To: T10 Technical Committee
From: Rob Elliott, Compaq Computer Corporation (Robert.Elliott@compaq.com)
Date: 16 April 2001
Subject: SAM-2, SPI-4, SBC-2 WAKEUP and reset cleanup

Revision History
Revision 0 (16 April 2001) first revision

Related Documents
spi4r03 – SCSI Parallel Interface-4 revision 3
sbc2r02 – SCSI Block Commands-2 revision 2
sip-r10 – SCSI Interlocked Protocol revision 10
01-128r0 - SPI-4 reset cleanup (Rob Elliott)
95-222r0 – Power condition mode page code (Ralph Weber)
91-014r6 – Power condition mode page?

Overview
SPI-4 and SBC-2 refer to two task management functions that SAM-2 does not describe – WAKEUP and RESET SERVICE DELIVERY SUBSYSTEM. No other protocol defines these. (FCP-2 briefly mentioned them until revision 5b).

SIP defined them as "non-message task management functions," distinct from normal task management functions:

> **5.3.1.4 Non-message task management functions**
> The task management functions are defined in the SCSI-3 Architecture Model Standard. This standard defines the services used by the SCSI-3 Interlocked Protocol to move the task management functions from the application client to the task manager. This standard does not define the binary values of the non-message task management functions.

The reason that WAKEUP exists is to enable ATA style power management in SCSI disk drives. ATA drives support these power management states:
- Active
- Idle - a little slower response time
- Standby - may take 30 sec to resume; spindle stopped
- Sleep - may take 30 sec to resume; spindle stopped and ATA interface inactive

States are selected by sending commands to the device (IDLE, STANDBY, and SLEEP). There is also a Standby Timer that causes the drive to go to Standby mode if it is idle for a certain period of time. Once a drive is in the Sleep state, only a bus reset or a register write to the soft reset register can wake it up. The register write can be detected via a simple decode of about 6 signals.

The SCSI block command set (SBC-2) defines the same power management states:
- Active
- Idle - a little slower response time
- Standby - media is stopped
- Sleep - media is stopped, SCSI interface is off (except for RST# receiver)

There are two ways to change states:
1. Run the START STOP UNIT command with the POWER CONDITION field indicating the requested state. This is equivalent to sending the ATA commands.

2. Use MODE SELECT to program the Power Conditions mode page 1Ah. It has four fields:
- Idle - enables the Idle Timer
- Standby - enables the Standby Timer

- Idle Timer
- Standby Timer

The device will go to Idle or Standby modes when inactive for the selected periods of time.

Once a SCSI device reaches the Sleep state, only a hard reset wakes it up. The only way another device can force a hard reset is by issuing a bus reset. A TARGET RESET message cannot be decoded; that requires too much of the parallel SCSI interface logic be powered (to track arbitration, selection, and message phases).

To clean up the standards, the suggested changes are:

- Remove references to RESET SERVICE DELIVERY SUBSYSTEM in SBC-2 and remove it from SPI-4. The existing "hard reset" references cover it in SBC-2.
- Alternative 1: WAKEUP does not fit well as a "task management function." SBC-2 does not say anywhere that it affects tasks, although the current SPI and ATA implementations using resets do have that effect. A protocol could be defined that had a separate wakeup signal which did not have any affect on tasks. Define a new class of "interconnect functions" in SAM-2 that covers WAKEUP and BUS RESET (a new name for RESET SERVICE DELIVERY SUBSYSTEM). Change wording in SBC-2 and SPI-4 to match.
- Alternative 2: Continue calling WAKEUP a task management function and document it in SAM-2. Mandate that it terminate all tasks like a hard reset. Forget about BUS RESET.
- Make a few reset cleanups throughout SBC-2 to parallel 01-128r0 (SPI-4 reset cleanup).
- Remove mode page 0Dh from SBC-2. SPC-2 has marked that page code obsolete since it documents mode page 1Ah as a replacement available for all device types.
- Remove redundant task management function descriptions in SAM-2 section 6.1. The summary descriptions had as many shalls as the subsequent per-function sections.

Other options:

- Obsolete the power condition mode page. This functionality does not have widespread use even in ATA – operating systems prefer to direct all power management transitions rather than have the devices change state themselves.
- Obsolete the power condition field in START STOP UNIT. This field is not well supported in SCSI. Idle mode can be handled automatically by the device, since it has no visible side effects. Stopping the unit with a plain START STOP UNIT command is essentially the same as requesting standby mode. Sleep mode yields little additional power savings over stopping the unit.

If both are obsoleted, then all sections referring to WAKEUP can be removed from SPI-4 and SBC-2 (4.2.1.5, parts of 5.1.19, table 97, 6.2.7, table 123) and SAM-2 doesn't need to mention it.

Both Alternatives: Suggested Changes to SBC-2

Remove section 6.2.7 power condition mode page.

Change the power condition page code in tables 97 and 123 from a self reference to SPC-2.

Update all the SPC references to SPC-2 (in those tables and elsewhere).

Alternative 1: Suggested Changes to SPI-4 revision 4

[01-128r1 suggests changes to these sections too. This proposal overrides those]

19.5.2 Task management functions

This standard handles task management functions as a four step confirmed service that provides the means to transfer task management functions to a task manager.

The task management functions are defined in the SCSI Architecture Model-2 standard. This standard defines the actions taken by the SCSI parallel interface service to carry out the requested task management functions.

…

~~**19.5.8 RESET SERVICE DELIVERY SUBSYSTEM**~~

~~The SCSI parallel interface services request the initiator issue a hard reset (see 12.3) to the selected SCSI device.~~

19.5.~~9~~ 8 TARGET RESET

The SCSI parallel interface services request the initiator issue a TARGET RESET message (see 16.5.7) to the selected SCSI device.

~~**19.5.10 WAKEUP**~~

~~The SCSI parallel interface services request the initiator issue a hard reset (see 12.3) to the selected SCSI device.~~

19.6 Interconnect services

19.6.1 Interconnect functions overview

The interconnect services shall be requested from the application client using a procedure call defined as:

Function name (IN (nexus), service response)

19.6.2 Interconnect functions

This standard handles interconnect functions as a four step confirmed service.

The interconnect functions are defined in the SCSI Architecture Model-2 standard. This standard defines the actions taken by the SCSI parallel interface service to carry out the requested interconnect functions.

19.6.3 BUS RESET

The SCSI parallel interface services request the initiator create a bus reset condition (see 16.5.7) on the bus containing the selected SCSI device.

19.6.4 WAKEUP

The SCSI parallel interface services request the initiator create a bus reset condition (see 16.5.7) on the bus containing the selected SCSI device.

Alternative 1: Suggested changes to SBC-2 Revision 2

3.1.1.x hard reset: A target action in response to a reset event in which the target port performs the operations described in SCSI Architecture Model-2.

3.1.1.x logical unit reset: A logical unit action in response to a logical unit reset event in which the logical unit performs the operations described in SCSI Architecture Model-2.

3.1.x logical unit reset event: An event that triggers a logical unit reset from a logical unit as described in SCSI Architecture Model–2.

3.1.1.x power cycle: Power off followed by power on.

3.1.1.x power on: Power being applied.

3.1.x reset event: An event that triggers a hard reset from a SCSI device as described in the protocol standard. Reset events include power on and other protocol-specific events.

4.2.1.1 Direct-access device type model overview

…

Blocks of data are stored by a process that causes localized changes or transitions within the medium. The changes made to the medium to store the blocks of data may be volatile (i.e., not retained through off/on power cycles) or non-volatile (i.e., retained through power off/on cycles). The medium may be divided in parts that are used for data blocks, parts that are reserved for defect management, and parts that are reserved for use by the controller for the management of the block device.

4.2.1.5 Power conditions

…

The lowest power consumption, with power applied, occurs in the Sleep condition. When in the Sleep condition a block device requires a WAKEUP ~~task management~~interconnect function or hard reset to be activated.

…

If implemented, the block device shall use the optional power condition page to control the power conditions after a ~~power on or a~~ WAKEUP ~~task management~~interconnect function or hard reset until a START STOP UNIT command is received with the POWER CONDITIONS field set to a value other than 0h or 7h. See 5.1.19 and 6.2.7.

…

Figure 2 shows the flow control between the different power conditions in a device that is setup to adjust itself automatically to the power condition that allows any command to execute.

…

f) Path 6: A WAKEUP ~~task management~~interconnect function or hard reset returns the device to the state defined by the saved power mode page parameters.

…

Figure 3 shows the flow control between the different power conditions in a device that is setup to only allow changing of the power condition by the application client. Any command received that requires more power than allowed by the most recent power condition setting shall be terminated with a sense key of ILLEGAL REQUEST and the additional sense code shall be set to LOW POWER CONDITION ACTIVE.

…

f) Path 6: A WAKEUP ~~task management~~ interconnect function or hard reset returns the device to the state defined by the saved power mode parameters.

…

4.2.1.6 Initialization

…

Block devices using a non-volatile medium may save the parameters and only need to be initialized once. However, some mode parameters may need to be initialized after each ~~power-on and/or~~logical unit reset. A catastrophic failure of the direct-access block device may require the FORMAT UNIT command to be reissued.

Block devices that use a volatile medium may need to be initialized at each ~~power-on and/or~~hard reset prior to the execution of read or write operations. Mode parameters may also need initialization.

4.2.3.7 XOR data retention requirements

The target shall retain XOR data while awaiting retrieval by an XDREAD command until performing one of the following events: a matching XDREAD command, ~~TARGET RESET, power cycle~~logical unit reset, CLEAR TASK SET, ABORT TASK if the task matches the pending XDREAD, ABORT TASK SET.

5.1.19 START STOP UNIT command

…

If the START STOP UNIT command is issued with the POWER CONDITIONS field set to 1h, 2h, or 3h the block device shall:

a) change power conditions only on receipt of another START STOP UNIT command or a ~~RESET task management function or RESET SERVICE DELIVERY SUBSYSTEM~~logical unit reset;

b) suspend any Power Condition timers (see 6.2.7) that are active on receipt of the START STOP UNIT command until another START STOP UNIT command is received that returns control of the power condition to the block device or a ~~RESET task management function or RESET SERVICE DELIVERY SUBSYSTEM~~logical unit reset occurs;

c) terminate any command received that requires more power than allowed by the START STOP UNIT command's most recent power condition setting with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST with the additional sense code set to LOW POWER CONDITION ACTIVE.

If the START STOP UNIT command is issued with the POWER CONDITION field set to 5h the server shall:
a) suspend any Power Condition timers that are active on receipt of the START STOP UNIT command until a WAKEUP ~~task management~~interconnect function is received by the device server or a hard reset;

b) not respond to a task requests until a WAKEUP ~~task management~~ interconnect function is received by the device server or a hard reset.

On receipt of a WAKEUP ~~task management~~ interconnect function or a hard reset any previously active power conditions timers shall be restored to those values indicated by the saved power condition mode page parameters. Before returning a function complete response the target shall place itself into a condition capable of receiving commands and task management functions and shall create a unit attention condition for all initiators. The sense key shall be set to UNIT ATTENTION and the additional sense code set to LOW POWER CONDITION ACTIVE.
…
In the Sleep condition the device server shall only respond to a WAKEUP ~~task management~~interconnect function or a hard reset. When a target has multiple logical units attached it shall enter the Sleep condition only after all the logical units have been placed into a Sleep condition.

Alternative 1: Suggested changes to SAM-2 revision 16
6 Task Management Functions

[Editor's note: remove the hanging paragraphs by adding an introduction section.]
6.1 Task Management Functions Introduction
Task management functions provide an initiator with a way to explicitly control the execution of one or more tasks. An application client invokes a task management function by means of a procedure call having the following format:

Service Response = Function name (IN (nexus))

Service Response:

One of the following protocol-specific responses shall be returned:

FUNCTION COMPLETE: Each SCSI protocol standard shall define the actual events comprising each of the above service responses. A task manager response indicating that the requested function is complete. The task manager shall unconditionally return this response upon completion of a task management request supported by the logical

unit or target device to which the request was directed. Upon receiving a request to execute an unsupported function, the task manager may return this response or the FUNCTION REJECTED response described below.

FUNCTION REJECTED: An optional task manager response indicating that the operation is not supported by the object to which the function was directed (e.g., the logical unit or target device).

SERVICE DELIVERY OR TARGET FAILURE: The request was terminated due to a service delivery failure or target malfunction. The target may or may not have successfully performed the specified function.

Each SCSI protocol standard shall define the actual events comprising each of the above service responses.

[Editor's note; remove the redundant "summary descriptions". The subsequent sections should fully specify each function. This section has some shalls that were not even mentioned in the subsequent sections.]

The task management functions are summarized ~~as follows~~ in Table xx ~~(see the clauses below for detailed definitions of each task management function):~~.

Table xx. Task Management Functions.

Task Management Function	Nexus
ABORT TASK	I_T_L_Q
ABORT TASK SET	I_T_L
CLEAR ACA	I_T_L
CLEAR TASK SET	I_T_L
LOGICAL UNIT RESET	I_T_L
TARGET RESET	I_T

~~**ABORT TASK (IN (I_T_L_Q Nexus))** - Abort the identified task. This function shall be supported if the logical unit supports tagged tasks and may be supported if the logical unit does not support tagged tasks.~~

~~**ABORT TASK SET (IN (I_T_L Nexus))** - Abort all tasks in the task set for the I_T_L nexus. This function shall be supported by all logical units.~~

~~**CLEAR ACA (IN (I_T_L Nexus))** - Clear auto contingent allegiance condition. This function shall be supported if the logical unit accepts a NACA bit value of one in the CDB CONTROL byte (see 5.1.2).~~

~~**CLEAR TASK SET (IN (I_T_L Nexus))** - Abort all tasks in the specified task set as described in 6.4. This function shall be supported by all logical units, except in the following cases, when support for this function is optional:~~
~~a) The logical unit does not support tagged tasks (see 4.9); or~~
~~b) The logical unit supports the basic task management model (see 7.2).~~

~~**LOGICAL UNIT RESET (IN (I_T_L Nexus))** - Perform a logical unit reset as described in 5.7.7 by aborting all tasks in the task set(s) and propagating the reset to all dependent logical units (see 3.1.22). Support for this function is mandatory.~~

~~**TARGET RESET (IN (I_T Nexus))** - Perform a logical unit reset as described in 5.7.7 for every logical unit.~~

Argument descriptions:

Nexus: A non-specific initiator-target nexus (see 4.10).

I_T Nexus: An initiator and target nexus (see 4.10).
I_T_L Nexus: An initiator, target, and logical unit nexus (see 4.10).
I_T_L_Q Nexus: An initiator, target, logical unit, and tag nexus (see 4.10).

NOTE 10 The LOGICAL UNIT RESET, TARGET RESET, CLEAR TASK SET, ABORT TASK and ABORT TASK SET functions provide a means to abort one or more tasks prior to normal completion.

All SCSI protocol standards shall provide the functionality needed for a task manager to implement all of the task management functions ~~defined above~~.
…

6.1 ABORT TASK
Function call:
Service Response = ABORT TASK (IN (I_T_L_Q Nexus))
Description:
This function shall be supported by a logical unit ~~that~~ if it supports tagged tasks and may be supported by a logical unit ~~that~~ if it does not support tagged tasks.

The task manager shall abort the specified task if it exists. Previously established conditions, including MODE SELECT parameters, reservations, and auto contingent allegiance shall not be changed by the ABORT TASK function.

If the logical unit supports this function, a response of FUNCTION COMPLETE shall indicate that the task was aborted or was not in the task set. In either case, the target shall guarantee that no further responses from the task are sent to the initiator.

6.2 ABORT TASK SET
Function Call:
Service Response = ABORT TASK SET (IN (I_T_L Nexus))
Description:
This function shall be supported by all logical units.

The task manager shall abort all tasks in the task set which were created by the initiator as described in 5.5.

The task manager shall perform an action equivalent to receiving a series of ABORT TASK requests. All tasks from that initiator in the task set serviced by the logical unit shall be aborted. Tasks from other initiators or in other task sets shall not be aborted. Previously established conditions, including MODE SELECT parameters, reservations, and auto contingent allegiance shall not be changed by the ABORT TASK SET function. A contingent allegiance (NACA=0) shall be cleared by the ABORT TASK SET function.

6.3 CLEAR ACA
Function Call
Service response = CLEAR ACA (IN (I_T_L Nexus))
Description:
This function shall ~~only be implemented~~ be supported by a logical unit that accepts a NACA bit value of one in the CDB CONTROL byte (see 5.1.2) and shall not be supported by a logical unit that does not accept a NACA bit value of one.

The initiator invokes CLEAR ACA to clear an auto contingent allegiance condition from the task set serviced by the logical unit according to the rules specified in 5.7.1.2. If successful, this function shall be terminated with a service response of FUNCTION COMPLETE.

If the task manager clears the auto contingent allegiance condition, any task within that task set may be completed subject to the rules for task set management specified in clause 7.

6.4 CLEAR TASK SET
Function Call:
Service response = CLEAR TASK SET (IN (I_T_L Nexus))
Description:
~~This function shall be supported by all logical units that support tagged tasks (see 4.9) and may be supported by logical units that do not support tagged tasks.~~ This function shall be supported by all logical units, except in the following cases, when support for this function is optional:
a) The logical unit does not support tagged tasks (see 4.9); or
b) The logical unit supports the basic task management model (see 7.2).

All tasks in the appropriate task set as defined by the TST field in the Control mode page (see SPC-2) shall be aborted as described in 5.5. The medium may have been altered by partially executed commands. All pending status and sense data for the appropriate task set shall be cleared.

Previously established conditions, including MODE SELECT parameters, reservations, and auto contingent allegiance (NACA=1, see 5.1.2) shall not be changed by the CLEAR TASK SET function. A contingent allegiance (naca=0) shall be cleared by the CLEAR TASK SET function.

6.5 LOGICAL UNIT RESET
Function Call:
Service Response = LOGICAL UNIT RESET (IN (I_T_L Nexus))
Description:
This function shall be supported by all logical units.

Before returning a FUNCTION COMPLETE response, the logical unit shall perform the logical unit reset functions specified in 5.7.7. A unit attention condition for all initiators that have access shall be created on the logical unit and dependent logical unit(s), if any, as specified in 5.7.5.
NOTE 11 Previous versions of this standard only required LOGICAL UNIT RESET support in logical units that supported hierarchical logical units.

6.6 TARGET RESET
Function Call:
Service Response = TARGET RESET (IN (I_T Nexus))
Description:
Before returning a FUNCTION COMPLETE response, the target port shall perform logical unit reset functions specified in 5.7.7 for every logical unit. A unit attention condition for all initiators that have access shall be created on each of these logical units as specified in 5.7.5.

An initiator should issue LOGICAL UNIT RESETs only to the logical units it is using rather than issuing a TARGET RESET. This avoids resetting logical units that other initiators may be using.

NOTE 12 Previous versions of this standard required TARGET RESET support in all targets. SCSI protocols may or may not require that TARGET RESET be supported. SCSI protocols may require additional actions beyond those specified here.

7 Interconnect Functions

7.1 Interconnect functions introduction

The protocol services described in this clause are used by an initiator and target to process an interconnect function.

This clause describes the protocol services that support interconnect functions. SCSI protocol specifications may define the protocol-specific requirements for implementing the service request, indication, response, and confirmation described in this clause.

Protocol Service Request sent by an initiator and application client to a target's interconnect manager:

Send Interconnect Request (IN (I_T Nexus, Function Identifier))

Input Arguments:

I_T Nexus: an optional initiator and target nexus (see 4.10).

Protocol Service Indication received by the task manager:

Interconnect Request Received (IN (Nexus, Function Identifier))

Protocol Service Response from task manager to initiator and application client:

Interconnect Function Executed (IN (Nexus, Service Response))

The **Service Response** parameter encodes a value representing one of the following:

FUNCTION REJECTED: The interconnect does not implement the requested function.

FUNCTION COMPLETE: The requested function has been completed.

Protocol Service Confirmation received by application client:

Received Function-Executed (IN (Nexus, Service Response))

7.2 BUS RESET

Request:

Service Response = Bus Reset (IN ([I_T Nexus]))

Description:

This function shall only by supported by initiator ports and target ports on protocols that support it.

This function shall only be supported by protocols whose interconnects support a shared reset signal or individual reset signals for each target port. If individual reset signals are supported, the I_T Nexus argument indicates which target port to reset.

This function causes a wakeup event to be sent to either:

a) the specified target port; or

b) all target ports connected to the interconnect.

7.3 WAKEUP

Request:

Service Response = Wakeup (IN ([I_T Nexus]))

Description:

This function shall only by supported by initiator ports and target ports on protocols that support it.

This function shall only be supported by protocols whose interconnects support a shared wakeup signal or individual wakeup signals for each target port. If individual wakeup signals are supported, the I_T Nexus argument indicates which target port to wakeup.

This function causes a wakeup event to be sent to either:

a) the specified target port; or

b) all target ports connected to the interconnect.

The wakeup function shall cause a hard reset in the recipient target port(s).

The wakeup function may be implemented with the same signals used to implement the BUS RESET function.

Note: SBC-2 defines a Sleep power state where the device powers off most of its protocol interface. The WAKEUP function is used to restore power to the protocol interface. It is detected using a minimal amount of hardware that remains powered on while the rest of the device is powered off. In parallel SCSI, the WAKEUP function is the bus reset condition, shared by all devices. In ATA/ATAPI, the WAKEUP function is either a bus reset shared by all devices on the bus, or a soft reset directed to one of the devices.

Alternative 2: Suggested Changes to SPI-4 revision 4

[01-128r1 suggests changes to these sections too. This proposal overrides those]

19.5.2 Task management functions

This standard handles task management functions as a four step confirmed service that provides the means to transfer task management functions to a task manager.

The task management functions are defined in the SCSI Architecture Model-2 standard. This standard defines the actions taken by the SCSI parallel interface service to carry out the requested task management functions.

…

~~**19.5.8 RESET SERVICE DELIVERY SUBSYSTEM**~~

~~The SCSI parallel interface services request the initiator issue a hard reset (see 12.3) to the selected SCSI device.~~

19.5.~~9~~ 8 TARGET RESET

The SCSI parallel interface services request the initiator issue a TARGET RESET message (see 16.5.7) to the selected SCSI device.

19.5.10 WAKEUP

The SCSI parallel interface services request the initiator ~~issue a hard reset (see 12.3)~~create a bus reset condition (see 16.5.7) ~~to the~~ on the bus containing the selected SCSI device.

Alternative 2: Suggested changes to SBC-2 Revision 2

3.1.1.x hard reset: A target action in response to a reset event in which the target port performs the operations described in SCSI Architecture Model-2.

3.1.1.x logical unit reset: A logical unit action in response to a logical unit reset event in which the logical unit performs the operations described in SCSI Architecture Model-2.

3.1.x logical unit reset event: An event that triggers a logical unit reset from a logical unit as described in SCSI Architecture Model–2.

3.1.1.x power cycle: Power off followed by power on.

3.1.1.x power on: Power being applied.

3.1.x reset event: An event that triggers a hard reset from a SCSI device as described in the protocol standard. Reset events include power on and other protocol-specific events.

4.2.1.1 Direct-access device type model overview

…

Blocks of data are stored by a process that causes localized changes or transitions within the medium. The changes made to the medium to store the blocks of data may be volatile (i.e., not retained through off/on power cycles) or non-volatile (i.e., retained through power

off/on cycles). The medium may be divided in parts that are used for data blocks, parts that are reserved for defect management, and parts that are reserved for use by the controller for the management of the block device.

4.2.1.5 Power conditions

…

The lowest power consumption, with power applied, occurs in the Sleep condition. When in the Sleep condition a block device requires a WAKEUP ~~-~~task management function or hard reset to be activated.

…

If implemented, the block device shall use the optional power condition page to control the power conditions after a ~~power on or a~~ WAKEUP task management function or hard reset until a START STOP UNIT command is received with the POWER CONDITIONS field set to a value other than 0h or 7h. See 5.1.19 and 6.2.7.

…

Figure 2 shows the flow control between the different power conditions in a device that is setup to adjust itself automatically to the power condition that allows any command to execute.

…

f) Path 6: A WAKEUP task management function or hard reset returns the device to the state defined by the saved power mode page parameters.

…

Figure 3 shows the flow control between the different power conditions in a device that is setup to only allow changing of the power condition by the application client. Any command received that requires more power than allowed by the most recent power condition setting shall be terminated with a sense key of ILLEGAL REQUEST and the additional sense code shall be set to LOW POWER CONDITION ACTIVE.

…

f) Path 6: A WAKEUP task management function or hard reset returns the device to the state defined by the saved power mode parameters.

…

4.2.1.6 Initialization

…

Block devices using a non-volatile medium may save the parameters and only need to be initialized once. However, some mode parameters may need to be initialized after each ~~power-on and/or~~logical unit reset. A catastrophic failure of the direct-access block device may require the FORMAT UNIT command to be reissued.

Block devices that use a volatile medium may need to be initialized at each ~~power-on and/or~~hard reset prior to the execution of read or write operations. Mode parameters may also need initialization.

4.2.3.7 XOR data retention requirements

The target shall retain XOR data while awaiting retrieval by an XDREAD command until performing one of the following events: a matching XDREAD command, ~~TARGET RESET, power cycle~~logical unit reset, CLEAR TASK SET, ABORT TASK if the task matches the pending XDREAD, ABORT TASK SET.

5.1.19 START STOP UNIT command

…

If the START STOP UNIT command is issued with the POWER CONDITIONS field set to 1h, 2h, or 3h the block device shall:

a) change power conditions only on receipt of another START STOP UNIT command or a ~~RESET task management function or RESET SERVICE DELIVERY SUBSYSTEM~~logical unit reset;

b) suspend any Power Condition timers (see 6.2.7) that are active on receipt of the START STOP UNIT command until another START STOP UNIT command is received that returns control of the power condition to the block device or a ~~RESET task management function or RESET SERVICE DELIVERY SUBSYSTEM~~logical unit reset occurs;

c) terminate any command received that requires more power than allowed by the START STOP UNIT command's most recent power condition setting with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST with the additional sense code set to LOW POWER CONDITION ACTIVE.

If the START STOP UNIT command is issued with the POWER CONDITION field set to 5h the server shall:
a) suspend any Power Condition timers that are active on receipt of the START STOP UNIT command until a WAKEUP task management function is received by the device server or a hard reset;

b) not respond to a task requests until a WAKEUP task management function is received by the device server or a hard reset.

On receipt of a WAKEUP task management function or a hard reset any previously active power conditions timers shall be restored to those values indicated by the saved power condition mode page parameters. Before returning a function complete response the target shall place itself into a condition capable of receiving commands and task management functions and shall create a unit attention condition for all initiators. The sense key shall be set to UNIT ATTENTION and the additional sense code set to LOW POWER CONDITION ACTIVE.
…
In the Sleep condition the device server shall only respond to a WAKEUP task management function or a hard reset. When a target has multiple logical units attached it shall enter the Sleep condition only after all the logical units have been placed into a Sleep condition.

Alternative 2: Suggested changes to SAM-2 revision 16
6 Task Management Functions

[Editor's note: remove the hanging paragraphs by adding an introduction section.]
6.1 Task Management Functions Introduction
Task management functions provide an initiator with a way to explicitly control the execution of one or more tasks. An application client invokes a task management function by means of a procedure call having the following format:

Service Response = Function name (IN (nexus))

Service Response:

One of the following protocol-specific responses shall be returned:

FUNCTION COMPLETE: Each SCSI protocol standard shall define the actual events comprising each of the above service responses. A task manager response indicating that the requested function is complete. The task manager shall unconditionally return this response upon completion of a task management request supported by the logical unit or target device to which the request was directed. Upon receiving a request to execute an unsupported function, the task manager may return this response or the FUNCTION REJECTED response described below.

FUNCTION REJECTED: An optional task manager response indicating that the operation is not supported by the object to which the function was directed (e.g., the logical unit or target device).

SERVICE DELIVERY
OR TARGET FAILURE: The request was terminated due to a service delivery failure or target malfunction. The target may or may not have successfully performed the specified function.

Each SCSI protocol standard shall define the actual events comprising each of the above service responses.

[Editor's note; remove the redundant "summary descriptions". The subsequent sections should fully specify each function. This section has some shalls that were not even mentioned in the subsequent sections.]
The task management functions are summarized ~~as follows~~ in Table xx ~~(see the clauses below for detailed definitions of each task management function):~~.

Table xx. Task Management Functions.

Task Management Function	Nexus
ABORT TASK	I_T_L_Q
ABORT TASK SET	I_T_L
CLEAR ACA	I_T_L
CLEAR TASK SET	I_T_L
LOGICAL UNIT RESET	I_T_L
TARGET RESET	I_T
WAKEUP	I_T

~~**ABORT TASK (IN (I_T_L_Q Nexus))** - Abort the identified task. This function shall be supported if the logical unit supports tagged tasks and may be supported if the logical unit does not support tagged tasks.~~

~~**ABORT TASK SET (IN (I_T_L Nexus))** - Abort all tasks in the task set for the I_T_L nexus. This function shall be supported by all logical units.~~

~~**CLEAR ACA (IN (I_T_L Nexus))** - Clear auto contingent allegiance condition. This function shall be supported if the logical unit accepts a NACA bit value of one in the CDB CONTROL byte (see 5.1.2).~~

~~**CLEAR TASK SET (IN (I_T_L Nexus))** - Abort all tasks in the specified task set as described in 6.4. This function shall be supported by all logical units, except in the following cases, when support for this function is optional:~~
~~a) The logical unit does not support tagged tasks (see 4.9); or~~
~~b) The logical unit supports the basic task management model (see 7.2).~~

~~**LOGICAL UNIT RESET (IN (I_T_L Nexus))** - Perform a logical unit reset as described in 5.7.7 by aborting all tasks in the task set(s) and propagating the reset to all dependent logical units (see 3.1.22). Support for this function is mandatory.~~

~~**TARGET RESET (IN (I_T Nexus))** - Perform a logical unit reset as described in 5.7.7 for every logical unit.~~

Argument descriptions:

Nexus: A non-specific initiator-target nexus (see 4.10).
I_T Nexus: An initiator and target nexus (see 4.10).
I_T_L Nexus: An initiator, target, and logical unit nexus (see 4.10).
I_T_L_Q Nexus: An initiator, target, logical unit, and tag nexus (see 4.10).

NOTE 10 The LOGICAL UNIT RESET, TARGET RESET, CLEAR TASK SET, ABORT TASK and ABORT TASK SET functions provide a means to abort one or more tasks prior to normal completion.

All SCSI protocol standards shall provide the functionality needed for a task manager to implement all of the task management functions ~~defined above~~.
…

6.1 ABORT TASK
Function call:
Service Response = ABORT TASK (IN (I_T_L_Q Nexus))
Description:
This function shall be supported by a logical unit ~~that~~ if it supports tagged tasks and may be supported by a logical unit ~~that~~ if it does not support tagged tasks.

The task manager shall abort the specified task if it exists. Previously established conditions, including MODE SELECT parameters, reservations, and auto contingent allegiance shall not be changed by the ABORT TASK function.

If the logical unit supports this function, a response of FUNCTION COMPLETE shall indicate that the task was aborted or was not in the task set. In either case, the target shall guarantee that no further responses from the task are sent to the initiator.

6.2 ABORT TASK SET
Function Call:
Service Response = ABORT TASK SET (IN (I_T_L Nexus))
Description:
This function shall be supported by all logical units.

The task manager shall abort all tasks in the task set which were created by the initiator as described in 5.5.

The task manager shall perform an action equivalent to receiving a series of ABORT TASK requests. All tasks from that initiator in the task set serviced by the logical unit shall be aborted. Tasks from other initiators or in other task sets shall not be aborted. Previously established conditions, including MODE SELECT parameters, reservations, and auto contingent allegiance shall not be changed by the ABORT TASK SET function. A contingent allegiance (NACA=0) shall be cleared by the ABORT TASK SET function.

6.3 CLEAR ACA
Function Call
Service response = CLEAR ACA (IN (I_T_L Nexus))
Description:
This function shall ~~only be implemented~~ be supported by a logical unit that accepts a NACA bit value of one in the CDB CONTROL byte (see 5.1.2).

The initiator invokes CLEAR ACA to clear an auto contingent allegiance condition from the task set serviced by the logical unit according to the rules specified in 5.7.1.2. If successful, this function shall be terminated with a service response of FUNCTION COMPLETE.

If the task manager clears the auto contingent allegiance condition, any task within that task set may be completed subject to the rules for task set management specified in clause 7.

6.4 CLEAR TASK SET
Function Call:
Service response = CLEAR TASK SET (IN (I_T_L Nexus))
Description:
~~This function shall be supported by all logical units that support tagged tasks (see 4.9) and may be supported by logical units that do not support tagged tasks.~~ This function shall be supported by all logical units, except in the following cases, when support for this function is optional:

a) The logical unit does not support tagged tasks (see 4.9); or
b) The logical unit supports the basic task management model (see 7.2).

All tasks in the appropriate task set as defined by the TST field in the Control mode page (see SPC-2) shall be aborted as described in 5.5. The medium may have been altered by partially executed commands. All pending status and sense data for the appropriate task set shall be cleared.

Previously established conditions, including MODE SELECT parameters, reservations, and auto contingent allegiance (NACA=1, see 5.1.2) shall not be changed by the CLEAR TASK SET function. A contingent allegiance (naca=0) shall be cleared by the CLEAR TASK SET function.

6.5 LOGICAL UNIT RESET
Function Call:
Service Response = LOGICAL UNIT RESET (IN (I_T_L Nexus))
Description:
This function shall be supported by all logical units.

Before returning a FUNCTION COMPLETE response, the logical unit shall perform the logical unit reset functions specified in 5.7.7. A unit attention condition for all initiators that have access shall be created on the logical unit and dependent logical unit(s), if any, as specified in 5.7.5.
NOTE 11 Previous versions of this standard only required LOGICAL UNIT RESET support in logical units that supported hierarchical logical units.

6.6 TARGET RESET
Function Call:
Service Response = TARGET RESET (IN (I_T Nexus))
Description:
Before returning a FUNCTION COMPLETE response, the target port shall perform logical unit reset functions specified in 5.7.7 for every logical unit. A unit attention condition for all initiators that have access shall be created on each of these logical units as specified in 5.7.5.

An initiator should issue LOGICAL UNIT RESETs only to the logical units it is using rather than issuing a TARGET RESET. This avoids resetting logical units that other initiators may be using.

NOTE 12 Previous versions of this standard required TARGET RESET support in all targets. SCSI protocols may or may not require that TARGET RESET be supported. SCSI protocols may require additional actions beyond those specified here.

7.3 WAKEUP
Request:
Service Response = Wakeup (IN (I_T Nexus))

Description:
This function shall only by supported by initiator ports and target ports on protocols that support it.

This function shall only be supported by protocols whose interconnects support a shared wakeup signal or individual wakeup signals for each target port. If individual wakeup signals are supported, the I_T Nexus argument indicates which target port to wakeup.

This function causes a wakeup event to be sent to either:
a) the specified target port; or
b) all target ports connected to the interconnect.

The wakeup function shall cause a hard reset in the recipient target port(s).

Note: SBC-2 defines a Sleep power state where the device powers off most of its protocol interface. The WAKEUP function is used to restore power to the protocol interface. It is detected using a minimal amount of hardware that remains powered on while the rest of the device is powered off. In parallel SCSI, the WAKEUP function is the bus reset condition, shared by all devices. In ATA/ATAPI, the WAKEUP function is either a bus reset shared by all devices on the bus, or a soft reset directed to one of the devices.