To: T10 Technical Committee
From: Rob Elliott, Compaq Computer Corporation (Robert.Elliott@compaq.com)
Date: 18 July 2001
Subject: SAM-2, SPC-3, SPI-4, SBC-2 WAKEUP and reset cleanup

Revision History

Revision 0 (16 April 2001) first revision

Revision 1 (26 June 2001) incorporate comments from CAP meeting. Expanded scope to include SPC-3. The May WG recommended keeping power features, so just obsoleting everything is not proposed. The May WG noticed that the mode page is available for all device types but the power conditions model is in SBC referring to block devices. To reconcile this, the model is moved into SPC-3. The May WG recommended keeping WAKEUP out of SAM-2 and making the SBC-2 references SPI and ATA specific. Instead, I've made the standards use these terms:

SPI-4: wakeup, wakeup event (SPI's definition = reset event)
SAM-2: wakeup, wakeup event* (something defined by the protocol that causes a wakeup)
SBC-3: wakeup
SPC-3: wakeup* (something that causes a power condition change per this spec)

Revision 2 (18 July 2001) incorporate comments from July CAP meeting: fix "device" to "port" a few times and change "wakeup or hard reset" to "wakeup". I made this change when referring to the sleep state but not globally, because a device not implementing the sleep power condition may still implement the power condition page and use it to enter idle or standby after power on (i.e., after hard reset).

Related Documents

spi4r05 – SCSI Parallel Interface - 4 revision 5
sbc2r03 – SCSI Block Commands - 2 revision 3
spc2r19 - SCSI Primary Commands - 2 revision 19
sip-r10 – SCSI Interlocked Protocol revision 10
01-128r1 - 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. Many devices also support an Idle Timer. 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, bus interface is off (except for RST# receiver in parallel SCSI)

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). This is what is referred to as a WAKEUP task management function in SPI-4.

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. There is no other use of the term, so defining it in SPI-4 has little value.
- Move the power conditions model from SBC-2 into SPC-2 and reword it to apply to any device type. The power condition page is currently in SPC-2 and refers to states that are not defined in that standard or SAM-2.
- Continue calling WAKEUP a task management function in SPI-4 and and document it in SAM-2. Mandate that it terminate all tasks like a hard reset. Define it as creating a "wakeup event" which causes a "wakeup."
- Make a few reset cleanups throughout SBC-2 to parallel 01-128r0 (SPI-4 reset cleanup) - use terms hard reset, power on, etc. consistently.
- Clarify how power conditions work in a target device with multiple logical units. A target device only enters sleep state when all its logical units have been programmed to sleep. The other states only affect individual logical units. Define separate logical unit power condition and target device power condition to keep this clear.
- 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.

Suggested changes to SPI-4, SAM-2, SBC-2, and SPC-2 follow.

<u>Suggested Changes to SPI-4 revision 5</u>
[Create "wakeup event" term.
Remove RESET SERVICE DELIVERY SUBSYSTEM.]

<u>**3.1.x wakeup:** A target port returning from the sleep power condition to the active power condition (see SPC-3).</u>

<u>**3.1.x wakeup event:** An event that triggers a wakeup in a SCSI target port as described in SPC-3. Wakeup events defined in this standard are reset events (see 12.5.1).</u>

12.5 Reset events

12.5.1 Reset events overview

When a SCSI device detects a reset event it shall initiate a hard reset (see 12.4). Each reset event (see 12.5) shall also cause a wakeup event in SCSI target ports.

12.6 Wakeup events

When a SCSI target port detects a wakeup event it shall initiate a wakeup (see SPC-3).

Wakeup events defined in this standard are reset events (see 12.5.1).

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 create a bus reset condition (see 12.3) on the selected SCSI bus containing the selected SCSI device. A bus reset condition is treated as a wakeup event (see 12.5.1 and 12.6).

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.

3.1.x wakeup: A target port returning from the sleep power condition to the active power condition (see SPC-3).

3.1.x wakeup event: An event that triggers a wakeup from a target port as described in SPC-3.

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 optional power conditions permit the application client to modify the behavior of a block device in a manner that may reduce the required power. There is no notification to the application client that a block device has entered one of the power conditions. The power conditions may be controlled by the START STOP UNIT command or the power condition page of the MODE SELECT command. If both methods are being used on the same target/logical unit combination then any START STOP UNIT commands power condition request shall override the power condition page's power control. See the START STOP UNIT command description and the power condition mode page description for more information. (See 5.1.22 and SPC-2.)~~

~~No power condition shall affect the SCSI bus.~~

~~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 to be activated.~~

~~In the Standby condition a block device is capable of accepting commands, but media is not immediately accessible (e.g., the spindle is stopped).~~

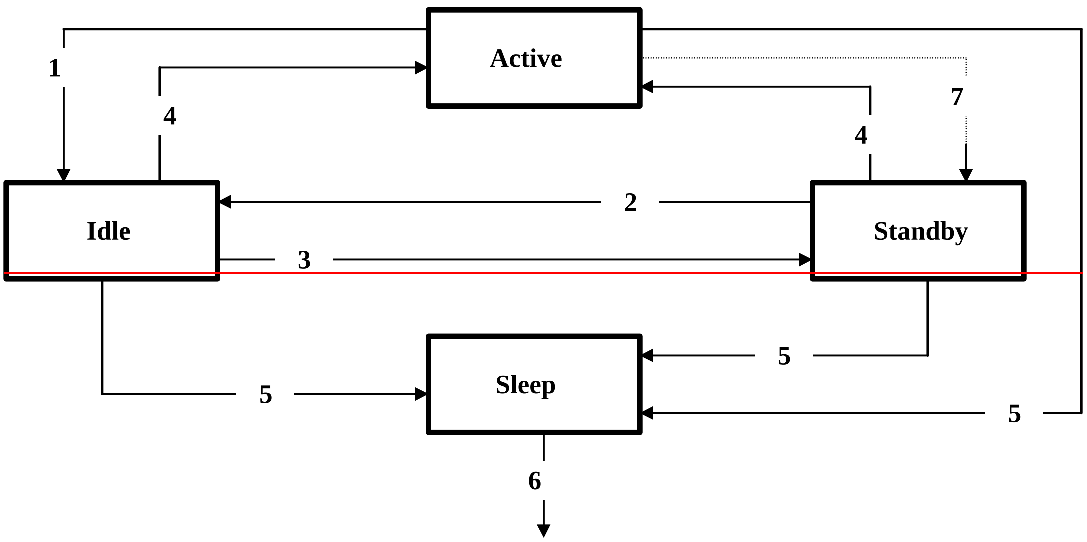
~~In the Idle condition a block device is capable of responding quickly to media access requests. However, a block device in the idle condition may take longer, than in the active condition, to complete the execution of a command because it may have to activate some circuitry.~~

In the Active condition a block device is capable of responding immediately to media access requests, and operations complete execution in the shortest time compared to the other power conditions.

Block devices that contain cache memory shall implicitly perform a SYNCHRONIZE CACHE command for the entire medium prior to entering any power condition that prevents access the media (e.g., the spindle being stopped).

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 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.

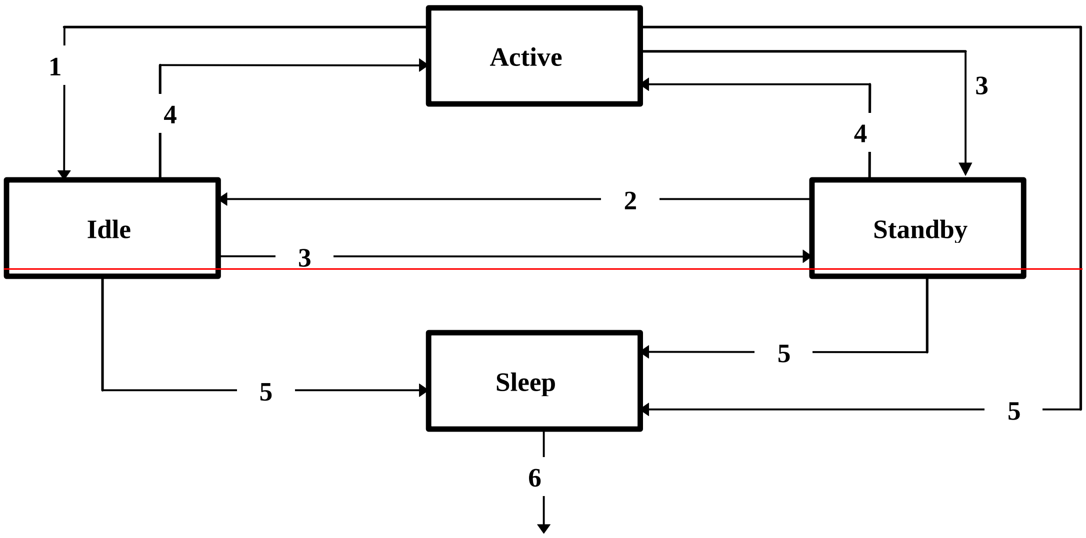


(Active, Idle, or Standby - see (f))

Path 1: An idle time-out or a START STOP UNIT command with a power condition code of Ah.
Path 2: Any command that can be executed within the power constraints of the Idle power condition.
Path 3: A standby time-out, or a START STOP UNIT command with a power condition code of Bh.
Path 4: Any command that exceeds the power constraints of the Idle power condition.
Path 5: A START STOP UNIT command with a power condition code of 5h.
Path 6: A WAKEUP task management function returns the device to the state defined by the saved power mode page parameters.
Path 7: This path only occurs if the idle bit is set equal to zero.

Figure 2 - SCSI power conditions flow control (automatic switching)

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.



~~(Active, Idle, or Standby - see (f))~~
~~Path 1: A START STOP UNIT command with a power condition code of 2h.~~
~~Path 2: A START STOP UNIT command with a power condition code of 2h.~~
~~Path 3: A START STOP UNIT command with a power condition code of 3h.~~
~~Path 4: A START STOP UNIT command with a power condition code of 1h.~~
~~Path 5: A START STOP UNIT command with a power condition code of 5h.~~
~~Path 6: A WAKEUP task management function returns the device to the state defined by the saved power mode parameters.~~

~~**Figure 3 - SCSI power conditions flow control (controlled switching)**~~

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~~ after each ~~power-on and/or~~logical unit 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~~SPC-3) 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~~ with the sense key ~~shall be~~ set to ILLEGAL REQUEST ~~with~~ and 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 device 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~~wakeup;

b) not respond to ~~a~~ commands and task ~~requests~~ management functions until a ~~WAKEUP task management function is received by the device server~~wakeup.

On receipt of a ~~WAKEUP task management function~~wakeup any previously active power condition~~s~~ timers shall be restored to those values indicated by the saved power condition mode page parameters. ~~Before returning a function complete response t~~The target port 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~~ with the additional sense code set to LOW POWER CONDITION ACTIVE.
…
In the s~~S~~leep power condition the device server shall only respond to a ~~WAKEUP task management function~~wakeup. When a target port has multiple logical units attached it shall enter the ~~Sleep~~ sleep power condition only after all the logical units have been placed into a ~~Sleep~~ sleep power condition.

Suggested changes to SAM-2 revision 18

[Changes include:
a) adding normative references to SPC-2 (which is referenced everywhere)
b) added WAKEUP
c) editorial suggestions to move redundant material out of the Introduction section]

2.1 Normative references

The following standards contain provisions that, by reference in the text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed below.

Copies of the following documents may be obtained from ANSI: approved ANSI standards, approved and draft international and regional standards (ISO, IEC, CEN/CENELEC, ITUT), and approved and draft foreign standards (including BSI, JIS, and DIN). For further information, contact ANSI Customer Service Department at 212-642-4900 (phone), 212-302-1286 (fax) or via the World Wide Web at http://www.ansi.org.

ISO/IEC *14776-312*, SCSI Primary Commands - 2 SPC-2 [ANSI NCITS.TBD:2001]

…

3.1.x wakeup: A target port returning from the sleep power condition to the active power condition (see SPC-3).

3.1.x wakeup event: An event that triggers a wakeup from a SCSI target port as described in SPC-3.

…

5.8.6 Hard reset

A hard reset is a target port action in response to a reset event within the service delivery subsystem. A wakeup event is also a reset event. The definition of additional reset events is protocol specific. Each SCSI protocol standard that defines reset events shall specify the target port's action in response to reset events.

~~The~~ A target port's response to a hard reset shall include initiating the equivalent of a logical unit reset for all logical units as described in 5.8.7 and shall include a wakeup as described in SPC-3.

While the task manager response to task management requests is subject to the presence of access restrictions, as managed by ACCESS CONTROL OUT commands (see SPC-3), a hard reset in response to a reset event within the service delivery subsystem shall be unaffected by access controls.

6 Task Management Functions

6.1 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, CLEAR TASK SET, LOGICAL UNIT RESET, TARGET RESET, and WAKEUP 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~~ if it 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:
Protocols may or may not define the WAKEUP function. This function may be supported by protocols whose interconnects support a shared wakeup signal or individual wakeup signals for each target port. This function may be supported by devices on protocols which support the function.

This function causes a wakeup event (see SPC-3) to be sent to either:
a) the specified target port, on protocols supporting individual wakeup signals; or
b) all target ports connected to the interconnect, on protocols supporting a shared wakeup signal.

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

Suggested changes to SPC-3

[Move SBC-2 section 4.2.1.5 modeling power conditions here. Due to extensive changes, change bars are not included.]

3.1.x wakeup: A target port returning from the sleep power condition to the active power condition (see 5.8).

3.1.x wakeup event: An event defined by the protocol that triggers a wakeup from a target port as described in 5.8.

5.8 Power conditions

The optional power conditions permit the application client to modify the behavior of a target port and/or logical unit in a manner that may reduce power consumption. There is no notification to the application client that one of the power conditions has been entered. Power conditions may be controlled by the START STOP UNIT command for some device types (see SBC-2 or RBC) or the power condition mode page for all device types (see 8.3.9). If both methods are being used on the same logical unit then any START STOP UNIT command's power condition request shall override the power condition mode page's power condition.

No power condition shall affect the service delivery subsystem.

The logical unit power conditions are described in Table 1.

Table 1- Logical unit power conditions

Logical unit power condition	Description
active	Device server is capable of responding to all its supported commands including media access requests, and operations complete processing in the shortest time compared to the other power conditions.
idle	Device server is capable of responding to all its supported commands including media access requests. However, a device server in the idle condition may take longer than in the active logical unit power condition to complete processing a command because it may have to activate some circuitry.
standby	Device server is not capable of processing media access commands (e.g., the spindle is stopped).
sleep	Device server is not capable of accepting or processing commands. The logical unit requires a wakeup to return to the active, idle, or standby power condition.

The logical unit sleep power condition shall only be supported on logical units accessed through target ports using protocols that define wakeup events.

The target port power conditions are described in **Table 2**.

Table 2- Target port power conditions

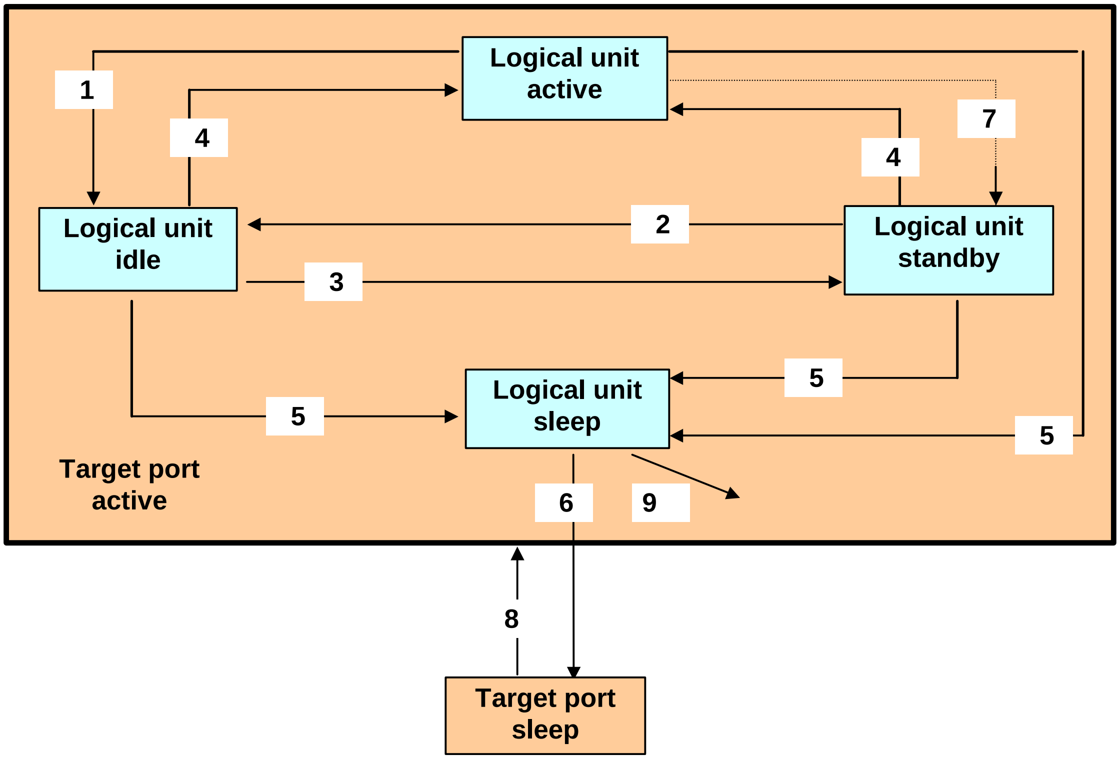
Target port power condition	Description
active	Logical units to which the target port has access that are not in the sleep logical unit power condition are capable of accepting commands routed from the target port.
sleep	Logical units to which the target port has access are not capable of accepting commands routed from the target port. The lowest power consumption, with power applied, occurs in the sleep power condition. The target port requires a wakeup ~~or hard reset~~ to return to the active power condition. The target port enters the sleep power condition only when all the logical units to which it has access have entered the sleep power condition.

The target port sleep power condition shall only be supported on target ports using protocols that define wakeup events.

Block devices that contain cache memory shall implicitly perform a SYNCHRONIZE CACHE command (see SBC-2 or RBC) for the entire medium prior to entering any power condition that prevents access the media (e.g., the spindle being stopped).

If implemented, the target port shall use the optional power condition mode page (see 8.3.9) to control the logical unit power conditions after a wakeup or hard reset until a START STOP UNIT command (see SBC-2 or RBC) is received with the POWER CONDITIONS field set to a value other than 0h or 7h.

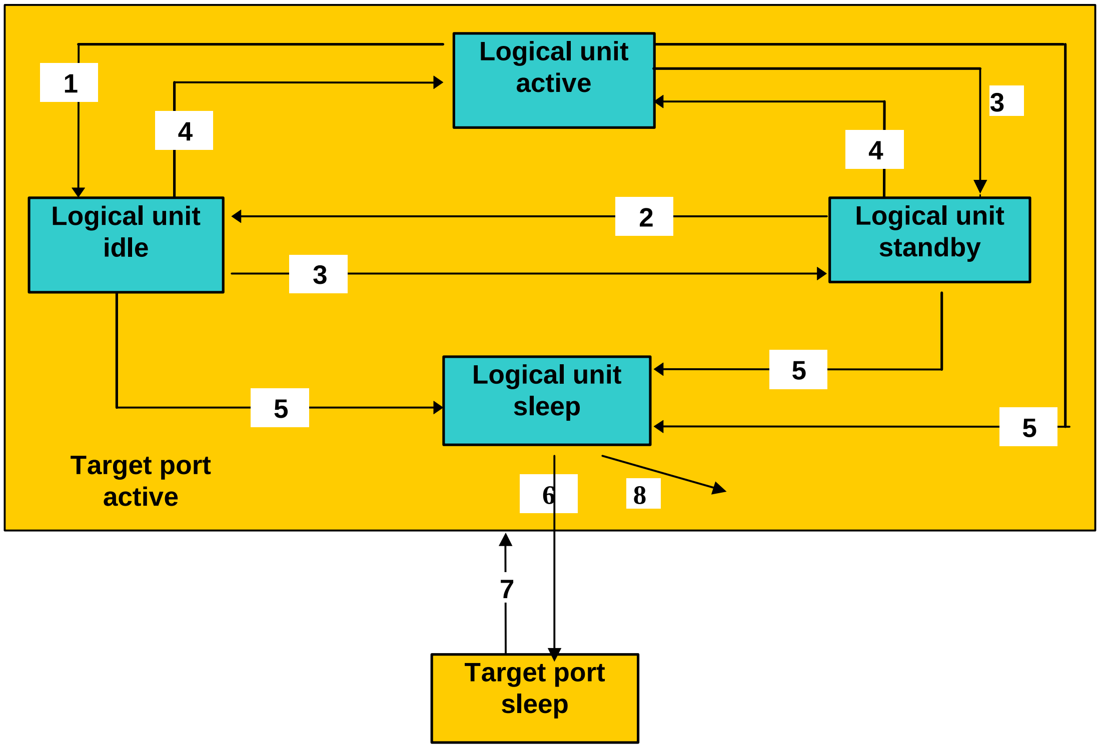
Figure 2 shows the transitions between the different power conditions in a logical unit and target port that are set up to adjust automatically to the power condition that allows any command to execute.



Path 1: An idle time-out or a START STOP UNIT command with a power condition code of Ah.
Path 2: Any command that can be executed within the power constraints of the idle power condition.
Path 3: A standby time-out, or a START STOP UNIT command with a power condition code of Bh.
Path 4: Any command that exceeds the power constraints of the idle power condition.
Path 5: A START STOP UNIT command with a power condition code of 5h.
Path 6: When all logical units in a target port reach the logical unit sleep power condition, the target port transitions to the target port sleep power condition.
Path 7: A standby time-out when the idle bit is set to zero.
Path 8: A wakeup or hard reset returns the target port to the active power condition and returns each logical unit to the power condition (active, idle, or standby) defined by the saved power condition mode page parameters.
Path 9: A wakeup or hard reset returns the logical unit to the power condition (active, idle, or standby) defined by the saved power condition mode page parameters.

Figure 2 - Power condition transitions (automatic switching)

Figure 3 shows the transitions between the different power conditions in a logical unit and target port that are setup to only allow changing of the power condition by the application client. Any command received that requires more power than allowed by the current power condition shall be terminated with a sense key of ILLEGAL REQUEST and the additional sense code shall be set to LOW POWER CONDITION ACTIVE.



Path 1: A START STOP UNIT command with a power condition code of 2h.
Path 2: A START STOP UNIT command with a power condition code of 2h.
Path 3: A START STOP UNIT command with a power condition code of 3h.
Path 4: A START STOP UNIT command with a power condition code of 1h.
Path 5: A START STOP UNIT command with a power condition code of 5h.
Path 6: When all logical units to which a target port has access reach the logical unit sleep power condition, the target port transitions to the target port sleep power condition.
Path 7: A wakeup or hard reset returns the target port to the active power condition and returns each logical unit to the power condition (active, idle, or standby) defined by the saved power condition mode page parameters.
Path 8: A wakeup or hard reset returns each logical unit to the power condition (active, idle, or standby) defined by the saved power condition mode page parameters.

Figure 2 - Power condition transitions (controlled switching)