Universal Serial Bus 3.2

Hub Compliance Test Specification

Revision 1.21

Date:	July 10, 2019
Revision:	1.21

Revision History

Revision	Issue Date	Comments
0.1	7/25/2013	Revision 0.1 release.
1.1		USB 3.1
1.2	3/6/2018	USB 3.2
1.21	2/7/2019	Update bus+self-powered section for TD 10.12

Significant Contributors:

Please send comments via electronic mail to: techadmin@usb.org or ssusbcompliance@usb.org

1 Hub Testing Requirements

3.2 Hubs must pass all of the following:

- Chapter 10 Super Speed Hub Tests
- SS Hub Loopback Tests
- 2.0 Hub Tests (HS)
- 2.0 Hub Tests (FS)
- Chapter 9 Framework Tests (SS)
- Chapter 9 Framework Tests (HS)
- Chapter 9 Framework Tests (FS)
- Interoperability Test
- Backwards Compatibility Test
- Electrical Tests
- Link Tests

2 Test Assertions

Unless otherwise noted, subsection references are to the USB3.2 Specification.

Assertion #	**Assertion Description**	**Test #**
Chapter 8 Test Assertions:		
Subsection reference: 8.4.2 Set Link Function		
8.4.2#2	Set Link Function LMP is sent by a hub to a device connected on a specific port when it receives a SetPortFeature(FORCE_LINKPM_ACCEPT) request.	LVS
Chapter 9 Test Assertions:		
Subsection reference: 9.4.14#1 Events and Their Effect on Device Parameters		
9.4.14#1	Downstream U1 Inactivity Timeout for a port should be reset if the hub downstream port is disconnected or downstream port Hot Reset or downstream port Warm Reset.	LVS
9.4.14#2	Downstream U2 Inactivity Timeout for a port should be reset if the hub downstream port is disconnected or downstream port Hot Reset or downstream port Warm Reset.	LVS
9.4.14#3	Downstream U1 and U2 Inactivity Timeout for a port should be reset if the hub upstream is disconnected or SetAddress(0) or upstream port Hot Reset or upstream port Warm Reset.	LVS

Assertion #	Assertion Description	Test #
9.4.14#4	Hub Depth should be reset if the hub upstream is disconnected or SetConfiguration or SetAddress(0) or Hot Reset or Warm Reset.	Untestable
Chapter 10 Test Assertions:		
Subsection reference: 10 Hub, Host Downstream Port, and Device Upstream Port Specification		
Subsection reference: 10.1 Hub Feature Summary		
10.1#1	All exposed downstream ports on an Enhanced SuperSpeed hub shall support both Gen X and USB 2.0 connections.	Implicitly tested when Hub goes through 3.0 Hub tests and 2.0 Hub tests Will be running most of the non-descriptor tests on every 3.0 port in 3.0 Hub tests and 2.0 Hub tests already run the tests on every port.
10.1#2	Hub detects device connect and disconnect	Multiple tests
10.1#3	When an Enhanced SuperSpeed hub connects on its upstream facing port at Gen 1, its downstream ports shall connect no higher than Gen 1.	LVS
10.1#4	Hubs shall only provide power to DS ports if the US port is connected, or the hub supports Battery Charging, Power Delivery, or USB Type-C current.	LVS
Subsection reference: 10.1.1 Connecting to an Enhanced SuperSpeed Capable Host		
10.1.1#1	An Enhanced SuperSpeed Hub must connect as a SuperSpeed or SuperSpeed Plus device and as a High Speed device.	TD 10.2
10.1.1#2	An Enhanced SuperSpeed hub shall have a unique Product ID value for the USB 2 and SuperSpeed hub portions of the device.	TD 10.2
10.1.1#3	The UFP will train at the fastest speed supported by its link partner.	LVS
10.1.1#4	DFPs must train at the fastest speed supported by its link partner.	10.26 (SS) LVS (SSP)
10.1.1#5	DS port power and training is simultaneous across DS ports.	TBD LVS – Not Tested

Assertion #	Assertion Description	Test #
10.1.1#6	If a downstream port trained at a higher speed than the upstream port then the downstream port shall retrain at a speed no faster than the upstream port.	LVS
Subsection reference: 10.1.2 Connecting to a USB 2.0 Host		
10.1.2#1	If hub supports one or more power applications and host is powered off, the hub provides power to its downstream ports.	Battery Charging Test
10.1.2#2	When Host is powered and there is no Gen X hardware support, hub connects and enumerates as a high speed device only.	Implicitly tested by Interoperability Test Topology Change 2. HUT is attached to High Speed hub.
Subsection reference: 10.1.3 Hub Connectivity		
Subsection reference: 10.1.3.1 Routing Information		
10.1.3.1#1	Packets received on the hub UFP are routed based on information contained in a 20-bit field (Route String) in the packet header.	Implicity tested.
10.1.3.1#2	The hub sets the route string to zero when the hub controller responds to a packet containing a route string routed to the root hub.	LVS
10.1.3.1#3	A hub shall assume all packets are routed directly to the hub until the hub enters the configured state and the hub's depth is set.	LVS
10.1.3.1#4	A hub shall set the route string of an upward flowing packet to 0 if the packet was originated by the hub controller.	LVS
10.1.3.1#5	A hub shall set the route string of an upward flowing packet to the route string of the corresponding downward flowing packet when the packet has been marked as deferred.	LVS
10.1.3.1#6	A hub shall set the route string of an upward flowing packet to the aggregate arbitration weight of the hub.	LVS – Not Tested
Subsection reference: 10.1.3.2 SuperSpeed Hub Packet Signaling Connectivity		
10.1.3.2#1	Packets transmitted upstream by a device attached to the hub are only sent to the upstream facing hub.	LVS
10.1.3.2#2	All packets except ITP are unicast in the downstream direction; SuperSpeed hubs operate using a direct connectivity model.	LVS
10.1.3.2#3	Isochronous Timestamp Packet is multicast in the downstream direction.	LVS
10.1.3.2#4	ITPs can only be sent to Ports in U0.	LVS

Assertion #	Assertion Description	Test #
10.1.3.2#5	The hub shall silently drop a header packet that is routed to a Downstream Facing Port (DFP) that is not enabled. The hub will perform link level acknowledgement in this case.	LVS Interop
Subsection reference: 10.1.3.3 SuperSpeedPlus Hub Packet Routing		
10.1.3.3#1	Packets transmitted upstream by a device attached to a SuperSpeedPlus hub are only sent to the upstream facing hub.	LVS
10.1.3.3#2	All packets except ITP are unicast in the downstream direction.	LVS
10.1.3.3#3	Isochronous Timestamp Packet is multicast in the downstream direction	LVS
10.1.3.3#4	ITPs can only be sent to Ports in U0.	LVS
10.1.3.3#5	The hub shall silently drop a header packet that is routed to a Downstream Facing Port (DFP) that is not enabled. The hub will perform link level acknowledgement in this case.	LVS Interop
Subsection reference: 10.1.4 Resume Connectivity		
10.1.4#1	Hub shall ensure that there are no race conditions in the case where wake signaling from multiple downstream ports overlap with each other and that wake signaling is propagated to all such ports.	Untestable
10.1.4#2	After receiving remote wakeup signaling from a downstream port, the hub shall ensure that it propagates the device wake notification upstream.	LVS
Subsection reference: 10.1.5 Hub Fault Recovery Mechanisms		
10.1.5#1	The Hub Controller shall adhere to the same rules as other USB devices, as described in Chapter 8 of the USB 3.2 specification.	Implicitly tested
Subsection reference: 10.1.6 Hub Header Packet Buffer Architecture		
Subsection reference: 10.1.6.1 SuperSpeed Hub Buffer Architecture		
Subsection reference: 10.1.6.1.1 SuperSpeed Hub Header Packet Buffer Architecture		
10.1.6.1.1#1	Header packets are kept in the hub header packet buffers until link level acknowledgement (LGOOD_n) for the header packet is received.	Link Tests TD 7.30

Assertion #	Assertion Description	Test #
10.1.6.1.1#2	When a downstream link is in a low power state, the header packet will be delivered when the link becomes active.	Interoperability, trace deferred packet and completed packet between hub and device. LVS
Subsection reference: 10.1.6.1.2 Hub Data Buffer Architecture		
10.1.6.1.2#1	Enhanced SuperSpeed hubs shall allow transactions in both directions to occur concurrently.	TD 5.19 Hub Loopback Tests
10.1.6.1.2#2	If data packets payloads are discarded because buffering is unavailable in the hub, the end-to-end protocol will recover by retrying the transaction.	TD 5.19 Hub Loopback Tests and Interoperability
Subsection reference: 10.1.6.2 SuperSpeedPlus Hub Data Buffer Architecture		
10.1.6.2#1	Downstream ports on a SuperSpeedPlus hub can operate at a different speed than the upstream port.	Interop
10.1.6.2#2	There can be multiple DPs simultaneously in transit on different downstream ports.	Interop
10.1.6.2#3	There can be multiple DPs buffered in a hub awaiting transmission.	Interop
Subsection reference: 10.2 Hub Power Management		
Subsection reference: 10.2.1 Link States		
10.2.1#1	Hub supports U0, U1, U2 and U3 on all upstream and downstream ports.	TD 10.9 Chapter 9 TD 9.24 LVS
Subsection reference: 10.2.2 Hub Downstream Port U1/U2 Timers		
10.2.2#1	The hub is required to have inactivity timers for both U1 and U2 on each DFP.	LVS
10.2.2#2	If a hub has received a valid packet on its upstream port that is routed to a downstream port, it shall reject U1 or U2 link entry attempts on the downstream port until the packet has been successfully transmitted.	LVS
10.2.2#3	Hub implementation ensures no race condition when a header packet that has not been deferred is queued for transmission on a downstream port with a link that is in U1, U2, or is in the process of entering U1, U2.	LVS
10.2.2#4	The default values for the U1/ U2 timeouts is zero.	LVS
10.2.2#5	The hub inactivity timers for U1 and U2 shall not be reset by an ITP.	LVS

Assertion #	Assertion Description	Test #
10.2.2#6	Down Stream Facing Ports U1/U2 timeout values must reset to default values at power on reset.	LVS
Subsection reference: 10.2.3 Downstream/Upstream Port Link State Transitions		
10.2.3#1	The hub shall evaluate the link power state of its DFPs such that it propagates the highest link state of any of its DFPs to its UFP when there is no pending upstream traffic. U0 is the highest link state, followed by U1, then U2, then U3, then Rx.Detect and then SS.Disabled.	LVS
10.2.3#2	The hub shall ensure that there is no race condition between putting a UFP into the highest link state of its DFPs and a DFP changing its state. For example if hub is transitioning the UFP to U1 because the last DFP went to U1 and another DFP is trying to go to U0 at the same time.	Microsoft LPM tool.
10.2.3#3	The hub never automatically attempts to transition the hub upstream port to U3 or lower state.	LVS
Subsection reference: 10.3 Hub Downstream Facing Port		
Subsection reference: 10.3.1 Hub Downstream Facing Port State Descriptions		
Subsection reference: 10.3.1.1 DSPORT.Powered-off		
10.3.1.1#1	The hub downstream port shall transition to DSPORT.Powered-off state from any state when local power is lost to the port.	Untestable
10.3.1.1#2	The hub downstream port shall transition to DSPORT.Powered-off state from any state when V_{BUS} is removed from the hub upstream port and the hub supports power switching on the DS ports.	LVS
10.3.1.1#3	The hub downstream port shall transition to DSPORT.Powered-off state from any state if the Hub upstream port link transitions to the SS. Disabled state and UFP VBUS is on.	LVS
10.3.1.1#4	The hub downstream port shall transition to DSPORT.Powered-off state from any state when the hubs UFP link has attempted eight consecutive Rx.Detect events without detecting far-end receiver terminations and the near-end receiver termination of the hubs DFP is ready to be turned off.	LVS
10.3.1.1#5	If a hub was configured while the local power supply was present and then if local power is lost, the hub shall place all ports in the Powered-off state if UFP VBus is on and bMaxPower > 0.	LVS
10.3.1.1#6	A DFP's termination is considered ready to turn off when warm reset signaling has completed.	LVS

Assertion #	Assertion Description	Test #
10.3.1.1#7	A DFP’s termination is optionally considered ready to turn off when the port is power switched and the USB 2.0 hub has also turned off power to the port.	LVS
10.3.1.1#8	A DFP’s termination is considered ready to turn off when the port is in a powered on state and not performing a warm reset.	LVS
10.3.1.1#9	A DFP’s termination is considered ready to turn off when no device is connected.	LVS
10.3.1.1#10	A port shall remain in the DSPORT.Powered-off state until the hub’s upstream port link has detected far-end receiver terminations, no “power-off” condition is true, and no Over-current condition is active.	LVS
10.3.1.1#11	In the DSPORT.Powered-off state, the port’s link is in the SS.Disabled state.	Implicit
10.3.1.1#12	If a hub was configured while local power was present and then power is lost, the hub shall place all DS ports in the powered-off state if power remains to run the hub controller.	LVS
10.3.1.1#13	If an Enhanced SuperSpeed hub provides power to DS ports all the time, it must ensure that Enhanced SuperSpeed devices attempt Enhanced SuperSpeed connection once upstream VBUS is seen by the hub.	LVS
Subsection reference: 10.3.1.2 DSPORT.Disconnected (Waiting for SS Connect)		
10.3.1.2#1	A port transitions to DSPORT.Disconnected state from SS.Inactive, Rx.Detect.Active, U1, U2 or U3 states when the port detects a disconnect.	LVS
10.3.1.2#2	A port transitions to DSPORT.Disconnected state from the DSPORT.Resetting state when a port’s link times out from Rx.Detect during a reset.	LVS
10.3.1.2#3	A port transitions to DSPORT.Disconnected state from the DSPORT.Disabled state when a SetPortFeature(PORT_LINK_STATE) Rx.Detext request is received for the port.	LVS
10.3.1.2#4	A port transitions to DSPORT.Disconnected state from the DSPORT.Resetting state if the port’s link times out from any Polling substate during a reset.	LVS
10.3.1.2#5	A port transitions to DSPORT.Disconnected state from the DSPORT.Training state if the port’s link times out from any Polling substate and the cPollingTimeout is less than 2 and the port is not enabled to enter compliance or Polling substate.	LVS
10.3.1.2#6	A port transitions to DSPORT.Disconnected state from the DSPORT.Loopback state if the port’s link performs a successful LFPS handshake in Loopback.Exit.	Not Tested

Assertion #	Assertion Description	Test #
10.3.1.2#7	A port transitions to DSPORT.Disconnected state from the DSPORT.Powered-off-reset state if the conditions for repowering are met and DSPORT.Powered-off-reset state has been maintained for treset.	LVS
10.3.1.2#8	A port transitions to DSPORT.Disconnected state from the DSPORT.Powered-off-detect state if the conditions for repowering are met.	LVS
10.3.1.2#9	A port transitions to DSPORT.Disconnected state from the DSPORT.Powered-off or DSPORT.Disabled state if the UFP receives a reset. The DFP shall issue a warm reset on the link if a far-end receiver is detected.	LVS
10.3.1.2#10	In DSPORT.Disconnected state, the port's link shall still perform connection detection normally from the Rx.Detect state if the hub upstream port's link is in U3.	LVS
10.3.1.2#11	In the DSPORT.Disconnected state, the port's link shall be in the Rx.Detect state.	Implicit
10.3.1.2#12	A port transitions to the DSPORT.Disconnected state from DSPORT.Powered-off state when the hub's Upstream Port link has detected far-end receiver terminations, Upstream Port VBUS is on, no power-off condition is met and no Over-current condition exists.	LVS
Subsection reference: 10.3.1.3 DSPORT.Training		
10.3.1.3#1	In DSPORT.Training state, the port's link shall be in the Polling state.	Untestable
Subsection reference: 10.3.1.4 DSPORT.Error		
10.3.1.4#1	A port shall transition to DSPORT.Error state only when an Enhanced SuperSpeed capable device is connected and a serious error condition occurred while attempting to operate the link.	Untestable
10.3.1.4#2	A port shall transition to DSPORT.Error state from the DSPORT.Enabled state if the link enters recovery and times out without recovering.	Not Tested
10.3.1.4#3	A port shall transition to DSPORT.Error state from the DSPORT.Resetting state if U1 or U2 exit fails.	Not Tested
10.3.1.4#4	A port shall transition to DSPORT.Error state from the DSPORT.Loopback state if the port is the loopback master and the LFPS handshake in Loopback.Exit fails.	Not Tested
10.3.1.4#5	A port shall transition to DSPORT.Error state from DSPORT.Enabled if Port Configuration fails as described in Section 8.4.6.	LVS

Assertion #	Assertion Description	Test #
10.3.1.4#6	A port shall transition to DSPORT.Error state from DSPORT.Training if the port's link times out from any Polling substate and cPollingTimeout is 2.	Not Tested
Subsection reference: 10.3.1.5 DSPORT.Enabled		
10.3.1.5#1	Port shall transition to DSPORT.Enabled state from the Training state when the port's link successfully enters U0.	Implicit
10.3.1.5#2	Port shall transition to DSPORT.Enabled state from the DSPORT.Resetting state when a reset completes successfully.	LVS
10.3.1.5#3	A port in the DSPORT.Enabled state will propagate packets in both the upstream and the downstream direction after its CCS is set.	Implicit
10.3.1.5#4	When the hub downstream port first transitions to the DSPORT.Enabled state after a power on or warm reset, it shall transmit a port configuration LMP as defined in Section 8.4.5.	LVS
10.3.1.5#5	If the hub UFP is in U3 when the DFP enters DSPORT.Enabled and the hub is not enabled for remote wakeup, the DFP shall initiate a transition to U3 on its link within tDSPortEnableToU3.	LVS
10.3.1.5#6	If CCS was set before entering the DSPORT.Enabled state, it will remain set.	LVS
10.3.1.5#7	If CCS was not set before entering DSPORT.Enabled state, it shall be set only after port configuration LMP exchange succeeds.	LVS
Subsection reference: 10.3.1.6 DSPORT.Resetting		
10.3.1.6#1	A downstream port shall transition to the DSPORT.Resetting state when the DFP receives SetPortFeature(PORT_RESET) unless the port is in the DSPORT.Powered-off, DSPORT.Powered-off-reset, DSPORT.powered-off-detect, DSPORT.Disabled or DSPORT.Disconnected state.	LVS
10.3.1.6#2	A downstream port shall transition to the DSPORT.Resetting state when the DFP receives SetPortFeature(BH_PORT_RESET) unless the port is in the DSPORT.Powered-off, DSPORT.Powered-off-reset, DSPORT.powered-off-detect, DSPORT.Disabled or DSPORT.Disconnected state.	LVS
10.3.1.6#3	If the downstream port is in the DSPORT.Powered-off, DSPORT.Powered-off-reset, DSPORT.powered-off-detect, DSPORT.Disabled or DSPORT.Disconnected state and receives a SetPortFeature reset request, the request is ignored.	LVS TD 10.14

Assertion #	Assertion Description	Test #
10.3.1.6#4	A downstream port shall transition to DSPORT.Resetting if the port state is DSPORT.Enabled and the port's link is in any state other than U3 when a SetPortFeature(PORT_RESET) request is received. The port shall initiate a hot reset on the link.	LVS
10.3.1.6#5	A downstream port shall transition to DSPORT.Resetting if the port state is DSPORT.Enabled and the port's link is in any state, if the port receives a SetPortFeature(BH_PORT_RESET) request. The port shall initiate a warm reset on the link.	LVS
10.3.1.6#6	If the port initiates a hot reset on the link and the hot reset TS1/TS2 handshake fails, a warm reset is automatically tried.	Link Test TD 7.31
10.3.1.6#7	When the downstream port link enters Rx.Detect.Active during a warm reset, the hub shall start a timer to count the time it is in Rx.Detect.Active or Rx.Detect.Quiet. If this timer exceeds tTimeForResetError while the link remains in Rx.Detect, the port shall transition to the DSPORT.Disconnected state.	LVS
10.3.1.6#8	A downstream port shall transition to the DSPORT.Resetting state when the UFP receives a hot reset unless the port is in the DSPORT.Powered-off, DSPORT.Powered-off-reset, DSPORT.powered-off-detect, DSPORT.Disabled or DSPORT.Disconnected state. The port shall transition before the UFP link reaches U0.	LVS
10.3.1.6#9	A downstream port shall transition to the DSPORT.Resetting state when the UFP receives a warm reset unless the port is in the DSPORT.Powered-off, DSPORT.Powered-off-reset, DSPORT.powered-off-detect, DSPORT.Disabled, or DSPORT.Disconnected state. The port shall initiate a warm reset on the link before the UFP link reaches U0.	LVS
10.3.1.6#10	A downstream port shall not transition into the DSPORT.Powered-off state when the UFP receives a hot or warm reset when the port is in the DSPORT.Powered-off, DSPORT.Powered-off-reset, DSPORT.powered-off-detect, DSPORT.Disabled or DSPORT.Disconnected state.	LVS
10.3.1.6#11	A downstream port shall transition to the DSPORT.Disconnected state when the UFP receives a hot or warm reset when the port is in the DSPORT.Powered-off, DSPORT.Powered-off-reset, DSPORT.powered-off-detect, DSPORT.Disabled or DSPORT.Disconnected state.	LVS
10.3.1.6#12	A downstream port shall transition to the DSPORT.Resetting state if the port state is DSPORT.Enabled and the port's link is in U3 when a SetPortFeature(PORT_RESET) request is received. The port shall initiate a warm reset on the link.	LVS

Assertion #	Assertion Description	Test #
10.3.1.6#13	If a DS port transitioned to DSPORT.Resetting due to a UFP Hot Reset, and the DS port was initially in DSPORT.Training or DSPORT.Enabled with PLS not in U3, the DS port initiates a Hot Reset on the link.	LVS
10.3.1.6#14	If a DS port transitioned to DSPORT.Resetting due to a UFP Hot Reset, and the DS port was initially in DSPORT.Training or DSPORT.Enabled with PLS in U3, or any other DSPORT state excluding DSPORT.Powered-off, DSPORT.Powered-off-reset, DSPORT.powered-off-detect, DSPORT.Disabled or DSPORT.Disconnected, then the DS port initiates a Warm Reset on the link.	LVS
10.3.1.6#15	If a DS port transitioned to DSPORT.Resetting due to a SetPortFeature(PORT_RESET), and the port did not transition from DSPORT.Enabled, the port shall initiate a Warm Reset on the link.	LVS
Subsection reference: 10.3.1.7 DSPORT.Compliance		
10.3.1.7#1	A downstream port transitions to DSPORT.Compliance state when the link enters the Compliance Mode state.	Untestable
Subsection reference: 10.3.1.8 DSPORT.Loopback		
10.3.1.8#1	A downstream port transitions to DSPORT.Loopback state from the DSPORT.Training state if the loopback bit is set in the received TS2 ordered sets.	Untestable
10.3.1.8#2	In DSPORT.Loopback state, the port’s link shall be in the Loopback state.	Untestable
Subsection reference: 10.3.1.9 DSPORT.Disabled		
10.3.1.9#1	A downstream port shall transition to DSPORT.Disabled state when the port receives a SetPortFeature(PORT_LINK_STATE) eSS.Disable request.	LVS
10.3.1.9#2	A downstream port’s link state shall be eSS.Disabled in the DSPORT.Disabled state.	LVS
Subsection reference: 10.3.1.10 DSPORT.Powered-off-detect		
10.3.1.10#1	A downstream port shall transition to the DSPORT.Powered-off-detect state from the DSPORT.Powered-off-reset state when treset has been met and conditions for Repowering are not met.	LVS
10.3.1.10#2	When in DSPORT.Powered-off-detect state, the port shall perform far-end receiver detection with the link state in Rx.Detect until a far-end receiver termination is detected or any power-off condition is met, or repower conditions are met for the port.	LVS
10.3.1.10#3	In the DSPORT.Powered-off-detect state, SS terminations are maintained and far-end receiver detection is enabled	LVS

Assertion #	Assertion Description	Test #
10.3.1.10#4	Upstream port VBUS is considered to have transitions from off to on when it is on at power-up.	Implicit
10.3.1.10#5	DSPORT.Powered-off-detect is entered regardless of previous state if no power-off condition is met and Over-current condition is detected either on this port or globally and Upstream port far-end receiver terminations are present and upstream VBUS is on.	Not Tested
10.3.1.10#6	DSPORT.Powered-off-detect is entered regardless of previous state if no power-off condition is met and upstream port VBUS is off and the hub does not support power switching.	LVS
10.3.1.10#7	DSPORT.Powered-off-detect is entered regardless of previous state if no power-off condition is met and the hub receives a ClearPortFeature(PORT_POWER) for this port.	LVS
10.3.1.10#8	DSPORT.Powered-off-detect is entered regardless of previous state if no power-off condition is met and the hub upstream port receives a SetConfiguration(0) request.	LVS
Subsection reference: 10.3.1.11 DSPORT.Powered-off-reset		
10.3.1.11#1	Terminations are maintained while in DSPORT.Powered-off-reset state.	Implicit
10.3.1.11#2	DSPORT.Powered-off-reset state is entered from DSPORT.Powered-off-detect whenever a far end receiver is detected.	LVS
10.3.1.11#3	When in the DSPORT.Powered-off-reset state, a port shall drive warm reset and remain in link substate Rx.Detect.Reset for tReset duration.	LVS
Subsection reference: 10.3.3 Labeling		
10.3.3#1	If a vendor provides labeling to identify individual DFPs, then each port connector shall be labeled with its respective port number.	TD 10.15
10.3.3#2	The port numbers assigned to a specific port by the hub shall be consistent between the USB 2.0 hub and Enhanced SuperSpeed hub.	TD 10.2
Subsection reference: 10.4 Hub Downstream Facing Port Power Management		
Subsection reference: 10.4.1 Downstream Facing Port PM Timers		
10.4.1#1	The PM timers are reset to 0 every time a SetPortFeature(PORT_U1_TIMEOUT) or SetPortFeature(PORT_U2_TIMEOUT) request is received.	LVS

Assertion #	Assertion Description	Test #
10.4.1#2	The inactivity timers shall be reset every time a packet of any type except isochronous timestamp packet is received or sent by the port's link.	LVS
10.4.1#3	The U1 timer shall be accurate to +1/-0 µs. The U2 timer shall be accurate to +500/-0 µs.	LVS
Subsection reference: 10.4.2 Hub Downstream Facing Port State Descriptions		
Subsection reference: 10.4.2.1 Enabled U0 States		
10.4.2.1#1	When U2_TIMEOUT = 0, the port’s link shall reject all U2 transition requests by the link partner.	LVS
10.4.2.1#2	When U1_TIMEOUT = X > 0, the port’s link shall accept U1 entry requests by its link partner unless the hub has one or more packets/link commands to transmit on the port.	LVS
10.4.2.1#3	If the U1 timeout is 0xFF, the port shall be disabled from initiating U1 entry but shall accept U1 entry requests by the link partner unless the hub has one or more packets/link commands to transmit on the port.	LVS
10.4.2.1#4	When U1_TIMEOUT = X > 0, U2_TIMEOUT = 0, if the U1 timeout X is not 0xFF but > 0 and the U1 timer reaches X, the port’s link shall initiate a transition to U1.	LVS
10.4.2.1#5	When U1_TIMEOUT = 0, the port's link shall reject all U1 transition requests by the link partner.	LVS
10.4.2.1#6	When U2_TIMEOUT = Y > 0, the port’s link shall accept U2 entry requests by its link partner unless the hub has one or more packets/link commands to transmit on the port.	LVS
10.4.2.1#7	If the U2 timeout is 0xFF, the port shall be disabled from initiating U2 entry but shall accept U2 entry requests by the link partner unless the hub has one or more packets/link commands to transmit on the port.	LVS
10.4.2.1#8	When U1_TIMEOUT = 0, U2_TIMEOUT = Y > 0, if the U2 timeout is not 0xFF and the U2 timer reaches Y, the port's link shall initiate a direct transition from U0 to U2. In this case, PORT_U2_TIMEOUT represents an amount of inactive time in U0.	LVS
10.4.2.1#9	A DFP shall transition to U0 states (depending on the U1 and U2 timeout values) from any U-state if the hub receives a SetPortFeature(PORT_LINK_STATE) U0 request.	LVS
10.4.2.1#10	Hub shall ensure that there is no race condition between a link partner initiating a U1/U2 request and transitioning to U0 based on SetPortFeature(PORT_LINK_STATE) U0 request.	LVS
10.4.2.1#11	A DFP shall transition to U0 states (depending on U1 and U2 timeout values) from U1 if the link partner successfully initiates a transition to U0.	Link Tests TD 7.18

Assertion #	Assertion Description	Test #
10.4.2.1#12	A DFP shall transition to U0 states (depending on the U1 and U2 timeout values) from U2 if the link partner successfully initiates a transition to U0.	Link Tests TD 7.19
10.4.2.1#13	A DFP shall transition to U0 states (depending on the U1 and U2 timeout values) from U1 if the hub successfully initiates a transition to U0 after receiving a packet routed to the port.	LVS
10.4.2.1#14	A DFP shall transition to U0 states (depending on the U1 and U2 timeout values) from U2 if the hub successfully initiates a transition to U0 after receiving a packet routed to the port.	LVS
Subsection reference: 10.4.2.2 Attempt U0 – U1 Transition		
10.4.2.2#1	A port shall transition to the U1 state if the hub receives a SetPortFeature(PORT_LINK_STATE) U1 request.	LVS
10.4.2.2#2	If the transition to U1 attempt fails, and transition is not result of SetPortFeature request, the port returns to the appropriate enabled U0 state.	Link Test TD 7.21
10.4.2.2#3	If the transition to U1 attempt fails and the state was entered due to a SetPortFeature request, the port continues to attempt the U0-U1 transition on its link.	Not being tested
Subsection reference: 10.4.2.3 Attempt U0 – U2 Transition		
10.4.2.3#1	A port shall attempt to transition to the U2 state if the hub receives a SetPortFeature(PORT_LINK_STATE) U2 request.	TD 10.17
10.4.2.3#2	If the transition to U2 attempt fails, and the transition is not the result of a SetPortFeature request, the port returns to the appropriate enabled U0 state.	Link Test TD 7.21.
10.4.2.3#3	If the transition to U2 attempt fails and the state was entered due to a SetPortFeature request, the port continues to attempt the U0-U2 transition on its link.	Not being tested
Subsection reference: 10.4.2.4 Link in U1		
10.4.2.4#1	Whenever a downstream port enters U1 and all downstream ports are now in U1 or a lower power state, the hub shall initiate a transition to U1 on the UFP within tHubPort2PortExitLat if the UFP is enabled for U1.	LVS
10.4.2.4#2	The U2 timer is reset to zero and started when the Link enters U1.	LVS
10.4.2.4#3	If the U2 timeout is not 0 or 0xFF and the U2 timer reaches Y, the DFP link shall initiate a direct transition from U1 to U2. In this case, PORT_U2_TIMEOUT represents an amount of time in U1.	TD 10.12

Assertion #	Assertion Description	Test #
10.4.2.4#4	Whenever a DFP or its link partner initiates a transition from U1 to one of the Enabled U0 states and the UFP is not in U0, the hub shall initiate a transition to U0 on the UFP within tHubPort2PortExitLat of when the transition was initiated on the DFP.	LVS
Subsection reference: 10.4.2.5 Link in U2		
10.4.2.5#1	Whenever a DFP enters U2 and all DFPs are now in U2 or a lower power state, the hub shall initiate a transition to U2 on the UFP within tHubPort2PortExitLat if the UFP is enabled for U2. If U2 is not enabled on the UFP, but U1 is enabled, the hub shall initiate a transition to U1 with the same time requirements.	LVS
10.4.2.5#2	Whenever the downstream port enters U2, if all the downstream ports are now in the U1 or the lower power state, the hub shall initiate a transition to U1 on the UFP within tHubPort2PortExitLat if the UFP is enabled for U1.	LVS
10.4.2.5#3	Whenever a DFP or its link partner initiates a transition from U2 to U0 and the hub UFP is in U2, the hub shall initiate a transition to U0 on the UFP within tHubPort2PortExitLat of when the transition was initiated on the DFP.	LVS
10.4.2.5#4	Hub shall ensure that there is no race condition between the link partner initiating a U0/U1/U2 transition and a U state timer expiring at the same time.	LPM tool.
10.4.2.5#5	Whenever a DFP or its link partner initiates a transition from U2 to U0 and the hub UFP is in U1, the hub shall initiate a transition to U0 on the UFP within tHubPort2PortExitLat + U2DevExitLat - U1DevExitLat of when the transition was initiated on the DFP.	LVS
Subsection reference: 10.4.2.6 Link in U3		
10.4.2.6#1	When a downstream port enters U3, if all downstream ports are now in the U2 or U3, the hub shall initiate a transition to the lowest enabled power state above U3 on the upstream port within tHubPort2PortExitLat.	LVS
10.4.2.6#2	When a downstream port enters U3, if all downstream ports are now in the U1 or lower power state, the hub shall initiate a transition to U1 on the upstream port within tHubPort2PortExitLat, if the upstream port is enabled for U1.	LVS
10.4.2.6#3	If the upstream port of the hub receives a packet that is routed to a downstream port that is in U3, the packet is silently discarded.	LVS
10.4.2.6#4	The hub shall perform normal link level acknowledgement of the header packet in case, the upstream port of the hub receives a packet that is routed to a downstream port that is in U3, the packet is silently discarded.	LVS
Subsection reference: 10.5 Hub Upstream Facing Port		
Subsection reference: 10.5.1 Upstream Facing Port State Descriptions		

Assertion #	Assertion Description	Test #
Subsection reference: 10.5.1.1 USPORT.Powered-off		
10.5.1.1#1	A port shall transition into USPORT_Powered-off from any state when VBUS is removed.	Untestable
10.5.1.1#2	A port shall transition into USPORT_Powered-off from any state if far-end receiver terminations are not detected.	Untestable
10.5.1.1#3	A port shall transition into USPORT_Powered-off from the USPORT.Connected state if the port configuration process fails.	Untestable
10.5.1.1#4	The default state for an upstream facing port shall be USPORT.Powered-off.	Untestable
Subsection reference: 10.5.1.2 USPORT.Powered-on		
10.5.1.2#1	The port shall transition from USPORT.Connected/Enabled state into USPORT.Power-On state when the link receives a Warm Reset.	LVS
10.5.1.2#2	The port shall transition from USPORT.Powered-off to USPORT.Powered-on when VBUS becomes valid.	Untestable
10.5.1.2#3	The port shall transition from USPORT.Error to USPORT.Powered-on when the link receives a warm reset.	Untestable
10.5.1.2#4	The port shall transition from USPORT.Training to USPORT.Powered-on if the port’s link times out from any polling substate.	Untestable
10.5.1.2#5	While in USPORT.Powered-on, the port’s link shall be in the Rx.Detect state.	LVS
10.5.1.2#6	While in USPORT.Powered-on, if the USB 2.0 portion of the hub enters the suspended state, the total hub current draw from VBUS shall meet the suspend current limit.	Electrical Tests
10.5.1.2#7	The port shall transition from USPORT.Error to USPORT.Powered-on when Far-end Terminations are removed.	Untestable
10.5.1.2#8	The port shall transition from USPORT.Training to USPORT.Powered-on if the port receives a Warm (LFPS) Reset.	Untestable
Subsection reference: 10.5.1.3 USPORT.Training		
10.5.1.3#1	A port shall transition to the USPort.Training state from the USPort.Powered-on state when SuperSpeed far-end receiver terminations are detected.	Implicit

Assertion #	Assertion Description	Test #
10.5.1.3#2	A port in the USPort.Training state shall have its port link in the Polling state.	Implicit
Subsection reference: 10.5.1.4 USPORT.Connected/Enabled		
10.5.1.4#1	In USPORT.Connect/Enabled state, the port's link shall be in the U0, U1, U2, U3 or Recovery states.	Implicit
10.5.1.4#2	The port shall remain in USPORT.Connected/Enabled state when the link receives a Hot Reset.	Untestable
10.5.1.4#3	The link shall be in U0 when the USPORT.Connected/Enabled state is entered.	Untestable
10.5.1.4#4	When the link enters USPORT.Connected/Enabled state, the port shall start the port configuration process.	LVS
Subsection reference: 10.5.1.5 USPORT.Error		
10.5.1.5#1	A port transitions to USPORT.Error state from the USPORT.Connected state if the link enters Recovery and times out without recovering.	Untestable
10.5.1.5#2	A port transitions to USPORT.Error state from the USPORT.Enabled state if the link enters Recovery and times out without recovering.	Untestable
10.5.1.5#3	A port shall exit the Error state only if a Warm Reset is received on the link.	Untestable
10.5.1.5#4	A port shall exit the USPORT.Error state if Far-end Receiver Terminations are removed.	Untestable
Subsection reference: 10.6 Upstream Facing Port Power Management		
Subsection reference: 10.6.1 Upstream Facing Port PM Timer		
10.6.1#1	The U2 inactivity timeout is set when a U2 Inactivity Timeout LMP is received.	LVS
10.6.1#2	The PM timer is reset when the hub upstream port link enters U1.	LVS
Subsection reference: 10.6.2 Hub Upstream Facing Port State Descriptions		
Subsection reference: 10.6.2.1 Enabled U0 States		
10.6.2.1#1	The upstream port shall accept U1 or U2 transitions from the link partner if the Force_LinkPM_Accept bit is set to one (refer to Section 8.4.2).	Chapter 9 TD 9.24. LVS
10.6.2.1#2	In the Enabled U0-only state the PM Timer values shall be ignored.	Implicit

Assertion #	Assertion Description	Test #
10.6.2.1#3	In the Enabled U0-only state the Upstream port's link shall accept U1 entry requests by its link partner unless the hub has one or more packets/link commands to transmit on the port or one or more of the hub downstream ports are in U0 or Recovery.	LVS
10.6.2.1#4	In the Enabled U0-only state the UFP port's link shall accept U2 entry requests by its link partner unless the hub has one or more packets/link commands to transmit on the port or one or more of the hub downstream ports has a link in U0, U1,or recovery.	LVS
10.6.2.1#5	When U1_ENABLE = 0 the Upstream port's link shall not attempt to initiate transitions to U1.	LVS
10.6.2.1#6	When U2_ENABLE = 0 the UFP port's link shall not initiate a U2 transition.	LVS
10.6.2.1#7	When U2_ENABLE = 1 the Port's link shall initiate a transition to U2 if all the hub downstream ports are in U2 or a low power state.	LVS
10.6.2.1#8	A U1 entry request shall not be accepted if one or more of the hub downstream ports have a link in U0 or recovery.	LVS
10.6.2.1#9	A U2 entry request shall not be accepted if one or more of the hub downstream ports have a link in U0, U1 or recovery.	LVS
10.6.2.1#10	When U1_ENABLE=1, the port's link shall initiate a transition to U1 if all the hub downstream ports are in U1 or a lower power state unless the requirements to enter U2 are met.	LVS
10.6.2.1#11	A UFP transitions to one of the Enabled U0 states (depending on the U1 and U2 Enable values) from U1 if the link partner successfully initiates a transition to U0.	LVS TD 10.10
10.6.2.1#12	A UFP transitions to one of the Enabled U0 states (depending on the U1 and U2 Enable values) from U2 if the link partner successfully initiates a transition to U0.	LVS TD 10.10
10.6.2.1#13	A UFP transitions to one of the Enabled U0 states (depending on the U1 and U2 Enable values) from U1 if there is a status change on a downstream port.	LVS TD 10.10
10.6.2.1#14	A UFP transitions to one of the Enabled U0 states (depending on the U1 and U2 Enable values) from U2 if there is a status change on a downstream port.	LVS TD 10.10
10.6.2.1#15	A UFP transitions to one of the Enabled U0 states (depending on the U1 and U2 Enable values) from U1 if a hub downstream port's link initiates a transition to U0.	LVS TD 10.10

Assertion #	Assertion Description	Test #
10.6.2.1#16	A UFP transitions to one of the Enabled U0 states (depending on the U1 and U2 Enable values) from U2 if a hub downstream port's link initiates a transition to U0.	LVS TD 10.10
10.6.2.1#17	A UFP transitions to one of the U0 Enabled state if there was an attempt to transition from the U0 to the U1 state and if the upstream port's link partner rejects the transition attempt.	TD 10.10 LVS
10.6.2.1#18	A UFP transitions to one of the U0 Enabled state if there was an attempt to transition from the U0 to the U2 state and if the upstream port's link partner rejects the transition attempt	LVS TD 10.10
10.6.2.1#19	A UFP transitions to one of the Enabled U0 states (depending on the U1 and U2 Enable values) from U3 if the upstream port of the hub receives wakeup signaling.	LVS TD 10.10
10.6.2.1#20	A UFP transitions to one of the Enabled U0 states (depending on the U1 and U2 Enable values) from U3 if there is a status change on a downstream port or a local power status change and remote wakeup is enabled for the corresponding event type.	LVS
10.6.2.1#21	The port's upstream link shall initiate a transition to U2 if all the hub downstream ports are in U2 or a lower link state. Note that if the port is already in U1, then the port shall transition to U0 before transitioning to U2.	LVS
Subsection reference: 10.6.2.2 Attempt U0 - U1 Transition		
10.6.2.2#1	If the UFP link transition from U0 to U1 fails (an LXU is received or the link goes to recovery), the port returns to the appropriate enabled U0 state.	LVS TD 10.10
Subsection reference: 10.6.2.3 Attempt U0 - U2 Transition		
10.6.2.3#1	If the UFP link transition from U0 to U2 fails (an LXU is received or the link goes to recovery), the port returns to the appropriate enabled U0 state.	LVS TD 10.10
Subsection reference: 10.6.2.4 Link in U1		
10.6.2.4#1	A UFP transitions to U1 after sending an LAU to accept a transition initiated by the link partner.	LVS
10.6.2.4#2	A UFP transitions to U1 after receiving an LAU from the link partner after initiating an attempt to transition the link to U1.	LVS
10.6.2.4#3	If the U2 inactivity timeout is not 0xFF or 0x00, and the PM timer reaches the U2 inactivity timeout, the upstream port's link shall initiate a transition from U1 to U2.	LVS
Subsection reference: 10.6.2.5 Link in U2		
10.6.2.5#1	A UFP transitions to U2 after sending an LAU to accept a transition initiated by the link partner.	LVS

Assertion #	Assertion Description	Test #
10.6.2.5#2	A UFP transitions to U2 after receiving an LAU from the link partner after initiating an attempt to transition the link to U2.	LVS
Subsection reference: 10.6.2.6 Link in U3		
10.6.2.6#1	A port transitions to U3 after sending an LAU to accept a transition initiated by the link partner.	LVS
10.6.2.6#2	If the hub UFP gets a request to go into U3, when one of its DFP port is in U0 and remote wake is not enabled for connect, the hub shall initiate U3 on that DFP.	LVS
Subsection reference: 10.7 SuperSpeed Hub Header Packet Forwarding and Data Repeater		
Subsection reference: 10.7.3 Interpacket Spacing		
10.7.3#1	When a hub forwards a header packet downstream and the downstream port link is in U0 when the header packet is received on the hub upstream port the propagation delay variation shall not be more than tPropagationDelayJitterLimit.	Untestable
Subsection reference: 10.7.4 SuperSpeed Header Packet Buffer Architecture		
10.7.4#1	A SuperSpeed hub starting with all header packet buffers empty shall be able to receive at least eight header packets directed to the same DFP that is not in U0 before its UFP runs out of credits.	LVS
10.7.4#2	A SuperSpeed hub that receives a header packet on its upstream port that is routed to a downstream port shall immediately route the header packet to the appropriate downstream port header packet buffer (if space in that buffer is available) regardless of the state of any other downstream port header packet buffers or the state of the upstream port Rx header packet buffer.	Interoperability
10.7.4#3	A SuperSpeed hub starting with all header packet buffers empty shall be able to receive at least eight header packets on the same downstream port directed for upstream transmission when the upstream port is not in U0.	Untestable
10.7.4#4	Header packets transmitted by a downstream port shall be transmitted in the order they were received on the upstream port.	Implicit
10.7.4#5	Header packets transmitted by an upstream port from the same downstream port shall be transmitted in the order they were received on that downstream port.	Implicit
10.7.4#6	A SuperSpeed hub shall have at least 1080 bytes of buffering for data packets received on the upstream port.	Implicit
10.7.4#7	A SuperSpeed hub shall have at least 1080 bytes of shared buffering for data packets received on all downstream ports.	Untestable

Assertion #	Assertion Description	Test #
Subsection reference: 10.7.5 SuperSpeed Packet Connectivity		
10.7.5#1	The SuperSpeed hub packet repeater/forwarder must reclock the packets in both directions.	Implicit
Subsection reference: 10.8 SuperSpeedPlus Store and Forward Behavior		
Subsection reference: 10.8.4 Upstream Flowing Buffering		
10.8.4#1	The SuperSpeedPlus hub shall provide buffering for 16 x 1KB Control/Bulk DPP buffers and 16 x 1KB Interrupt/Isochronous DPP buffers for each DFP receiver.	Untestable
10.8.4#2	The SuperSpeedPlus hub shall provide buffering for 16 x Control/Bulk header buffers and 16 TP/Interrupt/Isochronous header buffers per DFP receiver.	Untestable
10.8.4#3	A SuperSpeedPlus hub shall provide buffering for each downstream port.	Implicit
Subsection reference: 10.8.5 Downstream Flowing Buffering		
10.8.5#1	The SuperSpeedPlus hub shall provide enough buffering depending on the speed and number of lanes on the upstream port and the number of downstream ports. The number of buffers shall be calculated according to steps 1-3 in section 10.8.5 of the USB 3.2 specification.	Untestable
10.8.5#2	The SuperSpeedPlus hub shall provide sufficient buffering for Control/Bulk header buffers and TP/Interrupt/Isochronous header buffers per hub.	Untestable
10.8.5#3	Buffering for downstream flowing traffic shall allow for transmitting packets in a different order than they were received.	Implicit
Subsection reference: 10.7.6 SuperSpeedPlus Hub Arbitration of Packets		
Subsection reference: 10.8.6.1 Arbitration Weight		
10.8.6.1#1	The arbitration weight (AW) of DFP I shall be set to DFPi.link_speed / ArbitrationWeightBase.	Not Tested
Subsection reference: 10.8.6.2 Direction Independent Packet Selection		
10.8.6.2#1	The SuperSpeedPlus hub must select which packet to transmit next when there are multiple packets buffered.	Implicit
10.8.6.2#2	A TP shall only be considered for transmission after it has been fully received and validated.	Untestable
10.8.6.2#3	A buffered TP shall be selected for transmission before any buffered DPs.	LVS - Not Tested

Assertion #	Assertion Description	Test #
10.8.6.2#4	TPs shall be selected for transmission in the order they were buffered for a port (e.g. FIFO).	LVS - Not Tested
10.8.6.2#5	Buffered Interrupt or Isochronous DPs shall be selected for transmission before any buffered Control or Bulk DPs.	LVS - Not Tested
10.8.6.2#6	Once a hub starts transmitting a packet on a port, it shall continue transmitting that packet until the packet transmission is complete.	LVS - Implicit
10.8.6.2#7	If a DP is being received on a port and the port to which it is to be routed has no other packets buffered nor has a packet currently being transmitted, the hub shall begin transmitting the packet on the destination port before the DP is fully received.	Not Tested
10.8.6.2#8	Transmission of a DP shall not begin until sufficient bytes have been received to avoid transmitter underrun.	Implicit
Subsection reference: 10.8.6.3 Downstream Flowing Packet Reception and Selection		
10.8.6.3#1	Buffered Isochronous and Interrupt DPs destined for the same DS port shall be transmitted in the same order as they were received on the upstream port.	LVS – Not Tested
Subsection reference: 10.8.6.4 Upstream Flowing Packet Reception and Selection		
10.8.6.4#1	Any fully buffered packet from DFPs are candidates for the next packet to transmit upstream.	Untestable
10.8.6.4#2	To select the next DP for transmission on the upstream port, the Upstream Controller shall use a weighted round robin arbitration behavior to select the next Control/BulkDP buffered from DFPs.	LVS – Not Tested
Subsection reference: 10.8.6.4.1 Partially Buffered DP Selection Candidate		
10.8.6.4.1#1	A partially received DP can be a candidate for the next DP transmitted upstream if the remaining time to receive the DP is less than the time required to transmit a buffered DP and a sufficient amount of the incoming DP has already been received.	Untestable
Subsection reference: 10.8.6.4.2 Upstream Weighted Round Robin Arbitration		
10.8.6.4.2#1	When the Upstream Controller needs to select the next Control/Bulk DP to transmit on the upstream port, the Upstream Controller will use the selectPacket() algorithm described in 10.8.6.4.2	LVS – Not Tested
Subsection reference: 10.8.7 SuperSpeedPlus Upstream Flowing Packet Modifications		

Assertion #	Assertion Description	Test #
10.8.7#1	When the upstream port of a SuperSpeedPlus hub is operating at greater than SuperSpeedPlus mode and the hub Downstream Controller receives a valid IN/ACK TP that is routed to a SuperSpeed mode DFP, the Downstream Controller shall save the transfer type of the TP for that DFPi.	Untestable
10.8.7#2	When the hub Downstream Controller receives a valid DPH packet the Downstream Controller shall set the transfer type of the DP to the saved value if the DPH is not a deferred DPH.	LVS – Not Tested
10.8.7#3	When a hub Downstream Controller receives a valid DPH packet the Downstream Controller shall modify the AW field of the received DPH by setting DPH.AW to DFPi.AW if the AW field of the DPH is zero.	LVS – Not Tested
10.8.7#4	When a Downstream Controller modifies a DPH, it shall recompute the CRC-16 for the DPH.	Implicit
10.8.7#5	If a hub modifies the AW of a packet because the received DPH AW was zero, the modification shall be done when the packet is received.	Untestable
10.8.7#6	When the hub Upstream Controller selects a Control/Bulk packet to transmit and there are multiple DFPis with buffered Control/Bulk DPs awaiting transmission, the Upstream Controller shall change the AW as described in 10.8.7.	LVS – Not Tested
10.8.7#7	If the hub Upstream Controller changes the AW of a Control/Bulk packet based on the SUM_AW, the modification shall be done before the packet is routed to the upstream port for transmission.	LVS – Not Tested
Subsection reference: 10.8.8 SuperSpeedPlus Downstream Controller		
10.8.8#1	The Downstream Controller shall be responsible for updating the ITP fields as described in Section 8.4.8.8 before forwarding the ITP on all downstream ports in U0.	LVS – Not Tested
Subsection reference: 10.9 Port State Machines		
Subsection reference: 10.9.1 Port Transmit State Machine		
Subsection reference: 10.9.2 Port Transmit State Descriptions		
Subsection reference: 10.9.2.1 Tx IDLE		
10.9.2.1#1	A UFP shall transition to Tx IDLE, if from the Tx Data, Tx Data Abort, or Tx Header state after packet transmission is completed.	Untestable
10.9.2.1#2	A UFP shall transition to Tx IDLE, from the Tx Link Command state after a link command is transmitted and there are no other link commands awaiting transmission.	Untestable

Assertion #	Assertion Description	Test #
10.9.2.1#3	A UFP shall transition to Tx IDLE, as the default state when the link enters U0.	Untestable
Subsection reference: 10.9.2.2 Tx Header		
10.9.2.2#1	A port shall transition to the Tx Header state from the Tx IDLE state when there are one or more header packets queued for transmission and there are no link commands queued for transmission.	Untestable
Subsection reference: 10.9.2.3 Tx Data		
10.9.2.3#1	A hub shall not retransmit a DPP packet under any circumstances.	Implicit
10.9.2.3#2	A port transmitter shall transition to the Tx Data state from the Tx Header state when there is a DPP associated with the DPH that was transmitted.	Untestable
10.9.2.3#3	The DPP transmission shall begin immediately after transmission of the last symbol of the DPH.	Implicit
Subsection reference: 10.9.2.4 Tx Data Abort		
10.9.2.4#1	In the Tx Data abort state, the port transmitter aborts the normal transmission of a DPP by performing speed specific abort processing.	Untestable
10.9.2.4#2	After the port transmitter aborts the DPP it removes the DPP from hub storage.	Untestable
10.9.2.4#3	In the case where the hub is simultaneously receiving a DPP into the hub and transmitting the same DPP out of the hub, an upstream port transmitter shall transition to the Tx Data Abort state from the Tx Data state when the downstream port receiving the DPP detects a speed specific abort indication.	Untestable
10.9.2.4#4	In the case where the hub is simultaneously receiving a DPP into the hub and transmitting the same DPP out of the hub, a downstream port transmitter shall transition to the Tx Data Abort state from the Tx Data state when the upstream port receiving the DPP detects a speed specific abort indication.	Untestable
Subsection reference: 10.9.2.5 Tx Link Command		
10.9.2.5#1	A port shall transition to the Tx Link Command state from the Tx IDLE state when there are one or more link commands queued for transmission.	Untestable
10.9.2.5#2	A port shall transition to the Tx Link Command state from the Tx Link Command state when there are additional link commands queued for transmission.	Untestable
Subsection reference: 10.9.4 Port Receive State Descriptions		

Assertion #	Assertion Description	Test #
Subsection reference: 10.9.4.1 Rx Default		
10.9.4.1#1	A port receiver shall transition to the Rx Default state from the Rx Data state when a speed specific end of packet or abort indication is received.	Untestable
10.9.4.1#2	A port shall transition to the Rx Default state from the Rx Header state when the last symbol in a header packet is received.	Untestable
10.9.4.1#3	A port shall transition to the Rx Default state after receiving a link command.	Untestable
10.9.4.1#4	A port shall transition to the Rx Default state as the default state when the link enters U0.	Untestable
Subsection reference: 10.9.4.2 Rx Data		
10.9.4.2#1	A port shall transition to the Rx Data state when it receives a speed specific indication of the end of a packet or the occurrence of an abort condition.	Untestable
10.9.4.2#2	When the port detects an error before the end of the DPP as defined in Section 7.2.4.1.6, it performs speed specific abort processing.	Untestable
10.9.4.2#3	In the case where the hub is simultaneously receiving a DPP into the hub and transmitting the same DPP out of the hub, the corresponding port transmitter shall be given an indication of the abort condition so that it can perform speed specific abort processing.	Untestable
10.9.4.2#4	If the DPP is not being actively transmitted out of the hub, for the upstream port receiver, the hub shall buffer a speed specific aborted DP for the appropriate downstream port.	Untestable
10.9.4.2#5	If the DPP is not being actively transmitted out of the hub, for a downstream port receiver, the hub shall buffer a speed specific aborted DP on the upstream port.	Untestable
Subsection reference: 10.9.4.3 Rx Header		
10.9.4.3#1	A port shall transition to the Rx Header state when it detects the speed specific beginning of a header packet.	Untestable
10.9.4.3#2	The port shall finish validating CRC-16, the Link Control Word CRC-5, and check the route string (if this is an upstream port) and header packet type within four symbol times after the last symbol of the header packet is received.	Untestable
Subsection reference: 10.9.4.4 Process Header Packet		
10.9.4.4#1	When the final symbol for a header packet is received, the port shall perform all processing necessary for the header packet.	Untestable

Assertion #	Assertion Description	Test #
10.9.4.4#2	Any header packet processing shall not block the port from immediately returning to the Rx Default state.	Untestable
10.9.4.4#3	When the last symbol of a header packet is received in the Rx header packet state and either the header packet CRC-16 or Link Control Word CRC-5 is determined to be invalid, the link layer won't pass the header packet to the hub.	Untestable
Subsection reference: 10.9.4.4.1 SuperSpeed Hub Upstream Facing Port		
10.9.4.4.1#1	If a non-ITP or non-PING header packet is routed to a downstream port that is in U1 or U2, the hub initiates U0 entry on the appropriate downstream port link.	Chapter 9 TD 9.25 and Deferred packet trace in Interop LVS
10.9.4.4.1#2	If a non-ITP or non-PING header packet is routed to a downstream port that is in U1 or U2, U0 entry shall be initiated in tDownLinkStateChange from when the hub received the first symbol of the header packet.	LVS
10.9.4.4.1#3	If a deferred header packet is a DPH, the corresponding DPP is silently discarded.	Chapter 9 TD 9.25 and Deferred packet trace in Interop
10.9.4.4.1#4	If a non-ITP or non-PING header packet is routed to a downstream port that is in U1 or U2, and the header packet is not already marked as deferred, a copy of the header packet is modified to include the hub's hub depth, marked as deferred with the Link Control Word CRC-5 recalculated is queued for transmission on the upstream port.	Chapter 9 TD 9.25 and Deferred packet trace in Interop LVS
10.9.4.4.1#5	If a non-ITP or non-PING header packet is routed to a downstream port that is in U1 or U2, the deferred packet header is queued for transmission on the appropriate downstream port.	Chapter 9 TD 9.25 and Deferred packet trace in InteropLVS
10.9.4.4.1#6	If a PING header packet is routed to a downstream port that is in U1 or U2, the hub initiates U0 entry on the appropriate downstream link.	Chapter 9 TD 9.25
10.9.4.4.1#7	If a PING header packet is routed to a downstream port that is in U1 or U2, U0 entry shall be initiated no later than tDownLinkStateChange from when the hub received the first symbol of the header packet.	LVS – Not Tested
10.9.4.4.1#8	If a PING header packet is routed to a downstream port that is in U0, U1, U2 or Recovery, the header packet is queued for transmission on the appropriate downstream port.	Implicit

Assertion #	Assertion Description	Test #
10.9.4.4.1#9	If a non-ITP or non-PING header packet is routed to a downstream port that is in U0 or Recovery, if the downstream port Tx header packet buffer queue is not empty or no link credit is available, the header packet is marked as delayed and the Link Control Word CRC05 is recalculated for the modified header packet.	Untestable
10.9.4.4.1#10	If a non-ITP or non-PING header packet is routed to a downstream port that is in U0 or Recovery, the header packet is queued for transmission on the appropriate downstream port.	Implicit
10.9.4.4.1#11	ITP header packets are silently discarded for any downstream port with a link not in U0 and not in Recovery.	Implicit
10.9.4.4.1#12	An ITP for a downstream port shall have its Delay and Correction fields updated to account for the measured delay of propagating the ITP through the hub.	Not Tested
10.9.4.4.1#13	If the delay introduced by the hub to an ITP header packet exceeds the tPropagation DelayJitterLimit, then the header packet shall be marked Delayed and the correct Link Control Word CRC-5 is re-calculated for the modified header.	Untestable
10.9.4.4.1#14	If the delay introduced by the hub to an ITP header packet causes the Delay subfield to overflow, the ITP shall not be queued for downstream transmission.	Implicit / Untestable
10.9.4.4.1#15	An ITP header packet that does not have an overflowed Delta subfield shall be queued for transmission on each downstream port that has completed Port Configuration and is in U0 or in Recovery.	Implicit
10.9.4.4.1#16	If a header packet is routed to a disabled or nonexistent DS port or to a DS port that is not in U0, U1, U2 or Recovery, the header packet is removed from the RX header packet queue.	Implicit
10.9.4.4.1#17	If a header packet is routed to a disabled or nonexistent DS port or to a DS port that is not in U0,U1, U2 or Recovery, the header packet is silently discarded.	Implicit
10.9.4.4.1#18	If a header packet is routed to a disabled or nonexistent DS port or to a DS port that is not in U0, U1, U2 or Recovery and the header packet is a DPH, the corresponding DPP is silently discarded.	Implicit
10.9.4.4.1#19	If a header packet is routed to the hub controller, the header packet is processed by the hub controller.	Implicit
10.9.4.4.1#20	If a header packet is routed to the hub controller, the header packet is removed from the RX header packet queue.	Untestable
10.9.4.4.1#21	If a header packet is routed to the hub controller, a response to the header packet is queued for transmission on the upstream port, if required.	Implicit

Assertion #	Assertion Description	Test #
Subsection reference: 10.9.4.4.2 SuperSpeedPlus Hub Upstream Facing Port		
10.9.4.4.2#1	If a non-ITP or non-PING header packet is routed to a downstream port that is in U1 or U2, the hub initiates U0 entry on the appropriate downstream port link.	LVS
10.9.4.4.2#2	If a non-ITP or non-PING header packet is routed to a downstream port that is in U1 or U2, U0 entry shall be initiated in tDownLinkStateChange from when the hub received the first symbol of the header packet.	LVS
10.9.4.4.2#3	If a non-ITP or non-PING header packet is routed to a downstream port that is in U1 or U2, and the header packet is not already deferred, the header packet is marked deferred and the Link Control Word CRC-5 is recalculated.	Chapter 9 TD 9.25 and Deferred packet trace in Interop
10.9.4.4.2#4	If a deferred header packet is a DPH, the corresponding DPP is silently discarded.	Chapter 9 TD 9.25 and Deferred packet trace in Interop LVS
10.9.4.4.2#5	If a non-ITP or non-PING header packet is routed to a downstream port that is in U1 or U2, and the header packet is not already marked as deferred, a copy of the header packet is modified to include the hub's hub depth, marked as deferred with the Link Control Word CRC-5 recalculated is queued for transmission on the upstream port.	Chapter 9 TD 9.25 and Deferred packet trace in Interop LVS
10.9.4.4.2#6	If a non-ITP or non-PING header packet is routed to a downstream port that is in U1 or U2, the deferred packet header is buffered awaiting arbitration for transmission on the appropriate downstream port.	Chapter 9 TD 9.25 and Deferred packet trace in Interop LVS
10.9.4.4.2#7	If a PING header packet is routed to a downstream port that is in U1 or U2, the hub initiates U0 entry on the appropriate downstream link.	Chapter 9 TD 9.25
10.9.4.4.2#8	If a PING header packet is routed to a downstream port that is in U1 or U2, U0 entry shall be initiated no later than tDownLinkStateChange from when the hub received the first symbol of the header packet.	Not Tested
10.9.4.4.2#9	If a PING header packet is routed to a downstream port that is in U0, U1, U2 or Recovery, the header packet is buffered awaiting arbitration for transmission on the appropriate downstream port.	Implicit
10.9.4.4.2#10	If a non-ITP or non-PING header packet is routed to a downstream port that is in U0 or Recovery, if the downstream port Tx header packet buffer queue is not empty or no link credit is available, the header packet is marked as delayed and the Link Control Word CRC-5 is recalculated for the modified header packet.	Untestable

Assertion #	Assertion Description	Test #
10.9.4.4.2#11	If a non-ITP or non-PING header packet is routed to a downstream port that is in U0 or Recovery, the header packet is buffered awaiting arbitration for transmission on the appropriate downstream port.	Untestable / Implicit
10.9.4.4.2#12	ITP header packets are silently discarded for any downstream port with a link not in U0 and not in Recovery.	Implicit
10.9.4.4.2#13	An ITP for a downstream port shall have its Delay and Correction fields updated to account for the measured delay of propagating the ITP through the hub.	LVS – Not Tested
10.9.4.4.2#14	If the delay introduced by the hub to an ITP header packet exceeds the tPropagation DelayJitterLimit, then the header packet shall be marked Delayed and the correct Link Control Word CRC-5 is re-calculated for the modified header.	Untestable
10.9.4.4.2#15	If the delay introduced by the hub to an ITP header packet causes the Delay subfield to overflow, the ITP shall not be queued for downstream transmission.	Implicit
10.9.4.4.2#16	An ITP header packet that does not have an overflowed Delta subfield shall be buffered awaiting arbitration for transmission on each downstream port that has completed Port Configuration and is in U0 or in Recovery.	Implicit
10.9.4.4.2#17	If a header packet is routed to a disabled or nonexistent DS port or to a DS port that is not in U0, U1, U2 or Recovery, the header packet is removed from the Upstream Receive buffer.	Implicit
10.9.4.4.2#18	If a header packet is routed to a disabled or nonexistent DS port or to a DS port that is not in U0, U1, U2 or Recovery, the header packet is silently discarded.	Implicit
10.9.4.4.2#19	If a header packet is routed to a disabled or nonexistent DS port or to a DS port that is not in U0, U1, U2 or Recovery and the header packet is a DPH, the corresponding DPP is silently discarded.	Implicit
10.9.4.4.2#20	If the downstream port to which the packet is being routed is operating in SuperSpeed mode and the header packet is a valid IN/ACK, save the transfer type of the IN/ACK.	Untestable
10.9.4.4.2#21	A DFP.SAVE_TT is preserved until the next IN/ACK is received that is routed to the same downstream port.	Untestable
10.9.4.4.2#22	If a header packet is routed to the hub controller, the header packet is processed by the hub controller.	Implicit
10.9.4.4.2#23	If a header packet is routed to the hub controller, the header packet is removed from the Upstream Receive buffer.	Untestable
10.9.4.4.2#24	If a header packet is routed to the hub controller, a response to the header packet is buffered awaiting arbitration for transmission on the upstream port, if required.	Implicit

Assertion #	Assertion Description	Test #
Subsection reference: 10.9.4.4.3 SuperSpeed Hub Downstream Facing Port		
10.9.4.4.3#1	If a header packet is queued for transmission on the upstream port and the queue is full, the header packet is queued as soon as space is available in the upstream port queue.	Untestable
10.9.4.4.3#2	If a header packet is queued for transmission on the upstream port the hub shall process subsequent header packets while the upstream port queue is full.	Untestable
10.9.4.4.3#3	If header packets have been received on more than one downstream port or are queued to be sent by the hub controller when a space becomes available in the upstream port header packet queue, the hub shall prioritize a non-data packet header over a data packet header packet if one is waiting at the front of a downstream queue or from the hub controller.	Untestable
Subsection reference: 10.9.4.4.4 SuperSpeedPlus Hub Downstream Facing Port		
10.9.4.4.4#1	If a valid DP is received and the port is operating in SuperSpeed mode then set the transfer type of the DP to the value of DFP.SAVE_TT.	Untestable
10.9.4.4.4#2	If a valid DP is received and the transfer type is asynchronous and the AW field is zero, modify the AW field of the received DPH by setting DPH.AW field to DFP.AW.	LVS – Not Tested
Subsection reference: 10.9.4.5 Rx Link Command		
10.9.4.5#1	A port shall transition to the Rx Link Command state when it receives a valid speed specific indication of the beginning of a link command.	Untestable
Subsection reference: 10.9.4.6 Process Link Command		
10.9.4.6#1	Once the link command is received, the port shall perform all additional processing necessary for the link command and any such processing shall not block the port from immediately returning to the Rx Default state.	Untestable
Subsection reference: 10.10 Suspend and Resume		
10.10#1	Enhanced SuperSpeed hubs only support selective suspend and resume and do not support global suspend and resume. A suspend or resume action will only affect the port to which it is sent.	LVS

Assertion #	Assertion Description	Test #
10.10#2	When a hub downstream port link is in the U3 state, and the downstream port receives wakeup signaling from its link partner, and the hub upstream port's link is not in U3, the hub shall drive remote wakeup signaling on the downstream link where the wakeup signaling was received in tHubDriveRemoteWakeDownstream.	LVS
10.10#3	When a hub downstream port link is in the U3 state, and the downstream port receives wakeup signaling from its link partner, and the hub upstream port's link is also in U3, the hub shall drive wakeup signaling on its upstream port in tHubPropRemoteWakeUpstream.	LVS
10.10#4	When a hub upstream port's link is in the U3 state and it receives wakeup signaling from its link partner on the hub upstream port's link, the hub shall automatically drive remote wakeup to any downstream ports that are in U3 and have received remote wakeup signaling since entering U3.	LVS
10.10#5	When a hub receives wakeup signaling from its link partner on the hub upstream port's link, the hub shall not drive remote wakeup to any downstream ports that are in U3 and have not received remote wakeup signaling since entering U3.	LVS
10.10#6	When the hub receives a SetPortFeature(PORT_LINK_STATE) U0 for a downstream port with a link in U3, the hub shall drive remote wakeup signaling on the link in tHubDriveRemoteWakeDownstream.	LVS
10.10#7	When the hub receives a SetPortFeature(PORT_LINK_STATE) U0 for a downstream port with a link in U3, the hub shall not drive remote wakeup signaling on any other links , other than the one for which the request was sent.	LVS
10.10#8	When a hub upstream port's link enters the U3 state and one of its downstream links is in U0/U1/U2/Recovery and has received a remote wake, the hub shall automatically drive remote wakeup on the upstream port in tHubPropRemoteWakUpstream.	LVS
10.10#9	When a hub downstream port link is in the U3 state and it receives wakeup signaling from its link partner on that downstream port, if the hub upstream port is in the process of entering U3, the hub shall wait until the U3 entry is completed, before driving wakeup signaling on its upstream port in tHubPropRemoteWakeUpstream.	Will be tested when we have an updated compliance device LVS
10.10#10	If the hub upstream port's link is in U3, the hub shall drive wakeup signaling on its upstream port due to connect (when the downstream port enters DSPORT.Enabled) if the hub is enabled for remote wakeup.	LVS
10.10#11	If the hub upstream port's link is in U3, the hub shall drive wakeup signaling on its upstream port due to disconnect if the hub is enabled for remote wakeup.	LVS

Assertion #	Assertion Description	Test #
10.10#12	If the hub upstream port's link is in U3, the hub shall drive wakeup signaling on its upstream port due to Over-current events if the hub is enabled for remote wakeup.	LVS
Subsection reference: 10.11 Hub Upstream Port Reset Behavior		
10.11#1	A suspended hub shall interpret the start of reset as a wakeup event.	LVS
10.11#2	A suspended hub that receives a reset shall be awake and have completed its reset sequence by the end of reset signaling.	LVS
10.11#3	After completion of a Warm Reset on the UFP, the entire hub returns to the default state, clearing status bits.	LVS
10.11#4	After completion of a Hot Reset on the UFP, the hub returns to the default state except port configuration information is maintained for the upstream port, clearing status bits.	LVS
Subsection reference: 10.12 Hub Port Power Control		
10.12#1	A hub shall have individual power switches for all USB Type-C ports	
10.12#2	If a hub supports ganged power switching, then the power to all ports in a gang is turned on when power is required to be on for any port in the gang.	TD 10.11
10.12#3	The power to a gang is not turned off unless all ports in a gang are in a state that allows power to be removed as specified in Table 10-2.	Untestable
10.12#4	The power to a port is not turned on by a SetPortFeature(PORT_POWER) if both C_HUB_LOCAL_POWER and Local Power Status (in wHubStatus) are set to one at the time when the request is executed.	CV
10.12#5	A self-powered hub shall implement power switching for all downstream USB Type-C ports.	
10.12#6	A hub shall support the Powered-off states for all ports.	LVS
Subsection reference: 10.12.1 Multiple Gangs (Only supported for downstream USB Standard-A ports)		
10.12.1#1	Multiple gangs shall only be supported for downstream USB Standard-A ports.	
10.12.1#2	A hub that implements more than one Over-current gang shall set the Over-current Reporting Mode to indicate that Over-current reporting is on a per-port basis.	TD 10.8

Assertion #	Assertion Description	Test #
10.12.1#3	A hub that implements more than one power switching gang shall set the Logical Power Switching Mode to indicate that power switching is on a per port basis.	TD 10.8
10.12.1#4	When an Over-current condition occurs on an Over-current protection device and the hub supports Individual Over-current Protection, the Over-current shall signal on all ports that are protected by that device by setting C_PORT_OVER_CURRENT to one.	TD 10.23
10.12.1#5	When Over-current is signaled, all the ports in the Over-current group are placed in the DSPORT.Powered-off or DSPORT.Powered-off-reset state.	TD 10.23
10.12.1#6	When Over-current is signaled, the C_PORT_OVER_CURRENT field is set to one on all ports.	TD 10.23
10.12.1#7	When the Over-current is signaled and the hub supports Individual Over-current Protection, the PORT_OVER_CURRENT field of any port in the group shall be set to one as long as the Over-current condition exists.	TD 10.23
10.12.1#8	The C_PORT_OVER_CURRENT field shall be cleared in each port individually.	Implicit in TD 10.23
10.12.1#9	When multiple ports share a power switch, setting PORT_POWER on any port in the group shall not cause the other ports in that group to leave the DSPORT.Powered-off or DSPORT.Powered-off-reset state.	TD 10.11
10.12.1#10	When all the ports in a group are in the DSPORT.Powered-off state the power to the ports is turned off.	Untestable
10.12.1#11	When the hub is not configured, the power to the ports is turned off.	LVS
10.12.1#12	If an Over-current condition occurs and power switches are present, then all power switches associated with an Over-current protection circuit shall be turned off.	Untestable
10.12.1#13	If multiple Over-current protection devices are associated with a single power switch, then that switch will be turned off when any of the Over-current protection circuits indicates an Over-current condition.	Untestable
Subsection reference: 10.13 Hub Controller		
Subsection reference: 10.13.1 Endpoint Organization		
10.13.1#1	The Status Change Endpoint of the hub shall have the maximum burst size set to one.	LVS

Assertion #	Assertion Description	Test #
10.13.1#2	If the hub and port status change bits are not set, then the hub shall return an NRDY when the Status Change endpoint receives an IN (via and ACK TP) request.	Implicit in Interop LVS TD 10.10
10.13.1#3	When a status change bit is set, the hub will send an ERDY TP to the host.	Implicit in Interop LVS
Subsection reference: 10.11.2 Hub Information Architecture and Operation		
10.13.2#1	When port status change bits are set, they shall remain set until cleared by a hub reset or by USB system software through a ClearPortFeature request.	LVS
10.13.2#2	When the hub status change bit is set, it shall remain set until cleared by a hub reset or by USB system software through a ClearHubFeature request.	TD 10.12 TD 10.16
Subsection reference: 10.13.3 Port Change Information Processing		
10.13.3#1	A hub shall not report the same port change event again after software has cleared the port change bit.	LVS
Subsection reference: 10.13.4 Hub and Port Status Change Bitmap		
10.13.4#1	A hub shall not report a port status change for a port number greater than (bNbrPorts).	LVS
10.13.4#2	A hub shall report a port status change by setting the corresponding bit of the Hub and Port Status Change Bitmap (as described in 10.13.4) to one.	LVS
10.13.4#3	A hub shall report a hub status change by setting bit 0 of the Hub and Port Status Change Bitmap to one.	LVS
10.13.4#4	When the hub indicates that there has been a port status change on a particular port, a GetPortStatus on that port shall return a status with at least one of the port status change bits set.	LVS
10.13.4#5	When the hub indicates that there has been a hub status change, a GetHubStatus shall return a status with at least one of the hub status change bits set.	LVS
10.13.4#6	When an interrupt transfer is sent to the hub and there is already a pending change, the hub should complete the interrupt transfer immediately.	Untestable
Subsection reference: 10.13.5 Over-current Reporting and Recovery		
10.13.5#1	A hub that supports one or more USB Type-C ports shall report over-current on a per port basis.	

Assertion #	Assertion Description	Test #
10.13.5#2	If the hub has Over-current detection on a hub basis, then an Over-current condition on the hub shall cause all ports to enter the DSPORT.Powered-off-reset state	TD 10.23
10.13.5#3	If the hub has Over-current detection on a hub basis, then an Over-current condition on the hub shall not set C_PORT_OVER_CURRENT or PORT_OVER_CURRENT for the affected ports.	TD 10.23
10.13.5#4	A self-powered hub shall implement Over-current protection.	TD 10.23
Subsection reference: 10.13.6 Enumeration Handling		
10.13.6#1	A hub shall respond to a Get_Status(PORT) request with a PORT_CONNECTION indication and with the PORT_SPEED field set to zero if the downstream facing port has an Enhanced SuperSpeed device connected.	LVS
10.13.6#2	When a device is attached, the hub must detect device attach event and report on the Status Change endpoint.	LVS
10.13.6#3	When a device is detached from the port, the hub port must report the Status Change through the Status Change endpoint.	LVS
Subsection reference: 10.14 Hub Configuration		
10.14#1	A hub shall not report both the MaxPower and the Self-Powered bit of bmAttributes of the Hub Configuration Descriptor set to 0.	TD 10.3 TD 10.12
10.14#2	If the MaxPower field of the Configuration Descriptor has a value of zero and the self-powered bit in the bmAttributes field of the Configuration Descriptor is one, the hub shall not report a Hub Device Status value of zero.	TD 10.3 TD 10.12
10.14#3	If the MaxPower field of the Configuration Descriptor has a value of zero and the self-powered bit in the bmAttributes field of the Configuration Descriptor is one, the hub shall only report a Hub Device Status of one when it is a self-powered only hub and the local power supply is good.	TD 10.3 TD 10.12
10.14#4	If the MaxPower field of the Configuration Descriptor has a value greater than zero and the self-powered bit in the bmAttributes field of the Configuration Descriptor is zero, the hub shall operate as a bus powered only hub.	TD 10.3 TD 10.12
10.14#5	If the MaxPower field of the Configuration Descriptor has a value > 0 and the self-powered bit in the bmAttributes field of the Configuration Descriptor = 1, the hub shall be capable of operating in both bus powered or self-powered modes.	TD 10.3 TD 10.12

Assertion #	Assertion Description	Test #
10.14#6	If the MaxPower field of the Configuration Descriptor has a value > 0, the self-powered bit in the bmAttributes field of the Configuration Descriptor = 1 and the Hub Device Status is 0, the hub shall currently only be capable of operating as a bus powered hub.	TD 10.3 TD 10.12
10.14#7	If the MaxPower field of the Configuration Descriptor has a value > 0, the self-powered bit in the bmAttributes field of the Configuration Descriptor = 1 and the Hub Device Status is one, the hub shall currently be capable of operating as a self-powered hub.	TD 10.3 TD 10.12
10.14#8	A Self Powered hub shall draw no more than one unit load from its upstream connection.	Current Measurement in the Interop spec
10.14#9	When power is removed from a self-powered hub, the hub shall remain in the Configured state, transition all ports to the Powered-off state and notify the host of a hub status change.	TD 10.12
10.14#10	While local power is off on a self-powered hub, all port status (except for PLS) and change information shall be reported as zero.	TD 10.12
10.14#11	While local power is off on a self-powered hub, a hub shall ignore all SetPortFeature requests.	TD 10.12
10.14#12	A hub that draws power from USB PD (via its upstream port) or from USB Type-C Current shall report its operating power using the PD Consumer Port Descriptor capability in its BOS Descriptor.	
10.14#13	A bus powered hub shall be able to supply all available hub port power with any split across all exposed downstream ports.	Current Measurement in Interop test spec
10.14#14	Bus powered hubs shall not provide any downstream power until they are configured.	Current Measurement in Interop test spec
10.14#15	A self-powered hub shall ensure that it is able to provide at least six unit loads for each exposed downstream port on the hub.	
10.14#16	A self-powered hub that is powered by a non-USB connector shall move to the Powered state on both the USB 2.0 and USB 3.2 hubs when external power is applied.	
10.14#17	A self-powered hub that is USB PD powered via its upstream port shall move to the Powered state if it gets enough power to act as a self-powered hub.	

Assertion #	Assertion Description	Test #
10.14#18	If a self-powered hub that is USB PD powered via its upstream port does not get enough power to act as a self-powered hub, then it shall use PD mechanisms to inform the system of a Capability mismatch with insufficient power.	
10.14#19	A self-powered hub that is USB PD powered via its downstream port shall move to the Powered state if it gets enough power to act as a self-powered hub.	
10.14#20	If a self-powered hub that is USB PD powered via its downstream port does not get enough power to act as a self-powered hub, it shall move the USB 2.0 hub, but not the USB 3.2 hub, to the Powered state.	
10.14#21	If a self-powered hub that is USB PD powered via its downstream port does not get enough power to act as a self-powered hub, it shall set C_HUB_LOCAL_POWER and set the local power source field in the Hub Status to 1.	
10.14#22	If a self-powered hub that is USB PD powered via its downstream port does not get enough power to act as a self-powered hub, it shall set the Downstream PD Capability Mismatch field in the Hub Status to 1.	
10.14#23	A self-powered hub that is powered by USB Type-C current via its upstream port shall move to the Powered state if it gets enough power to act as a self-powered hub.	
10.14#24	If a self-powered hub that is powered by USB Type-C current via its upstream port does not get enough power to act as a self-powered hub, it shall move the USB 2.0 hub, but not the USB 3.2 hub, to the Powered state.	
10.14#25	If a self-powered hub that is powered by USB Type-C current via its upstream port does not get enough power to act as a self-powered hub, it shall set C_HUB_LOCAL_POWER and set the local power source field in the Hub Status to 1.	
10.14#26	If a self-powered hub that is powered by USB Type-C current via its upstream port does not get enough power to act as a self-powered hub, it shall set the Insufficient USB Type-C current field in the Hub Status to 1.	
Subsection reference: 10.15 Descriptors		
10.15#1	A hub shall support the Get Descriptor (BOS) when operating at either the Gen X speed or at USB 2.0 speeds.	TD 10.7
Subsection reference: 10.15.1 Standard Descriptors for Hub Class		
10.15.1#1	The descriptor returned in response to a GetDescriptor(Device) request must have a bLength of 0x12 bytes.	TD 10.2

Assertion #	Assertion Description	Test #
10.15.1#2	The descriptor returned in response to a GetDescriptor(Device) request must return a value of 0x01 in the bDescriptorType field.	TD 10.2
10.15.1#3	The descriptor returned in response to a GetDescriptor(Device) request must return a value of 0x20 in the lower Byte of the bcdUSB field.	TD 10.2
10.15.1#4	The descriptor returned in response to a GetDescriptor(Device) request must return a value of 0x03 in the higher Byte of the bcdUSB field.	TD 10.2
10.15.1#5	The descriptor returned in response to a GetDescriptor(Device) request must return a value of HUB_CLASSCODE (0x09) in the bDeviceClass field.	TD 10.2
10.15.1#6	The descriptor returned in response to a GetDescriptor(Device) request must return a value of 0x00 in the bDeviceSubClass field.	TD 10.2
10.15.1#7	The descriptor returned in response to a GetDescriptor(Device) request must return a value of 0x03 in the bDeviceProtocol field.	TD 10.2
10.15.1#8	The descriptor returned in response to a GetDescriptor(Device) request must return a value of 0x09 in the bMaxPacketSize0 field.	TD 10.2
10.15.1#9	The descriptor returned in response to a GetDescriptor(Device) request must return a value of 0x01 in the bNumConfigurations field.	TD 10.2
10.15.1#10	The descriptor returned in response to a GetDescriptor(BOS) request must return a value of 0x05 in the bLength field.	TD 10.7
10.15.1#11	The descriptor returned in response to a GetDescriptor(BOS) request must return a value of 0x0F in the bDescriptorType field.	TD 10.7
	The descriptor returned in response to a GetDescriptor(BOS) request must return a value of 0x49 in the lower byte of the wTotalLength field if the hub is Gen 2-capable.	10.15.1#12 *(removed because, with the advent of PD, this field is no longer fixed for hubs.)*
	The descriptor returned in response to a GetDescriptor(BOS) request must return a value of 0x00 in the higher byte of the wTotalLength field.	10.15.1#13 *(removed because, with the advent of PD, this field is no longer fixed for hubs.)*

Assertion #	Assertion Description	Test #
	The descriptor returned in response to a GetDescriptor(BOS) request must return a value of 0x05 in the bNumDeviceCaps field if the hub is Gen 2-capable.	10.15.1#14 *(removed because, with the advent of PD, this field is no longer fixed for hubs.)*
10.15.1#15	The USB 2.0 Extension descriptor returned in response to a GetDescriptor(BOS) request must return value of 0x07 in the bLength field.	TD 10.7
10.15.1#16	The USB 2.0 Extension descriptor returned in response to a GetDescriptor(BOS) request must return value of 0x10 in the bDescriptorType field.	TD 10.7
10.15.1#17	The USB 2.0 Extension descriptor returned in response to a GetDescriptor(BOS) request must return value of 0x02 in the bDevCapabilityType field.	TD 10.7
10.15.1#18	The USB 2.0 Extension descriptor returned in response to a GetDescriptor(BOS) request must return value of 0x02 in the bmAttributes field.	TD 10.7
10.15.1#19	The SuperSpeed USB Device Capability descriptor returned in response to a GetDescriptor(BOS) request must return value of 0x0A in the bLength field.	TD 10.7
10.15.1#20	The SuperSpeed USB Device Capability descriptor returned in response to a GetDescriptor(BOS) request must return value of 0x10 in the bDescriptorType field.	TD 10.7
10.15.1#21	The SuperSpeed USB Device Capability descriptor returned in response to a GetDescriptor(BOS) request must return value of 0x03 in the bDevCapabilityType field.	TD 10.7
10.15.1#22	The SuperSpeed USB Device Capability descriptor returned in response to a GetDescriptor(BOS) request must return value of 0xE in the lower byte of wSpeedsSupported field.	TD 10.7
10.15.1#23	The SuperSpeed USB Device Capability descriptor returned in response to a GetDescriptor(BOS) request must return value of 0x00 in the higher byte of wSpeedsSupported field.	TD 10.7
10.15.1#24	The SuperSpeed USB Device Capability descriptor returned in response to a GetDescriptor(BOS) request must return value of 0x01 in the bFunctionalitySupport field.	TD 10.7
10.15.1#26	The container ID descriptor returned in response to a GetDescriptor(BOS) request must return value of 0x14 in the bLength field.	TD 10.7
10.15.1#27	The container ID descriptor returned in response to a GetDescriptor(BOS) request must return value of 0x10 in the bDescriptorType field.	TD 10.7

Assertion #	**Assertion Description**	**Test #**
10.15.1#28	The container ID descriptor returned in response to a GetDescriptor(BOS) request must return value of 0x04 in the bDevCapabilityType field.	TD 10.7
10.15.1#29	The container ID descriptor returned in response to a GetDescriptor(BOS) request must return value of 0x00 in the bReserved field.	TD 10.7
10.15.1#30	The descriptor returned in response to a GetDescriptor(Configuration) request must return a value of 0x09 in the bLength field.	TD 10.3
10.15.1#31	The descriptor returned in response to a GetDescriptor(Configuration) request must return a value of 0x02 in the bDescriptorType field.	TD 10.3
10.15.1#32	The descriptor returned in response to a GetDescriptor(Configuration) request must return a value of 0x1F in the lower byte of wTotalLength field.	TD 10.3
10.15.1#33	The descriptor returned in response to a GetDescriptor(Configuration) request must return a value of 0x00 in the higher byte of wTotalLength field.	TD 10.3
10.15.1#34	The descriptor returned in response to a GetDescriptor(Configuration) request must return a value of 0x01 in the bNumInterfaces field.	TD 10.3
10.15.1#35	The descriptor returned in response to a GetDescriptor(Configuration) request must have the value of maximum amount of bus power the hub will consume in this configuration in the bMaxPower field.	Current Measurement Testing
10.15.1#36	The Interface descriptor returned in response to a GetDescriptor(Configuration) request must return a value of 0x09 in the bLength field.	TD 10.4
10.15.1#37	The Interface descriptor returned in response to a GetDescriptor(Configuration) request must return a value of 0x04 in the bDescriptorType field.	TD 10.4
10.15.1#38	The Interface descriptor returned in response to a GetDescriptor(Configuration) request must return a value of 0x00 in the bInterfaceNumber field.	TD 10.4
10.15.1#39	The Interface descriptor returned in response to a GetDescriptor(Configuration) request must return a value of 0x00 in the bAlternateSetting field.	TD 10.4
10.15.1#40	The Interface descriptor returned in response to a GetDescriptor(Configuration) request must return a value of 0x01 in the bNumEndpoints field.	TD 10.4

Assertion #	Assertion Description	Test #
10.15.1#41	The Interface descriptor returned in response to a GetDescriptor(Configuration) request must return a value of HUB_CLASSCODE (0x09) in the bInterfaceClass field.	TD 10.4
10.15.1#42	The Interface descriptor returned in response to a GetDescriptor(Configuration) request must return a value of 0x00 in the bInterfaceSubClass field.	TD 10.4
10.15.1#43	The Interface descriptor returned in response to a GetDescriptor(Configuration) request must return a value of 0x00 in the bInterfaceProtocol field.	TD 10.4
10.15.1#44	The Endpoint descriptor returned in response to a GetDescriptor(Configuration) request must return a value of 0x07 in the bLength field.	TD 10.5
10.15.1#45	The Endpoint descriptor returned in response to a GetDescriptor(Configuration) request must return a value of 0x05 in the bDescriptorType field.	TD 10.5
10.15.1#46	The Endpoint descriptor returned in response to a GetDescriptor(Configuration) request must return a value of 1 in bit 7 of the bEndpointAddress field of the Endpoint Descriptor	TD 10.5
10.15.1#47	The Endpoint descriptor returned in response to a GetDescriptor(Configuration) request must return a value of 0x13 in the bmAttributes field.	TD 10.5
10.15.1#48	The Endpoint descriptor returned in response to a GetDescriptor(Configuration) request must return a value of 0x02 in the lower byte of wMaxPacketSize field.	TD 10.5
10.15.1#49	The Endpoint descriptor returned in response to a GetDescriptor(Configuration) request must return a value of 0x00 in the high byte of wMaxPacketSize field.	TD 10.5
10.15.1#50	The Endpoint descriptor returned in response to a GetDescriptor(Configuration) request must return a value of 0x08 in the bInterval field.	TD 10.5
10.15.1#51	The SuperSpeed Endpoint Companion descriptor returned in response to a GetDescriptor(Configuration) request must return a value of 0x06 in the bLength field.	4TD 10.6
10.15.1#52	The SuperSpeed Endpoint Companion descriptor returned in response to a GetDescriptor(Configuration) request must return a value of 0x30 in the bDescriptorType field.	TD 10.6
10.15.1#53	The SuperSpeed Endpoint Companion descriptor returned in response to a GetDescriptor(Configuration) request must return a value of 0x00 in the bMaxBurst field.	TD 10.6
10.15.1#54	The SuperSpeed Endpoint Companion descriptor returned in response to a GetDescriptor(Configuration) request must return a value of 0x00 in the bmAttributes field.	TD 10.6

Assertion #	**Assertion Description**	**Test #**
10.15.1#55	The SuperSpeed Endpoint Companion descriptor returned in response to a GetDescriptor(Configuration) request must return a value of 0x02 in the low byte of the wBytesPerInterval field.	4TD 10.6
10.15.1#56	The SuperSpeed Endpoint Companion descriptor returned in response to a GetDescriptor(Configuration) request must return a value of 0x00 in the high byte of the wBytesPerInterval field.	4TD 10.6
10.15.1#57	The descriptor returned in response to a GetDescriptor(BOS) request must return a value of 0x49 in the lower byte of the wTotalLength field if the hub is operating at Gen 2 speed.	TD 10.7
10.15.1#58	An Enhanced SuperSpeed hub operating at Gen 2 speed must have both a SuperSpeed USB Device Capability descriptor and a SuperSpeedPlus USB Device Capability descriptor.	TD 10.7
10.15.1#59	The SuperSpeedPlus USB Device Capability descriptor returned in response to a GetDescriptor(BOS) request must return value of 0x1C in the bLength field.	TD 10.7
10.15.1#60	The SuperSpeedPlus USB Device Capability descriptor returned in response to a GetDescriptor(BOS) request must return value of 0x10 in the bDescriptorType field.	TD 10.7
10.15.1#61	The SuperSpeedPlus USB Device Capability descriptor returned in response to a GetDescriptor(BOS) request must return value of 0x0A in the bDevCapabilityType field.	TD 10.7
10.15.1#62	The SuperSpeedPlus USB Device Capability descriptor returned in response to a GetDescriptor(BOS) request must return value of 0 in the bReserved field.	TD 10.7
10.15.1#63	The SuperSpeedPlus USB Device Capability descriptor returned in response to a GetDescriptor(BOS) request must return value of 0x23 in the bmAttributes field.	TD 10.7 *Note: the USB Spec has this mistakenly set to 0x13.*
	The SuperSpeedPlus USB Device Capability descriptor returned in response to a GetDescriptor(BOS) request must return value of 0x1100 in the bFunctionalitySupport field.	10.15.1#64 *(removed because, with the advent of PD, this field is no longer fixed for hubs.)*
	The SuperSpeedPlus USB Device Capability descriptor returned in response to a GetDescriptor(BOS) request must return value of 0x00050030 in the bmSublinksSpeedAttr[0] field.	10.15.1#65 *(removed because, with the advent of PD, this field is no longer fixed for hubs.)*

Assertion #	Assertion Description	Test #
	The SuperSpeedPlus USB Device Capability descriptor returned in response to a GetDescriptor(BOS) request must return value of 0x000500B0 in the bmSublinksSpeedAttr[1] field.	10.15.1#66 *(removed because, with the advent of PD, this field is no longer fixed for hubs.)*
	The SuperSpeedPlus USB Device Capability descriptor returned in response to a GetDescriptor(BOS) request must return value of 0x000A0031 in the bmSublinksSpeedAttr[2] field.	10.15.1#67 *(removed because, with the advent of PD, this field is no longer fixed for hubs.)*
	The SuperSpeedPlus USB Device Capability descriptor returned in response to a GetDescriptor(BOS) request must return value of 0x000A00B1 in the bmSublinksSpeedAttr[3] field.	10.15.1#68 *(removed because, with the advent of PD, this field is no longer fixed for hubs.)*
10.15.1#69	The Precision Time Measurement descriptor returned in response to a GetDescriptor(BOS) request must return value of 0x03 in the bLength field.	TD 10.7
10.15.1#70	The Precision Time Measurement descriptor returned in response to a GetDescriptor(BOS) request must return value of 0x10 in the bDescriptorType field.	TD 10.7
10.15.1#71	The Precision Time Measurement descriptor returned in response to a GetDescriptor(BOS) request must return value of 0x0B in the bDevCapabilityType field.	TD 10.7
	The descriptor returned in response to a GetDescriptor(BOS) request must return a value of 0x2a in the lower byte of the wTotalLength field if the hub is Gen 1-only capable and does not support Precision Time Management.	10.15.1#72 *(removed because, with the advent of PD, this field is no longer fixed for hubs.)*
	The descriptor returned in response to a GetDescriptor(BOS) request must return a value of 0x2d in the lower byte of the wTotalLength field if the hub is Gen 1-only capable and does support Precision Time Management.	10.15.1#73 *(removed because, with the advent of PD, this field is no longer fixed for hubs.)*
	The descriptor returned in response to a GetDescriptor(BOS) request must return a value of 0x03 in the bNumDeviceCaps field if the hub is Gen 1-only capable and does not support Precision Time Management.	10.15.1#74 *(removed because, with the advent of PD, this field is no longer fixed for hubs.)*

Assertion #	Assertion Description	Test #
	The descriptor returned in response to a GetDescriptor(BOS) request must return a value of 0x04 in the bNumDeviceCaps field if the hub is Gen 1-only capable and does support Precision Time Management.	10.15.1#75 *(removed because, with the advent of PD, this field is no longer fixed for hubs.)*
Subsection reference: 10.15.2 Class-specific Descriptors		
Subsection reference: 10.15.2.1 Hub Descriptor		
10.15.2.1#1	The descriptor returned in response to a GetHubDescriptor request must return the value of 0x0C in the bDescLength field.	TD 10.8
10.15.2.1#2	The descriptor returned in response to a GetHubDescriptor request must return the value of 0x2A in the bDescriptorType field.	TD 10.8
10.15.2.1#3	The descriptor returned in response to a GetHubDescriptor request must have the value of number of downstream facing ports that this hub supports in the bNbrPorts field.	TD 10.8 TD 10.15
10.15.2.1#4	Bit 1:0 in the wHubCharacteristics field in the descriptor returned in response to a GetHubDescriptor request must be 00 if the hub is ganged power switching.	TD 10.8 TD 10.11
10.15.2.1#5	Bit 1:0 in the wHubCharacteristics field in the descriptor returned in response to a GetHubDescriptor request must be 01 if the hub is individual port power switching.	TD 10.8 TD 10.11
10.15.2.1#6	Bit 1 in the wHubCharacteristics field in the descriptor returned in response to a GetHubDescriptor request is reserved and shall never be 1.	TD 10.8
10.15.2.1#7	Bit 2 in the wHubCharacteristics field in the descriptor returned in response to a GetHubDescriptor request must be 0 if the hub is not part of a compound device.	TD 10.8
10.15.2.1#8	Bit 2 in the wHubCharacteristics field in the descriptor returned in response to a GetHubDescriptor request must be 1 if the hub is part of a compound device.	TD 10.8
10.15.2.1#9	Bit 4:3 in the wHubCharacteristics field in the descriptor returned in response to a GetHubDescriptor request must be 00 if the hub provides global Over-current protection.	TD 10.8
10.15.2.1#10	Bit 4:3 in the wHubCharacteristics field in the descriptor returned in response to a GetHubDescriptor request must be 01 if the hub provides individual port Over-current protection.	TD 10.8
10.15.2.1#11	Bit 4 in the wHubCharacteristics field in the descriptor returned in response to a GetHubDescriptor request must be 1 if the hub does not provide Over-current protection, for bus powered hubs only.	TD 10.8

Assertion #	Assertion Description	Test #
10.15.2.1#12	Bits 15:5 in the wHubCharacteristics field in the descriptor returned in response to a GetHubDescriptor are reserved and must be 0.	TD 10.8
10.15.2.1#13	The descriptor returned in response to a GetHubDescriptor request must contain the time from the time the power-on sequence begins on a port until power is good on that port in the bPwrOn2PwrGood field.	TD 10.8
10.15.2.1#14	The descriptor returned in response to a GetHubDescriptor request must set to zero in the bPwrOn2PwrGood field if the hub does not support power-switching.	TD 10.8 TD 10.11
10.15.2.1#15	The descriptor returned in response to a GetHubDescriptor request must have the Maximum current requirements of the Hub Controller electronics when the hub is operating on both USB 2.0 and Enhanced SuperSpeed in bHubContrCurrent field.	Current Measurement
10.15.2.1#16	The descriptor returned in response to a GetHubDescriptor request must contain hub packet header decode latency in bHubHdrDecLat field.	LVS
10.15.2.1#17	The descriptor returned in response to a GetHubDescriptor request must have the bHubHdrDecLat field set to a value between 0 and 0x0AH	TD 10.8
10.15.2.1#18	The descriptor returned in response to a GetHubDescriptor request must contain the maximum delay that the hub introduces while forwarding packets in wHubDelay.	LVS
10.15.2.1#19	The descriptor returned in response to a GetHubDescriptor request must have a maximum value of 400ns (tHubDelay) in the wHubDelay field.	TD 10.8
10.15.2.1#20	The descriptor returned in response to a GetHubDescriptor request must have a value of 1 set in the corresponding bitmap location of the DeviceRemovable field if the device attached to that port is non-removable, or the port is otherwise inaccessible.	TD 10.24
10.15.2.1#21	The descriptor returned in response to a GetHubDescriptor request must have a value of 0 set in the corresponding bitmap location of the DeviceRemovable field if any device attached to that port is removable.	TD 10.24
10.15.2.1#22	The descriptor returned in response to a GetHubDescriptor request must have a value of 0 set in the corresponding bitmap location of the DeviceRemovable field if no port exists for the corresponding port number.	TD 10.8
10.15.2.1#23	wHubDelay must be greater than 0.	TD 10.8
10.15.2.1#24	The descriptor returned in response to a GetHubDescriptor request must have a value less than or equal to 15 in bNbrPorts field.	TD 10.8

Assertion #	Assertion Description	Test #
Subsection reference: 10.16 Requests		
Subsection reference: 10.16.1 Standard Requests		
10.16.1#1	For requests without a data stage, the hub shall complete them within 50ms.	LVS
10.16.1#2	For requests with a data stage, the hub shall complete the first data stage of a request within 50ms of receiving the setup packet.	LVS
10.16.1#3	For requests with a data stage, the hub shall meet the 50ms completion timing requirement between each subsequent data stage.	LVS
10.16.1#4	For requests with a data stage, the hub shall complete each subsequent data stage of a request within 50ms of the previous one.	LVS
10.16.1#5	For requests with a data stage, the hub shall complete the status stage of the request within 50ms of receiving the last data stage of the request.	LVS
10.16.1#6	(It is recommended that) hubs average response times be less than 5ms for all requests.	Untestable
10.16.1#7	A hub shall accept all standard requests without error.	Multiple TDs
10.16.1#8	Optional requests that are not implemented shall return a STALL in the Data stage or Status stage of the request.	Chapter 9 tests
10.16.1#9	GET_INTERFACE bRequest is undefined for Hubs as Hubs are allowed to support only one interface.	Untestable
10.16.1#10	SET_INTERFACE bRequest is undefined for Hubs as Hubs are allowed to support only one interface.	Untestable
10.16.1#11	SYNCH_FRAME bRequest is undefined for Hubs as Hubs are not allowed to have isochronous endpoints.	Untestable
10.16.1#12	A hub is required to accept all “Standard” requests without error.	Chapter 9 tests
10.16.1#13	A hub shall not respond with a request error to a well-formed SET_ISOCH_DELAY request. A hub is not required to retain or process the delay value.	Chapter 9 tests
Subsection reference: 10.16.2 Class-specific Requests		
Subsection reference: 10.16.2.1 Clear Hub Feature		
10.16.2.1#1	The hub shall clear C_HUB_LOCAL_POWER bit to zero after receiving a ClearHubFeature(C_HUB_LOCAL_POWER) request.	TD 10.30

Assertion #	Assertion Description	Test #
10.16.2.1#2	The hub shall clear C_HUB_OVER_CURRENT bit to zero after receiving a ClearHubFeature(C_HUB_OVER_CURRENT) request.	TBD
10.16.2.1#3	If the wValue field in a ClearHubFeature request is not a valid feature selector value, the hub shall return a STALL in response to the next data stage transaction or in the status stage.	TD 10.9
10.16.2.1#4	If the wIndex field in a ClearHubFeature request is not zero, the hub shall return a STALL in response to the next data stage transaction or in the status stage.	TD 10.9
10.16.2.1#5	If the wLength field in a ClearHubFeature request is not zero, the hub shall return a STALL in response to the next data stage transaction or in the status stage.	TD 10.9
Subsection reference: 10.16.2.2 Clear Port Feature		
10.16.2.2#1	The hub shall clear PORT_POWER bit in wPortStatus field and cause the port to be placed in DSPORT.Powered-off state after receiving a ClearPortFeature(PORT_POWER).	LVS
10.16.2.2#2	The hub shall set C_PORT_CONFIG_ERROR bit to zero after receiving ClearPortFeature(C_PORT_CONFIG_ERROR) request.	TD 10.13
10.16.2.2#3	The hub shall cause the port to de-assert the Force_LinkPM_Accept bit in Set Link Function LMPs after receiving a ClearPortFeature(FORCE_LINKPM_ACCEPT).	LVS
10.16.2.2#4	If the wValue field in a ClearPortFeature request is not a valid feature selector value, the hub shall return a STALL in response to the next data stage transaction or in the status stage.	TD 10.9
10.16.2.2#5	If the wIndex field in a ClearPortFeature request specifies a port that does not exist, the hub shall return a STALL in response to the next data stage transaction or in the status stage.	TD 10.9
10.16.2.2#6	If the wLength field in a ClearPortFeature request is not zero, the hub shall return a STALL in response to the next data stage transaction or in the status stage.	TD 10.9
10.16.2.2#7	Hub shall ignore (treat as a no-op) any standard request, other than GetPortStatus and SetPortFeature(PORT_POWER), received in the DSPORT.Powered-off, the DSPORT.Powered-off-detect or the DSPORT.Powered-off-reset state.	LVS
Subsection reference: 10.16.2.3 Get Hub Descriptor		
10.16.2.3#1	All hubs shall implement one hub descriptor, with descriptor index zero.	TD 10.8

Assertion #	Assertion Description	Test #
10.16.2.3#2	The hub shall respond with only the actual length of the descriptor if the wLength is larger than the actual length.	TD 10.9
10.16.2.3#3	The hub shall respond with only the first wLength bytes of the descriptor if the wLength is less than the actual length.	TD 10.9
10.16.2.3#4	The hub shall respond with zero bytes of the descriptor if the wLength is zero.	TD 10.9
10.16.2.3#5	If the wValue field in a GetHubDescriptor request is not 0x2A00H, the hub shall return a STALL in response to the next data stage transaction or in the status stage.	TD 10.9
10.16.2.3#6	If the wIndex field in a GetHubDescriptor request is not 0x0, the hub shall return a STALL in response to the next data stage transaction or in the status stage.	TD 10.9
Subsection reference: 10.16.2.4 Get Hub Status		
10.16.2.4#1	If hub power is provided by external source, the hub shall return Local power source bit equal to zero in wHubStatus field after receiving a GetHubStatus request.	TD 10.12
10.16.2.4#2	If hub power is provided by USB, the hub shall return Local power source bit equal to one in wHubStatus field after receiving a GetHubStatus request.	TD 10.12
10.16.2.4#3	When performing a GetHubStatus request, the hub shall only report the C_HUB_LOCAL_POWER bit of the wHubChange field set to 1 if there has been either a change in the local power source since the last time software cleared this bit to 0, or it has received a SetHubFeature request that results in a change of the C_HUB_LOCAL_POWER bit on the hub since the last time software cleared this bit to 0.	TD 10.12
10.16.2.4#4	When performing a GetHubStatus request, the hub shall only report the C_HUB_OVER_CURRENT bit of the wHubChange field set to 1 if either an Over-current condition has occurred on the hub since the last time software cleared this bit to 0, or it has received a SetHubFeature request that results in a change of the C_HUB_OVER_CURRENT bit on the hub since the last time software cleared this bit to 0.	Implicit in 10.23
10.16.2.4#5	If the wValue field in a GetHubStatus request is not zero, the hub shall return a STALL in response to the next data stage transaction or in the status stage.	TD 10.9
10.16.2.4#6	If the wIndex field in a GetHubStatus request is not zero, the hub shall return a STALL in response to the next data stage transaction or in the status stage.	TD 10.9
10.16.2.4#7	If the wLength field in a GetHubStatus request is not four, the hub shall return a STALL in response to the next data stage transaction or in the status stage.	TD 10.9

Assertion #	Assertion Description	Test #
10.16.2.4#8	If the hub has Over-current detection capabilities and supports Over-current reporting for the whole hub and if the sum of all the ports' current has exceeded the specified maximum the hub shall return a value of '1' in bit 1 of the wHubStatus field after receiving a GetHubStatus request	Untestable
10.16.2.4#9	If bit 1 of the wHubStatus field has a value of '1' the hub shall place all ports in the DSPORT.Powered-off-reset or DSPORT.Powered-off-detect state and GetPortStatus(PORT_POWER) shall be '0' for all downstream ports.	LVS
10.16.2.4#10	If the hub does not have Over-current detection capability the hub shall return a value of '0' in bit 1 of the wHubStatus field after receiving a GetHubStatus request	TD 10.23
10.16.2.4#11	If the hub does not report Over-current on a hub basis, the hub shall return a value of '0' in bit 1 of the wHubStatus field after receiving a GetHubStatus request	TD 10.23
10.16.2.4#12	The hub shall return a value of '0' in bits 2:15 of the wHubStatus field after receiving a GetHubStatus request	TD 10.12
10.16.2.4#13	C_HUB_OVER_CURRENT is set when a change has occurred in the Over-Current field in wHubStatus	TD 10.23
10.16.2.4#14	The hub shall return a value of '0' in bits 2:15 of the wHubChange field after receiving a GetHubStatus request	TD 10.12
10.16.2.4#15	C_HUB_LOCAL_POWER field is initialized to zero when the hub receives a bus reset.	TD 10.12
10.16.2.4#16	C_HUB_OVER_CURRENT field is initialized to zero when the hub receives a bus reset.	LVS
Subsection reference: 10.16.2.5 Get Port Error Count		
10.16.2.5#1	The hub shall return the number of link errors detected by the hub on the port indicated by wIndex after receives GetPortErrorCount() request.	LVS
10.16.2.5#2	The hub shall reset the number of link errors to zero, when the downstream port is powered on or is reset.	LVS
10.16.2.5#3	Hub shall return a STALL if the wValue field in GetPortErrorCount is not zero either in Data state or Status Stage.	TD 10.9
10.16.2.5#4	Hub shall return a STALL if the wLength field in GetPortErrorCount is not two either in Data Stage or the Status Stage.	TD 10.9
10.16.2.5#5	Hub Shall return a STALL if the wIndex field in GetPortErrorCount specifies a port that does not exist either in Data Stage or in the Status Stage.	TD 10.9
Subsection reference: 10.16.2.6 Get Port Status		

Assertion #	Assertion Description	Test #
10.16.2.6#1	Hub shall report the PORT_CONNECTION bit in the port status as 1 when the port is in the DSPORT.Enabled state.	LVS
10.16.2.6#2	Hub shall report the PORT_CONNECTION bit in the port status as 0 when the port is in any state other than DSPORT.Enabled, DSPORT.Resetting after DSPORT.Enabled or DSPORT.Error after DSPORT.Enabled states.	LVS
10.16.2.6#3	SetPortFeature(PORT_CONNECTION) shall be treated as no-operation requests by hubs.	LVS
10.16.2.6#4	ClearPortFeature(PORT_CONNECTION) shall be treated as no-operation requests by hubs.	LVS
10.16.2.6#5	Hub shall report the PORT_ENABLE bit in the port status bits as 1 when the downstream port is in the DSPORT.Enabled state.	LVS
10.16.2.6#6	Hub shall report the PORT_ENABLE bit in the port status bits as 0 when the downstream port is in any state other than DSPORT.Enabled.	LVS
10.16.2.6#7	Hub shall report PORT_OVER_CURRENT bit as 1 while an Over-current condition exists on the port and zero otherwise.	TD 10.23
10.16.2.6#8	Hub shall report PORT_OVER_CURRENT bit as 0 while no Over-current condition exists on the port.	LVS
10.16.2.6#9	SetPortFeature(PORT_OVER_CURRENT) shall be treated as no-operation requests by hubs.	LVS
10.16.2.6#10	ClearPortFeature(PORT_OVER_CURRENT) shall be treated as no-operation requests by hubs.	LVS
10.16.2.6#11	Hub shall report the PORT_RESET bit as 1 while the port is in the DSPORT.Resetting state and as zero in all other downstream port states.	LVS
10.16.2.6#12	Hub shall report the PORT_RESET bit as 0 while the port is in any state other than DSPORT.Resetting.	LVS
10.16.2.6#13	ClearPortFeature(PORT_RESET) shall be treated as no-operation requests by hubs.	LVS
10.16.2.6#14	ClearPortFeature(PORT_LINK_STATE) shall be treated as no-operation requests by hubs.	LVS
10.16.2.6#15	Hub shall report PORT_POWER bit as zero when the port is in DSPORT.Powered-off state, DSPORT.Powered-off-detect state or DSPORT.Powered-off-reset state.	LVS

Assertion #	Assertion Description	Test #
10.16.2.6#16	Hub shall report PORT_POWER bit as one when the port is in any state other than DSPORT.Powered-off state, DSPORT.Powered-off-detect state or DSPORT.Powered-off-reset state.	LVS
10.16.2.6#17	SetPortFeature(PORT_POWER) will set PORT_POWER bit to one.	LVS
10.16.2.6#18	Hub shall report PORT_SPEED field value as zero to indicate that an Enhanced SuperSpeed device attached to this port.	LVS
10.16.2.6#19	If the wValue field in a GetPortStatus request is not 0, 1 or 2, the hub shall return a STALL in response to the next data stage transaction or in the status stage.	TD 10.9
10.16.2.6#20	If the wIndex field in a GetPortStatus request specifies a port that does not exist, the hub shall return a STALL in response to the next data stage transaction or in the status stage.	TD 10.9
10.16.2.6#21	If the wLength field in a GetPortStatus request is not 4 (for PORT_STATUS type), 8 (for PD_STATUS type) or 8 (for EXT_PORT_STATUS type), the hub shall return a STALL in response to the next data stage transaction or in the status stage.	TD 10.9
10.16.2.6#22	A hub shall not report a Port status where bits 5-8 of the wPortStatus field are in the range of 0xC-0xF inclusive.	Add a generic routine for checking the link state that makes sure link state is never an invalid value, and call this whenever we check link states.
10.16.2.6#23	In the DSPORT.Resetting or DSPORT.Error state, the PORT_CONNECTION bit maintains the value from the prior state.	TD 10.13
10.16.2.6#24	A hub shall return a PORT_STATUS to a GetPortStatus with wValue of 0x00.	TBD
10.16.2.6#25	A GetPortStatus with wValue of 0x01 (PD_STATUS) is deprecated and shall not be used.	TBD
10.16.2.6#26	A hub shall return an EXT_PORT_STATUS to a GetPortStatus with wValue of 0x02.	TBD
10.16.2.6#27	The first word of PORT_STATUS or EXT_PORT_STATUS contains the wPortStatus field.	TBD
10.16.2.6#28	The second word of PORT_STATUS or EXT_PORT_STATUS data contains the wPortChange field.	TBD
10.16.2.6#29	An EXT_PORT_STATUS request shall return an additional dword of data that contains the dwExtPortStatus field.	TBD

Assertion #	Assertion Description	Test #
10.16.2.6#30	The GetPortStatus wValue high order byte shall be 0.	TBD
Subsection reference: 10.16.2.6.1 Port Status Bits		
Subsection reference: 10.16.2.6.2 PortStatus Change Bits		
10.16.2.6.2#1	Hub shall report C_PORT_CONNECTION bit as one when the PORT_CONNECTION bit changes. Note that this applies even when the device was already connected when the port was being powered on and no reset has occurred on the port.	LVS
10.16.2.6.2#2	Hub shall maintain the prior value of the PORT_CONNECTION bit when the downstream port transitions from any state to DSPORT.Resetting or DSPORT.Error.	LVS
10.16.2.6.2#3	C_PORT_CONNECTION bit shall be set to zero by a ClearPortFeature(C_PORT_CONNECTION) request.	LVS
10.16.2.6.2#4	Hub shall report C_PORT_CONNECTION as 0 if the port is in DSPORT.PoweredOff State, DSPORT.Powered-off-detect state or DSPORT.Powered-off-reset state.	LVS TD 10.23
10.16.2.6.2#5	Hub shall report C_PORT_OVER_CURRENT bit as one when the PORT_OVER_CURRENT bit changes from zero to one or from one to zero.	TD 10.23
10.16.2.6.2#6	C_PORT_OVER_CURRENT bit shall be set to zero by a ClearPortFeature(C_PORT_OVER_CURRENT) request.	TD 10.23
10.16.2.6.2#7	C_PORT_OVER_CURRENT bit shall be set to one when the port is in the DSPORT.Powered-off state, DSPORT.Powered-off-detect state or DSPORT.Powered-off-reset state due to an Over-current condition on another port.	TD 10.23
10.16.2.6.2#8	C_PORT_RESET bit is set to one when the port transitions from the DSPORT.Resetting state to the DSPORT.Enabled state for any type of reset.	LVS
10.16.2.6.2#9	C_PORT_RESET bit shall be set to zero by a ClearPortFeature(C_PORT_RESET) request.	LVS
10.16.2.6.2#10	C_PORT_RESET bit shall be set to zero when logical port power is off.	LVS TD 10.23
10.16.2.6.2#11	C_BH_PORT_RESET bit is set to one when the port transitions from the DSPORT.Resetting state to the DSPORT.Enabled state for a Warm Reset only.	LVS

Assertion #	Assertion Description	Test #
10.16.2.6.2#12	Hub shall report C_PORT_RESET bit as one whenever it reports C_BH_PORT_RESET as one, as a result of transitioning from DSPORT.Resetting state to the DSPORT.Enabled state.	LVS
10.16.2.6.2#13	C_BH_PORT_RESET bit shall be cleared by a ClearPortFeature(C_BH_PORT_RESET) request.	LVS
10.16.2.6.2#14	C_PORT_BH_RESET bit shall be set to zero when logical port power is off.	LVS TD 10.23
10.16.2.6.2#15	C_PORT_LINK_STATE bit is set to one when the port's link completes a transition from the U3 state to the U0 state as a result of a SetPortFeature(Port_Link_State) request.	LVS
10.16.2.6.2#16	C_PORT_LINK_STATE bit is set to one when the port's link completes a transition from any of the U-states to SS.Inactive or Loopback.	Untestable
10.16.2.6.2#17	C_PORT_LINK_STATE bit is not set to one due to transition from U3 to U0 as a result of remote wakeup signaling received on a downstream facing port.	LVS
10.16.2.6.2#18	C_PORT_LINK_STATE bit will be cleared by a ClearPortFeature(C_PORT_LINK_STATE) request.	LVS
10.16.2.6.2#19	C_PORT_LINK_STATE bit will be cleared when logical port power is off.	LVS TD 10.23
10.16.2.6.2#20	C_PORT_CONFIG_ERROR bit is set to one if the link connected to the port could not be successfully configured. In addition, the port shall transition to the DSPORT.Error state.	TD 10.13
10.16.2.6.2#21	C_PORT_CONFIG_ERROR bit will be cleared when logical port power is off.	TD 10.13 TD 10.23
10.16.2.6.2#22	The hub shall make sure that the all change bits are reported as 0, unless they need to be 1 as dictated by the above asserts.	TD 10.1
10.16.2.6.2#23	Port status change bits shall be cleared with a directed hub reset.	TD 10.13
Subsection reference: 10.16.2.6.3 Extended Port Status Bits		
10.16.2.6.3#1	The extended port status bits are returned only for Get Port Status request of type EXT_PORT_STATUS.	TD 10.28
10.16.2.6.3#2	For Enhanced SuperSpeed device the Port Speed is the Link Speed multiplied by Lane Count.	untestable
10.16.2.6.3#3	Bits 0-3 of dwExtPortStatus represent the negotiated Rx Sublink Speed of an attached Enhanced SuperSpeed device.	TD 10.27

Assertion #	Assertion Description	Test #
10.16.2.6.3#4	Bits 4-7 of dwExtPortStatus represent the negotiated Tx Sublink Speed of an attached Enhanced SuperSpeed device.	TD 10.27
10.16.2.6.3#5	Bits 8-11 of dwExtPortStatus represent the Rx Lane Count of an attached Enhanced SuperSpeed device.	TD 10.27
10.16.2.6.3#6	Bits 12-15 of dwExtPortStatus represent the Tx Lane Count of an attached Enhanced SuperSpeed device.	TD 10.27
10.16.2.6.3#7	Bits 12-31 of dwExtPortStatus are reserved and must be 0.	TD 10.27
Subsection reference: 10.16.2.7 Set Hub Descriptor		
10.16.2.7#1	For the Hubs that support this Set Hub Descriptor, the hubs shall buffer all the bytes received from this request to ensure that the entire descriptor has been successfully transmitted from the host.	Untestable
10.16.2.7#2	For the Hubs that support Set Hub Descriptor, if the wIndex is not zero it results in a Request Error.	Untestable
10.16.2.7#3	For the Hubs that support Set Hub Descriptor, if the wLength does not match the amount of data sent by the host, it results in a Request Error.	Untestable
10.16.2.7#4	The Hubs that do not support Set Hub Descriptor must respond with a STALL in the data phase of the request.	Not Testing
Subsection reference: 10.16.2.8 Set Hub Feature		
Subsection reference: 10.16.2.9 Set Hub Depth		
10.16.2.9#1	It is a Request Error if wValue field in SetHubDepth is greater than four.	TD 10.9
10.16.2.9#2	It is a Request Error if wIndex field in SetHubDepth is not zero.	TD 10.9
10.16.2.9#3	It is a Request Error if wLength field in SetHubDepth is not zero.	TD 10.9
Subsection reference: 10.16.2.10 Set Port Feature		
10.16.2.10#1	The hub shall set the Timeout value for U1 inactivity timer with wIndex field when receive a SetPortFeature (PORT_U1_TIMEOUT).	TD 10.12
10.16.2.10#2	The hub shall set the Timeout value for U2 inactivity timer with wIndex field when receive a SetPortFeature (PORT_U2_TIMEOUT).	TD 10.12

Assertion #	Assertion Description	Test #
10.16.2.10#3	The port's link shall send a U2 Inactivity Timeout LMP to its link partner with the specified timeout value after receiving a Set Port Feature request with the PORT_U2_TIMEOUT feature selector.	LVS
10.16.2.10#4	If both U1 and U2 timeout values are set, then the downstream will initiate a U1 transition after U1 timeout and then will initiate a U2 transition after U2 timeout has expired. U2 timeout should be counted after link has entered U1.	LVS
10.16.2.10#5	The hub shall transition the link to U0 from any of U states when it receives a SetPortFeature(PORT_LINK_STATE) and the value of the most significant byte in wIndex is 0.	LVS
10.16.2.10#6	The hub shall transition the link to U1 from U0 when it receives a SetPortFeature(PORT_LINK_STATE) in U0 and the value of the most significant byte in wIndex is 1.	TD 10.17 LVS
10.16.2.10#7	The hub shall transition the link to U3 from U0 when it receives a SetPortFeature(PORT_LINK_STATE) in U0 and the value of the most significant byte in wIndex is 3.	LVS
10.16.2.10#8	When receiving a SetPortFeature(PORT_LINK_STATE = 3) for a port that is in U1 or U2, the hub shall transition the port first to U0 and then to U3.	LVS
10.16.2.10#9	The hub shall transition the link to SS.Disabled state when it receives a SetPortFeature(PORT_LINK_STATE) and the value of the most significant byte in wIndex is 4.	LVS
10.16.2.10#10	The hub shall transition the link to RX.Detect state when it receives a SetPortFeature(PORT_LINK_STATE), the value of the most significant byte in wIndex is 5 and the downstream port is in DSPORT.Disabled state.	LVS
10.16.2.10#11	The hub shall not signal a remote wake up due to a connect event on a port if it has received SetPortFeature (PORT_REMOTE_WAKE_MASK) and the Bit0 (Conn_RWEnable) of the most significant byte in wIndex is 0 for the port.	LVS
10.16.2.10#12	The hub shall signal a remote wake up due to a connect event on a port if it has received SetPortFeature (PORT_REMOTE_WAKE_MASK) and the Bit0 (Conn_RWEnable) of the most significant byte in wIndex is 1, and if Function Remote Wake is also enabled for the port.	LVS
10.16.2.10#13	The hub shall not signal a remote wake up due to a disconnect event on a port if it has received SetPortFeature (PORT_REMOTE_WAKE_MASK) and the Bit1 (Disconn_RWEnable) of the most significant byte in wIndex is 0 for the port.	LVS
10.16.2.10#14	The hub shall signal a remote wake up due to a disconnect event on a port if it has received SetPortFeature (PORT_REMOTE_WAKE_MASK) and the Bit1 (Disconn_RWEnable) of the most significant byte in wIndex is 1, and if Function Remote Wake is also enabled for the port.	LVS

Assertion #	Assertion Description	Test #
10.16.2.10#15	The hub shall not signal a remote wake up due to an Over-current event on a port if it has received SetPortFeature (PORT_REMOTE_WAKE_MASK) and the Bit2 (OC_RWEnable) of the most significant byte in wIndex is 0 for the port.	TBD
10.16.2.10#16	The hub shall signal a remote wake up due to an Over-current event on a port if it has received SetPortFeature (PORT_REMOTE_WAKE_MASK) and the Bit2 (OC_RWEnable) of the most significant byte in wIndex is 1, and if Function Remote Wake is also enabled for the port.	TBD
10.16.2.10#17	When a port transitions to powered on state or when the hub is reset, the hub shall set the remote wake mask to zero.	TD 10.16
10.16.2.10#18	The hub shall initiate a warm reset on the port when it receives a SetPortFeature(BH_PORT_RESET).	LVS
10.16.2.10#19	On completion of a BH_PORT_RESET, the hub shall set the C_BH_PORT_RESET field to one in the PortStatus for this port.	LVS
10.16.2.10#20	The hub shall treat a SetPortFeature(PORT_RESET) request as a functional no-operation if the port is in the Powered-off state.	LVS
10.16.2.10#21	The hub shall treat a SetPortFeature(PORT_LINK_STATE) U3 request as a functional no-operation if the port is not in the Enabled state.	LVS
10.16.2.10#22	The hub shall treat a SetPortFeature(PORT_POWER) request as a functional no-operation if the port is not in the Powered-off state.	LVS
10.16.2.10#23	The hub shall treat a SetPortFeature (FORCE_LINKPM_ACCEPT) request as a functional no-operation if the port is not in the Enabled state.	LVS
10.16.2.10#24	It is a Request Error if wValue is not a feature selector listed in Table 10.7.	TD 10.9
10.16.2.10#25	It is a Request Error if wIndex specifies a port that does not exist.	TD 10.9
10.16.2.10#26	It is a Request Error if wLength is not as specified.	TD 10.9
10.16.2.10#27	A hub that does not support per-port Over-current detection/reporting shall signal remote-wakeup for an Over-current event if at least one port has OC-RWEnable set to 1.	TBD
10.16.2.10#28	When a hub receives a SetPortFeature(PORT_LINK_STATE) and the value of the most significant byte in wIndex is 10, it shall enable entry into Compliance Mode for the next attach, if the downstream port is currently in the DSPORT.Disconnected state.	Link Tests TD 7.34

Assertion #	Assertion Description	Test #
10.16.2.10#29	Entry into Compliance Mode via SetPortFeature(PORT_LINK_STATE=10) on a downstream port is disabled once the link enters Compliance Mode or Polling.LFPS succeeds.	Link Tests TD 7.34
Subsection reference: 10.17 Host Root (Downstream) Ports		
Subsection reference: 10.18 Peripheral Device Upstream Ports		
Subsection reference: 10.19 Hub Chapter Parameters		
10.19#1	An Enhanced SuperSpeed hub can have no more than nMaxHubPorts (15) ports.	TD 10.8
10.19#2	tHubDelay (400 ns) This timing defines the maximum delay in nanoseconds a hub can introduce while forwarding packets in either direction. The time is measured from receipt of the last symbol of the packet by the receiving port until the transmitting port sends the first framing of the packet, when both the receiving and transmitting links are in U0 and the following conditions are met: No Link Commands or SKP ordered sets or other packets are in flight. Remote Rx Header Buffer Credit Count of the transmitting port is not zero. Tx Header Buffer of the transmitting port is empty.	LVS
10.19#3	tDSPortEnabledToU3 (max 1 s) Time from when a downstream port enters DSPORT.ENABLED when the upstream hub port is in U3 and remote wakeup is disabled to when the downstream port shall initiate a transition to the U3 link state.	LVS
Subsection reference: 11.4.1 Classes of Devices and Connections		
11.4.1#1	Self powered hubs shall provide six unit loads of power to each port.	Electrical Drop Test
Subsection reference: 11.4.1.1 Self-powered Hubs		
11.4.1.1#1	If a self-powered hub draws power for its upstream facing port from Vbus, it shall not draw more than one unit load	Current Measurement Tests
Subsection reference: 11.4.1.1 Over-current Protection		
11.4.1.1.1#1	The Over-current limit value of a hub cannot exceed 5.0A and must be sufficiently higher than the maximum allowable port current or time delayed such that transient currents do not trip the Over-current protector.	TBD Requires new Over-current fixture. Untestable for now.
11.4.1.1.1#2	The Over-current limiting mechanism within a hub shall be resettable without user mechanical intervention	TBD
Subsection reference: 11.4.2 Steady-State Voltage Drop Budget		

Assertion #	Assertion Description	Test #
11.4.2#1	The voltage supplied at the connector of a hub shall be between 4.45V to 5.25V	Electrical Drop/Droop Test
Subsection reference: 11.4.3 Power Control During Suspend/Resume		
11.4.3#1	When a hub is in the Suspend state, it shall still be able to provide the maximum current per port.	Electrical Drop/Droop Test

3 DFP Directed Reset Behavior

DFP State	Warm Reset Capable	Hot Reset Capable
SS.Disabled		
Rx.Detect		
Polling	yes	Yes
Loopback	yes	
Compliance	yes	
SS.Inactive	yes	
U0	yes	Yes
U1	yes	yes
U2	yes	yes
U3	yes	
Recovery	yes	yes
HotReset	yes	yes

DFP State	DFP Behavior if UFP Warm Reset	DFP Behavior if UFP Hot Reset
DSPORT.Disabled	Warm Reset - after Rx.Detect	Warm Reset - after Rx.Detect
DSPORT.Powered-off	none	none
DSPORT.Powered-off-detect	none	none
DSPORT.Disconnected	none	none
DSPORT.Powered-off-reset	no change in behavior	no change in behavior
DPORT.ERROR (ss.inactive)	Warm Reset	Warm Reset
DSPORT.Enabled - U0	Warm Reset	Hot Reset
DSPORT.Enabled - U1	Warm Reset	Hot Reset
DSPORT.Enabled - U2	Warm Reset	Hot Reset
DSPORT.Enabled - U3	Warm Reset	Warm Reset
DSPORT.Enabled - Recovery	Warm Reset	Hot Reset
Hot Reset	Warm Reset	no change in behavior
Rx.Detect	none	none
Polling/Training	Warm Reset	Hot Reset
DSPORT.Loopback	Warm Reset	Warm Reset
DSPORT.Compliance	Warm Reset	Warm Reset

DFP State	DFP Behavior if SetPortFeature(BH_PORT_RESET) and hub is configured	DFP Behavior if SetPortFeature(PORT_RESET) and hub is configured
DSPORT.Disabled	none	none
DSPORT.Powered-off	none	none
DSPORT.Powered-off-detect	none	none
DSPORT.Disconnected	none	none
DSPORT.Powered-off-reset	no change in behavior	no change in behavior
DPORT.ERROR (ss.inactive)	Transition to DSPORT.Resetting and issue Warm Reset.	Transition to DSPORT.Resetting and issue Warm Reset.
DSPORT.Enabled - U0	Warm Reset	Hot Reset
DSPORT.Enabled - U1	Warm Reset	Hot Reset
DSPORT.Enabled - U2	Warm Reset	Hot Reset
DSPORT.Enabled - U3	Warm Reset	Warm Reset
DSPORT.Enabled - Recovery	Warm Reset	Hot Reset
Hot Reset	Warm Reset	Hot Reset
Rx.Detect	none	none
DSPORT.Training	Warm Reset	Hot Reset
DSPORT.Loopback	Warm Reset	Warm Reset
DSPORT.Compliance	Warm Reset	Warm Reset

4 Test Descriptions

Port Under Test: For tests that require a device to be connected to a DS port, the Port Under Test is the port the device is connected to.

Auxiliary Ports: These are all other DS ports, usually unpopulated.

Chapter 9 Assertions

On occasion, the tests that follow record assertions from Chapter 9 of the USB 3.2 spec. (This primarily happens during test initialization, for example when GET_DESCRIPTOR requests fail). These assertions are listed here rather than in the tests themselves.

Assertions Covered

9.2.5.2#1, 9.3#1, 9.4#1, 9.4.3#2, 9.4.3#4, 9.4.3#6, 9.4.3#9, 9.4.5#33, 9.4.9#14, 9.6.1#8, 9.6.1#10, 9.6.1#11, 9.6.2#1, 9.6.2.3#1, 9.6.2.3#2, 9.6.5#17, 9.6.6#1, 9.6.7#1, 9.6.7#2.

TD 10.1 Port Status Change Bits Default Value Test

This test verifies the default value of port change bits.

Assertions Covered

10.16.2.6.2#22

Starting Configuration

Port Under Test: No device attached

Auxiliary Ports: No device attached

Test Steps

1. If HUT is embedded, then exit test with a pass.
2. Start test with no hub attached.
3. Prompt user to power on hub (if self-powered).
4. Prompt user to connect hub.
5. Enumerate and configure the Hub.
6. If hub has no exposed ports, then the test passes. Exit the test.
7. Prompt user to attach a SS device to a downstream port.
8. Prompt user to detach all DS devices.
9. Prompt user to detach Hub Under Test
10. Prompt user to reattach Hub Under Test
11. Enumerate and configure the hub.
12. Issue GetPortStatus for all exposed ports.
13. Test fails if any of the Change bits are set to 1.(10.16.2.6.2#22)
    a. C_PORT_CONNECTION
    b. C_PORT_OVER_CURRENT
    c. C_PORT_RESET
    d. C_BH_PORT_ RESET
    e. C_PORT_LINK_STATE.
    f. C_PORT_CONFIG_ERROR

TD 10.2 Hub Device Descriptor Test

This test verifies hub descriptors for both SS and 2.0 hub components.

Assertions Covered

10.1.1#1, 10.1.1#2, 10.15.1#1 – 10.15.1#9

Starting Configuration

Port Under Test: No device attached

Auxiliary Ports: No device attached

Test Steps

1. Enumerate and configure the Hub.
2. Find the HS Hub at the same physical location as the Hub Under Test. The test fails if there isn't one (10.1.1#1)
3. Get the Container ID for the SS Hub and the HS Hub.
    a. The test fails if this does not succeed. (9.4.3#9, 9.6.2#1)
    b. The test fails if either Container ID does not exist. (9.6.2.3#1)
    c. The test fails if the two Container IDs are not equal. (9.6.2.3#2)
4. Send GetDescriptor (Device) request to the SS hub. Parse the data returned.
    a. Test fails if this does not succeed. (9.4.3#2, 9.6.1#8, 9.3#1)
5. Send GetDescriptor (Device) request to the HS hub. Parse the data returned.
    a. Test fails if this does not succeed. (9.4.3#2)
    b. Test fails if bcdUSB is not 0x210. (9.6.1#10, 9.6.1#11)
6. Check the data returned Device Descriptor for the following values:
   - thebLength field of returned descriptor is 0x12. (10.15.1#1)
   - the bDescriptorType field of returned descriptor is 0x01. (10.15.1#2)
   - the lower byte of bcdUSB field of returned descriptor is 0x10. (10.15.1#3)
   - the higher byte of bcdUSB field of returned descriptor is 0x03. (10.15.1#4)
   - the bDeviceClass field of returned descriptor is 0x09. (10.15.1#5)
   - the bDeviceSubClass field of returned descriptor is 0x00. (10.15.1#6)

- if the bDeviceProtocol field of returned descriptor is 0x03. (10.15.1#7)
- if the bMaxPacketSize field of returned descriptor is 0x09. (10.15.1#8)
- the bNumConfigurations field of returned descriptor is 0x01. (10.15.1#9)

7. Check the HS hub data returned for the following values:
   - the lower byte of bcdUSB field of returned descriptor is 0x10. (9.6.1#10)
   - the higher byte of bcdUSB field of returned descriptor is 0x02. (9.6.1#11)
8. The product IDs for the SS and HS hubs must not be the same. (10.1.1#2)

TD 10.3 Configuration Descriptor Test

This test verifies that the Configuration Descriptor is correct.

Assertions Covered

10.14#1 – 10.14#3, 10.15.1#30 – 10.15.1#34

Starting Configuration

Port Under Test:	No device attached
Auxiliary Ports:	No device attached

Test Steps

1. Enumerate and configure the Hub.
2. Send GetDescriptor (Configuration) request.
   a. Test fails if this doesn't succeed. Record an abort and exit the test. (9.3#1, 9.4.3#6)
3. Check the returned Configuration Descriptor:
   a. If the MaxPower field is greater than zero and the self-powered bit in the bmAttributes field of the Configuration Descriptor is zero, abort the test. (We do not certify bus powered hubs at this time.)
4. Sent GetHubStatus() to the HUT. Retain the hub status value returned for the remainder of the test.
   a. If MaxPower.is greater than zero, the self-powered bit in the bmAttributes field is one, and the LocalPowerSource bit of the hub status is one, then abort the test.
5. Send GetStatus() to the HUT. Retain the status value returned for the remainder of the test..
6. Check the returned Configuration Descriptor for the following values:
   - bLength field of returned descriptor is 0x09 (10.15.1#30)
   - bDescriptorType field of returned descriptor is 0x02. (10.15.1#31)
   - the lower byte of wTotalLength field of returned descriptor is 0x1F. (10.15.1#32)
   - the higher byte of wTotalLength field of returned descriptor is 0x00. (10.15.1#33)
   - the higher byte of bNumInterfaces field of returned descriptor is 0x01. (10.15.1#34)
   - the MaxPower field and the Self-Powered bit of are not both 0. (10.14#1)
7. Check the GetStatus() field. Test fails if:
   a. MaxPower has a value of
   b. the self powered bit in the bmAttributes field is one
   c. the SelfPowered bit of the Device Status field is zero. (10.14#2)
8. Check the GetHubStatus() field. Test fails if:
   a. MaxPower has a value of zero
   b. the self powered bit in the bmAttributes field is one
   c. the LocalPowerSource bit of the Hub Status field is one. (10.14#3)

TD 10.4 Interface Descriptor Test

This test verifies that the Interface Descriptor is correct.

Assertions Covered

10.15.1#36 – 10.15.1#43

Starting Configuration

Port Under Test: No device attached

Auxiliary Ports: No device attached

Test Steps

1. Enumerate and configure the Hub.
2. Send GetDescriptor (Configuration) request. Parse the data returned.
3. Check the returned Interface Descriptor for the following values.
   a. bLength field of returned descriptor is 0x09. (10.15.1#36)
   b. bDescriptorType field of returned descriptor is 0x04. (10.15.1#37)
   c. bInterfaceNumber field of returned descriptor is 0x00. (10.15.1#38)
   d. bAlternateSetting field of returned descriptor is 0x00. (10.15.1#39)
   e. bNumEndpoints field of returned descriptor is 0x01. (10.15.1#40)
   f. bInterfaceClass field of returned descriptor is 0x09. (10.15.1#41)
   g. bInterfaceSubClass field of returned descriptor is 0x00. (10.15.1#42)
   h. bInterfaceProtocol field of returned descriptor is 0x00. (10.15.1#43)

TD 10.5 Endpoint Descriptor Test

This test verifies that the Endpoint Descriptors are correct.

Assertions Covered

10.15.1#40, 10.15.1#44 – 10.15.1#50.

Starting Configuration

Port Under Test: No Device Attached

Auxiliary Ports: No Device Attached

Test Steps

1. Enumerate and configure the Hub.
2. Send GetDescriptor (Configuration) request. Parse the data returned.
3. Check the returned Endpoint Descriptor for the following values.
   a. bLength field of returned descriptor is 0x07. (10.15.1#44)
   b. bDescriptorType field of returned descriptor is 0x05. (10.15.1#45)
   c. bit 7 of the bEndpointAddress field is 1. (10.15.1#46)
   d. bmAttributes field of returned descriptor is 0x13. . (10.15.1#47)
   e. lower byte of wMaxPacketSize field of returned descriptor is 0x02. (10.15.1#48)
   f. higher byte of wMaxPacketSize field of returned descriptor is 0x00. (10.15.1#49)
   g. bInterval field of returned descriptor is 0x8. (10.15.1#50)
4. Verify that there is only 1 endpoint besides the default endpoint. (10.15.1#40)

TD 10.6 Endpoint Companion Descriptor Test

This test verifies correct Endpoint Companion Descriptor reporting.

Assertions Covered

10.15.1#51 – 10.15.1#56

Starting Configuration

Port Under Test: No Device Attached

Auxiliary Ports: No Device Attached

Test Steps

1. Enumerate and configure the Hub.
2. Send GetDescriptor (Configuration) request. Parse the data returned.
3. Check the returned Endpoint Companion Descriptor for the following values.
    a. bLength field of returned Endpoint Companion descriptor is 0x06. . (10.15.1#51)
    b. bDescriptorType field of returned Endpoint Companion descriptor is 0x30. (10.15.1#52)
    c. bMaxBurst field of returned Endpoint Companion descriptor is 0x00. . (10.15.1#53)
    d. bmAttributes field of returned Endpoint Companion descriptor is 0x00. (10.15.1#54)
    e. 0x02 in the low byte of the wBytesPerInterval. . (10.15.1#55)
    f. 0x00 in the high byte of the wBytesPerInterval. (10.15.1#56)

TD 10.7 BOS Descriptor Test

This test verifies that the BOS Descriptors are correct for both SS and 2.0 hub components.

Assertions Covered

10.1.1#1, 10.15.1#10 – 10.15.1#29, 10.15.1#57 – 10.15.1#71

Starting Configuration

Port Under Test: No Device Attached

Auxiliary Ports: No Device Attached

Test Steps

1. Enumerate and configure the Hub.
2. Send GetDescriptor (BOS) request. Parse the data returned. (10.1.1#1)
3. Check the returned BOS Descriptor for the following values:
    a. bLength field of returned descriptor is 0x05. (10.15.1#10)
    b. bDescriptorType field of returned descriptor is 0x0F. (10.15.1#11)
    c. wTotalLength field of returned descriptor is as follows:
        i. 0x0049 if the BOS descriptor contains a SuperSpeedPlus USB Device Capability descriptor. (10.15.1#12, 12.15.1#13)
        ii. 0x002a if the BOS descriptor does not contain a SuperSpeedPlus USB Device Capability descriptor and does not contain a PTM Capability descriptor. (10.15.1#72)
        iii. 0x002d if the BOS descriptor does not contain a SuperSpeedPlus USB Device Capability descriptor and does contain a PTM Capability descriptor. (10.15.1#72)
    d. bNumDeviceCaps field of returned descriptor is as follows:
        i. 0x05 if the BOS descriptor contains a SuperSpeedPlus USB Device Capability descriptor. (10.15.1#14)
        ii. 0x03 if the BOS descriptor does not contain a SuperSpeedPlus USB Device Capability descriptor and does not contain a PTM Capability descriptor. (10.15.1#74)
        iii. 0x04 if the BOS descriptor does not contain a SuperSpeedPlus USB Device Capability descriptor and does contain a PTM Capability descriptor. (10.15.1#75)
    e. bLength field of returned USB 2.0 Extension descriptor is 0x07. (10.15.1#15)
    f. bDescriptorType field of returned USB 2.0 Extension descriptor is 0x10. (10.15.1#16)
    g. bDevCapabilityType field of returned USB 2.0 Extension descriptor is 0x02. (10.15.1#17)
    h. bmAttributes field of returned USB 2.0 Extension descriptor is 0x02. (10.15.1#18)
    i. bLength field of returned SuperSpeed USB Device Capability descriptor is 0x0A. (10.15.1#19)
    j. bDescriptorType field of returned SuperSpeed USB Device Capability descriptor is 0x10. (10.15.1#20)
    k. bDevCapabilityType field of returned SuperSpeed USB Device Capability descriptor is 0x03. (10.15.1#21)
    l. lower byte of wSpeedsSupported field of returned SuperSpeed USB Device Capability descriptor is 0x0E. (10.15.1#22)
    m. higher byte of wSpeedsSupported field of returned SuperSpeed USB Device Capability descriptor is 0x00. (10.15.1#23)

  n. bFunctionalitySupport field of returned SuperSpeed USB Device Capability descriptor is 0x01. (10.15.1#24)
  o. bLength field of returned container ID descriptor is 0x14. (10.15.1#26)
  p. bDescriptorType field of returned container ID descriptor is 0x10. (10.15.1#27)
  q. bDevCapabilityType field of returned container ID descriptor is 0x04. (10.15.1#28)
  r. bReserved field of returned container ID descriptor is 0x00. (10.15.1#29)
  s. If the BOS descriptor contains a SuperSpeedPlus USB Device Capability descriptor:
    i. The bLength field of the returned SuperSpeedPlus USB Device Capability descriptor is 28. (10.15.1#59)
    ii. The bDescriptorType field of the returned SuperSpeedPlus USB Device Capability descriptor is 0x10. (10.15.1#60)
    iii. The bDevCapabilityType field of the returned SuperSpeedPlus USB Device Capability descriptor is 0x0A. (10.15.1#61)
    iv. The bReserved field of the returned SuperSpeedPlus USB Device Capability descriptor is 0 (10.15.1#62).
    v. The bmAttributes field of the returned SuperSpeedPlus USB Device Capability descriptor is 0x23. (10.15.1#63)
    vi. The wFunctionalitySupport field of the returned SuperSpeedPlus USB Device Capability descriptor is 0x1100. (10.15.1#64)
    vii. The field bmSublinksSpeedAttr[0] is 0x00050030. (10.15.1#65)
    viii. The field bmSublinksSpeedAttr[0] is 0x000500B0. (10.15.1#66)
    ix. The field bmSublinksSpeedAttr[0] is 0x000A0031. (10.15.1#67)
    x. The field bmSublinksSpeedAttr[0] is 0x000A00B1. (10.15.1#68)
  t. If the BOS descriptor contains a PTM Capability descriptor (note: this is mandatory if BOS descriptor contains a SuperSpeedPlus USB Device Capability descriptor):
    i. The bLength field of the returned Precision Time Measurement descriptor is 3. (10.15.1#69)
    ii. The bDescriptorType field of the returned Precision Time Measurement descriptor is 0x10. (10.15.1#70)
    iii. The bDevCapabilityType field of the returned Precision Time Measurement descriptor is 0x0B. (10.15.1#71)

2. Repeat above steps for the corresponding USB2 hub. (10.1.1#1, 10.15#1)

TD 10.8 Hub Descriptor Test

This test verifies correct Hub Descriptor reporting.

Assertions Covered

10.15.2.1#1 – 10.15.2.1#14, 10.15.2.1#17, 10.15.2.1#19 – 10.15.2.1#24, ~~10.12.1#1, 10.12.1#2,~~ 10.16.2.3#1

Starting Configuration

Port Under Test:	No device attached
Auxiliary Ports:	No device attached

Test Steps

1. Enumerate and configure the Hub.
2. Send GetHubDescriptor(SSHubDescriptor) request. Parse the data returned.
3. Check the returned Hub Descriptor for the following values:
   a. bDescLength field must be 0x0C. (10.15.2.1#1)
   b. bDescriptorType field.must be 0x2A. (10.15.2.1#2)
   c. The number of downstream facing ports must be equal to the bNbrPorts field. (10.15.2.1#3)
   d. Ask the user if the hub supports power switching.
      i. If the hub supports power switching, ask the user if the hub supports global power switching, more than one power switching gang, or per-port power switching.
      ii. Bit 1:0 in the wHubCharacteristics field request must be 00 if the hub uses global power switching (all ports power at once). (10.15.2.1#4)

        iii. Bit 1:0 in the wHubCharacteristics field must be 01 if the hub uses individual port power switching or more than one power switching gang. (10.15.2.1#5, 10.12.1#2)
        iv. Bit 1 in the wHubCharacteristics field is reserved and shall never be 1. (10.15.2.1#6)
    e. Ask the user if the hub is a compound device.
        i. Bit 2 in the wHubCharacteristics field must be 0 if the hub is not part of a compound device. (10.15.2.1#7)
        ii. Bit 2 in the wHubCharacteristics field must be 1 if the hub is part of a compound device.(10.15.2.1#8)
    f. Ask the user if the hub supports Over-current protection.
        i. If the hub supports Over-current protection, ask the user if the hub supports global Over-current protection, more than one Over-current protection gang, or per-port Over-current protection.
        ii. Bit 4:3 in the wHubCharacteristics field must be 00 if the hub provides global Over-current protection.(hub basis protection) (10.15.2.1#9)
        iii. Bit 4:3 in the wHubCharacteristics field must be 01 if the hub provides individual port Over-current protection (per-port basis protection) or more than one Over-current gang (10.15.2.1#10, 10.12.1#1)
    g. If Bit 4 in the wHubCharacteristics field is 1 the hub must be a bus-powered hub that does not provide Over-current protection.(10.15.2.1#11)
    h. Bits 15:5 in the wHubCharacteristics field are reserved and must be 0. (10.15.2.1#12)
    i. The descriptor in response to a GetHubDescriptor request must set the bPwrOn2PwrGood field to a value greater than 0 if the Hub supports power-switching. (10.15.2.1#13)
    j. The descriptor returned in response to a GetHubDescriptor request must set bPwrOn2PwrGood field to zero if the hub does not support power-switching. (10.15.2.1#14)
    k. bHubHdrDecLat field must be between 0 and 0x0AH. (10.15.2.1#17)
    l. wHubDelay field must have a value in the range of 1ns to 400ns (tHubDelay). (10.15.2.1#19, 10.15.2.1#23)
    m. The DeviceRemovable field must have 0 set for the corresponding bitmap of all non-existent ports. (10.15.2.1#22)
    n. bNbrPorts field must be less than or equal to nMaxHubPorts (10.15.2.1#24, 10.17#1)
4. Issue a GetHubDescriptor request with the descriptor index set to one. Test fails if the request succeeds. (10.16.2.3#1)

TD 10.9 Invalid Standard Request Parameters Test

This test verifies that the hub correctly handles invalid requests.

Assertions Covered

10.16.2.1#3 – 10.16.2.1#5, 10.16.2.2#4 – 10.16.2.2#6, 10.16.2.9#1 – 10.16.2.9#3, 10.16.2.10#24 – 10.16.2.10#26, 10.16.2.5#3 – 10.16.2.5#5, 10.16.2.3#2 – 10.16.2.3#6, 10.16.2.8#3 – 10.16.2.8#5, 10.16.2.6#19, – 10.16.2.6#21, 10.16.1#13, 10.16.2.4#5 – 10.16.2.4#7

Starting Configuration

Port Under Test:	No Device Attached
Auxiliary Ports:	No Device Attached

Test Steps

1. Enumerate and configure the Hub.

Testing Hub Class Request

2. Issue SetHubDepth with wValue is five. Test fails if the Hub does not return a STALL. (10.16.2.9#1)
3. Issue SetHubDepth with wIndex is not zero. Test fails if the Hub does not return a STALL. (10.16.2.9#2)
4. Issue SetHubDepth with wLength is not zero. Test fails if the Hub does not return a STALL. (10.16.2.9#3)
5. Issue GetHubStatus with wValue is not zero. Test fails if the Hub does not return a STALL. (10.16.2.4#5)
6. Issue GetHubStatus with wIndex is not zero. Test fails if the Hub does not return a STALL. (10.16.2.4#6)
7. Issue GetHubStatus with wLength is not four. Test fails if the Hub does not return a STALL. (10.16.2.4.9#7)
8. Issue ClearPortFeature with wValue is not listed in Table 10-7. Test fails if the Hub does not return a STALL. (10.16.2.2#4)
9. Issue ClearPortFeature with wIndex specifies an invalid port. Test fails if the Hub does not return a STALL. (10.16.2.2#5)
10. Issue ClearPortFeature with wLength is not zero. Test fails if the Hub does not return a STALL.(10.16.2.2.#6)
11. Issue SetPortFeature with wValue is not listed in Table 10.7. Test fails if the Hub does not return a STALL(10.16.2.10#24)
12. Issue SetPortFeature with wIndex is an invalid port number. Test fails if the Hub does not return a STALL. (10.16.2.10#25)
13. Issue SetPortFeature with wLength is not zero. Test fails if the Hub does not return a STALL. (10.16.2.10#26)
14. Issue GetPortErrorCount with wValue is not zero. Test fails if the Hub does not return a STALL. (10.16.2.5#3)
15. Issue GetPortErrorCount with wLength is not two. Test fails if the Hub does not return a STALL. (10.16.2.5#4)
16. Issue GetPortErrorCount with wIndex specifies an invalid port. Test fails if the Hub does not return a STALL. (10.16.2.5#5)
17. Issue GetHubDescriptor with wValue is 2a01H. Test fails if the Hub does not return a STALL. (10.16.2.3#5)
18. Issue GetHubDescriptor with wIndex is not zero. Test fails if the Hub does not return a STALL. (10.16.2.3#6)
19. Issue GetHubDescriptor with wLength of zero. Test fails if the request fails, or the Hub returns more than zero bytes. (10.16.2.3#4)
20. Issue GetHubDescriptor with wLength of one. Test fails if the request fails, or the Hub does not return exactly one byte. (10.16.2.3#3)
21. Issue GetHubDescriptor with wLength of 24. Test fails if the request fails, or the Hub does not return exactly 12 bytes. (10.16.2.3#2)

22. Issue ClearHubFeature with wValue is 5. Test fails if the hub does not return STALL. (10.16.2.1#3)
23. Issue ClearHubFeature with wIndex is not zero. Test fails if the Hub does not return STALL.(10.16.2.1#4)
24. Issue ClearHubFeature with wLength not 0. Test fails if the Hub does not return a STALL. (10.16.2.1#5)
25. Issue GetPortStatus with wValue is 0xFF. Test fails if the hub does not return STALL. (Assert: 10.16.2.6#19)
26. Issue GetPortStatus with wIndex of 0x11. Test fails if the Hub does not return STALL.(10.16.2.6#20)
27. Issue GetPortStatus with wIndex of 0x00. Test fails if the Hub does not return a STALL. (10.16.2.6#20)
28. Issue GetPortStatus with wIndex of bNbrPorts+1. Test fails if the Hub does not return STALL. (10.16.2.6#20)
29. Issue GetPortStatus with wLength 5. Test fails if the Hub does not return a STALL. (10.16.2.6#21)

Testing Standard Request:

30. Issue SetIsochDelay with valid parameters. Test fails if the Hub returns a STALL (10.16.1#13)

TD 10.10 US Ports State Transition to U0 Test

This test verifies correct behavior of UFP entering and leaving U0. Testing requires a SS device that supports function remote wake and U1/U2.

Assertions Covered

10.6.2.2#1, 10.6.2.3#1, 10.13.1#2, 10.6.2.1#11 – 10.6.2.1#19

Starting Configuration

Port Under Test:	Device attached
Auxiliary Ports:	No device attached

Test Steps

1. Enumerate and configure the Hub.
2. Attach a SS Device to Downstream port of hub.
3. Clear all port status change bits for all DS ports of hub.
4. Initiate IN transaction to Interrupt EP. Hub must respond with NRDY.

U1/U2->U0 (hub command)

5. Put the DS port close to the device in U1 state.
6. Test fails if the US port does not transition to U1 state after 70ms of being idle
7. Initiate a transition to U0 by issuing GetPortStatus command to the Hub.
8. Test fails if US port of the hub does not successfully wakeup to U0 from U1(10.6.2.1#11)

9. Put the DS port close to the device in U2 state.
10. Test fails if US port of the hub does not transition to U2 state after 70ms of being idle.
11. Initiate a transition to U0 by issuing a GetPortStatus command to the hub.
12. Test fails if the US port of the hub does not successfully wakeup to U0 from U2. (10.6.2.1#12)

U1/U2->U0 (port status change)

13. Put the hub US port and DS port in U1 state.
14. Generate a port status changes on the downstream device by detaching a SS device.
15. Test fails if it does not successfully wake the US port to U0 (10.6.2.1#13, 10.6.2.1#15)
16. Re-attach the SS device to the DS port under test.
17. Put the hub US port and DS ports in U2 state.
18. Generate a port status changes on the downstream by detaching the SS device.
19. Test fails if f US port’s link does not successfully transition to U0. (10.6.2.1#14, 10.6.2.1#16)

U1/U2->U0 (U0 initiated by DFP)

20. Attach a remote wake capable device downstream.
21. Put the hub US port and DS port close to the device in U1 state.
22. Resume downstream link to U0 by doing a remote wake from the device.
23. Test fails if UFP does not initiate a transition to U0. (10.6.2.1#15)
24. Put the hub US port and DS port close to the device in U2 state.
25. Resume downstream link to U0 by doing a remote wake from the device.
26. Test fails if UFP does not initiate a transition to U0. (10.6.2.1#16)

US link remains in U0

27. Set the U1_TIMEOUT=0 U2_TIMEOUT=0 on the US port’s link Partner (root port to which the Hub is attached).
28. Put the DS port to which device is attached in U1.
29. Test fails if US ports do not initiate U1 Entry (Need Analyzer to see the LGO_U1)
30. Test fails if the link does not remain in U0 as a result of the US port link partner rejecting the state transition attempt. (10.6.2.1#17, 10.6.2.2#1)
31. Put the DS port under test in U2.

32. Test fails if the US ports do not initiate U2 Entry (need analyzer to verify LGO_U2)
33. Test fails if the US link is not in U0 (10.6.2.1#18, 10.6.2.3#1)

U3->U0 (upstream traffic=> wakeup signaling)

34. Put the Hub DS port under test and US port in U3 state.
35. Resume US port of the Hub by issuing a SetPortFeature command to U0 from the link partner of the US port.
36. Test fails if the US port does not resume back to U0. (10.6.2.1#19)

TD 10.11 Port Power Test

This test verifies…

Assertions Covered

10.12#1, 10.12.1#8, 10.12.1#18

Starting Configuration

Port Under Test: SS device attached

Auxiliary Ports: No device attached

Test Steps

1. Enumerate and configure the Hub.
2. Issue GetHubStatus request.
3. Test fails if:
    a. Local Power Status = 1
    b. C_HUB_LOCAL_POWER = 1
4. Issue SetPortFeature(PORT_POWER) request to each port.
5. Wait for 512ms (maximum value of time to power-good)
6. Issue GetPortStatus. Test fails if PORT_POWER is still zero.
7. If the hub is self-powered verify the following:
    a. ClearPortFeature(PORT_POWER) (PORT_POWER=OFF).
    b. Test fails if subsequent GetPortStatus does not read Port Power=0. Note: Follow the rules of Setting Port Feature to PORT_POWER.
8. If the hub has no power switches
    a. to make sure that port is functional with no delay, with a device connected, do a get port status. Test fails if it does not return expected information.
    b. Send GetHubDescriptor(SSHubDescriptor) request. Test fails if the value of bPwrOn2PwrGood field is not zero.
9. If multiple ports share the power switch:
10. Ask the User if the Hub has multiple ports supports sharing the same power switch.
11. If the user responds with the ports that are sharing the same power switch,
    a. Issue ClearPortFeature(PORT_POWER) to one port sharing the power switch.
    b. Issue GetPortStatus to all ports sharing the same power switch. Test fails if any other port on the power switch indicates PORT_POWER is off (10.12.1#18)
    c. Issue USB3 and USB2 ClearPortFeature(PORT_POWER) to all ports sharing the power switch.
    d. Issue SetPortFeature(PORT_POWER) to one port sharing the power switch.
    e. Issue GetPortStatus to all ports sharing the same power switch. Test fails if any other port for these set of ports indicate PORT_POWER is ON. (10.12.1#8)
    f. Test fails if for any other port in the gang:
        i. PP != 0 (10.12.1#8)
        ii. PLS != 5 (10.12.1#8)
        iii. CCS != 0 (10.12.1#8)

            iv. PR != 0 (10.12.1#8)
            v. PE != 0 (10.12.1#8)
    g. Ask user to insert VBus Test dongle to all ports in the gang. The test fails if Vbus is off for any of the ports (10.12#1)

12. When PORT_POWER is still on, SetConfiguration(0) on hub.
13. *** We cannot get status of DS ports; we need another way to see if power is off on ports (DHR)
14. Configure hub.
15. SetPortFeature(PORT_POWER=1) for all ports. Test fails if requests fail.
16. GetPortStatus to all ports.
17. Test fails if PORT_POWER for any ports is OFF.
18. If hub is self AND bus powered:
    a. Remove local power (unplug external power source):
    b. Issue GetConfiguration request.
        i. Test fails if hub does not return correct configuration value.
    c. Issue Interrupt-In transfer for Status Change Endpoint
        i. Test fails if request times out.
        ii. Test fails if bit 0 of Hub and Port Status Change Bitmap is not set to 1 (assert)
    d. Issue GetHubStatus()
        i. Test fails if wHubStatus bit 0 is 0 (assert)
        ii. Test fails if wHubChange biti 0 is 0 (assert)

    e. If the hub reports any Port status changes (should not – correct?)
    f. If GetStatus() shows that any status change bits are on.
    g. If the hub responds to any SetPortFeature requests.
    h. Restore power to hub

TD 10.12 Hub Power Operating Mode Test

This test verifies that hub power descriptors are correct.

Assertions Covered

10.3.1.1#, 10.13.2#2, 10.13.4#5, 10.14#1, 10.14#2, 10.14.1#3, 10.14#4, 10.14#6, 10.14#7, 10.14#9, 10.14#10, 10.16.2.1#1, 10.16.2.4#1 - 10.16.2.4#3, 10.16.2.4#12, 10.16.2.4#15.

Starting Configuration

Port Under Test: No Device Attached

Auxiliary Ports: No Device Attached

Test Steps

1. If Supported, Start supplying power from the external power source to the HUT.
2. Enumerate and configure the Hub.
3. Issue GetHubStatus() to the Hub under test.
4. Verify that hub returns the following:
    a. A value of 0 in the bits 2:15 of the wHubStatus. (10.16.2.4#12)
    b. A value of 0 in the bits 2:15 of the wHubStatusChange. (10.16.2.4#14)

5. Send GetDescriptor (Configuration) request. Parse the data returned.
6. Perform one of the following steps depending on the bMaxPower and Self-powered bit of bmAttributes reported by Hub Configuration Descriptor.

7. bMaxPower=0 and Self-powered=0 **(Invalid)**
    a. NOTE: This state is USB 3.0 spec, Table10-3, Row 1
    b. Test is failed. (10.14#1)

8. bMaxPower>0 and Self-powered=0 **(Bus Power mode Only)**
    a. NOTE: This state is USB 3.0 spec, Table10-3, Row 4
    b. Test fails if GetStatus(Recipient=Device) request not report
        i. SelfPower bit = 0. (10.14#4)
    c. Test fails if GetHubStatus() request not report:
        i. Local power source bit = 1. (10.16.2.4#2)
        ii. C_HUB_LOCAL_POWER = 0. (10.16.2.4#3)

9. bMaxPower=0 and Self-powered=1 **(Self Power mode Only)**
    a. NOTE: This state is USB 3.2 spec, Table10-4, Row 3
    a. Test fails if GetStatus(Recipient=Device) request not report:
    b. SelfPower bit = 1. (,2)
    b. Test fails if GetHubStatus() request not report:
        i. Local power source bit = 0. (10.16.2.4#1)
        ii. C_HUB_LOCAL_POWER = 0. (10.16.2.4#3)
    c. If HUT is not embedded:
        i. Prompt user to remove the external power source of the HUT.
        ii. NOTE: This state is USB 3.2 spec, Table10-4, Row 2
        iii. Issue GetHubStatus() to the HUT.
        iv. Test fails if the request does not timeout.

10. bMaxPower>0 and SelfPower=1 **(Both Power modes supported)**
    NOTE: This state is USB 3.2 spec, Table10-4, Row 6
    a. Issue GetStatus(Recipient=Device).
        i. Verify that SelfPower bit = 1. (10.14#7)
    b. Issue GetHubStatus() request.
        i. Verify that Local power source bit = 0. (10.16.2.4#1)
        ii. Verify that C_HUB_LOCAL_POWER = 0. (10.16.2.4#3)
    c. If HUT is not embedded:
        i. Prompt user to remove the external power source of the HUT.
        NOTE: This state is USB 3.2 spec, Table10-4, Row 5
        ii. Issue GetConfiguration() to HUT.
            1) Test fails if hub is not configured (10.14#9)
        iii. Issue GetStatus(Recipient=Device) request.
            1) Verify SelfPower bit = 0. (10.14#6)
        iv. Issue GetHubStatus() request.. (10.14.2.4#1)
            1) Verify Local power source bit = 1. (10.16.2.4#2)
            2) Verify C_HUB_LOCAL_POWER = 1 (10.16.2.4#3)
        v. Issue GetHubStatus() request again.
            1) Test fails if C_HUB_LOCAL_POWER does not maintain the value of 1. (10.13.2#2)
        vi. Initiate IN transaction to the Interrupt EP.
            1) Test fails if the Hub does not report with the Hub Status change bit set. (10.13.4#3).
        vii. Issue ClearHubFeature(C_HUB_LOCAL_POWER) to hub UFP.
        viii. Issue GetHubStatus() request.
            1) Verify Local power source bit = 1. (10.16.2.4#2)
            2) Verify C_HUB_LOCAL_POWER = 0. (10.16.2.1#1)
        ix. For each port, issue a SetPortFeature(PORT_POWER)
            1) Test fails if hub does not ignore these requests (10.14#11)
        x. For each port, issue a GetPortStatus() request.
            1) Test fails if the Port Power bit is set for any port. (10.14#9)
            2) Test fails if the Current Connect Status bit is set for any port. (10.14#10)
        xi. Prompt user to attach the external power source of the HUT.

NOTE: This state is USB 3.2 spec, Table10-4, Row 6

xii. Issue GetStatus(Recipient=Device) request.
  1) Verify SelfPower bit = 1. (10.14#7)

xiii. Issue GetHubStatus() request.
  1) Verify Local power source bit = 0. (10.16.2.4#1)
  2) Verify C_HUB_LOCAL_POWER = 1. (10.16.2.4#3)

xiv. Issue WarmReset to hub UFP.

xv. Enumerate and configure the Hub.

xvi. Issue GetStatus(Recipient=Device) request.
  1) Verify SelfPower bit = 1. (10.14#7)

xvii. Issue GetHubStatus() request.
  1) Verify Local power source bit = 0. (10.16.2.4#1)
  2) Verify C_HUB_LOCAL_POWER = 0. (10.16.2.4#15)

TD 10.13 Port Config Error Test

This test requires the use of an A-to-A cable.

Assertions Covered

10.3.1.2#4, 10.3.1.6#15, 10.16.2.2#2, 10.16.2.6.2#2, 10.16.2.6.2#5, 10.16.2.6.2#20, 10.16.2.6.2#21, 10.2.6.2#23

Starting Configuration

<u>Port Under Test:</u>	No Device Attached
<u>Auxiliary Ports:</u>	No Device Attached

Test Steps

1. Enumerate and configure the Hub.
2. Connect port under test with the DS port of a SuperSpeed Hub using a standard A-A cable. NOTE: the second port can either be a different port on the HUT, or a port on a second hub.
3. Issue GetPortStatus.
4. Check that port under test state is as follows:

a.	PORT_POWER	1
b.	PORT_LINK_STATE	6 (SS.Inactive and DSPORT.Error) (10.16.2.6.2#20)
c.	C_PORT_CONFIG_ERROR	1 (10.16.2.6.2#20)
d.	PORT_CONNECTION	0 (10.16.2.6.2#2)

   Note: Transition from DSPORT.Enabled to DSPORT.Error by port config error.

5. Issue ClearPortFeature(C_PORT_CONFIG_ERROR)
6. Test fails if GetPortStatus() reports C_PORT_CONFIG_ERROR set to 1 (10.16.2.2#2)

Testing port reset:

7. Start capture.
8. Issue SetPortFeature(PORT_RESET) request
9. Immediately issue GetPortStatus.
10. If PORT_RESET bit is still set to 1, verify the following:

a.	PORT_CONNECTION	0 (10.16.2.6.2#2)
b.	PORT_LINK_STATE	5 (Rx.Detect)

   c. NOTE: If reset completion is very fast, PORT_RESET=1 may not be read even though GetPortStatus request is immediately issued.

11. Wait 360ms.

12. Repeat issuing GetPortStatus() requests to DS port under test and wait until the reset completes. Verify the following: (10.3.1.2#4)
    a. PORT_RESET 0
    b. PORT_CONNECTION 0.
13. Stop capture and verify that hub DS port issued warm reset (10.3.1.6#15)
14. Disconnect the standard A-A cable then reconnect after more than 1 second.
15. Issue GetPortStatus request to DS port under test and wait until it transitions to DSPORT.Error.
16. Start capture.
17. Issue SetPortFeature(BH_PORT_RESET) request
18. Immediately issue GetPortStatus request
19. If PORT_RESET bit is still set to 1, verify the following
    a. PORT_CONNECTION 0 (10.16.2.6.2#2)
    b. PORT_LINK_STATE 5 (Rx.Detect).
    c. NOTE: If reset completion is very fast, PORT_RESET=1 may not be read even though GetPortStatus request is issued immediately.

20. Wait 360ms
21. Repeat issuing GetPortStatus()request to DS port under test and wait until the reset completes. Verify the following: (10.3.1.2#4)
    a. PORT_RESET 0
    b. PORT_CONNECTION 0
22. Stop capture and verify that the hub DS port issued warm reset (10.3.1.6#15)

23. Disconnect standard A-A cable.
24. Issue GetPortStatus request to DS port under test and verify that C_PORT_CONFIG_ERROR is still set to one.
25. Issue Hot Reset to parent port of SS hub.
26. Enumerate and configure the SS hub. (Do not issue ClearPortFeature() request at this enumeration.)
27. Test fails if GetPortStatus request reports C_PORT_CONFIG_ERROR set to 1 (10.16.2.6.2#23)

28. Connect port under test with DSPort of an SS Hub by standard A-A cable.

29. Repeat issuing GetPortStatus request and wait until DS port under test transitions to DSPORT.Error.
30. Disconnect standard A-A cable.
31. Issue Warm Reset to parent port of SS hub.
32. Enumerate and configure the SS hub. (Do not issue ClearPortFeature() request at this enumeration.)
33. Test fails if GetPortStatus request reports C_PORT_CONFIG_ERROR set to 1 (10.16.2.6.2#23)

Testing in Powered off state:

34. Connect port under test with DSPort of an SS Hub by standard A-A cable.

35. Repeat issuing GetPortStatus request and wait until DS port under test transitions to DSPORT.Error.
36. Issue ClearPortFeature(PORT_POWER) request
37. Test fails if GetPortStatus request reports C_PORT_CONFIG_ERROR bit set to 1 (10.16.2.6.2#21) and PORT_POWER bit set to 1
38. Disconnect the standard A-A cable.

TD 10.14 Reset Powered-Off Ports Test

This test verifies that a reset sent to a powered-off DFP is ignored.

Assertions Covered

10.3.1.6#3

Starting Configuration

Port Under Test:	SS Device attached
Auxiliary Ports:	No Device Attached

Test Steps

1. Enumerate and configure the Hub.
2. Attach SS device to port under test.
3. Issue ClearPortFeature(PORT_POWER) to all DFPs.
4. Send SetPortFeature(PORT_RESET) to Hub for port under test.
5. Wait 1s.
6. Send a GetPortStatus request to the hub for the port under test.
7. Read the GetPortStatus response. The test fails if: (10.3.1.6#3)
    a. The GetPortStatus request fails
    b. PLS != 5
    c. PP != 0
    d. CCS != 0
    e. PR != 0
    f. PE != 0
    g. Any status change bit is set
8. Send SetPortFeature(BH_PORT_RESET) to Hub for port under test.
9. Wait 1s.
10. Send a GetPortStatus request to the hub for the port under test.
11. Read the GetPortStatus response. The test fails if: (10.3.1.6#3)
    a. The GetPortStatus request fails
    b. PLS != 5
    c. PP != 0
    d. CCS != 0
    e. PR != 0
    f. PE != 0
    g. Any status change bit is set

TD 10.15 Hub Port Numbering Test

This test verifies that USB 2 and Enhances SuperSpeed port numbers are reported correctly.

Assertions Covered

9.6.2.3#1, 9.6.2.3#2, 10.1.1#1, 10.3.3#1, 10.3.3#2

Starting Configuration

Port Under Test:	No Device Attached
Auxiliary Ports:	No Device Attached

Test Steps

1. Enumerate and configure the Hub.
2. Compare the bNbrPorts fields in the Hub's USB 2.0 and SuperSpeed Hub Descriptors.
    a. The test fails if either GetDescriptor call fails (10.1.1#1)
    b. The test fails if they are not equal in value.
3. Ask the user if the Hub's ports are labeled. Save the result.
4. Do the following steps for the number of ports that the Hub reports in the bNbrPorts field in its Hub Descriptors:
    a. If port number corresponding DeviceRemovable bit is 1, continue to next port.
    b. Prompt user to insert a 2nd USB Enhanced SuperSpeed Hub into a port on the Hub under test (HUT).
    c. Send GetPortStatus requests to both the USB 2 and Enhanced SuperSpeed portions of the HUT for each port.
    d. The test fails if:
        i. Anything other than one removable port reported an attached downstream device (CCS = 1) for the USB 2 portion of the HUT.
        ii. Anything other than one removable port reported an attached downstream device (CCS = 1) for the Enhanced SuperSpeed portion of the HUT.
        iii. The USB 2 and Enhanced SuperSpeed portions of the HUT reported a device attached on different port numbers. (10.3.3#2)
        iv. The USB 2 and Enhanced SuperSpeed portions of the attached do not report the same ContainerID. (9.6.2.3#1, 9.6.2.3#2)
        v. If the HUT's ports are labeled, ask the user whether the port number the 2nd Hub was inserted into matches the hub port label. The test fails if this value does not match the value that the HUT reports for the attached device. (10.3.3#1)

TD 10.16 Remote Wake Mask Reset Test

This test verifies that remote wake mask is reset to zero at initial start up or reset of the hub.

Assertions Covered

10.16.2.10#17

Starting Configuration

Port Under Test:	No Device Attached
Auxiliary Ports:	No Device Attached

Test Steps

1. Enumerate and configure the Hub.
2. Skip the following steps if the Remote Wakeup bit is set to zero in the bmAttributes field of the Configuration Descriptor.
3. Issue a SetPortFeature(REMOTE_WAKE_MASK) request to all DS ports with all the mask bits (Conn_RWEnable, Disconn_RWEnable and OC_RWEnable) are set to one.
4. Perform hot reset to the parent port of the hub.
5. Enumerate and configure the hub.
6. Issue a SetFeature(FUNCTION_SUSPEND) request to enable Function Remote Wake on hub and suspend the hub UFP link to U3.
7. For each removable DS port:
    a. Connect SS Device to DS port. Test fails if US port resume to U0. (10.16.2.10#17)
    b. Disconnect SS Device to DS port. Test fails if US port resume to U0. (10.16.2.10#17)
8. If the hub supports Over-current protection, for each removable DS port:
    a. Generate an Over-current event on DS port under test.
    b. Test fails if US port resumes to U0. (10.14.2.10#17)
9. Repeat from step 3 to step 8. In step 4, perform warm reset on the parent port of the hub instead of issuing hot reset.
10. If HUT is not embedded, repeat from step 3 to step 8. In step 4, power cycle the hub instead of issuing hot reset.

TD 10.17 Group Global Remote Wakeup Test

Assertions Covered

10.1.4#2, 2.2#40, 2.2#1

Starting Configuration

Port Under Test:	Remote Wake capable device attached	Port enabled
Auxiliary Ports:	Any device attached	Port enabled

Test Steps

1. Enumerate and configure the Hub.
2. Issue GetPortStatus() to all ports other than the PUT and save the statuses.
3. Check the number of populated ports on the HUT. The test fails (but continues) if:
    a. The number of populated ports != bNbrPorts from the HUT's Hub Descriptor
4. Check what the HUT reports for compound device and having non-removable device(s) in its hub descriptor. The test emits a warning but continues if:
    a. Compound device = 1 AND Non-removable device != 0.
5. Send a SetFeature.FUNCTION_SUSPEND(FUNCTION_REMOTE_WAKE = 1) to the HUT's interface 0.
6. Send SetFeature.FUNCTION_SUSPEND(FUNCTION_REMOTE_WAKE = 1) to the interface of the device connected to the PUT that supports remote wake. If the device is a hub, then also unmask remote wake for connect and disconnect for the PUT.
7. Issue GetPortStatus() to the PUT and save the status.
8. Issue SetPortFeature(PORT_LINK_STATE = U3) to the HUT for all populated ports.
9. Wait for the PUT to enter U3.
10. Put the HUT's UFP link into U3.
11. Wait for the HUT's UFP link to get to U3.
12. Prompt the user to initiate remote wake from the device connected to the PUT. The test fails if:
    a. The host does not detect a remote wake event within the timeout period or before the user clicks "OK". (10.1.4#2)
13. Read the HUT's status change endpoint. The test fails if:
    a. The HUT doesn't respond with NRDY.
14. Issue GetPortStatus() to the PUT. The test fails if:
    a. CCS != 1
    b. PE != 1
    c. PP != 1
    d. C_PLS != 0
    e. C_BH_RESET != 0
    f. C_CCS != 0
    g. C_OC != 0
    h. PORT_CONFIG_ERROR != 0
    i. PORT_RESET != 0
    j. C_PORT_RESET != 0
    k. OVERCURRENT != 0
    l. PLS != 0
    m. PORT_SPEED != 0
15. Issue GetPortStatus() for all ports other than the PUT. The test fails if:
    a. Any of the status bits changed compared to step (2). (2.2#1)

TD 10.18 Group Remote Wakeup Test

This test verifies that.

Assertions Covered

Starting Configuration

Port Under Test:	Remote Wake capable device attached	Port enabled
Auxiliary Ports:	Any device attached	Port enabled

Test Steps

1. Enumerate and configure the Hub.
2. Issue GetPortStatus() to all ports other than the PUT and save the statuses.
3. Send a SetFeature.FUNCTION_SUSPEND(FUNCTION_REMOTE_WAKE = 1) to the HUT's interface 0.
4. Send SetFeature.FUNCTION_SUSPEND(FUNCTION_REMOTE_WAKE = 1) to the interface of the device connected to the PUT that supports remote wake. If the device is a hub, then also unmask remote wake for connect and disconnect for the PUT.
5. Issue SetPortFeature(PORT_LINK_STATE = U3) to the HUT for the PUT.
6. Wait for the PUT to enter U3.
7. Prompt user to initiate remote wake on the device connected to the PUT.
8. The test fails if:
    a. The host does not detect a remote wake event within the timeout period or before the user clicks "OK".
9. Read the HUT's status change endpoint. The test fails if:
    a. The HUT doesn't respond with NRDY.
10. Issue GetPortStatus() to the PUT. The test fails if:
    a. CCS != 1
    b. PE != 1
    c. PP != 1
    d. C_PLS != 0
    e. C_BH_RESET != 0
    f. C_CCS != 0
    g. C_OC != 0
    h. PORT_CONFIG_ERROR != 0
    i. PORT_RESET != 0
    j. C_PORT_RESET != 0
    k. OVERCURRENT != 0
    l. PLS != 0
    m. PORT_SPEED != 0
11. Issue GetPortStatus() for all ports other than the PUT. The test fails if:
    a. Any of the status bits changed compared to step (2).

TD 10.19 Port Remote Wakeup Test

Assertions Covered

2.2#1, 2.2#40

Starting Configuration

Port Under Test:	Remote Wake capable device attached	Port enabled
Auxiliary Ports:	No device attached	Port disabled

Test Steps

1. Enumerate and configure the Hub.
2. Issue GetPortStatus() to all ports other than the PUT and save the statuses.
3. Send a SetFeature.FUNCTION_SUSPEND(FUNCTION_REMOTE_WAKE = 1) to the HUT's interface 0.
4. Send SetFeature.FUNCTION_SUSPEND(FUNCTION_REMOTE_WAKE = 1) to the interface of the device connected to the PUT that supports remote wake. If the device is a hub, then also unmask remote wake for connect and disconnect for the PUT.
5. Issue SetPortFeature(PORT_LINK_STATE = U3) to the HUT for the PUT.
6. Wait for the PUT to enter U3.
7. Prompt user to initiate remote wake on the device connected to the PUT.

8. The test fails if:
    a. The host does not detect a remote wake event within the timeout period or before the user clicks “OK”.
9. Read the HUT’s status change endpoint. The test fails if:
    a. The HUT doesn’t respond with NRDY.
10. Issue GetPortStatus() to the PUT. The test fails if:
    a. CCS != 1
    b. PE != 1
    c. PP != 1
    d. C_PLS != 0
    e. C_BH_RESET != 0
    f. C_CCS != 0
    g. C_OC != 0
    h. PORT_CONFIG_ERROR != 0
    i. PORT_RESET != 0
    j. C_PORT_RESET != 0
    k. OVERCURRENT != 0
    l. PLS != 0
    m. PORT_SPEED != 0
11. Issue GetPortStatus() for all ports other than the PUT. The test fails if:
    a. Any of the status bits changed compared to step (2). (2.2#1)

TD 10.20 Port Hot Plug Test

Assertions Covered

10.13.6#2

Starting Configuration

Port Under Test:	No Device Attached	Port enabled
Auxiliary Ports:	No Device Attached	Port disabled

Test Steps

1. Enumerate and configure the Hub.
2. Issue GetPortStatus() to all ports other than the PUT and save the statuses.
3. Prompt user to connect a SS device to the PUT. The test fails if:
    a. The hub does not report status change for the plug on its status change endpoint. (#10.11.6.2)
4. Issue GetPortStatus() for all ports other than the PUT. The test fails if:
    a. Any of the status bits for any of these ports changed.
5. Issue GetPortStatus() for the PUT. The test fails if:
    a. PP != 1
    b. CCS != 1
    c. PE != 1
    d. Port Speed != 0
    e. OC != 0
    f. PLS != 0
    g. C_CCS != 1
    h. C_PLS != 0
    i. C_OC != 0
    j. C_PORT_CONFIG_ERROR != 0

TD 10.21 Port Power Off Suspended Port Test

Assertions Covered

10.16.2.2#1

Starting Configuration

Port Under Test:	Any device attached	Port enabled
Auxiliary Ports:	No Device Attached	Port disabled

Test Steps

1. Enumerate and configure the Hub.
2. Issue GetPortStatus() to all ports other than the PUT and save the statuses.
3. Issue SetPortFeature(PLS) = 3 for the PUT.
4. Issue GetPortStatus for the PUT. The test fails if:
    a. PLS !=3
5. Issue ClearPortFeature(PORT_POWER) for the PUT.
6. Issue GetPortStatus() for all ports other than the PUT. The test fails if:
    a. Any of the status bits for any of these ports changed.
7. Issue GetPortStatus() for the PUT. The test fails if: (10.14.2.2#1)
    a. PP != 0
    b. CCS != 0
    c. PE != 0
    d. PR != 0
    e. OC != 0
    f. PLS != 5
    g. C_CCS != 0
    h. C_PLS != 0
    i. C_OC != 0
    j. C_PORT_CONFIG_ERROR != 0
    k. C_PR != 0
    l. C_BH_PR != 0

TD 10.22 Port Detach Suspended Device Test

Assertions Covered

2.1#1, 2.1#410.13.6#3, 1.2#1

Starting Configuration

Port Under Test:	Any device attached	Port enabled
Auxiliary Ports:	No Device Attached	Port disabled

Test Steps

1. Enumerate and configure the Hub.
2. Issue GetPortStatus() to all ports other than the PUT and save the statuses.
3. Issue SetPortFeature(PORT_LINK_STATE = U3) to the HUT for the PUT.
4. Wait for the PUT to enter U3.
5. Prompt user to disconnect device from the PUT. The test fails if:
    a. A status change notification is not detected from the HUT. (#10.13.6.3)
6. Issue GetPortStatus() for all ports other than the PUT. The test fails if:
    a. Any of the status bits changed compared to step (2).
7. Issue GetPortStatus() to the PUT. The test fails if: (#10.3.1.2.1)
    a. CCS != 0
    b. PE != 0
    c. PP != 1
    d. C_PLS != 0
    e. C_BH_RESET != 0
    f. C_CCS != 1
    g. C_OC != 0
    h. PORT_CONFIG_ERROR != 0
    i. PORT_RESET != 0
    j. C_PORT_RESET != 0
    k. OVERCURRENT != 0

        l. PLS != 5

TD 10.23 Over-Current Test

This test verifies that an Over-Current condition on a downstream port is handled correctly.

Required Device Resource

Over-current fixture

Assertions Covered

10.12.1#3 - 10.12.1#5, 10.13.5#1 - 10.13.5#3, 10.16.2.4#11, 10.16.2.4#13, 10.16.2.6#7, 10.16.2.6#8, 10.16.2.6.2#4, 10.16.2.6.2#5, 10.16.2.6.2#10, 10.16.2.6.2#14, 10.16.2.6.2#19, 10.16.2.6.2#21.

Starting Configuration

Port Under Test:	No device attached	Port enabled
Auxiliary Ports:	No device attached	Port enabled

Test Steps

*A port is in a **powered-off** state, if all of the following are true:*

- *PORT_POWER = 0*
- *CURRENT_CONNECT_STATUS = 0*
- *PORT_RESET = 0*
- *PORT_ENABLE = 0*
- *PORT_LINK_STATE = 4 or 5.*
- *C_PORT_CONNECTION = 0 (10.16.2.6.2#4)*
- *C_PORT_RESET = 0 (10.16.2.6.2#10)*
- *C_PORT_BH_RESET = 0 (10.16.2.6.2#14)*
- *C_PORT_LINK_STATE = 0 (10.16.2.6.2#19)*
- *C_PORT_CONFIG_ERROR = 0 (10.16.2.6.2#21)*

1. Enumerate and configure the Hub.
2. Determine whether HUT is self-powered.
    a. Get the Config descriptor for HUT.
        i. If bit D6 of bmAttributes is 1, then the device is SELF-POWERED.
        ii. If bit D6 of bmAttributes is 0, then the device is BUS-POWERED.
3. Determine Over-Current Protection mode:
    a. Get the Hub Descriptor for HUT. Examine bits D4:D3 of wHubCharacteristics:
        i. If they are 00, then HUT supports OC protection on a GLOBAL (HUB) BASIS.
        ii. if they are 01, then HUT supports OC protection on a PER-PORT BASIS.
        iii. If they are 10 or 11, then HUT DOES NOT SUPPORT OC protection

4. If HUT does not support OC protection, then exit the test.
    a. If HUT is self-powered then the test fails. (10.13.5#3).

5. Prompt user to generate Over-current condition on Port Under Test.
6. Wait for change notification from HUT

7. If HUT supports OC protection on a global basis:
    a. Do GetHubStatus()..
        i. Test fails if C_HUB_OVER_CURRENT is not 1. (10.16.2.4#13)
        ii. Do ClearHubFeature(C_HUB_OVER_CURRENT).
    b. For each port, do GetPortStatus().
        i. Test fails if bit PORT_OVER_CURRENT is not 0. (10.13.5#2)
        ii. Test fails if C_PORT_OVER_CURRENT is not 0. (10.13.5#2)
        iii. Test fails if port is not in a powered-off state. (10.13.5#1)

        - iv. If necessary, send ClearPortFeature(C_PORT_OVER_CURRENT) to port.

8. If HUT supports OC protection on a per-port basis:
    - a. Do GetPortStatus() for Port Under Test:
        - i. Test fails if C_PORT_OVER_CURRENT is not 1. (10.12.1#3, 10.12.1#5, 10.16.2.6.2#5)
        - ii. Test fails if port is not in a powered-off state. (10.12.1#4)
    - b. For all other ports:
        - i. Do GetPortStatus()
        - ii. if C_PORT_OVER_CURRENT is 1, then test fails if port is not in a powered-off state. (10.12.1#4)
        - iii. if C_PORT_OVER_CURRENT is 1, then do ClearPortFeature(C_PORT_OVER_CURRENT) for that port.
    - c. Do GetHubStatus():
        - i. Test fails if HUB_OVER_CURRENT is not 0. (10.16.2.4#11)
        - ii. Test fails if C_HUB_OVER_CURRENT is not 0. (10.16.2.4#13)
        - iii. If necessary, do ClearHubFeature(C_HUB_OVER_CURRENT).

9. Prompt user to remove Over-current condition.

10. If HUT supports OC protection on a global basis:
    - a. Do GetHubStatus()..
        - i. Test fails if HUB_OVER_CURRENT is 1.
        - ii. Do ClearHubFeature(C_HUB_OVER_CURRENT).
    - b. For each port, do GetPortStatus().
        - i. Test fails if bit PORT_OVER_CURRENT is not 0. (10.13.5#2)
        - ii. Test fails if C_PORT_OVER_CURRENT is not 0. (10.13.5#2)
        - iii. If necessary, send ClearPortFeature(C_PORT_OVER_CURRENT) to port.

11. If HUT supports OC protection on a per-port basis:
    - a. For each port, do GetPortStatus():
        - i. Test fails if bit PORT_OVER_CURRENT is 1. (10.16.2.6#7, 10.16.2.6#8)
        - ii. If C_PORT_OVER_CURRENT is 1, then do ClearPortFeature(C_PORT_OVER_CURRENT).
    - b. Do GetHubStatus():
        - i. Test fails if HUB_OVER_CURRENT is not 0. (10.16.2.4#11)
        - ii. Test fails if C_HUB_OVER_CURRENT is not 0. (10.16.2.4#13)
        - iii. If necessary, do ClearHubFeature(C_HUB_OVER_CURRENT).

12. For each port, issue a SetPortFeature(PORT_POWER).
13. Find the corresponding USB 2 hub.
    - a. Issue a GetHubStatus().
        - i. If necessary, do a ClearHubFeature(C_HUB_OVER_CURRENT).
14. For each port on the USB 2 hub:
    - a. Issue a GetPortStatus().
        - i. If necessary do a ClearPortFeature(C_PORT_OVER_CURRENT).
    - b. For each port, issue a SetPortFeature(PORT_POWER).

15. Repeat for all accessible ports.

TD 10.24 Non-Removable Device Notification Test

This test verifies that the Hub Under Test correctly identifies those ports that have non-removable devices attached.

Assertions Covered

10.15.2.1#20, 10.15.2.1#21.

Starting Configuration

<u>Port Under Test:</u> No device attached Port enabled

Auxiliary Ports:	No device attached	Port enabled

Test Steps

1. Enumerate and configure the Hub.
2. Prompt user to provide the port numbers of the ports with non-removable devices attached, or that are otherwise inaccessible.
3. Do a GetDescriptor(HUB) request.
4. Compare the values given in step 2 with the values given in the *wDeviceRemovable* field of the Hub Descriptor. For each port:
    a. If the descriptor states that it has a non-removable device attached or is otherwise inaccessible, and the user states that it does NOT, then the test fails. (10.15.2.1#20, 10.15.2.1#21)
    b. If the descriptor states that the port is accessible and/or does NOT have a non-removable device attached, and the user states that it does, then test provides warnings. (10.15.2.1#20, 10.15.2.1#21)

TD 10.25 Unacknowledged Connect Remote Wake Test

This test verifies that a Suspended hub with a downstream device with an unacknowledged port connect change will wake up the host.

Assertions Covered

Assertions….

Starting Configuration

Port Under Test:	No device attached	Port enabled
Auxiliary Ports:	No device attached	Port enabled

Test Steps

1. Enumerate and Configure the Hub.
2. Prompt user to attach device to Port Under Test.
3. Call GetPortStatus() on the Port Under Test.
    a. Test fails if PORT_CONNECTED and C_PORT_CONNECTED are not set.
4. Call GetPortStatus() on the other ports.
    a. Test fails if any of the change bits are set.
5. Enable Remote Wakeup on Connect on the Port Under Test, by calling SetPortFeature(PORT_REMOTE_WAKE_MASK) with bit 0 set to 1.
    a. Test fails if this does not succeed.
6. Enable Function Remote Wake on Hub Under Test and also do a FUNCTION_SUSPEND by calling SetFeature(FUNCTION_SUSPEND) with Interface set to 0, and Suspend Options set to 3.
7. Suspend HUT by putting its parent port into U3.
8. Wait for HUT to generate Remote Wake on upstream port.
    a. Test fails if this does not happen in 5 seconds.

Repetitions

Repeat for all ports.

TD 10.26 Gen2 Hub Upstream Port Operating at Gen1 Test

This test verifies that the downstream port of a Gen2 capable hub will only operate a speed less than or equal to the upstream hub port.

Assertions Covered

10.1#3, 10.1.1#3, 10.1.1#4

Starting Configuration

Port Under Test:	SuperSpeedPlus Device

Auxiliary Ports: No Device Attached

Test Steps

1. Enumerate and configure the Hub Under Test.
2. If Hub Under Test is not Gen2 capable, then record success and exit.
3. If parent port of Hub Under Test is not a host port, then communicate to user that this test requires that the Hub Under Test must be directly plugged into a host port and then record success and exit.
4. If parent port of Hub Under Test is operating at Gen2 speed, then communicate to user that this test requires a Gen1 speed parent port and then record success and exit.
5. Prompt user to attach an Enhanced SuperSpeed device with Gen2 capability to a hub DS Port Under Test.
6. Enumerate the attached device.
7. Test fails if DS device does not enumerate as an Enhanced SuperSpeed device. (10.1.1#4)
8. Test fails if DS device did not enumerate at Gen1 speed as described in the following steps. (10.1 #3)
    a. Read the BOS device descriptor
    b. Find the SSP USB Device Capability Descriptor.
    c. Save the bmSublinkSpeedAttr array for reference.
    d. Read Extended Port Status bits for the port under test.
    e. Using the RX_SUBLINK_SPEED_ID from the Extended Port Status Bits:
        i. Locate the bmSublinkSpeedAttr with a matching SSID.
        ii. If bmSublinkSpeedAttr.LSE != 3 (Gb/s), test fails.
        iii. If bmSublinkSpeedAttr.LSM != 5 (Gen1), test fails.
    f. Using TX_SUBLINK_SPEED_ID from the Extended Port Status Bits:
        i. Locate the bmSublinkSpeedAttr with matching SSID.
        ii. If bmSublinkSpeedAttr LSE != 3 (Gb/s), test fails.
        iii. if bmSublinkSpeedAttr LSM != 5 (Gen1), test fails.
    g. Test fails if RX_LANE_COUNT != 0.
    h. Test fails if TX_LANE_COUNT != 0.

TD 10.27 Gen2 Downstream Facing Ports Test

This test verifies that the downstream port of a Gen2 capable hub will only operate a speed less than or equal to the upstream hub port.

Assertions Covered

8.5.6.7#7, 10.1.1#4, 10.16.2.6.3#3 - 10.16.2.6.3#7

Starting Configuration

Port Under Test: SuperSpeedPlus Device

Auxiliary Ports: No Device Attached

Test Steps

1. Enumerate and configure Hub Under Test.
2. If parent port of Hub Under Test is not operating at Gen2, then record success and exit.
3. Read the full BOS descriptor of Hub Under Test.
4. Look for the SSP USB Device Capability Descriptor.
    a. If there is no SSP USB Device Capability Descriptor, then Hub Under Test is not Gen2 capable. Record success and exit.
5. Save the bmSublinkSpeedAttr array for reference.
6. Prompt user to attach Gen2 capable device to Port Under Test.
7. Enumerate DS device.
8. Test fails if DS device does not enumerate at Gen2 as described in the following steps. (10.1.1#4)
    a. Read Extended Port Status bits for the port under test.
    b. If bits 12-31 of the extended Port Status bits are not equal to 0, then test fails. (10.16.2.6.3#7)
    c. Using the RX_SUBLINK_SPEED_ID from the Extended Port Status Bits, locate the bmSublinkSpeedAttr with a matching SSID.
        i. If no such bmSublinkSpeedAttr exists, then test fails. (10.16.2.6.3#3)
        ii. If bmSublinkSpeedAttr.LSE != 3 (Gb/s), test fails.
        iii. If bmSublinkSpeedAttr.LSM != 10 (Gen2), test fails.
    d. Using TX_SUBLINK_SPEED_ID from the Extended Port Status Bits, locate the bmSublinkSpeedAttr with matching SSID.

            i. If no such bmSublinkSpeedAttr exists, then test fails. (10.16.2.6.3#4)
            ii. If bmSublinkSpeedAttr LSE != 3 (Gb/s), test fails.
            iii. If bmSublinkSpeedAttr.LSM != 10 (Gen2), test fails.
    e. Test fails if RX_LANE_COUNT != 0. (8.5.6.7#8, 10.16.2.6.3#5)
    f. Test fails if TX_LANE_COUNT != 0. (8.5.6.7#8, 10.16.2.6.3#6)
    g. Test fails if RX_LANE_COUNT != TX_LANE_COUNT. (8.5.6.7#7)
9. Prompt user to detach DS device.
10. Prompt user to attach Enhanced SuperSpeed device which is not Gen2 capable.
11. Enumerate DS device.
12. Test fails if DS device does not enumerate at Gen1 as described in the following steps. (10.1.1#4)
    a. Read Extended Port Status bits for the port under test.
    b. If bits 12-31 of the extended Port Status bits are not equal to 0, then test fails. (10.16.2.6.3#7)
    c. Using the RX_SUBLINK_SPEED_ID from the Extended Port Status Bits, locate the bmSublinkSpeedAttr with a matching SSID.
        i. If no such bmSublinkSpeedAttr exists, then test fails. (10.16.2.6.3#3)
        ii. If bmSublinkSpeedAttr.LSE != 3 (Gb/s), test fails.
        iii. If bmSublinkSpeedAttr.LSM != 5 (Gen1), test fails.
    d. Using TX_SUBLINK_SPEED_ID from the Extended Port Status Bits, locate the bmSublinkSpeedAttr with matching SSID.
        i. If no such bmSublinkSpeedAttr exists, then test fails. (10.16.2.6.3#4)
        ii. If bmSublinkSpeedAttr LSE != 3 (Gb/s), test fails.
        iii. If bmSublinkSpeedAttr.LSM != 5 (Gen1), test fails.
    e. Test fails if RX_LANE_COUNT != 0. (8.5.6.7#8, 10.16.2.6.3#5)
    f. Test fails if TX_LANE_COUNT != 0. (8.5.6.7#8, 10.16.2.6.3#6)
    g. Test fails if RX_LANE_COUNT != TX_LANE_COUNT. (8.5.6.7#7)

Repetitions

Repeat for all ports.

TD 10.28 Extended Port Status Bits Test

This test verifies that the Hub Under Test returns the extended port status bits when appropriately requested.

Assertions Covered

10.16.2.6.3#1.

Starting Configuration

Port Under Test:	No Device Attached.
Auxiliary Ports:	No Device Attached

Test Steps

1. Enumerate and configure Hub Under Test.
2. Read the full BOS descriptor of Hub Under Test.
3. Look for the SSP USB Device Capability Descriptor.
    a. If there is no SSP USB Device Capability Descriptor, then Hub Under Test does not support Extended Port Status bits. Record success and exit.
4. Send a GetPortStatus() request to the Hub Under Test, wherein the Port Status Type is set to 0, the Port is set to the port number of the Port Under Test, and Port Status Length is set to 8.
    a. Test fails if request does not return 4 bytes. (10.16.2.6.3#1)
5. Send a GetPortStatus() request to the Hub Under Test, wherein the Port Status Type is set to 2, the Port is set to the port number of the Port Under Test, and Port Status Length is set to 8.
    a. Test fails if request does not return 8 bytes. (10.16.2.6.3#1)
6. Prompt tester to attach Enhanced SuperSpeed device to Port Under Test.
7. Repeat steps 4-5.

Repetitions

Repeat for all ports.

TD 5.1 Control Loopback Test

This test performs functional testing of control transfers.

Required Device Resource

3.0 compliance device

Overview of Test Steps

The test performs the following steps.

1. Initialize and start the xHC.
2. Prompt a user to attach a USB compliance device to a port of the HUT the host if it is not attached. HUT must be directly attached to the Host.
3. Configure the compliance device for control endpoint loopback.
4. Schedule a control write transfer to the control endpoint using compliance device vendor command.
5. Check the result of the transfer.
6. Schedule a control read transfer to the control endpoint using compliance device vendor command.
7. Check the result of the transfer.
8. Verify that the OUT data matches the IN data (same data in the same order).

Repetitions

Repeat for all allowed control transfer sizes (from 1 to EP0 MaxPacketSize).
And repeat for all allowed max packet size for control transfers:

- For SS: 512 bytes

And repeat for all byte repeating data patterns.
Run test with and without Event Data TRB.

TD 5.2 Bulk Loopback Tests

This test performs functional testing of bulk transfers.

Required Device Resource

3.0 compliance device

Overview of Test Steps

The test performs the following steps.

1. Initialize and start the xHC.
2. Prompt a user to attach a USB compliance device to a port of the HUT the host if it is not attached. HUT must be directly attached to the Host.
3. Configure the compliance device for bulk endpoint loopback.

TD 5.2.1 Bulk Loopback using Scatter-Gather List

4 Perform loopback using scatter-gather list of data buffers.
    4.1 Schedule a bulk OUT transfer to the bulk OUT endpoint.
    4.2 Check the result of the transfer.
    4.3 Schedule a bulk IN transfer to the bulk IN endpoint.
    4.4 Check the result of the transfer.
    4.5 Repeat for 10 iterations.
    4.6 Verify that the OUT data matches the IN data (same data in the same order).

TD 5.2.2 Bulk Loopback using Contiguous Data Buffer

5 Perform loopback using a physically contiguous data buffer.
    5.1 Schedule a bulk OUT transfer to the bulk OUT endpoint.
    5.2 Check the result of the transfer.
    5.3 Schedule a bulk IN transfer to the bulk IN endpoint.
    5.4 Check the result of the transfer.
    5.5 Repeat for 10 iterations.

Repetitions

Transfer size for scatter-gather iterations is 16K for SS.
Transfer size for contiguous buffer iteration is 64K.
And repeat for all possible page offset (from 0 to 4095) for scatter-gather iterations.
And repeat for all allowed max packet size for bulk transfers:

- For SS: 1024 bytes

And repeat for all allowed max burst size for SS bulk transfers (from 1 to 16).
And repeat for all byte repeating data patterns.
Run test with and without Event Data TRB.

TD 5.3 Interrupt Loopback Test

This test performs functional testing of interrupt transfers.

Required Device Resource

3.0 compliance device

Overview of Test Steps

The test performs the following steps.

1 Initialize and start the xHC.
2 Prompt a user to attach a USB compliance device to a port of the HUT the host if it is not attached. HUT must be directly attached to the Host.
3 Configure the compliance device for interrupt endpoint loopback.

TD 5.3.1 Interrupt Loopback using Scatter-Gather List

4 Perform loopback using scatter-gather list of data buffers.
  4.1 Schedule an interrupt OUT transfer to the interrupt OUT endpoint.
  4.2 Check the result of the transfer.
  4.3 Schedule an interrupt IN transfer to the interrupt IN endpoint.
  4.4 Check the result of the transfer.
  4.5 Verify that the OUT data matches the IN data (same data and in the same order).
  4.6 Repeat for 10 iterations.

TD 5.3.2 Interrupt Loopback using Contiguous Data Buffer

5 Perform loopback using physically contiguous data buffer.
  5.1 Schedule a bulk OUT transfer to the bulk OUT endpoint.
  5.2 Check the result of the transfer.
  5.3 Schedule a bulk IN transfer to the bulk IN endpoint.
  5.4 Check the result of the transfer.
  5.5 Repeat for 10 iterations.

Repetitions

Transfer size for scatter-gather iterations are 16K for SS.
Transfer size for contiguous buffer iteration is 64K.
And repeat for all possible page offset (from 0 to 4095) for scatter-gather iterations.
And repeat for all allowed max packet size for interrupt transfers (1 to 1024, dependent on burst size).
And repeat for all allowed max burst size for SS or HS interrupt transfers (1 to 3).
And repeat for all byte repeating data patterns.
Run test with and without Event Data TRB.

TD 5.4 Isochronous Loopback Test

This test performs functional testing of isochronous transfers.

Required Device Resource

3.0 compliance device

Overview of Test Steps

The test performs the following steps.

1 Initialize and start the xHC.
2 Prompt a user to attach a USB compliance device to a port of the HUT the host if it is not attached. HUT must be directly attached to the Host.

TD 5.4.1 Test isochronous OUT.

- 2.1 Configure compliance device for isochronous endpoint OUT looped back to bulk IN.
- 2.2 Write data buffer of MaxPacketSize * Burst. This will be contained in 1 isochronous TD.
- 2.3 Read data buffer back.
- 2.4 Verify that the OUT completed successfully. xHCI *CompletionCode* field should be set to TrbSuccess.
- 2.5 Verify that the IN completed successfully.
- 2.6 Verify that the OUT data matches the IN data (same data and in same order).
- 2.7 Verify that the host generated a Transfer Event with *CompletionCode* set to Ring Underrun for the OUT endpoint (4.10.3.1#2).
- 2.8 Verify that the EP State field is set to Running in the EndpointContext for the OUT endpoint (4.11.2.3#1).
- 2.9 Repeat for 10 iterations.
- 2.10 Add steps for 4.11.2.3#5 and 4.11.2.3#6

TD 5.4.3 Test Isochronous Streaming OUT

- 2.11 Configure compliance device for isochronous endpoint OUT looped back to bulk IN.
- 2.12 Stream continuous isochronous traffic of MaxPacketSize * Burst for 30 minutes.
- 2.13 Verify that the Isochronous OUT completed successfully.
- 2.14 Verify that bulk INT data matches the IN data.
- 2.15 Test fails if any Ring Overrun or Ring Underrun events are generated.
- 2.16 After last Isochronous packet verify that a Ring Underrun is generated.

Repetitions

Repeat for transfer size of (MaxPacketSize * Burst / 2) + 1.
And repeat for all allowed max packet size for isochronous transfers:
And repeat for all allowed max burst size for SS or HS isochronous transfers (1 to 16 for SS and 1 to 3 for HS).
And repeat for all allowed number of bursts (mult) for SS.
Run test with and without Event Data TRB.

TD 5.19 Mixed Traffic Test

This test generates concurrent traffic of different types and speeds.

Required Device Resource

USB 2.0 and 3.0 compliance devices

Overview of Test Steps

The test performs the following steps:

1 Initialize and start the xHC.
2 Prompt a user to attach a LS, FS, HS and SS compliance devices below a SS hub.
3 Concurrently run the following loopback tests:
    3.1 TD 5.1 Control Loopback Test on LS compliance device
    3.2 TD 5.2 Bulk Loopback Test on FS compliance device
    3.3 TD 5.3 Interrupt Loopback Test on HS compliance device – subset of full test
    3.4 TD 5.4 Isochronous Loopback Test on SS compliance device – subset of full test
4 Test fails if any of these loopback tests fail.

Repetitions

Repeat with FS: Interrupt loopback, HS: Isochronous Loopback, SS: Bulk Loopback.
Repeat with FS: Isochronous loopback, HS: Bulk Loopback, SS: Interrupt Loopback.