ENDL
TEXAS

Date: 8 January 2003
To: T10 Technical Committee
From: Ralph O. Weber
Subject: Remove SPI from SAM-3 and SPC-3

In keeping with the plan that SAM-2 be the last SCSI Architecture Model that includes features specific to the parallel SCSI bus, this proposal specifies the removal of the following from SAM-3:

- AutoSense is made mandatory and assumed to be the principal mechanism of sense data delivery (changes 1 to 19).
- Contingent Allegiance is removed completely (changes 20 to 31 and change 10). N.B. Contingent Allegiance is removed completely by this proposal and nothing is presented to replace it.

In preparing the changes in this proposal it was noted that introduction of required autosense support renders the REQUEST SENSE command sufficiently useless that one has to wonder whether its implementation needs to remain mandatory for all devices (see change 13).

Most changes affect only SAM-3, but changes 13 to 19 and change 30 affect only SPC-3. Both SAM-3 and SPC-3 are affected by changes 1 to 2 and changes 20 to 21.

Revision History

r0 Initial proposal
r1 Revise deferred errors wording that requires use of a REQUEST SENSE command following a CHECK CONDITION status (see change 15). Since there is no longer the concept of a 'solicited' REQUEST SENSE command, modify the description of MRIE code 6h (see change 18). Fix 8.6.3 (ASCII Information VPD page) text that treats REQUEST SENSE as the only source of sense data (see change 19).

Specific Changes

All proposed changes reference SAM-3 r04 and SPC-3 r10.

18484 Preston Road, Suite 102 PMB 178, Dallas, TX 75252 214-912-1373 972-596-2775 Fx ENDL_TX@computer.org

Change 1 [autosense in SAM-3 and SPC-3]: In SAM-3, replace the current definition of 'sense data':

> **3.1.111 sense data:** Data returned to an application client as a result of an autosense operation or REQUEST SENSE command (see 5.9.4). Fields in the sense data are referenced by name in this standard. See SPC-2 for a complete sense data format definition.

with:

> **3.1.111 sense data:** Data returned to an application client concurrently with a CHECK CONDITION status ~~as a result of an autosense operation or REQUEST SENSE command~~ (see 5.9.4). Fields in the sense data are referenced by name in this standard. See SPC-2 for a complete sense data format definition. Sense data may also be retrieved using the REQUEST SENSE command.

In SPC-3, remove the definition of autosense in glossary entry 3.1.14 and replace the current definition of 'sense data':

> **3.1.88 sense data:** Data describing an error or exceptional condition that a device server delivers to an application client as a result of an autosense operation (see 3.1.14), asynchronous event report (see 3.1.12), or REQUEST SENSE command (see 7.24). The format of sense data is the format defined for parameter data returned by the REQUEST SENSE command in 7.24.2.

with:

> **3.1.88 sense data:** Data describing an error or exceptional condition that a device server delivers to an application client concurrently with a CHECK CONDITIONS status, or as a result of an ~~autosense operation (see 3.1.14),~~ asynchronous event report (see 3.1.12), or REQUEST SENSE command (see 7.24). The format of sense data is the format defined for parameter data returned by the REQUEST SENSE command in 7.24.2.

Note that asynchronous event reporting also needs to be removed from the above, but that is a matter for a different proposal.

Change 2 [autosense in SAM-3 and SPC-3]: In SAM-3 Clause 5, remove all instances of the **Autosense Request** remote procedure call parameter, specifically:

- Remove it from the **Execute Command** procedure call and Input Arguments list in 5.1;
- Remove it from the **Send SCSI Command** procedure call and Input Arguments list in 5.4.2;
- Remove it from the **SCSI Command Received** procedure call and Input Arguments list in 5.4.2;

Other SAM-3 autosense related changes in 5.4.2 are described in change 6.

In SPC-3 subclause 4.2, make the following changes:

> Service response = Execute Command (IN (I_T_L_x Nexus, CDB, [Task Attribute], [Data-In Buffer Size], [Data-Out Buffer], [Data-Out Buffer Size], ~~[Autosense Request],~~ [Command Reference Number]), OUT ([Data-In Buffer], [Sense Data], Status))
>
> SAM-3 ~~SAM-2~~ defines all of the inputs and outputs in the procedure call above. As they may apply to any SCSI device, this standard defines the contents of the following procedure inputs and outputs; CDB, Data-Out Buffer, Data-In Buffer, and ~~Autosense~~ Sense Data. This standard does not define all possible instances of these procedure inputs and outputs. This standard defines only those instances that may apply to any SCSI device or to processor type SCSI devices. Instances of the procedure inputs and outputs that apply to specific SCSI device models are defined in the applicable SCSI command standards (see 3.1.19).

Change 3 [autosense in SAM-3]: Since sense data arrives concurrently with any CHECK CONDITION status, modify the third sentence in the description of the **Data-In Buffer** parameter in 5.1 as follows:

> While some valid data may be present for other values of status, the application client should rely on ~~obtain~~ additional information from the logical unit, such as sense data, to determine the state of the buffer contents.

Change 4 [autosense in SAM-3]: Since the sense data buffer always contains the sense data after a CHECK CONDITION status, modify the description of the **Sense Data** parameter in 5.1 as follows:

> A buffer to contain sense data returned concurrently with a CHECK CONDITION status (see 5.9.4) ~~by means of the autosense mechanism (see 5.9.4.2)~~.

Change 5 [autosense in SAM-3]: Since a CHECK CONDITION status always indicates that sense data has been delivered, replace the description of CHECK CONDITION status in 5.3.1:

> **CHECK CONDITION.** This status indicates that an CA or ACA condition has occurred (see 5.9.1). Autosense data may be delivered (see 5.9.4.2).

with:

> **CHECK CONDITION.** This status indicates that sense data has been delivered in the buffer defined by the **Sense Data** argument to the **Execute Command** procedure call (see 5.9.4). Additional actions that are required when CHECK CONDITION status is returned are described in 5.9.1.

Change 6 [autosense in SAM-3]: Because returned sense data is no longer called autosense data, change the descriptions of the **Sense Data** parameter in 5.4.2 as follows:

- **Send Command Complete** procedure call — **Sense Data:** If present, this argument instructs the SCSI target port to return sense information to the SCSI initiator port automatically (see 5.9.4 ~~5.9.4.2~~).
- **Command Complete Received** procedure call — **Sense Data:** Sense data returned concurrently with a CHECK CONDITION status (see 5.9.4). ~~Autosense data (see 5.9.4.2).~~

Other autosense related changes in 5.4.2 are described in change 2.

Change 7 [autosense in SAM-3]: Since the only sense data of concern in the architecture is autosense data, modify the definition of background operations in 5.5 as follows:

> Background operations may generate deferred errors that are reported ~~with autosense or~~ in sense data for a subsequent completed command (see SPC-3).

Change 8 [autosense in SAM-3]: Subclause 5.9.3 (Incorrect logical unit selection) describes the behavior of the REQUEST SENSE command in three locations. Since sense data is always autosense data else where in the revised SAM-3, these descriptions need to be modified to clarify that autosense data is **not** being discussed in these particular instances. Adding statements explaining that the REQUEST SENSE command completes with GOOD status will help differentiate sense data returned concurrently with a CHECK CONDITION status and sense data returned as REQUEST SENSE parameter data. Similar changes are described for 5.9.5 in change 11.

In 5.9.3, modify list entry b) as follows:

> In response to a REQUEST SENSE command, the SCSI target device shall return GOOD status and parameter data that contains sense data. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to LOGICAL UNIT NOT SUPPORTED.

In 5.9.3, modify list entry c) as follows:

In response to REQUEST SENSE, the SCSI target device shall return GOOD status and parameter data that contains sense data appropriate to the condition that is making the logical unit not operational.

In 5.9.3, modify list entry d) as follows:

In response to a REQUEST SENSE command, the SCSI target device shall return ~~the REQUEST SENSE~~ GOOD status and parameter data that contains sense data with a sense key of NO SENSE.

Change 9 [autosense in SAM-3]: Rewrite subclause 5.9.4 (Sense Data) as follows:

5.9.4 Sense data

~~**5.9.4.1 Sense data introduction**~~

Sense data shall be made available by the logical unit in the event a command completes with a CHECK CONDITION status or other conditions. The format, content and conditions under which sense data shall be prepared by the logical unit are specified in this standard, SPC-3 ~~SPC-2~~, the applicable command set standard and applicable SCSI transport protocol standard.

Sense data shall be preserved by the logical unit for the SCSI initiator port until:

a) It is transferred;
~~b) Another task from the SCSI initiator port for which the sense data is being preserved is entered into the task set;~~
c) A logical unit reset (see 6.3.3) occurs; or
d) An I_T nexus loss (see 6.3.4) involving the SCSI initiator port occurs.

~~The sense data may be transferred to the SCSI initiator device through any of the following methods:~~

~~a) The REQUEST SENSE command (see SPC-2); or~~
~~b) Autosense delivery (see 5.9.4.2).~~

~~**5.9.4.2 Autosense**~~

~~Autosense is the automatic return of sense data to the application client coincident with the completion of a SCSI command under the conditions described in this subclause. All SCSI transport protocols shall support autosense.~~

~~If supported by the SCSI transport protocol and logical unit and requested by the **Execute Command** remote procedure call (see 5.1), the device server shall only return sense data in this manner coincident with the completion of a command with a status of CHECK CONDITION. After autosense data is sent the following shall be cleared:~~

~~a) The CA condition (see 5.9.1.6), if any; and~~
~~b) The sense data, except sense data associated with an unit attention condition when the UA_INTLCK_CTRL field in the Control mode page (see SPC-3) contains 10b or 11b.~~

When a command completes with a CHECK CONDITION status, sense data shall be returned concurrently with the CHECK CONDITION status. After the sense data is returned, it shall be cleared except when it associated with an unit attention condition and the UA_INTLCK_CTRL field in the Control mode page (see SPC-3) contains 10b or 11b.

~~Autosense shall not affect ACA (see 5.9.1) or the sense data associated with an unit attention condition when the UA_INTLCK_CTRL field contains 10b or 11b.~~

The return of sense data concurrently with a CHECK CONDITION status shall not affect ACA (see 5.9.1) or the sense data associated with an unit attention condition when the UA_INTLCK_CTRL field contains 10b or 11b.

~~SCSI transport protocol standards that support autosense shall require an autosense implementation to:~~

~~a) Notify the logical unit when autosense data has been requested for a command; and~~
~~b) Inform the application client when autosense data has been returned upon command completion (see 5.1).~~

~~It is not an error for the application client to request the automatic return of sense data when autosense is not supported by the SCSI transport protocol or logical unit implementation. If the application client requested the return of sense data through the autosense facility and the SCSI transport protocol layer does not support this feature, then the confirmation returned by the SCSI initiator port should indicate that no sense data was returned. If the SCSI transport protocol layer supports autosense but the logical unit does not, then the SCSI target device should indicate that no sense data was returned. In either case, sense information shall be preserved and the application client may issue a command to retrieve it.~~

Change 10 [autosense & remove CA in SAM-3]: Subclause 5.9.5 (Unit attention condition) describes the behavior of the REQUEST SENSE command. Since sense data is always autosense data else where in the revised SAM-3, this description needs to be modified to clarify that autosense data is **not** being discussed in this particular instance. Adding a statement explaining that the REQUEST SENSE command completes with GOOD status will help differentiate sense data returned concurrently with a CHECK CONDITION status and sense data returned as REQUEST SENSE parameter data. These changes are similar to those described in change 8.

If a REQUEST SENSE command enters the enabled task state while an unit attention condition exists for the SCSI initiator port that sent the REQUEST SENSE command, then the logical unit shall return GOOD status and either:

a) Report any pending sense data as parameter data and preserve all unit attention conditions on the logical unit; or,
b) Report an unit attention condition for the SCSI initiator port that sent the REQUEST SENSE command as parameter data. The logical unit may discard any pending sense data and shall clear the reported unit attention condition for that SCSI initiator port.

The paragraph following those shown above includes a reference to CA that should be removed as follows:

If the logical unit has already generated the ~~CA or~~ ACA condition for an unit attention condition, the logical unit shall report the unit attention condition (i.e., option b) above).

Change 11 [autosense in SAM-3]: In 5.9.5 (Unit attention condition), the description of unit attention conditions that are reported using CHECK CONDITION status needs several changes, as follows:

If a command other than INQUIRY, REPORT LUNS, or REQUEST SENSE enters the enabled task state while an unit attention condition exists for the SCSI initiator port that sent the command, the logical unit shall terminate the command with a CHECK CONDITION status. The logical unit shall provide sense data that reports an unit attention condition for the SCSI initiator port that sent the command.

If a logical unit reports an unit attention condition with a CHECK CONDITION status ~~autosense (see 5.9.4.2)~~ and the UA_INTLCK_CTRL field in the Control mode page contains 00b (see SPC-3), then the logical unit shall clear the reported unit attention condition for that SCSI initiator port on the logical unit. If the UA_INTLCK_CTRL field in the Control mode page contains 10b or 11b, the logical unit shall not clear unit attention conditions reported a CHECK CONDITION status ~~with autosense~~.

Change 12 [autosense in SAM-3]: Since sense data is most commonly delivered concurrently with a CHECK CONDITION status, the discussion of 'retrieving sense data' in 6.2 (Establishing a unit attention condition subsequent to detection of an event) should be modified as follows:

A logical unit may use the I_T NEXUS LOSS OCCURRED additional sense code when establishing a unit attention condition if:

a) The SCSI initiator port being used to deliver ~~retrieving~~ the sense data is the SCSI initiator port that was involved in the I_T nexus loss, and the logical unit has maintained all state information specific to that SCSI initiator port since the I_T nexus loss; and
b) The I_T nexus being used to deliver ~~retrieve~~ the sense data is the same I_T nexus that was lost, and the logical unit has maintained all state information specific to that I_T nexus since the I_T nexus loss.

Change 13 [autosense in SPC-3]: Because sense data is always delivered concurrently with the CHECK CONDITION status, replace the SPC-3 subclause describing its use as a mandatory for all devices command as follows:

5.2.3 Using the REQUEST SENSE command

~~Whenever a command completes with a CHECK CONDITION status and autosense data is not provided, the application client that received the error status should issue a REQUEST SENSE command to receive the sense data describing the cause of the condition. If the application client issues a command other than REQUEST SENSE, the sense data is lost.~~

The REQUEST SENSE command may be used by an application client to poll the status of some background operations and to clear interlocked unit attention conditions (see 8.4.6).

It is difficult to see why support for the REQUEST SENSE command needs to be mandatory except for historical reasons and the somewhat esoteric reasons listed above.

Change 14 [autosense in SPC-3]: Revise the first paragraph of 7.24.2.1 (Sense data introduction) as follows:

Sense data shall be returned concurrently with a CHECK CONDITION status and may be returned by the REQUEST SENSE command~~, with the SCSI status in autosense (see SAM-2),~~ or with AER (see SAM-2). The REQUEST SENSE command may be used to request either the fixed format sense data or the descriptor format sense data. ~~Autosense~~ Sense data returned concurrently with a CHECK CONDITION status and AER return either fixed or descriptor format sense data format based on the value of the D_SENSE bit in the Control mode page (see 8.4.6).

Note that AER also needs to be removed from the above, but that is a matter for a different proposal.

Change 15 [autosense in SPC-3]: Revise the third paragraph of 7.24.4 (Deferred errors) as follows:

If AER is not supported, the deferred error may be indicated by returning CHECK CONDITION status to an application client on the appropriate initiator as described later in this subclause. ~~A subsequent REQUEST SENSE command shall return the deferred error sense information.~~

Note that AER also needs to be removed from the above, but that is a matter for a different proposal.

Change 16 [autosense in SPC-3]: Revise the description of the D_SENSE bit in 8.4.6 (Control mode page) as follows:

A descriptor format sense data (D_SENSE) bit of zero indicates that the device server shall return the fixed format sense data (see 7.24.2.3) concurrently with a CHECK CONDITION status ~~during autosense (see SAM-2)~~ and AER (see SAM-2). A D_SENSE bit of one indicates that the device server shall return descriptor format sense data (see 7.24.2.2) concurrently with a CHECK CONDITION status ~~during autosense~~ and AER.

Note that AER also needs to be removed from the above, but that is a matter for a different proposal.

Change 17 [autosense in SPC-3]: Revise the description of the UA_INTLCK_CTRL field in 8.4.6 (Control mode page) as follows:

The unit attention interlocks control (UA_INTLCK_CTRL) field (see table 225) controls the clearing of unit attention conditions reported concurrently with a CHECK CONDITION status ~~with autosense~~ or via asynchronous event reporting and whether returning a status of BUSY, TASK SET FULL or RESERVATION CONFLICT results in the establishment of a unit attention condition (see SAM-3 ~~SAM-2~~).

Table 225 — Unit attention interlocks control (UA_INTLCK_CTRL) field

Value	Definition
00b	The logical unit shall clear any unit attention condition reported concurrently with a CHECK CONDITION status ~~with autosense~~ or via asynchronous event reporting and shall not establish a unit attention condition when a task is terminated with BUSY, TASK SET FULL, or RESERVATION CONFLICT status.
01b	Reserved
10b	The logical unit shall not clear any unit attention condition reported concurrently with a CHECK CONDITION status ~~with autosense~~ or via asynchronous event reporting and shall not establish a unit attention condition when a task is terminated with BUSY, TASK SET FULL, or RESERVATION CONFLICT status.
11b	The logical unit shall not clear any unit attention condition reported concurrently with a CHECK CONDITION status ~~with autosense~~ or via asynchronous event reporting and shall establish a unit attention condition when a task is terminated with BUSY, TASK SET FULL, or RESERVATION CONFLICT status. Depending on the status, the device server shall set the additional sense code to PREVIOUS BUSY STATUS, PREVIOUS TASK SET FULL STATUS, or PREVIOUS RESERVATION CONFLICT STATUS. Until it is cleared by a REQUEST SENSE command, a unit attention condition shall be established only once for a BUSY, TASK SET FULL, or RESERVATION CONFLICT status regardless to the number of commands terminated with one of those status values.
NOTE 1 Even when UA_INTLCK_CTRL equals 10b or 11b, a REQUEST SENSE command still clears any unit attention condition that it reports.	

Note that AER also needs to be removed from the above, but that is a matter for a different proposal.

Change 18 [autosense in SPC-3]: In Table 232 (Method of reporting informational exceptions (MRIE) field), modify code value 6h as follows:

Only report informational exception condition on request: This method instructs the device server to preserve the informational exception(s) information. To find out about information exception conditions the application client polls the device server by issuing ~~an unsolicited~~ a REQUEST SENSE command. The sense key shall be set to NO SENSE and the additional sense code shall indicate the cause of the informational exception condition.

Change 19 [autosense in SPC-3]: Subclause 8.6.3 (ASCII Information VPD page) treats REQUEST SENSE as the only source of sense data.

Revise the first sentence as follows:

> The ASCII Information VPD page (see table 267) contains information for the field replaceable unit code returned in the ~~REQUEST SENSE~~ sense data (see 7.24.2).

Revise second paragraph after Table 267 (ASCII Information VPD page) as follows:

> The PAGE CODE field contains the same value as in the PAGE OR OPERATION CODE field of the INQUIRY CDB (see 7.4) and is associated with the FIELD REPLACEABLE UNIT CODE field returned in the sense data. ~~by the REQUEST SENSE command.~~

Change 20 [remove CA in SAM-3 and SPC-3]: In SAM-3, revise the definition of ACA from:

> **3.1.4 auto contingent allegiance (ACA):** One of the possible conditions of a task set following the return of a CHECK CONDITION status. See 5.9.1.

to:

> **3.1.4 auto contingent allegiance (ACA):** The task set condition established following the return of a CHECK CONDITION status when the NACA bit is set to one in the CONTROL byte. See 5.9.1.

In SPC-3, revise ACA definition from:

> **3.1.13 auto contingent allegiance (ACA):** One of the conditions of a task set following the return of a CHECK CONDITION status. A detailed definition of ACA may be found in SAM-2.

to:

> **3.1.13 auto contingent allegiance (ACA):** The task set condition established following the return of a CHECK CONDITION status when the NACA bit is set to one in the CONTROL byte. A detailed definition of ACA may be found in SAM-3 ~~SAM-2~~.

Change 21 [remove CA in SAM-3 and SPC-3]: In SAM-3, remove the contingent allegiance (CA) glossary entry, 3.1.9, and acronym definition in 3.2. Remove all uses of the CA acronym in phrases such as 'CA or ACA', 'CA and ACA', etc. The following subclauses and figures containing such references:

- 3.1.6 (blocked task state)
- 3.1.132 (task set)
- 4.8 (Logical units)
- 6.3.3 (Logical unit reset)
- 6.3.4 (I_T nexus loss)
- 7.2 (ABORT TASK)
- 7.5 (CLEAR TASK SET)
- 8.4.3 (Blocked task state)
- Figure 35 (Task states) [three times]
- 8.7.1 Task set management examples introduction [twice]

In SPC-3, remove the contingent allegiance (CA) glossary entry 3.1.21, and acronym definition in 3.2. Remove all uses of the CA acronym in the following subclauses and tables:

- 3.1.104 (task set)
- 5.5.2.7.5 (Preempting and aborting) [3 times]
- 7.24.1 (REQUEST SENSE command introduction)

Change 22 [remove CA in SAM-3]: The terms 'faulted initiator port', 'faulted task set', 'faulting command', and 'faulting task' now apply only when ACA is used. Reword their definitions as follows:

3.1.33 faulted initiator port: The SCSI initiator port to which a CHECK CONDITION status was returned that resulted in the establishment of an ACA. The faulted initiator port condition is cleared when the ~~CA or~~ ACA condition ~~resulting from the CHECK CONDITION status~~ is cleared.

3.1.34 faulted task set: A task set that contains a faulting task. The faulted task set condition is cleared when the ~~CA or~~ ACA condition resulting from the CHECK CONDITION status is cleared.

3.1.35 faulting command: A command that completed with a status of CHECK CONDITION that resulted in the establishment of an ACA.

3.1.36 faulting task: A task that has completed with a status of CHECK CONDITION that resulted in the establishment of an ACA.

Change 23 [remove CA in SAM-3]: In 5.2.3, the definition of the naca bit must be substantially rewritten to remove discussion of CA. Change from:

The NACA (Normal ACA) bit is used to select whether a contingent allegiance (CA) or an auto contingent allegiance (ACA) is established if the command returns with CHECK CONDITION status. An NACA bit set to one specifies that an ACA shall be established. An NACA bit set to zero specifies that a CA shall be established. The actions for CA and ACA are specified in 5.9.1.2. All logical units shall implement support for the NACA value of zero (i.e., CA) and may support the NACA value of one (i.e., ACA). The ability to support a NACA value of one is indicated with the NORMACA bit in the standard INQUIRY data (see SPC-2).

If the NACA bit is set to one but the logical unit does not support ACA, the logical unit shall complete the command with a CHECK CONDITION status, sense key of ILLEGAL REQUEST, an additional sense code of INVALID FIELD IN CDB and establish a CA condition. The requirements for handling the resulting CA condition shall be as described in 5.9.1.

to:

The NACA (Normal ACA) bit specifies ~~is used to select~~ whether ~~a contingent allegiance (CA) or~~ an auto contingent allegiance (ACA) is established if the command returns with CHECK CONDITION status. An NACA bit set to one specifies that an ACA shall be established. An NACA bit set to zero specifies that an ACA shall not ~~a CA shall~~ be established. The actions for ~~CA and~~ ACA are specified in 5.9.2 ~~5.9.1.2~~. Actions that may be required when an ACA is not established are described in 5.9.1. All logical units shall implement support for the NACA value of zero ~~(i.e., CA)~~ and may support the NACA value of one ~~(i.e., ACA)~~. The ability to support a NACA value of one is indicated with the NORMACA bit in the standard INQUIRY data (see SPC-3 ~~SPC-2~~).

If the NACA bit is set to one but the logical unit does not support ACA, the logical unit shall complete the command with a CHECK CONDITION status, sense key of ILLEGAL REQUEST, an additional sense code of INVALID FIELD IN CDB~~. and establish a CA condition. The requirements for handling the resulting CA condition shall be as described in 5.9.1.~~

Change 24 [remove CA in SAM-3]: In 5.7.1 (Mechanisms that cause tasks to be aborted), change the description of CA/ACA events that cause tasks to be aborted from:

The following actions affect only the task(s) created via the SCSI initiator port that transmits the action:

a) …;
b) …;
c) An CA or ACA condition was established (see 5.9.1.2) and the QERR field was set to 01b or 11b in the Control mode page (see SPC-2); or

to:

The following actions affect only the task(s) created via the SCSI initiator port that transmits the action:

a) …;
b) …;
c) A command completes with a CHECK CONDITION status without establishing an ACA condition (see 5.9.1.3) ~~An CA or ACA condition was established (see 5.9.1.2)~~ and the QERR field was set to 01b or 11b in the Control mode page (see SPC-3 ~~SPC-2~~); or
d) An ~~CA or~~ ACA condition was established (see 5.9.2.2 ~~5.9.1.2~~) and the QERR field was set to 01b or 11b in the Control mode page ~~(see SPC-2)~~; or

Also change from:

The following actions affect the task(s) created via the SCSI initiator port that transmits the action and/or task(s) created via other SCSI initiator ports:

a) …;
b) An CA or ACA condition was established (see 5.9.1.2) and the QERR field was set to 01b in the Control mode page (see SPC-2);

to:

The following actions affect the task(s) created via the SCSI initiator port that transmits the action and/or task(s) created via other SCSI initiator ports:

a) …;
b) A command completes with a CHECK CONDITION status without establishing an ACA condition (see 5.9.1.3) ~~An CA or ACA condition was established (see 5.9.1.2)~~ and the QERR field was set to 01b in the Control mode page (see SPC-3 ~~SPC-2~~);
c) An ~~CA or~~ ACA condition was established (see 5.9.2.2 ~~5.9.1.2~~) and the QERR field was set to 01b in the Control mode page ~~(see SPC-2)~~;

Change 25 [remove CA in SAM-3]: In 7.3 (ABORT TASK SET), remove discussion of CA by deleting the following sentence:

~~A CA shall be cleared by the ABORT TASK SET function from the faulted initiator port (see 5.9.1.6).~~

Change 26 [remove CA in SAM-3]: In 7.4 (CLEAR ACA), remove discussion of CA by deleting the following paragraph:

~~While a CA is in effect (see 5.9.1), a logical unit that supports the CLEAR ACA task management function shall ignore all CLEAR ACA requests and shall return a service response of FUNCTION COMPLETE.~~

Change 27 [remove CA in SAM-3]: Remove discussion of CA in 8.2 (Controlling task set management) as follows:

The basic task management model requires the following task set management behaviors:

a) …;
b) …;
c) All the tasks shall be aborted when a CHECK CONDITION status is returned or an ~~CA or~~ ACA condition is established;

Change 28 [remove CA in SAM-3]: Remove discussion of CA in 8.3 (Task management events) as follows:

The following describe the events that cause changes in task state.

…

~~CA or~~ ACA establishment:	A ~~CA or~~ ACA condition has been established (see 5.9.1).
task abort:	A task has been aborted as described in 5.7.
task completion:	…
task ended:	…
~~CA cleared:~~	~~An CA condition has been cleared (see 5.9.1.6).~~

Note: No additional discussion is necessary regarding tasks aborted due to the completion of a task with a CHECK CONDITION status because the topic is covered in 5.7, specifically in 5.7.1 (see change 24).

Change 29 [remove CA in SAM-3]: Remove discussion of CA in the state transition specifications in 8.6 (Task state transitions) by changing 'CA or ACA' to 'ACA' in the following state transition descriptions:

- S1:S2 [twice]
- S2:S3
- S3:S2 [twice]

Note: No additional discussion is necessary regarding tasks aborted due to the completion of a task with a CHECK CONDITION status because the topic is covered in 5.7, specifically in 5.7.1 (see change 24) and the S2:S4 description references 5.7 as follows:

Transition S2:S4: A task that has completed (see 8.3) or aborted (see 8.3 and 5.7) shall enter the ended task state. This is the only state transition that applies to an ACA task.

Change 30 [remove CA in SPC-3]: The SPC-3 definition of QERR field contains assumptions about CHECK CONDITION status always resulting in blocked tasks. That is no longer true with the removal of CA. Modify the QERR field description as follows:

The queue error management (QERR) field (see table 224) specifies how the device server shall handle ~~blocked and dormant~~ other tasks ~~(see SAM-2)~~ when ~~another~~ one task receives a CHECK CONDITION status ~~(see table 224)~~ (see SAM-3). The task set type (see the TST field definition above in this subclause) defines which other tasks are affected ~~blocked or dormant~~. If TST field equals 000b, then all tasks from all initiators are affected ~~blocked or dormant~~. If TST field equals 001b, then only tasks from the initiator that receives the CHECK CONDITION status are affected ~~blocked or dormant~~.

Table 224 — Queue error management (QERR) field

Value	Definition
00b	~~Blocked and dormant~~ Affected tasks in the task set shall resume after an ACA ~~or CA~~ condition is cleared (see SAM-3 ~~SAM-2~~).
01b	All the affected ~~blocked and dormant~~ tasks in the task set shall be aborted when the CHECK CONDITION status is sent. If the TAS bit is zero, a unit attention condition (see SAM-3 ~~SAM-2~~) shall be generated for each initiator that had ~~blocked and dormant~~ tasks aborted except for the initiator to which the CHECK CONDITION status was sent. The device server shall set the additional sense code to COMMANDS CLEARED BY ANOTHER INITIATOR. If the TAS bit is one, all affected tasks ~~blocked or dormant~~ for initiators other than the initiator for which the CHECK CONDITION status was sent shall be completed with a TASK ABORTED status and no unit attention shall be generated.
10b	Reserved
11b	~~Blocked and dormant~~ Affected tasks in the task set belonging to the initiator to which a CHECK CONDITION status is sent shall be aborted when the status is sent.

Change 31 [remove CA in SAM-3]: Because CA is being removed completely, it is necessary to restructure 5.9.1 as two subclauses so that the requirements associated with returning a CHECK CONDITION status are clearly separated from the ACA requirements. This restructuring includes moving and copying some out of the middle of the current 5.9.1 and this moved or copied text appears in green.

Rewrite 5.9.1 as follows:

5.9.1 Commands that complete with CHECK CONDITION status

5.9.1.1 Overview

When a command completes with a CHECK CONDITION status, the application client may request that device server alter command processing by establishing an ACA condition. The application client requests establishment of an ACA condition by setting NACA bit to one in the CONTROL byte of the CDB that specified the command. The ACA condition is described in 5.9.2. Requirements that apply when the ACA condition is not in effect are described in 5.9.1.2.

When the NACA bit is set to zero in the CONTROL byte of the CDB specifying a command that returns in a CHECK CONDITION status, the device server shall not establish an ACA condition. However, tasks other than the task for the command returning the CHECK CONDITION status may be aborted as described in 5.9.1.3.

5.9.1.2 Handling tasks when ~~neither CA or~~ ACA is not in effect

Table x225 describes the handling of tasks when ~~neither a CA nor~~ an ACA condition is not in effect for the task set. The number of SCSI initiator ports in the task set is influenced by the TST field in the Control mode page (see SPC-3 ~~SPC-2~~).

Table x225 — Task handling when ~~neither CA nor~~ ACA is not in effect

New Task Properties		Device Server Action	~~Condition~~ ACA Established if New Task Terminates with a CHECK CONDITION status
Attribute [a]	NACA Value [b]		
Any Attribute Except ACA	0	Process the task. [c]	~~CA~~ No
	1		~~ACA~~ Yes
ACA	0	Terminate the command with CHECK CONDITION status, sense key of ILLEGAL REQUEST and additional sense code of INVALID MESSAGE ERROR.	~~CA~~ No
	1		~~ACA~~ Yes

[a] Task attributes are described in 8.5.
[b] The NACA bit is in the CONTROL byte in the CDB (see 5.2.3).
[c] All the conditions that affect the processing of commands (e.g., reservations) still apply.

5.9.1.3 Abort other tasks when CHECK CONDITION status is returned without an ACA

When a CHECK CONDITION status is returned for a command where the NACA bit is set to zero in the command's CDB CONTROL byte (i.e. when an ACA condition is not established), tasks in the dormant or enabled task state (see 8.4) ~~shall~~ may be ~~blocked or~~ aborted based on the contents of the TST and QERR field in the Control mode page (see SPC-3) as shown in table x226. The TST (task set type) Control mode page field specifies the type of task set in the logical unit (see SPC-3). The QERR (queue error management) Control mode page field specifies how the device server handles blocked and dormant tasks when another task receives a CHECK CONDITION status (see SPC-3).

Table x226 — Aborting tasks when a an ACA is not established

QERR	TST	Action
00b	000b	Tasks other than the task returning CHECK CONDITION status shall not be aborted.
	001b	
01b	000b	All enabled and dormant tasks from all SCSI initiator ports shall be aborted (see 5.7).
	001b	All enabled and dormant tasks from the faulted initiator port shall be aborted (see 5.7). All tasks from SCSI initiator ports other than the faulted initiator port shall not be aborted. ~~affected by the establishment of this CA or ACA condition.~~
~~11b~~	~~000b~~	~~All enabled and dormant tasks from the faulted initiator port shall be aborted (see 5.7). All enabled tasks from SCSI initiator ports other than the faulted initiator port shall transition to the blocked task state (see 8.6). All dormant tasks from SCSI initiator ports other than the faulted initiator port shall remain in the dormant task state.~~
	~~001b~~	~~All enabled and dormant tasks from the faulted initiator port shall be aborted (see 5.7). All tasks from SCSI initiator ports other than the faulted initiator port shall not be affected by the establishment of this CA or ACA condition.~~
11b	000b	All enabled and dormant tasks from the faulted initiator port shall be aborted (see 5.7). All tasks from SCSI initiator ports other than the faulted initiator port shall not be aborted.
	001b	

5.9.2 ~~Contingent allegiance (CA) and auto~~ Auto contingent allegiance (ACA)

5.9.2.1 Overview

When a command is terminated with a CHECK CONDITION status, the application client may request that device server alter command processing by establishing an ACA condition. When the NACA bit in the CONTROL byte of the CDB that specified the command is set to one, the device server shall establish an ACA condition as described in 5.9.2.2. Upon establishment of the ACA condition, some tasks other than the task returning the CHECK CONDITION status may be aborted and continued processing of other tasks may be blocked as described in 5.9.2.2.

While the ACA condition is in effect, new tasks received by the logical unit from the faulted initiator port are not allowed to enter the task set unless they have the ACA task attribute (see 8.5.4). One of the results of the ACA task attribute requirement is that commands in-flight when the CHECK CONDITION status occurs are returned unprocessed to the SCSI initiator port with an ACA ACTIVE status. Multiple commands may be sent one at a time using the ACA task attribute to recover from the CHECK CONDITION that caused the ACA condition without clearing the ACA.

While the ACA condition is in effect:

a) New tasks from the faulted initiator port shall be handled as described in 5.9.1.4, and
b) New tasks from SCSI initiator ports other than the faulted initiator port shall be handled as described in 5.9.1.5.

The methods for clearing an ACA condition are described in 5.9.2.7.

~~There are two mechanisms for returning sense data when a command is terminated with a CHECK CONDITION status: autosense (see 5.9.4.2) and the REQUEST SENSE command (see SPC-2). There are two mechanisms for altering task processing when a command is terminated with a CHECK CONDITION status: CA and ACA. CA alters task processing so that sense data is preserved for subsequent delivery. ACA alters task processing until a CLEAR ACA task management function (see 7.4) is requested. Table 25 provides an overview of how autosense, CA, and ACA interact.~~

~~Table 25 — Autosense, CA, and ACA Interactions~~

~~Autosense Requested~~ [a]	**~~NACA Value~~** [b]	**~~Tasks Blocked~~** [c]	
		~~From~~	**~~To~~** [d]
~~No~~	~~0 (i.e., CA)~~	~~Termination of a command with CHECK CONDITION status~~	~~Receipt of a command~~ [e]
	~~1 (i.e., ACA)~~		~~Receipt of CLEAR ACA~~ [g]
~~Yes~~	~~0 (i.e., CA)~~		~~Transmission of autosense data~~ [f]
	~~1 (i.e., ACA)~~		~~Receipt of CLEAR ACA~~ [g]

~~a Autosense is requested via the **Execute Command** remote procedure call (see 5.1).~~

~~b The NACA bit is in the CONTROL byte in the CDB (see 5.2.3).~~

~~c The blocking of tasks is described in 5.9.1.2. If the QERR field in the Control mode page (see SPC-2) contains 01b or 11b, tasks are aborted instead of being blocked. If the TST field in the Control mode page contains 000b, tasks from all SCSI initiator ports are blocked or aborted. If the TST field in the Control mode page contains 001b, only tasks from the faulted initiator port are blocked or aborted.~~

~~d This table covers only the normal methods for clearing a CA or ACA as seen by the faulted initiator port. Exception handling methods for clearing CA and ACA are described in 5.9.1.6 and 5.9.1.7.~~

~~e The intent is that the next command from the faulted initiator port be a REQUEST SENSE command but the next command received clears the CA condition, regardless of what command that is.~~

~~f Since the autosense data is transmitted coincident with the delivery of the CHECK CONDITION status (see 5.9.4.2), the interval during which tasks are blocked is not detectable by the application client. If the QERR field in the Control mode page (see SPC-2) contains 01b or 11b, the specified blocked tasks are aborted, an action that makes the CA condition detectable by the application client.~~

~~g The CLEAR ACA task management function is described in 7.4. During ACA new tasks received by the logical unit are not allowed to enter the task set unless they have the ACA task attribute (see 8.5.4) and are from the faulted initiator port. One of the results of the ACA task attribute requirement is that commands in flight when the CHECK CONDITION status occurs are returned unprocessed to the SCSI initiator port with an ACA ACTIVE status. Multiple commands may be sent one at a time using the ACA task attribute to recover from the CHECK CONDITION that caused the ACA condition without clearing the ACA.~~

5.9.2.2 Establishing ~~a CA or~~ an ACA

When a device server terminates a command with a CHECK CONDITION status~~, either an CA or ACA condition is established within the task set. If~~ and the NACA bit ~~was zero in the CONTROL byte (see 5.2.3) of the faulting command, the device server shall create a CA condition. If the NACA bit~~ was set to one in the CONTROL byte of the faulting command, the device server shall create an ACA condition.

When a CA or ACA condition is established, tasks in the dormant or enabled task state (see 8.4) shall either be aborted or blocked based on the contents of the TST and QERR field in the Control mode page (see SPC-2) as shown in table 26. The TST (task set type) Control mode page field specifies the type of task set in the logical unit (see SPC-2). The QERR (queue error management) Control mode page field specifies how the device server handles blocked and dormant tasks when another task receives a CHECK CONDITION status (see SPC-2).

Table 26 — Blocking and aborting tasks when a CA or ACA is established

QERR	TST	Action
00b	000b	All enabled tasks from all SCSI initiator ports shall transition to the blocked task state (see 8.6). All dormant tasks from all SCSI initiator ports shall remain in the dormant task state.
	001b	All enabled tasks from the faulted initiator port shall transition to the blocked task state (see 8.6). All dormant tasks from the faulted initiator port shall remain in the dormant task state. All tasks from SCSI initiator ports other than the faulted initiator port shall not be affected by the establishment of this CA or ACA condition.
01b	000b	All enabled and dormant tasks from all SCSI initiator ports shall be aborted (see 5.7).
	001b	All enabled and dormant tasks from the faulted initiator port shall be aborted (see 5.7). All tasks from SCSI initiator ports other than the faulted initiator port shall not be affected by the establishment of this CA or ACA condition.
11b	000b	All enabled and dormant tasks from the faulted initiator port shall be aborted (see 5.7). All enabled tasks from SCSI initiator ports other than the faulted initiator port shall transition to the blocked task state (see 8.6). All dormant tasks from SCSI initiator ports other than the faulted initiator port shall remain in the dormant task state.
	001b	All enabled and dormant tasks from the faulted initiator port shall be aborted (see 5.7). All tasks from SCSI initiator ports other than the faulted initiator port shall not be affected by the establishment of this CA or ACA condition.

~~After the CA or ACA condition is established:~~

~~a) New tasks from the faulted initiator port shall be handled as described in 5.9.1.4, and~~
~~b) New tasks from SCSI initiator ports other than the faulted initiator port shall be handled as described in 5.9.1.5.~~

~~A CA or~~ An ACA condition shall not cross task set boundaries and shall be preserved until it is cleared as described in ~~5.9.1.6 or~~ 5.9.1.7. ~~If requested by the application client and supported by the SCSI transport protocol and logical unit, sense data shall be returned via autosense as described in 5.9.4.2.~~

If the SCSI transport protocol does not enforce state synchronization as described in 4.6.2, there may be a time delay between the occurrence of the ~~CA or~~ ACA condition and the time at which the application client becomes aware of the condition.

~~5.9.2.3 Handling tasks when neither CA or ACA is in effect~~

~~Table 27 describes the handling of tasks when neither a CA nor an ACA condition is in effect for the task set. The number of SCSI initiator ports in the task set is influenced by the TST field in the Control mode page (see SPC-2).~~

~~Table 27 — Task handling when neither CA nor ACA is in effect~~

New Task Properties		Device Server Action	Condition Established if New Task Terminates with a CHECK CONDITION status
Attribute [a]	NACA Value [b]		
Any Attribute Except ACA	0	Process the task. [c]	CA
	1		ACA
ACA	0	Terminate the command with CHECK CONDITION status, sense key of ILLEGAL REQUEST and additional sense code of INVALID MESSAGE ERROR.	CA
	1		ACA

[a] Task attributes are described in 8.5.
[b] The NACA bit is in the CONTROL byte in the CDB (see 5.2.3).
[c] All the conditions that affect the processing of commands (e.g., reservations) still apply.

5.9.2.4 Handling new tasks from the faulted initiator port when ~~CA or~~ ACA is in effect

~~Table 28 describes the handling of new tasks from the faulted initiator port when CA is in effect.~~

~~Table 28 — Handling for new tasks from a faulted initiator port during CA~~

New Task Properties		Device Server Action	Condition Established If New Task Terminates with a CHECK CONDITION status [e]
Attribute [a]	NACA Value [b]		
Any Attribute Except ACA	0	Process the task. [d]	CA
	1		ACA
ACA	0	Terminate the command with CHECK CONDITION status, sense key of ILLEGAL REQUEST, and additional sense code of INVALID MESSAGE ERROR.	CA
	1		ACA

[a] Task attributes are described in 8.5.
[b] The NACA bit is in the CONTROL byte in the CDB (see 5.2.3).
[e] The CA condition is cleared upon completion of any new task regardless of status. Termination of that new task with CHECK CONDITION status shall result in the establishment of a new CA or ACA based on the value of the NACA bit.
[d] All the conditions that affect the processing of commands (e.g., reservations) still apply.

Table 29 describes the handling of new tasks from the faulted initiator port when ACA is in effect.

Table 29 — Handling for new tasks from a faulted initiator port during ACA

New Task Properties		ACA Task Present in the Task Set	Device Server Action	~~Condition~~ ACA Established If New Task Terminates with a CHECK CONDITION status
Attribute [a]	NACA Value [b]			
ACA	0	No	Process the task. [d]	~~CA~~ No [c]
	1	No		~~ACA~~ Yes [c]
	0 or 1	Yes	Terminate the task with ACA ACTIVE status.	n/a
Any Attribute Except ACA	0 or 1	n/a	Terminate the task with ACA ACTIVE status.	n/a

[a] Task attributes are described in 8.5.

[b] The NACA bit is in the CONTROL byte in the CDB (see 5.2.3).

[c] If a task with the ACA attribute terminates with a CHECK CONDITION status, the existing ACA condition shall be cleared and the value of the NACA bit shall control the establishment of a new ACA condition. ~~a new CA or ACA condition shall be established based on the value of the NACA bit.~~

[d] All the conditions that affect the processing of commands (e.g., reservations) still apply.

5.9.2.5 Handling new tasks from non-faulted initiator ports when ~~CA or~~ ACA is in effect

5.9.2.5.1 Commands permitted from non-faulted initiator ports during ~~CA or~~ ACA

The device server shall process a PERSISTENT RESERVE OUT command with a PREEMPT AND ABORT service action (see SPC-3 ~~SPC-2~~) while ~~a CA or~~ an ACA condition is established when the command is received from a SCSI initiator port other than the faulted initiator port.

NOTE 1 - The processing of specific commands (e.g., PERSISTENT RESERVE OUT command with a PREEMPT AND ABORT service action) the from SCSI initiator ports other than the faulted initiator port while ~~a CA or~~ an ACA condition is in effect provides SCSI initiator ports other than the faulted initiator port the opportunity to recover from error conditions that the faulted initiator port cannot recover from itself.

5.9.2.5.2 Handling new tasks from non-faulted initiator ports when ~~CA or~~ ACA is in effect

The handling of tasks created by SCSI initiator ports other than the faulted initiator port depends on the value in the TST field in the Control mode page (see SPC-3 ~~SPC-2~~).

~~Table 30 describes the handling of new tasks from SCSI initiator ports other than the faulted initiator port when CA is in effect.~~

~~Table 30 — Handling for new tasks from non-faulted initiator ports during CA~~

~~tst Field Value in Control mode page~~	~~New Task Properties~~		~~New Command Permitted During CA [e]~~	~~Device Server Action~~	~~Condition Established If New Task Terminates with a CHECK CONDITION status~~
	~~Attribute [a]~~	~~NACA Value [b]~~			
~~000b~~	~~ACA~~	~~n/a~~	~~n/a~~	~~Terminate the task with BUSY status.~~	~~n/a~~
	~~Any Attribute Except ACA~~	~~0 or 1~~	~~No~~	~~Terminate the task with BUSY status.~~	~~n/a~~
		~~0~~	~~Yes~~	~~Process the task.~~	~~CA [d]~~
		~~1~~	~~Yes~~		~~ACA [d]~~
~~001b~~	~~ACA~~	~~0~~	~~n/a~~	~~Terminate the command with CHECK CONDITION status, sense key of ILLEGAL REQUEST and additional sense code of INVALID MESSAGE ERROR.~~	~~CA~~
		~~1~~			~~ACA~~
	~~Any Attribute Except ACA~~	~~0 or 1~~	~~n/a~~	~~Process the task. [e]~~	~~See 5.9.1.3.~~

~~[a] Task attributes are described in 8.5.~~

~~[b] The NACA bit is in the CONTROL byte in the CDB (see 5.2.3).~~

~~[c] See 5.9.1.5.1.~~

~~[d] If a permitted command terminates with a CHECK CONDITION status, the existing CA condition shall be cleared and a new CA or ACA condition shall be established for a new faulted initiator port based on the value of the NACA bit.~~

~~[e] When the TST field in the Control mode page contains 001b, commands from SCSI initiator ports other than the faulted initiator port shall be processed as if the CA condition does not exist (see 5.9.1.3). In this case, the logical unit shall be capable of handling concurrent CA conditions and sense data for all SCSI initiator ports.~~

Table 31 describes the handling of new tasks from SCSI initiator ports other than the faulted initiator port when ACA is in effect.

Table 31 — Handling for new tasks from non-faulted initiator ports during ACA

TST Field Value in Control mode page	New Task Properties		New Command Permitted During ACA [c]	Device Server Action	~~Condition~~ ACA Established If New Task Terminates with a CHECK CONDITION status
	Attribute [a]	NACA Value [b]			
000b	ACA	n/a	n/a	Terminate the task with ACA ACTIVE status.	n/a
	Any Attribute Except ACA	0	No	Terminate the task with BUSY status.	n/a
		1	No	Terminate the task with ACA ACTIVE status.	n/a
		0	Yes	Process the task.	~~CA~~ No [d]
		1	Yes		~~ACA~~ Yes [d]
001b	ACA	0	n/a	Terminate the command with CHECK CONDITION status, sense key of ILLEGAL REQUEST and additional sense code of INVALID MESSAGE ERROR.	~~CA~~ No
		1			~~ACA~~ Yes
	Any Attribute Except ACA	0 or 1	n/a	Process the task. [e]	See 5.9.1.3.

[a] Task attributes are described in 8.5.

[b] The NACA bit is in the CONTROL byte in the CDB (see 5.2.3).

[c] See 5.9.1.5.1.

[d] If a permitted command terminates with a CHECK CONDITION status, the existing ACA condition shall be cleared and the value of the NACA bit shall control the establishment of a new ACA condition. ~~a new CA or ACA condition shall be established for a new faulted initiator port based on the value of the NACA bit.~~

[e] When the TST field in the Control mode page contains 001b, commands from SCSI initiator ports other than the faulted initiator port shall be processed as if the ACA condition does not exist (see 5.9.1.3). In this case, the logical unit shall be capable of handling concurrent ACA conditions and sense data for all SCSI initiator ports.

~~**5.9.2.6 Clearing a CA condition**~~

~~A CA condition shall only be cleared:~~

~~a) As a result of a reset condition (see 6.3.2) or logical unit reset (see 6.3.3);~~
~~b) By an ABORT TASK SET task management function (see 7.3) from the faulted initiator port;~~
~~c) By a CLEAR TASK SET task management function (see 7.5) from any SCSI initiator port including the faulted initiator port if the TST field in the Control mode page (see SPC-2) contains 000b;~~

~~d) By a CLEAR TASK SET task management function from the faulted initiator port if the TST field in the Control mode page (see SPC-2) contains 001b;~~
~~e) By a PERSISTENT RESERVE OUT command with a PREEMPT AND ABORT service action from another SCSI initiator port that clears the tasks of the faulted initiator port (see SPC-2);~~
~~f) When a PERSISTENT RESERVE OUT command with a PREEMPT AND ABORT service action from another SCSI initiator port terminates in a CHECK CONDITION status;~~
~~g) Upon completion of a subsequent REQUEST SENSE command for the I_T_L nexus;~~
~~h) Upon accepting any subsequent command other than a REQUEST SENSE command for the I_T_L nexus; or~~
~~i) Upon sending sense data by means of the autosense mechanism (see 5.9.4.2).~~

~~Case f) results in the establishment of a new CA or ACA for a new faulted initiator port based on the value of the NACA bit.~~

~~When a CA condition is cleared and no new CA or ACA condition is established, the state of all tasks in the task set shall be modified as described in clause 8.~~

5.9.2.7 Clearing an ACA condition

An ACA condition shall only be cleared:

a) As a result of a reset condition (see 6.3.2) or logical unit reset (see 6.3.3);
b) By a CLEAR ACA task management function (see 7.4) from the faulted initiator port;
c) By a PERSISTENT RESERVE OUT command with a PREEMPT AND ABORT service action with the ACA task attribute from the faulted initiator port that clears the tasks of the faulted initiator port (see SPC-2);
d) By a PERSISTENT RESERVE OUT command with a PREEMPT AND ABORT service action with a task attribute other than ACA from a SCSI initiator port other than the faulted initiator port that clears the tasks of the faulted initiator port;
e) When a command with the ACA task attribute from the faulted initiator port terminates with a CHECK CONDITION status; or
f) When a PERSISTENT RESERVE OUT command with a PREEMPT AND ABORT service action terminates in a CHECK CONDITION status.

Cases e) and f) may result in the establishment of a new ~~CA or~~ ACA based on the value of the NACA bit.

When an ACA condition is cleared and no new ~~CA or~~ ACA condition is established, the state of all tasks in the task set shall be modified as described in clause 8.