control of the target. The target has the freedom to optimize the sequence of I/O Process completion unless the initiator chooses to override the optimization.

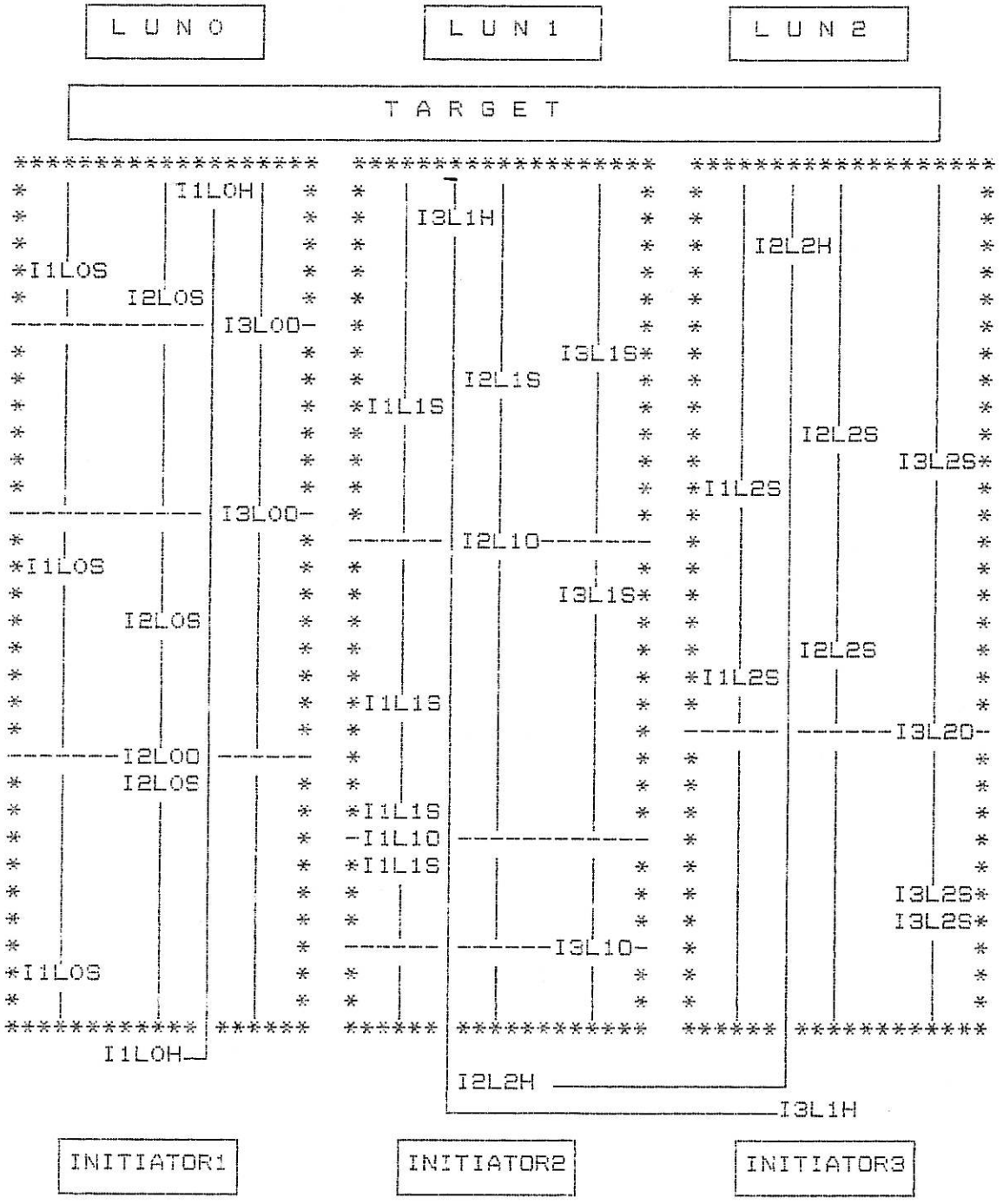
1.5 Tagged I/O Processes

Tagged I/O Processes are those which establish an I_T_L_Q nexus.

2. Ordering Boundaries

The ordering boundaries within a target shall be per logical unit. The target shall manage all the I/O Processes from all initiators on all ports received for each logical unit as a single set. Each set of I/O Processes shall be independent of all other sets.

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) (SAM document)

The Auto Contingent Allegiance Condition shall exist for a Logical Unit across all initiators following the return of a Check Condition or Command Terminated status. The Auto Contingent Allegiance Condition shall not cross set boundaries. The Auto Contingent Allegiance Condition shall be preserved for the I_T_x_y nexus until it is cleared.

When an I/O Process clears the Auto Contingent Allegiance the clearing I/O Process shall not be treated any differently than if it had been received in the absence of an Auto Contingent Allegiance.

3.1.2 Clearing Auto Contingent Allegiance Condition (ACA) (SIP document)

The Auto Contingent Allegiance Condition shall be cleared upon the receipt of:

- a power on condition,
- a hard reset,
- an Abort message for the I_T_L nexus,
- a Clear Queue message,
- a Release Recovery message for the I_T_x_y nexus,
- a Bus Device Reset message, or
- the receipt of any subsequent non-ACA I/O Processes for the Initiator/Logical Unit combination.

> Implementors Note: In order to maintain compatibility with SCSI-2, it is recommended that if the Auto Contingent Allegiance Condition is cleared by an untagged I/O Process, then that untagged I/O Process should be treated as if it was a head of queue I/O Process.

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

3.1.3 Response to Auto Contingent Allegiance Condition (SIP document)

While the Auto Contingent Allegiance exists the target shall respond to any other request for access to the logical unit from any device, other than the one which has the Auto Contingent Allegiance Condition, with an ACA Active status. The target shall not allow any existing Pending I/O Processes to become Current I/O Processes.

An initiator which has an outstanding Auto Contingent Allegiance Condition may allow other devices to use the target by sending a Limited Release message for the affected I_T_x_y nexus. After the target receives the Limited Release message Pending I/O Processes

for unaffected I_T_x_y nexuses may become Current I/O Processes.

3.1.4 Clearing Auto Contingent Allegiance Condition (ACA) (SBP document)

The Auto Contingent Allegiance Condition shall be cleared upon the receipt of:

-a power on condition,
-a hard reset,
-an Abort message for the I_T_L nexus,
-a Clear Queue message,
-a Release Recovery message for the I_T_x_y nexus,
-a Bus Device Reset message.

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

3.1.5 Response to Auto Contingent Allegiance Condition (SIP document)

While the Auto Contingent Allegiance exists the target shall only allow one ACA I/O Process to become Pending I/O Process. The target shall respond to any other request for access to the logical unit from any device including the one which has the Auto Contingent Allegiance Condition, with an ACA Active status. The target shall not allow any existing Pending I/O Processes to become Current I/O Processes.

An initiator which has an outstanding Auto Contingent Allegiance Condition may allow other devices to use the target by sending a Limited Release message for the affected I_T_x_y nexus. After the target receives the Limited Release message Pending I/O Processes for the released I_T_x_y nexuses may become Current I/O Processes.

3.2 Exception Checking

If an I/O Process is received with an duplicate tag the target shall create an Auto Contingent Allegiance with a key of Illegal Request, an ASC of Tagged Overlapped Commands, and an ASCQ which contains the value of the illegal tag. (Note: This requires a new ASC; 4Dh is recommended.)

The target shall abort all I/O Processes within the set which received the duplicate nexus. The target shall only create one Auto Contingent Allegiance within the set the duplicate nexus error occurred regardless of how many I/O Processes are aborted.

4. Current I/O Process

4.1 Current I/O Process (SAM document)

A Current I/O Process is an I/O Process which has information which is being sent or received on a physical transport system.

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

- A device on a physical transport system may have a Current I/O Process which is not the same Current I/O Process as another device on the same physical transport system.

- A device may simultaneously send information for one or more I/O Processes and receive information for one or more different I/O Processes.

4.2 Current I/O Process (SIP document)

A Current I/O Process begins when arbitration is won and ends when the SCSI bus next goes to the BUS FREE phase.

4.3 Current I/O Process (SBP document)

A Current I/O Process begins at the start of an information packet transfer and ends at the end of an information packet transfer. At any given time there may be multiple packets within the physical transport system.

5. Pending I/O Process

An I/O Process which is not a Current I/O Process, but was, and is not yet complete.

6. Simple I/O Process (Simple Tag)

A Simple I/O Process is one that is tagged with a Simple Queue Tag. If only Simple I/O Processes are Pending I/O Processes in the Target for the I_T_L nexus then any Simple I/O Process may become a Current I/O Process at any time.

7. Ordered I/O Process (Ordered Tag)

When an I/O Process is tagged as an Ordered I/O Process the target shall not:

-Write media information for the Ordered I/O Process,
-Read media information for the Ordered I/O Process,
-Perform any non-volatile action related to the I/O Process, or
-Complete the Ordered I/O Process

until all I/O Processes received before an Ordered I/O Process have completed for the I_T_L nexus.

If Simple I/O Processes are received after an Ordered I/O Process the target shall not:

-Write media information for any of the new Simple I/O Processes,
-Read media information for any of the new Simple I/O Processes, or
-Complete any of the new Simple I/O Processes

before the Ordered I/O Process completes.

8. Head Of Queue I/O Process (Head of Queue Tag)

When an I/O Process is tagged as a Head Of Queue I/O Process any I/O Processes, except a Head Of Queue I/O Process, received after

a Head Of Queue I/O Process shall not:

-Write media information for any of the new I/O Processes,
-Read media information for any of the new I/O Processes, or
-Complete any of the new I/O Processes

before the Head Of Queue I/O Process completes.

Any I/O Process which was received before a Head Of Queue I/O Process may complete before the Head Of Queue I/O Process completes.

9. ACA I/O Process (ACA Queue Tag) <<Need a better name -GOP>>

When an I/O Process is tagged as an ACA I/O Process, the target shall reject the ACA Queue Tag unless an Auto Contingent Allegiance Condition exists for the I_T_L nexus.

Note: In a SIP environment, after the ACA Queue Tag is rejected the Target shall go to a Bus Free phase. In a Serial environment the packet containing the ACA Queue Tag information shall be rejected.

If an Auto Contingent Allegiance Condition exists for the I_T_L nexus the ACA I/O Process shall complete before any other I/O Processes for that I_T_L nexus.

If another ACA I/O Process becomes the Current I/O Process while an ACA I/O Process is pending for that Initiator/Logical Unit combination the Target shall indicate an incorrect initiator connection condition occurred.

<<GOP - In SIP an incorrect initiator connection is indicated by a BUS FREE phase. I do not know how it is indicated in SBP.>>

10. Limited Release Message

The Limited Release message is sent from an initiator to a target during an Auto Contingent Allegiance Condition to allow other devices to use the target without clearing the Auto Contingent Alliance for the affected I_T_x_y nexus.

After the target receives the Limited Release message Pending I/O Processes for unaffected I_T_x_y nexuses may become Current I/O Processes.

Appendix A - Ordering Boundaries

A.1. Ordering Boundary Disclaimer

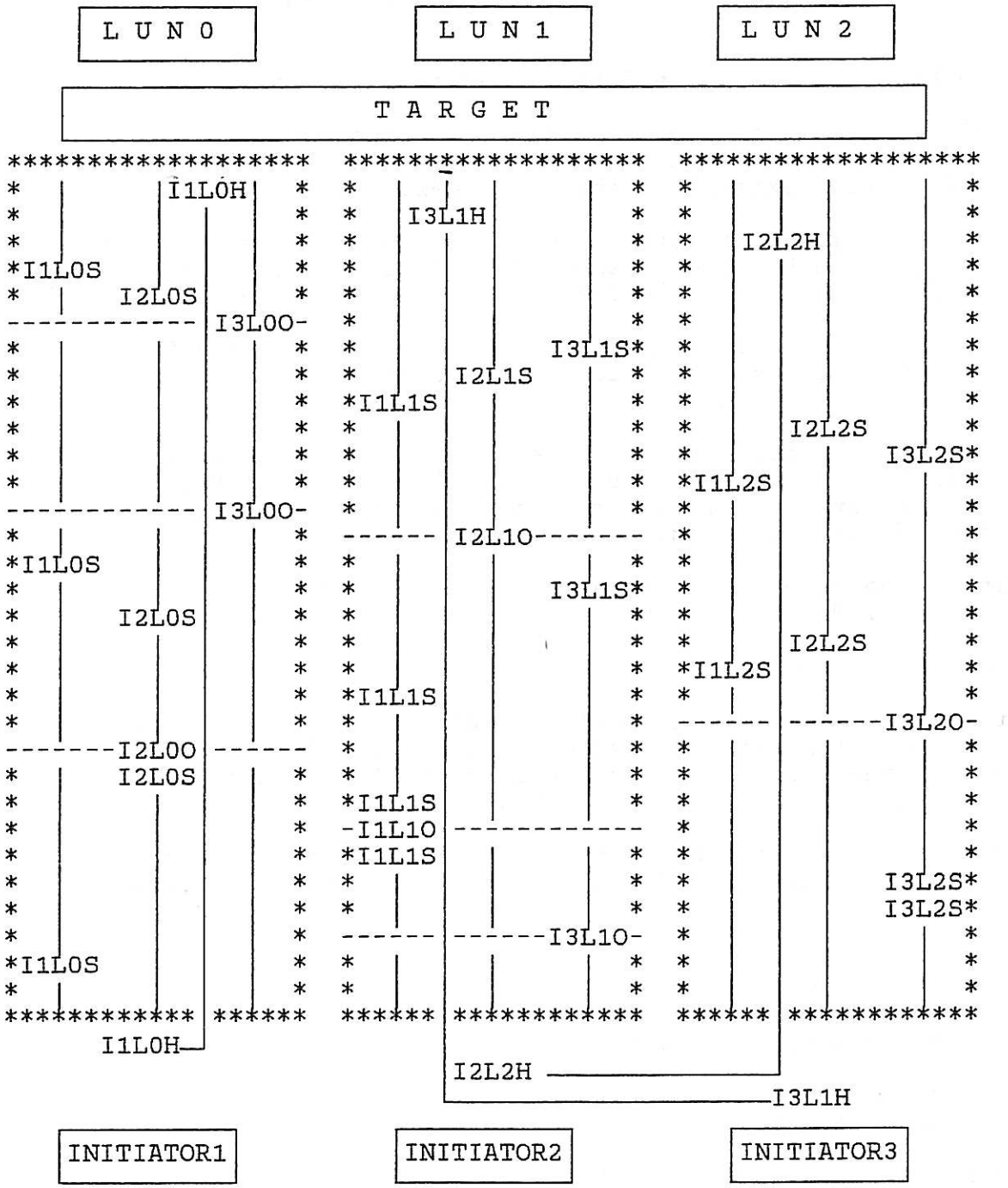
This appendix recognizes that there are other ways to order boundaries than the one chosen by the Standards Committee.

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

A.2 Per Logical Unit Ordering Boundaries

The ordering boundaries within a target shall be per logical unit. The target shall manage all the I/O Processes from all initiators on all ports received for each logical unit as a single set. Each set of I/O Processes shall be independent of all other sets.

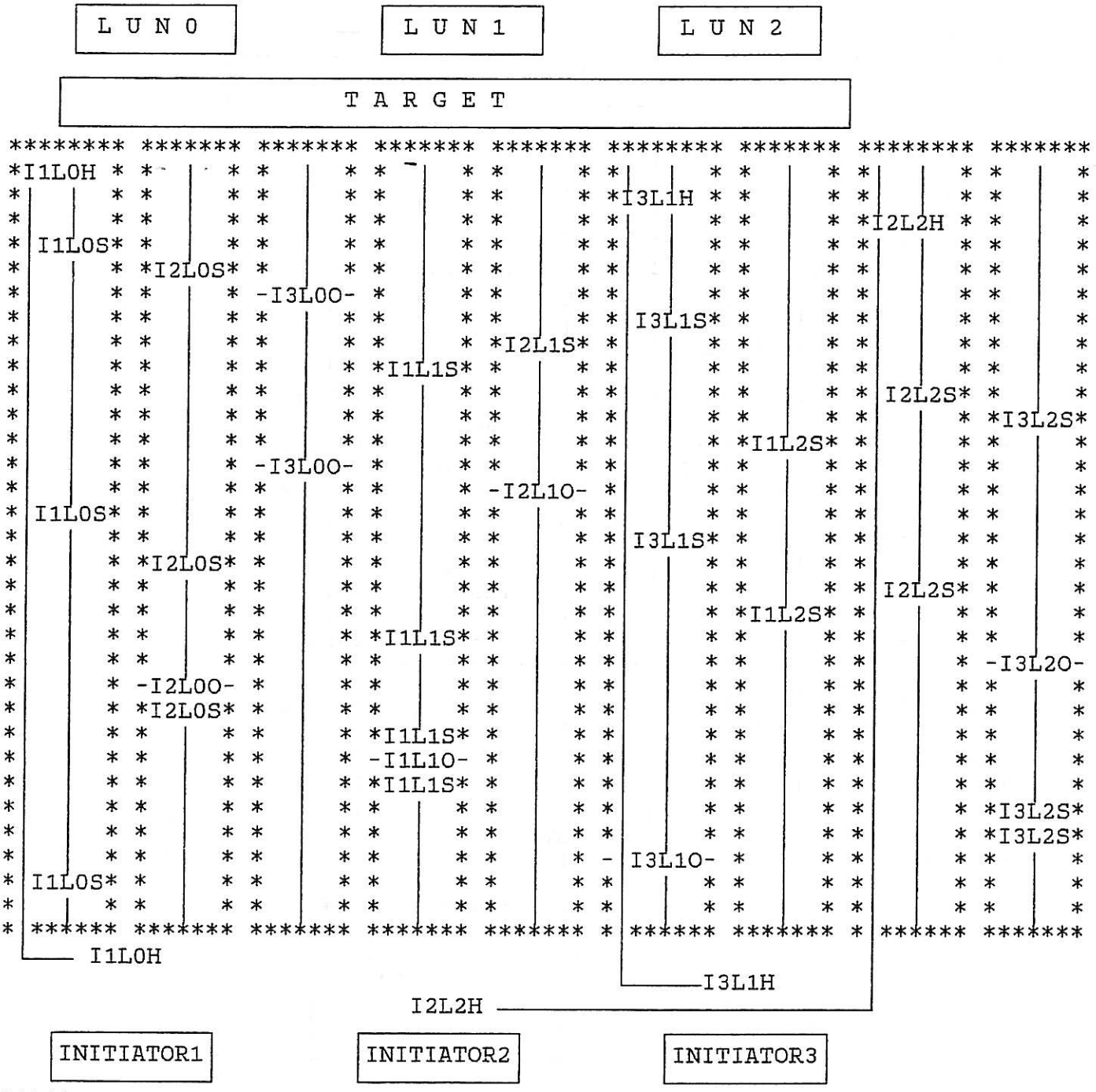
Table 1 - Per logical unit ordering boundaries



A.3 Per Initiator/Logical Unit Ordering Boundaries

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

Table 2 - Per initiator/logical unit ordering boundaries



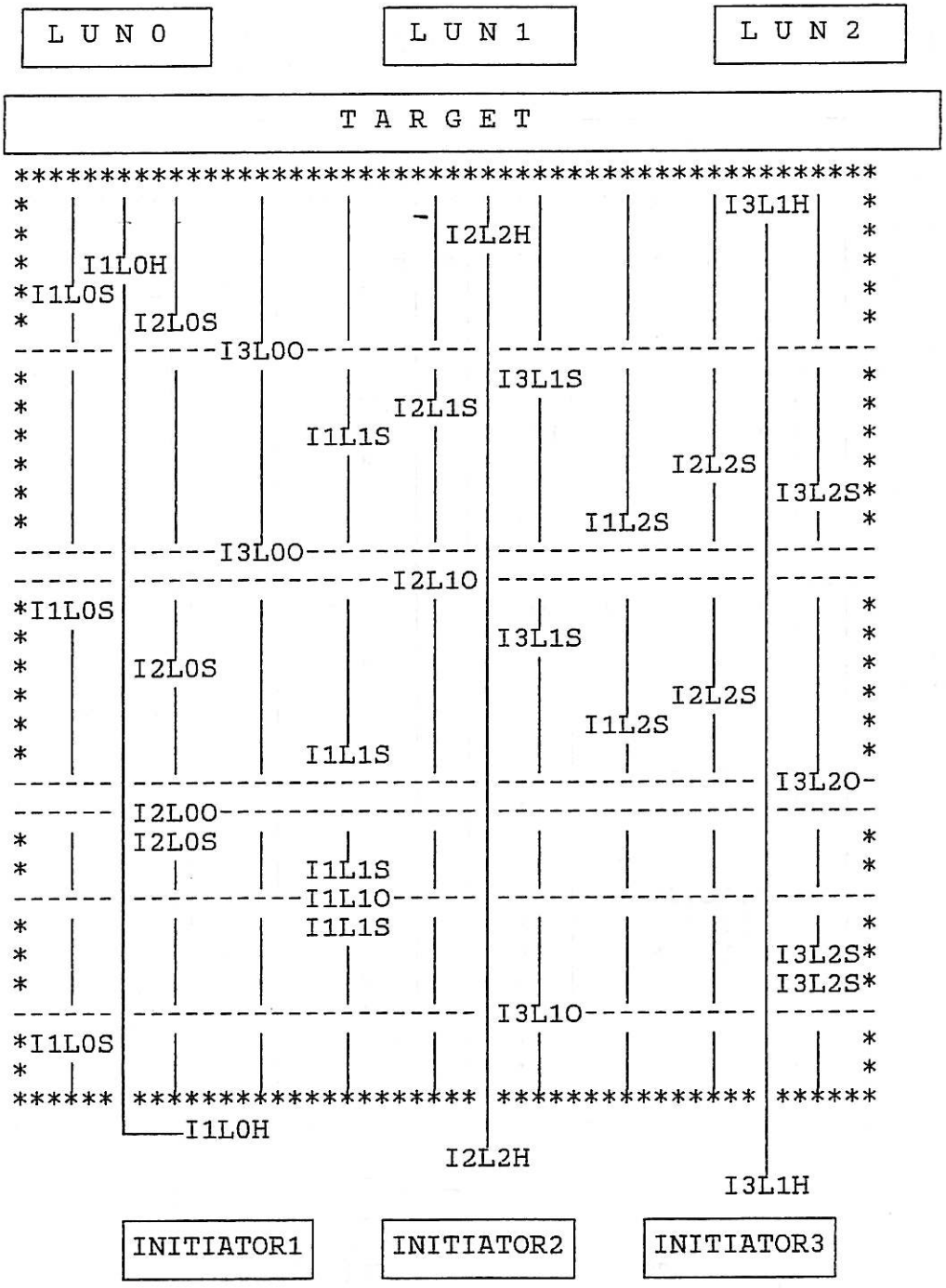
***** - Set Boundary

----- - Ordered Blocking Boundary

A.4 Per Target Ordering Boundaries

When the ordering boundary is per target the target manages all the I/O Processes from all initiators on all ports as a single set.

Table 3 - Per target ordering boundaries



*****
* * - Set Boundary
*****
----- - Ordered Blocking Boundary

A.5 Per Initiator Ordering Boundaries

The target manages all the I/O Processes from each initiator, on all ports, as a single set. Each set of I/O Processes shall be independent of all other sets.

Table 4 - Per initiator ordering boundaries