This proposal represents work in progress for the P1394pm Suspend/Resume effort currently underway. A great deal of the content of this proposal has a direct impact on PHY silicon. Though this work has not yet been completed, sufficient content contained herein will provide the reader with a firm understanding of the mechanism required to support suspend and resume on a port-to-port and a bus wide basis.

Our many thanks to those who, so willingly, gave of their time to assist in making this effort what it is so far!

The P1394pm Working Task Force expects this work to be complete before weeks end (July 24th, 1997) and therefore encourage one and all to provide their feedback as soon as possible.

Best Regards,

Steve Bard
p1394pm Acting Editor

6.7 Port Suspend

Port suspend mechanisms provide a facility for implementing a power conservation state while maintaining a port-to-port connection between a (possible) active bus segment and a (possible) inactive bus segment. While in this state, a port is unable to transmit or receive data transaction packets. However, a suspended port is capable of generating or receiving a resume event - restoring the port-to-port connection to normal, full-power, operation.

The glossary table below provides definitions of specific terms unique to the suspend and resume mechanism and protocol process:

Suspend refers to a low power state in which normal data processing is not possible.
Resume: refers to a state of transition from suspend to full power, normal data processing capability.
Suspend initiator: refers to a port in a node which has been selected to start (initiate) the suspended-connected state protocol process between itself and the port to which it is connected.
Suspend target: refers to the port connected to a suspend initiator and is the recipient (target) of the initiated suspended-connected protocol.
Resume initiator: a port which starts (initiates) a resume (exit from suspend) event to an attached, suspended, port and to any other suspended ports in the node of which it is resident.
Resume target: refers the suspended port connected to a port from which a resume event is generated.
Suspended-connected: a low power state connection between two nodes.
Suspend Sub-Net: three nodes (minimum) in which one node has at least two suspended-connected ports - one to each of the other two nodes
Suspend Domain: one or more suspend sub-nets joined via a port-to-port connection in a suspended-connected state.
Bus Manager: within the context of suspend/resume, the Bus Manager includes suspend management functionality.
Boundary Node: refers to a node with one or more connected and active ports and only one port in a suspended-connected state.

6.7.1 Suspended PHY Port Behavior

A leakage current is supplied via TPA* on one suspended-connected port to another suspended-connected port's TPB* through a cable connection. The TPA* side of the port-to-port connection is connected to a voltage-sensing circuit. If the cable connection is removed, the voltage input to the sensing circuit will rise, generating a disconnect notification.

An informative example follows (actual circuit implementation may vary):

Leakage current is supplied through a cable connection to a destination port's TPB* from a 100 kilohm resistor connected between an *always on* power rail and the output of the disabled TpBias generator. The input to a schmidt trigger buffer/inverter is connected to the junction of the 100 kilohm resistor and the TpBIAS generator output. This mechanism will create a signal (active low) on the output of the buffer/inverter (monitored only during a suspended-connected state when the TpBIAS generator is disabled) when the cable connection between the target and initiator is opened (see figure 6.11).

While in the suspended-connected state, a node's port is configured as follows (refer to section xxxx for a detailed description of a node's port configuration and control registers):

— The port drivers are in the high impedance state.
— The TpBIAS generators are turned off

— The port differential receivers are in a disabled state, including the arbitration, connection and speed comparators.
— The port common-mode connection status bit, Con, remains set (logic 1) and the associated port variable maintained by the PHY connection state machine indicates a connected port.
— The port disabled status bit, ***Disable***, is reset (logic 0 - i.e. port enabled).
— The port suspended status bit, ***Suspend***, is set (logic 1 - i.e. port is suspended)
— The port resume enable bit, ***Resume_Disable*** may be either state (logic 1 or 0) depending upon the desired action to take upon detection of a resume event.
— The port suspend propagation block bit, ***Suspend_Block*** may be either state (logic 1 or 0) depending upon the whether this port had been configured to propagate the suspended-connected event.
— The port suspend enable bit, ***Suspend_Disable*** is reset (logic 0 - i.e. the port is enabled to respond to suspended-connected event notification).

6.7.2 Suspend Model and Process Overview

The IEEE 1394-1995.A suspend mechanism allows pairs of directly-connected ports to be put into a low-power, suspended-connected, state. Any connected and active port may receive notification to either initiate or enter a suspended-connected state. Notification to a port may be originated from the bus manager via an extended PHY packet, or from the link (connected to the PHY in which the port resides) via a PHY register write. The port notified by the bus manager or link becomes the suspend initiator. The initiator notifies the port to which it is connected (the target) to enter into the suspended-connected state.

The bus manager configures a port to become a suspend initiator by generating an extended PHY-packet of type “PHY register write” with the node ID and port ID or the port chosen to become the initiator and the port register ID value of 100b (register value is *don’t care* (e.g. it will be ignored). Bus manager originated suspend initiator selection can be sent to any connected and active port on the bus. Link originated suspend initiator selection can only be generated to connected and active ports on the PHY associated with the link generating the initiator selection.

A suspend sub-net is created when a port in a suspend-connected state is connected to a port on a node containing a minimum of two ports in a suspended-connected state - one to which it is connected and one which is connected to a port on a third node (also in a suspended-connected state).

A suspend domain is created when one (or more) suspend sub-nets are connected via a port-to-port connection in a suspended-connected state.

A suspend target will propagate suspend initiation to all active (not suspended) ports in the node in which it resides - dependent upon the state of the target port’s ***Suspend_Block*** bit.

Ports in a suspended-connected state within a suspend sub-net or suspend domain may be restored to normal operation via a “resume” event - initiated by any port in a suspended-connected state.

A port in a suspended-connected state which detects a resume event becomes a resume target and will propagate the resume event to all other ports in its node that are in a suspended-connected state - depending upon the state of the resume target port’s ***Resume_Disable*** bit.

6.7.3 Port Suspend Circuitry

Figure 6.11 shows the cable media signal interface configuration - including circuitry used while a port is in the suspended-connected state.

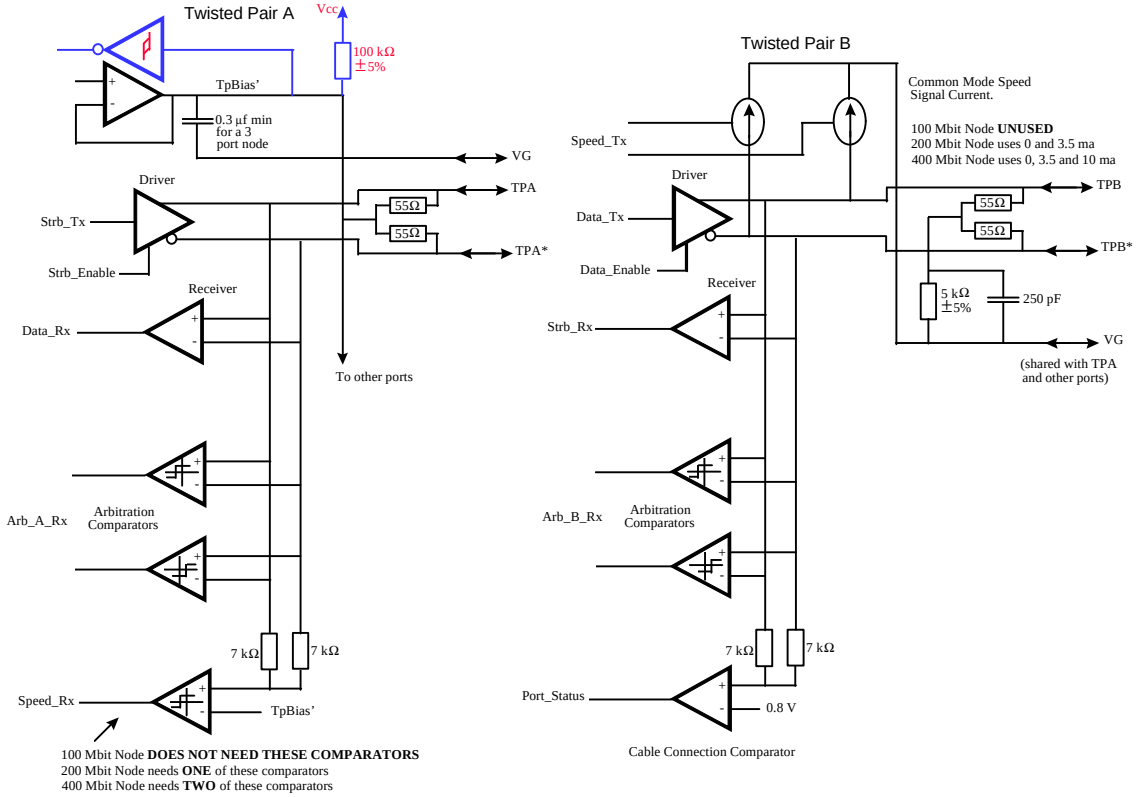

Figure 6-11 – Cable media signal interface configuration with Suspended-Connected Circuitry

TpBias is used by both the suspend initiator and suspend target to establish either a suspend-connected state or generate a resume event. Connect and disconnect state information is obtained via a leakage current source (attached to the TpBias generator output signal line) into a current sink at the connected port in a suspended-connected state. When one port or the other is disconnected from the cable, a rise in voltage on the TpBias generator's output signal line occurs. The rise in voltage is detected by a voltage sensing circuit connected to the TpBias generator output. The voltage sensing circuit generates connect and disconnect notification. Connect or disconnect notification results in a resume event.

6.7.4 PHY Register Addressing

PHY registers are accessed using an extended PHY packet with the following structure:

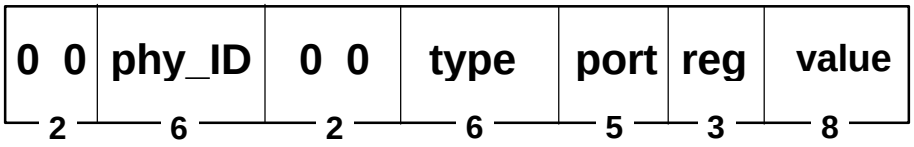

Figure 6-12 – PHY Control Packet

The fields in this packet are defined as follows:

phy_ID: 6-bit node address of the ports PHY;

type: 6-bit field identifying the type of PHY-register operation (000000b=read, 000001b=write);

port: 5-bit port-select value. Values 0-26_{10} inclusive select a specific port in the node; values 27_{10} through 30_{10} are reserved; value 31_{10} selects the 8 node registers.

reg: 3-bit field to select a particular port or node control or status register. For port values of 0-26_{10} inclusive: register 000b (control) and 001b (alternate control) are read-write for link access but are read-only for an extended PHY packet command; reads from 010b or 011b return the current value of the port's status register; writes to 010b set corresponding bits (as specified in the 'value' field) in the port's status register; writes to 011b clear the corresponding bits (as specified in the 'value' field) in the port's status register; a write to 100b selects the port as a suspend initiator. Extended PHY packet accesses to any node registers (i.e. port=31_{10}) will return the content of the node control register accessed (e.g. a read only operation of node registers 0 through 8_{10}).

value: 8-bit field which will either return the contents of the selected register or contain a pattern of bit(s) to be set or cleared at the selected register.

When a valid extended PHY packet of either type read or write is sent to a node, the node PHY returns a response packet formatted as shown in figure 6.13

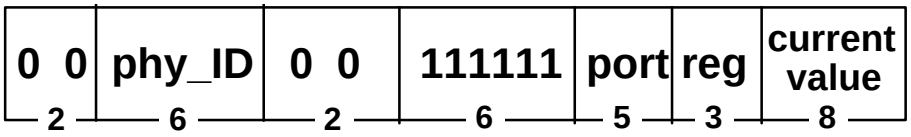

Figure 6-13 – PHY-Access Response Packet

The fields in this packet are defined as follows:

phy_id: node ID of node PHY generating the response packet

type: 111111b

port: port register value equal to the port register value provided in the extended PHY packet for which the response is being generated

reg: register value equal to the register value provided in the extended PHY packet for which the response is being generated

value: the 8 bit value currently contained in the register addressed by the port and reg fields. If an invalid (e.g. reserved) register is addressed via the port and reg fields, a value of 0 is returned.

6.7.5 Entry into Suspend

A connected and active port will initiate the suspended-connected process protocol when its port control register 4 is written to (data not defined) from either: 1) an extended PHY packet of type 'PHY register write' (000001b), or 2) the link associated with the node in which the port resides. Each method is described in the following sections.

Once a suspended-connected state has been established between a suspend initiator port and a suspend target port, both initiator node and the target node (respectively) remove power from: 1) the initiator's and target's TpBias generator; 2) the initiator's and target's differential transmitter drivers and receivers; 3) the initiator's and target's speed and arbitration comparators. A port's **Port_Status** will indicate **CONNECTED** for any port in a suspended-connected state.

6.7.5.1 Suspend Initiator selected via an Extended PHY Packet

An extended PHY packet of type 'PHY Register Write" (000001b) to a node's port control register four (100b) will select that port as a suspend initiator (the port must be connected and active). The port connected to the suspend initiator becomes the suspend target.

If the port selected to become the suspend initiator has its ***Suspend_Disable*** bit set (logic 1) in its ***Port Control*** register (000b), the port will not begin the suspended-connected process. It will, however, generate a PHY packet response with the register value field initialized with the current contents of the ***Port Status*** register.

A port's ***Suspend_Disable*** bit may only be set (logic 1) or cleared (logic 0) by the link through the PHY-Link interface. A port's ***Suspend_Disable*** bit is located in the port's ***Port Control*** register (000b), bit position 6.

Assuming the initiator's ***Suspend_Disable*** bit is cleared (logic 0), upon receiving notification to begin the suspended-connected process, the initiator sends **TX_DATA_END** to all ports in the node in which it resides (except the target port) followed by an **IDLE**. The initiator then transmits **TX_SUSPEND_NOTIFY** (Arb_A_Tx = 'Z,' Arb_B_Tx = '1') to the target. The target interprets the **TX_SUSPEND_NOTIFY** as an *out-of-context* **RX_IDENT_DONE** and, therefore, understands the *out-of-context* **RX_IDENT_DONE** to be an initiator suspend request. The initiator holds **TX_SUSPEND_NOTIFY** for **TBD** µs, waiting for the target to respond with **TX_SUSPEND_NOTIFY** and subsequent drop in the target TpBias. Upon receiving the **TX_SUSPEND_NOTIFY** from the target (received by the initiator as an *out-of-context* **RX_IDENT_DONE**) the initiator completes the suspended-connected protocol process by dropping its TpBias to the target to below 0.4 volts within 200 µs of receiving **RX_IDENT_DONE** and will then set its own **Suspend** bit in its **PHY Port Status** register (010b).

A port's **Port_Status** will provide **CONNECTED** status for any port in a suspended-connected state.

The target, upon seeing the *out of context* **RX_IDENT_DONE** (and if the port's ***Suspend_Disable*** bit is reset - i.e. logic 0) transmits **TX_SUSPEND_NOTIFY** back to the initiator followed by dropping its TpBias to the initiator to below 0.4 volts within 200 µs of transmitting **TX_SUSPEND_NOTIFY** to the initiator and will then set its own **Suspend** bit in its **PHY Port Status** register (010b). If the target port's ***Suspend_Block*** bit is reset (i.e. logic 0) the target node will transmit a **TX_SUSPEND_NOTIFY** to all of its remaining connected and active ports (selecting them as suspend initiators).

If the initiator's **RX_IDENT_DONE** timer timeouts out after **TBD** (i.e. the target does not respond by transmitting **TX_SUSPEND_NOTIFY** and does not drop its TpBias to the initiator) the initiator will set it's own port ***Disable*** bit immediately dropping initiator TpBias. The target will

see the loss of the initiator's TpBias and interpret the loss as a disconnect resulting in the target node asserting bus reset to all of its remaining connected ports. A target disconnected from an initiator in this fashion will report a **Port_Status** state of **DISCONNECTED**.

Upon completing the suspended-connected process (whether or not the initiator became suspended or disabled), the initiator responds to the PHY register write packet by returning a response packet containing the updated value of the initiator's ***Port Status*** register (010b) after which the initiator node asserts a bus reset on all of its other active and connected ports.

6.7.5.2 Suspend Initiation from the Link

A link selects a port to become a suspend initiator by issuing a PHY register write to the PHY's port control register 100b (data in 8-bit data field for register 100b is not defined).

The initiator proceeds with the execution of the suspended-connected process as previously described with the exception the initiator node does not transmit a PHY packet response to the bus upon completion of the suspended-connected state transition.

6.7.5.3 Target Response to Suspend Request

A target can be configured to not respond to an initiator request for suspend when the target's **Port Control** register (000b) has its ***Suspend_Disable*** bit set (logic 1). The port's ***Suspend_Disable*** bit can only be set (logic 1) or cleared (logic 0) by the link through the PHY-Link interface. A port's ***Suspend_Disable*** bit is located in the port's ***Port Control*** register (000b), bit position 6.

A target node can be configured to not propagate an initiator request for suspend to its other active and connected ports when the target **Port Status** register (010b) has its ***Suspend_Block*** bit set (logic 1). The port's ***Suspend_Block*** bit can be set (logic 1) or cleared (logic 0) by the link through the PHY-Link interface or via an extended PHY packet of type 'PHY Register Write' (000001b) to the target's **Port Status** register. NOTE: A PHY write to target's **Port Status** register 010b with bit 5 set (logic 1) in the value field will set (logic 1) the target's **Suspend_Block** bit, while a PHY write to the target's **Alternate Port Status** register 011b with bit 5 set (logic 1) in the value field will clear (logic 0) the target port's **Suspend_Block** bit.

When an initiator transmits a **TX_SUSPEND_NOTIFY** (Arb_A_Tx = 'Z,' Arb_B_Tx = '1') to the target, it will be received as an **RX_IDENT_DONE**. Under normal operating circumstances, the target would receive an **RX_IDENT_DONE** as the result of a child PHY completing transmission of its self-ID. In this circumstance, an event has **not** occurred in which the target would be expecting to receive an **RX_IDENT_DONE** from a child PHY, therefore, the target interprets the *out-of-context* **RX_IDENT_DONE** as a suspend notification event from an initiator.

When a target's ***Suspend_Disable*** bit is reset (logic 0) the target transmits **TX_SUSPEND_NOTIFY** back to the initiator. The target removes its TpBias to the initiator to below 0.4 volts within 200 μs of transmitting **TX_SUSPEND_NOTIFY** to the initiator. Upon dropping its TpBias to the initiator, the target sets (logic 1) its own **Suspend** bit in its **Port Status** register (010b).

When target's ***Suspend_Disable*** bit is set (logic 1) the target will **NOT** transmit a **TX_SUSPEND_NOTIFY** to the initiator and will **NOT** removes its TpBias to the initiator. The initiator will timeout due to a lack of response from the target. When the initiator times out, the initiator will remove its TpBias to the target. The target node, upon seeing a loss of TpBias from the initiator, will assert bus reset to all of its other connected ports.

When the target's ***Suspend_Block*** bit is clear (logic 0), and as soon as the target's **Suspend** bit is set (logic 1), the target node will configure all of its remaining connected and active ports as suspend initiators (within the constraints provided herein).

When the target's ***Suspend_Block*** bit is set (i.e. logic 1) the target node will **NOT** propagate suspend notification to any of its other ports (e.g. it will not configure any of its remaining connected and active ports as suspend initiators).

A port with its ***Suspend_Disable*** bit set (logic 1) must also have its ***Suspend_Block*** bit set (logic 1). A target's ***Suspend_Block*** bit cannot be cleared (logic 0) if its ***Suspend_Disable*** bit is set.

6.7.6 Resume (Exit from a Suspended-Connected State)

A port in a suspended-connected state will generate a resume event when the ***Suspend*** bit in its ***Port Status*** register is cleared (logic 0). A port which generates a resume event becomes a resume initiator. A resume event is generated when the resume initiator powers on and enables its TpBias generator - asserting TpBias to the suspended-connected port the resume initiator is connected to.

The ***Suspend*** bit in the ***Port Status*** register of a port in a suspended-connected state may be cleared (logic 0) by the link via the PHY-Link interface or from an extended PHY packet of type 'PHY Register Write' (000001b) to any active port status register in the node where the suspended-connected port resides. NOTE: A PHY write to port status register 010b with bit 6 set (logic 1) in the value field will set (logic 1) the port's **Suspend** bit, while a PHY write to port status register 011b with bit 6 set (logic 1) in the value field will clear (logic 0) the port's **Suspend** bit.

The ***Suspend*** bit may not be cleared (logic 0) if the port's ***Resume_Disable*** bit in its ***Port Control*** register (000b) is set (logic 1).

A port's ***Resume_Disable*** bit may only be cleared (logic 0) or set (logic 1) by the link through the PHY-Link interface. A port's ***Resume_Disable*** bit is located in the port's ***Port Control*** register (000b), bit position 7.

A port in a suspended-connected state which has TpBias asserted on its TPB/TPB* pair, becomes a resume target.

If the resume target's ***Resume_Disable*** bit is clear (logic 0), the resume target will respond to the resume event by powering on and enabling its own TpBias generator - restoring TpBias to the resume initiator port.

Upon restoring its TpBias to the resume initiator, the resume target clears (logic 0) its ***Suspend*** bit in its own ***Port Status*** register.

If the resume target port's ***Resume_Disable*** bit is set (logic 1), the resume target will **not** respond to the resume event and will **not** power on or enable its own TpBias generator - thus leaving TpBias off to the resume initiator port. The resume target's ***Suspend*** bit will remain set (logic 1).

A resume initiator will return to the suspended-connected state if it does not see TpBias restored by the resume target within one debounce interval.

If the ***Resume_Block*** bit of the resume target is clear (logic 0) the resume event is propagated by the resume target node to all of its ports in a suspended-connected state - each of them becoming a resume initiator. A node with ports participating in the resume process must have

restored TpBias to all of its participating port connections within 10 milliseconds of receiving resume event notification.

If the ***Resume_Block*** bit of the resume target is set (logic 1) the resume target will respond (assuming its ***Resume_Disable*** bit is clear - logic 0) to the resume event by powering on and enabling its own TpBias generator - restoring TpBias to the resume initiator port and so forth (as previously outlined). However, the resume target node will not propagate the resume event to any other port in the node.

A port's ***Resume_Block*** bit can be set (logic 1) or cleared (logic 0) by the link through the PHY-Link interface or via an extended PHY packet of type 'PHY Register Write' (000001b) to the node's port's status register. NOTE: A PHY write to port status register 010b with bit 4 set (logic 1) in the value field will set (logic 1) the port's **Resume_Block** bit, while a PHY write to port status register 011b with bit 4 set (logic 1) in the value field will clear (logic 0) the port's **Resume_Block** bit.

A resume event will not propagate through an active port (e.g. a resume event in one suspend domain (or suspend sub-net) will not propagate into another suspend domain (or suspend sub-net) through an active boundary node.

A resume initiator will wait five debounce intervals (measured from the time the resume target cleared its ***Suspend*** bit in its own ***Port Status*** register) after which it will generate a long reset - providing no other bus reset has occurred during the 5 debounce time delay. This delay provides opportunity for a resume target (which may be connected to a boundary node as far away as 16 hops from the originating resume initiator) to win arbitration of the bus in order to generate an arbitrated short reset. Resume propagation can take as long as 10 milliseconds per node. A single debounce interval is approximately 85 milliseconds. At the end of two debounce intervals, the farthest node from the resume initiator will have TpBias restored to all of its resuming ports. If that farthest node had a connection to an active bus segment through a port that did not have to resume, that node code begin arbitration for the active bus segment in order to generate an arbitrated reset. Arbitration may take as long as two debounce intervals, therefore, the far node may not be able to generate a short reset until the resume initiator has entered into its fifth debounce interval. If a bus reset does not occur after five debounce intervals, the resume initiator may assume there is no connection to an active bus segment, therefore it asserts a long reset.

A resume target in a node containing a connected and active port, will signal to the recently resumed suspend domain or suspend sub-net) that it desires to take control of the resume completion process. The node desiring to complete the resume process asserts **TX_REQUEST** to the resuming suspend domain (or suspend sub-net) **TX_REQUEST** will propagate to all resuming ports and will prevent the originating resume initiator from generating a bus reset. The node desiring to complete the resume process will assert **TX_REQUEST** within one debounce interval of responding the its resume event, however, it will not begin arbitration for the active bus segment until two debounce intervals measured from the time the resume event was first detected. If the node desiring to complete the resume process does not see a bus reset from the resumed segment for two debounce delay intervals, it will begin bus arbitration on the active bus segment for the purpose of generating an arbitrated short reset.

When multiple nodes are asserting **TX_REQUEST**, the node that first winds bus arbitration with its active bus segment will generate a reset on the resumed segment. When this reset reaches another node that had also asserted **TX_REQUEST**, that node will generate a long reset on bot the active segment to which it is attached and the resume segment.

6.7.6.1 Resume Propagation

6.7.6.2 Reset Concluding Resume

Once all of the suspended ports in a resuming suspend domain have restarted their TpBias generators, these ports can again participate in normal bus traffic. To merge the suspend domain with adjoining active bus segments, and thus make the resumed ports “visible” for usage, the resume activity must conclude with a reset of the resumed domain.

6.7.7 Detach Detection During Suspend

When a port is suspended, the initiator enables a single-ended low-voltage source to drive TPA. The initiator also enables a comparator to monitor the voltage that the initiator port receives on its TPB. If TPB drops below ***XXX***v while the port is suspended, this indicates that the attached node has been disconnected. Similarly, if the attached target port detects that its TPB drops below ***XXX***v, this indicates that the initiator has been detached.

Port hardware records the occurrence of a detach event on a suspended port by clearing the ***Con*** (Connected) bit of that port’s ***Port Status PHY*** to 0.

6.7.8 Connect Detection During Suspend

If a node is attached to a suspended port, the attachment event generates a reset in the suspended port. Reset causes the suspended port to resume, and resume processing then occurs as described above.

6.7.9 Target Power Loss During Suspend

The initiator port is unable to differentiate between loss of power on the target port, and detachment of the target port. In either case, the initiator sees the voltage on TPB go to zero. No hardware indication of power loss is available.

6.7.10 Reset Propagation

6.7.11 PHY-Core Suspend/Resume Control

A PHY contains both circuitry which is dedicated to each port, and shared functions such as clock generation. Port-to-port suspend affects the power-state of PHY ports, but does not affect the power state of the PHY core. The latter can be controlled from either the link, or from the bus. In both cases, the state of the entire PHY is affected.

When the last port in a node is suspended, the SLEEP bit in node control register 5 is set. The core functions of a node enter a very low power state when the SLEEP bit becomes set (e.g. clocks and all high current consuming logic is powered off). The node only maintains sufficient active logic to detect a resume event. A resume event on any port in a sleeping node will reset the nodes SLEEP bit - restoring the nodes core to normal, full-power operation. A link may reset the nodes SLEEP bit (resuming the node’s core) without causing a port in the node to generate a resume event.

6.7.11.1 PHY-Core Control from the Link

A node-resident device can use the node’s link to control the associated PHY power state. Typically, this would be done after each of the PHY’s

6.7.12 Resume/Resume Collisions

6.7.13 Suspend/Resume Collisions

6.7.14 Boundary Nodes

6.7.15 Extended PHY Packet of type 'Resume'

6.7.16 Disabled Nodes

Other notes on the way to make certain all of the bases are covered:

1) **Address any discrepancies between port control and alternate port control in this proposal against the PHY port registers in the 0.09 draft covered in section 5.2.1 - specifically on page 50. Pay particular attention to the port "DIS" bit in register 100.**
2) **Address conflicts with this proposals port addressing mechanism, port assignments and bit assignments against any that may exist in the P1394a 0.09 draft in section 5.2.1**
3) **Address the issue of core suspending. Suspending is "easy," its the resuming that seems a little tricky.**
4) **Address the need to notify the link or PC system software stack that a port resume event has occurred. Link On cannot be used for this because the Link may not be off - unless Link On uses an edge to bring the Link On as opposed to a level (meaning, once a link on has been sent, take it back to an active state and then, if the link is already on, subsequent transitions of link on are indications of resume events in the node…**