Table of Contents

Table of Figures

Date	Revision	Comment
08/14/97	.08	• Corrected Arbitration state for TX_DATA_PREFIX • Added Revision Table • Added Table of Contents and List of Tables/Figures • Removed “Connected and active” from glossary • removed “and to any other ports in it’s node which are in a suspend state;” from the glossary definition for “Resume Initiator” • Added clarification for Suspend Bit status when a Control Set or Control Clear register write with a value of Suspend is performed • Added power up state for the Fault bit in the Port Control register • Corrected state of Resume when the PHY does NOT generate a LinkOn
09/07/97	.09	• Clarified intent of resume target in a multiple port node • Corrected editorial errors in figure 0-4 and added enable/disable “bubble” for the TpBias generator • Altered and added extended PHY packet Types • Redefined bits for Extended PHY Packet Type field and Port field (changes PHY register addressing) • Eliminated TX_REQUEST • Swapped defined arbitration states for TX_SUSPEND and TX_DISABLE • Added "signals-on-the-wire" diagram for suspend and disable operations • Restored power-on state of the Disable bit to be that of a strap option • Changed nomenclature of the suspend process to show the ***Bias*** bit as the suspend state indicator while the ***Control*** register ***Suspend*** and ***Disable*** bits indicate their respective “*pending*” operation • Changed assertion time for **TX_SUSPEND**
		•

Figure 0-1 –Document Revision Table

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 below provides definitions, unique in some instances, for terms used in the discussion of suspend and resume mechanisms, protocol, and process:

suspend state: a power state in which power consumption is as low as possible without turning power off - normal data processing is not possible;
Active: a port state in which the port is connected but not suspended or disabled;
Resume: a transition process from suspend to full power in which normal data processing is possible – normal transaction packet processing does not occur in a port/node involved in the process of resuming from a suspend state;
Suspended Fault state: the state of a port which has it's ***Fault*** bit set and it's ***Bias*** bit set. A port in this state has the same power consumption restrictions as a port in a suspend state;
Resume Fault state: the state of a port which it's ***Fault*** bit set and it's ***Bias*** bit clear. A port in this state has the same power consumption restrictions as a port in a suspend state;
Suspend initiator: an active port in a node which has been selected to initiate suspend notification to an active port in another node to which it is connected;
Suspend target: an active port in a node which receives suspend notification from a port in another node to which it is connected;
Resume initiator: a suspended port in a node which has been selected to initiate a resume event to a suspended port in another node to which it is connected;
Resume target: a suspended port in a node which detects a resume event from a suspended port in another node to which it is connected;
Suspend Connection: a port-to-port connection between two nodes in which both ports are in a suspend state;
Suspend Domain: three or more nodes joined via port-to-port connections in which all ports may be in a suspended state;
Bus Manager: provides (among other functions) suspend mechanism management;
Boundary Node: a node with one or more active ports and one or more suspend connections;
Set: logic 1, not clear, a logic high state;
Clear: logic 0, not set, reset, a logic low state;
$\textit{\textbf{Notify}}_{\textit{\textbf{hold}}}$: A period of time defined as 8,192 SCLKs - approximately 166.67 µs;
$\textit{\textbf{Bias}}_{\textit{\textbf{hold}}}$; A period of time defined as 8,192 SCLKs - approximately 166.67 µs;
$\textit{\textbf{Detect}}_{\textit{\textbf{min}}}$: A period of time defined as 8,192 SCLKs - approximately 166.67 µs;
$\textit{\textbf{time}}_{\textit{\textbf{res\_notify}}}$: A period of time defined as 16,384 SCLKs - approximately 333.33 µs;
RESET_DETECT interval: 80.0 milliseconds, maximum 85.3 milliseconds
Short_Reset Interval: 1.6 microseconds
***TX_DISABLE*:** Arb_A_Tx = Z, Arb_B_Tx = 1
TX_SUSPEND: Arb_A_Tx = '0,' Arb_B_Tx = '0'

> ***NOTE:*** All timing values are to be interpreted as the *maximum* time allowed for an operation to complete, a response to occur, or a stimulus to be applied. In actual application, all operations are expected to complete in the shortest time possible, responses are to be provided in the earliest time period possible, and stimulus must be applied as soon as feasible.

6.7.1 Port Register Map

The register map in Figure 0-2 identifies a port's register bits. The suspend/resume process and mechanism specifically make use of the bits: ***Con***, ***Bias***, ***Chg_int_en***, ***Initiate Suspend***, ***Initiate Resume***, ***Initiate Disable***, ***Initiate Enable***, ***Disabled***, ***Fault Set***, and ***Fault Clear***.

Figure 0-2 -- **Port *Control Set*, *Control Clear*, *Status*, and *Alternate Status* Register map**

	Contents							
Address*	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
(1)000$_b$	Astat		Bstat		Ch	Con	Bias	Disabled
(1)001$_b$	Negotiated_speed			Rsrvd				
(1)010$_b$	Chg_int_en	Initiate Suspend	Initiate Disable	Enab_token (optional)	Fault Set	rsrvd		
(1)011$_b$	Chg_int_en	Initiate Resume	Initiate Enable	Enab_token (optional)	Fault Clear	rsrvd		
(1)100$_b$	rsrvd							
(1)101$_b$	rsrvd							
(1)110$_b$	rsrvd							
(1)111$_b$	rsrvd							

Port register addresses are referred to using the following naming conventions:

Port Register Address*	**Port Reference**
(1)000$_b$	***Status A***
(1)001$_b$	***Status B***
(1)010$_b$	***Control Set***
(1)011$_b$	***Control Clear***

*Note: The most significant bit of the address (parenthesized) is used only by the link during a PHY register read/write (Lreq). The most significant bit is not used when addressing a port via an extended PHY packet.

Port registers ***Status A*** and ***Status B*** are read only. A read of a port's ***Control Set*** or ***Control Clear*** registers will return the current state of each bit in the ***Control*** register.

The readable register contents of the ***Control Set*** and ***Control Clear*** are identical.

A bit in the ***Control*** register is set (logic 1) when the ***Control Set*** register (address 010$_b$) is written with a bit set in the data field (whose bit position corresponds to the bit position of the bit to be set). Multiple bits may be set with a single register write.

A bit in the ***Control*** register is cleared (logic 0) when the ***Control Clear*** register (address 011$_b$) is written with a bit set (logic 1) in the data field (whose bit position corresponds to the bit position of the bit to be cleared). Multiple bits may be cleared with a single register write.

1 Figure 0-3 – Port Register Map Bit Field Definitions

Register (1)000$_b$ - ***Status A***

Field	Size	Type*	Description
Astat	2	Ro	TPA line state for the port: 00_2 = invalid 01_2 = 1 10_2 = 0 11_2 = Z
Bstat	2	Ro	TPB line state for the port (same encoding as Astat)
Ch	1	Ro	If set, the port is a child, else a parent. The meaning of this bit is undefined from the time a bus reset is detected until the PHY transitions to state T1: Child Handshake during the tree identify process (see 4.4.2.2 in IEEE Std 1394-1995).
Con	1	Ro	If set, the port is connected, else disconnected. The value reported by this bit is filtered by hysteresis logic to reduce multiple status changes caused by contact scrape when a connector is inserted or removed.
Bias	1	Ro	If set, bias voltage is detected (possible connection). The value reported by this bit is filtered by hysteresis logic to reduce multiple status changes caused by contact scrape when a connector is inserted or removed.
Disabled	1	Ro	When set, a port is disabled (the port is in a low power state); a disabled port will not respond to or generate bus resets, suspend notification, or resume events; this bit assumes the state of its power-on strap option upon a power-up reset.

Register (1) 001$_b$ - ***Status B***

Field	Size	Type*	Description
Negotiated_speed	3	Ro	Indicates the maximum speed negotiated between this PHY port and it's immediately connected port; the encoding is the same as for the *xspd* bit in self-ID packet 7 (see clause 6.2.1 of the IEEE 1394-1995.A draft specification).
Rsrvd	5	Ro	reserved - must not be used or interpreted as having a meaningful value

Register: (1) 010$_b$ and (1) 011$_b$ – ***Control Set*** and ***Control Clear*** (respectively)

Field	Size	Type*	Description
Chg_int_en	1	Rw	***ch***_ange_ ***int***_errupt_ ***en***_able_ - if set, the PHY will interrupt the link with a 4-bit PHY Status Packet when one or more of the following bits change state: ***Con***, ***Bias***, ***Suspend***, ***Fault***, or ***Disable***; this bit is clear subsequent to a power reset.
Initiate Suspend (Control Set); Initiate Resume (Control Clear)	1	Wo	When setting this bit via a write to register ***Control Set***, the port is selected as a suspend initiator; setting this bit via a write to register ***Control Clear*** selects the port as a resume initiator; this bit is clear (logic 0) subsequent to a power reset.
Initiate Disable (Control Set); Initiate Enable (Control Clear	1	Wo	When setting this bit via a write to register ***Control Set***, the port is selected to be disabled; setting this bit via a write to register ***Control Clear*** selects the port to be enabled; this bit is clear (logic 0) subsequent to a power reset.
Enab_token (optional)	1	Rw	Enable token-style arbitration. When set, the enhancements specified in clause 6.5 of the IEEE 1394-1995.A draft specification shall be enabled for this port.
Fault	1	Rw	Set when a port is not able to successfully interact with it's connected port during either a suspend or resume process; When set, writing to the ***Control Clear*** register with the value ***Fault*** in the data field select the port as a resume initiator; writing to the ***Control Set*** register with the value ***Fault*** in the data field is an undefined operation; this bit is clear subsequent to a power reset.
Rsrvd	3	Ro	Reserved - must not be used or interpreted as having a meaningful value

Registers: $(1)100_b$, $(1)101_b$, $(1)110_b$, $(1)111_b$

Field	Size	Type*	Description
Rsrvd	8	Ro	Reserved - must not be used or interpreted as having a meaningful value

***Type** definition: Ro=Read Only, Rw=Read/Write, Wo=Write Only – from the perspective of extended PHY packet access.

6.7.2 Suspend Model and Process Overview

The suspend mechanism allows pairs of directly-connected ports to be placed into a low-power, suspended state. Any active port may be selected to initiate suspend state notification. A port is selected to become a suspend notification initiator when a PHY register write packet is sent to the port's ***Control Set*** register with the value ***Initiate Suspend*** in the data field. A suspend initiator notifies the port to which it is connected (the target) to enter into the suspend state.

A suspend domain is created when three (or more) nodes are joined via a port-to-port connection in which each port is in a suspended state. Two nodes with a port-to-port connection (each in a suspend state) constitutes a ***suspend connection*** and can be the attach point between an active (i.e. not suspended) domain and a suspend domain.

A suspend domain is restored to normal operation when a suspended port in the domain detects a "resume" event (within the constraints outlined in proceeding sections). A resume event may be initiated by any connected port in a suspend state whose ***Disabled*** bit is clear.

A connected port in a suspend state which detects a resume event becomes a resume target and will propagate the resume event to all other connected and suspended ports in it's node (within the constraints outlined in proceeding sections).

A port which is in a low power state (e.g. has its' ***Disable*** or ***Fault*** bit set or its' ***Bias*** bit clear, or does not have a port connection) will be reported as no-connection in the nodes' self-ID packet.

A port connected to a 1394-1995 port may enter into a low power state equivalent to a suspend state. This state is indicated by the port's ***Fault*** bit set and its' ***Bias*** bit set. This state may be referred to as the ***Suspended Fault*** state. A port in this state can be selected as a resume initiator by writing the port's ***Control Clear*** register with the value ***Fault Clear***.

6.7.2.1 Port Suspend Circuitry

A common mode current is supplied from the TPA pair on a node's port to the TPB pair on a peer port through a cable connection between the two ports. The level of the current will result in a voltage drop across a 5 kilohm current sink on the TPB pair of the peer port no greater than 0.4 volts. The TPA side of the port-to-port connection has a connection from the output of the TpBias generator to the input of a voltage-sensing circuit. If the cable connection is removed, the voltage input to the sensing circuit will rise, generating a disconnect notification on its' output.

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

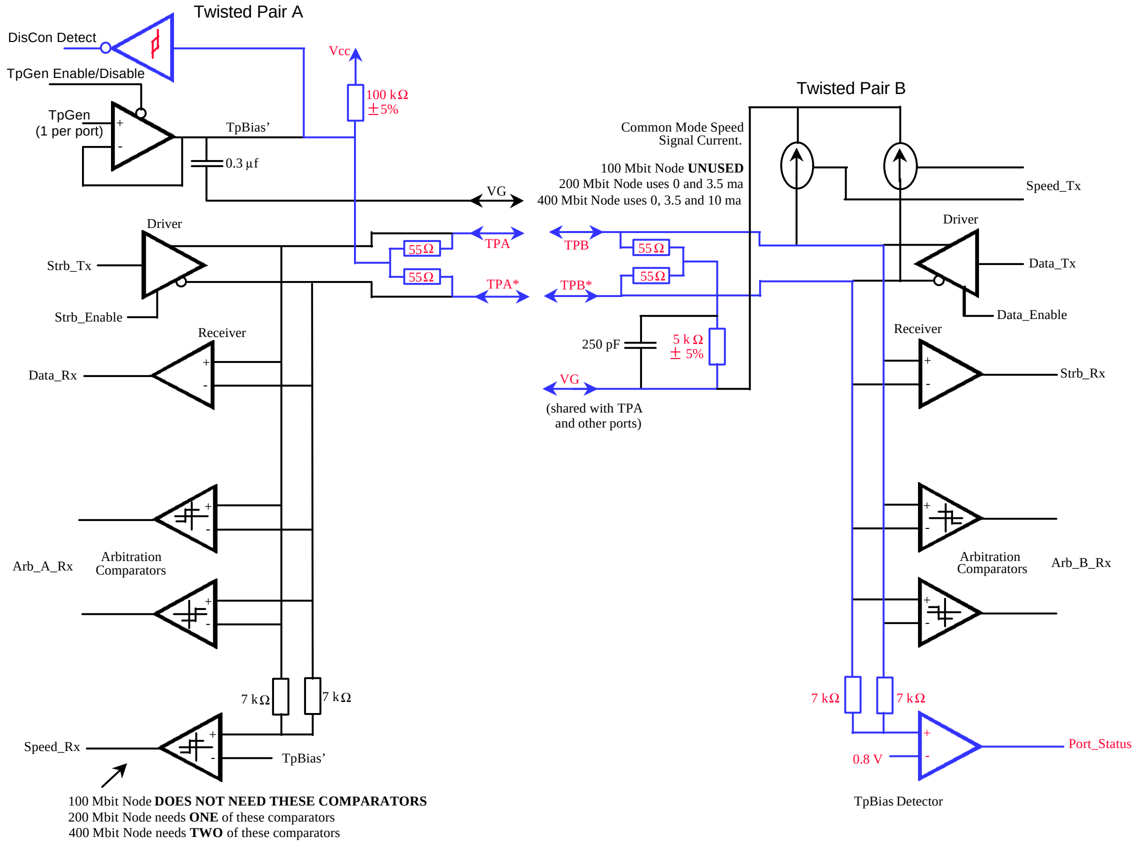
Current is supplied through a cable connection to a peer port's TPB pair through a 100 kilohm resistor connected between an *always on* power rail (3.3 volts) and the output of the disabled TpBias generator. The input to a schmidt trigger buffer is connected to the junction of the 100-kilohm resistor and the TpBias generator output. This mechanism will create a signal (active logic low) on the output of the buffer when the cable connection between the target and initiator is opened (see figure 0-4). Note: The output of the buffer is monitored only when a port is in a suspended state - e.g. TpBias generator (among other circuitry) has been disabled and it's output is in a high impedance state.

While in the suspended state, a port is configured as follows (refer to section 6.7.1 for a detailed description of a port's ***Status*** and ***Control*** registers):

— Only circuitry associated with the port's voltage-sense and **Port_Status** TpBias detector is active;
— The port's ***Disable*** bit is clear (i.e. port enabled);
— The port's ***Bias*** bit is clear (i.e. port is suspended);

Figure 0-4 shows the cable media signals interface configuration – shaded circuitry is required to be active in a port in a suspend state.

Figure 0-4 – Cable media signal interface configuration with Suspend Connection Circuitry



Common mode signaling on the TPA port is used by both the suspend initiator and suspend target to establish either a suspend connection or generate a resume event.

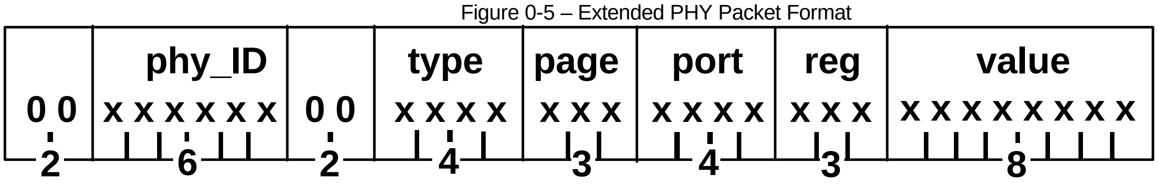
Connect and disconnect detection while a port is in a suspended state is obtained via the output of voltage sense circuitry whose input is connected to the TpBias generator output. A leakage current source (attached to the TpBias generator output of the TPA port) is delivered into a current sink on the TPB peer port through a cable connection. When one suspended port or the other is disconnected from the cable, a rise in voltage occurs on the voltage sensing circuit input - generating a disconnect notification signal on the output on the voltage sense circuitry.

The input to the voltage sense circuitry will remain in a low state while the current sink is attached through the cable connection. Connect notification results in a resume event (the port's ***Con*** bit becomes set - among other actions as outlined in proceeding sections). Disconnect notification results in the port's ***Con*** bit being cleared - a resume event may be indirectly generated – dependent upon the state of the ports' ***chg_int_en*** bit and the resultant actions of the PHY's attached link.

6.7.3 PHY Register Addressing

PHY registers are accessed using an extended PHY packet consisting of the following fields:

Figure 0-5 – Extended PHY Packet Format



The fields in the extended PHY packet are defined as follows:

phy_ID: 6-bit node address of the node's PHY;

type: 4-bit field identifying the type of PHY register operation

0000$_b$=Ping
0001$_b$ =Node Core Register Read
0010$_b$ =Reserved
0011$_b$ =Node Core Register Read PHY Response Packet
0100$_b$ =Resume
0101$_b$ =Indexed Register Read
0110$_b$ =Indexed Register Write*
0111$_b$ =Indexed Register PHY Response Packet
1000$_b$ through 0111b=Reserved
1111$_b$ =Not Implemented

page: 3-bit page select. Page 000$_b$ selects the page providing access to the PHY's 16 port register banks.

port: 4-bit port value selects one of 16 register banks - each bank contains 8 registers

reg: 3-bit register select field - selects 1 of the 8 registers in a port register bank on a page.

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

The first eight core node registers can only be read when using an extended PHY packet to access them. Only the Link can write to these registers using methods outlined in IEEE Std. 1394-1995 and 1394a.

When reading the nodes first eight core registers, the bits in the extended PHY packet fields *page* and *port* are in a cleared state while the *reg* field bits are configured to address one of the eight node core registers (000$_b$ through 111$_b$).

6.7.3.1 *Ping*

Refer to revision 1.1 of IEEE 1394-1995a draft specification clause 7.3.5 for specific information on the details for extended PHY packet of type 0000$_b$ (***Ping***).

6.7.3.2 Node Core Register Read

This extended PHY packet type facilitates a read only operation of the first eight registers in the PHY's register map (referred to as the node core registers).

Bit field values for ***page*** and ***port*** are equal to zero when accessing the node core registers.

Bit field values for ***reg*** range from 000$_b$ to 111$_b$ (inclusive) – depending upon which node core register is being read.

*All PHY registers may be read with an extended PHY packet, however, only *reg* field values 010$_b$ and 011$_b$ (port ***Control Set*** and ***Control Clear*** registers -respectively) may be written to and then only when accessing *page* 000$_b$ with *port* values 0000$_b$ through 1111$_b$

Refer to revision 1.1 of IEEE 1394-1995a draft specification clause 6.1 for definitive detail of the PHY register map.

6.7.3.3 Node Core Register Read PHY Response Packet

This PHY response packet contains the exact same field values as the extended PHY packet of type ***Node Core Register Read*** (type 0001_b) with the exception that the field ***value*** contains the contents of the node core register being read.

6.7.3.4 Resume

The ***page*** and ***port*** fields for the ***Resume*** PHY packet type are clear (logic 0). The ***reg*** field addresses the port ***Control Clear*** register (011_b). The ***value*** field contains the value for ***Initiate Resume***.

A node which receives a ***Resume*** PHY packet configures all its connected and suspended ports to be resume initiators - as if each port had the value ***Initiate Resume*** written to it's ***Control Clear*** register.

Each resuming port in the node will generate a resume event to it's connected port.

A response packet is not returned for an extended PHY packet of type ***Resume***.

6.7.3.5 Indexed Register Read

All PHY registers (except the node core registers) are read by this extended PHY packet type.

The ***page*** field selects which register page. The ***port*** field selects one of sixteen register ports (each port provides access to eight registers). The ***reg*** field selects one of eight registers in a port on a register page.

The registers pertinent to this proposal are the first four registers in each of the sixteen PHY port registers located in ***page*** zero. These are (specifically): ***Status A***, ***Status B***, ***Control Set***, and ***Control Clear*** (***reg*** values 000_b, 001_b, 010_b, and 011_b – respectively).

A read from 010_b (***Control Set***) or 011_b (***Control Clear***) returns the current value of the port's ***Control*** register - the contents returned for a read of either the ***Control Set*** or the ***Control Clear*** registers will be identical (note: the bits ***Initiate Suspend***, ***Initiate Resume***, ***Initiate Disable***, and ***Initiate Enable***.are write only bits and will always return a value of zero. The ***Fault Set*** and ***Fault Clear*** bit will always return a value of one if the port is in a fault state, otherwise, it will return a value of zero.

A read from 001_b or 010_b (***Status A*** and ***Status B*** respectively) will return current values of the defined bits. Any undefined or reserved bits will always return a value of zero.

6.7.3.6 Indexed Register Write

Only a ports ***Control Set*** or ***Control Clear*** registers (010_b and 011_b - respectively) may be written to via an extended PHY packet. The ***Control Set*** and ***Control Clear*** registers are only available in *page* zero, *port* register banks 0000_b through 1111_b.

Writes to 010_b (***Control Set***) set corresponding bits (as specified in the '*value*' field) in the port's ***Control*** register

Writes to 011_b (***Control Clear***) clears the corresponding bits (as specified in the '*value*' field) in the port's ***Control*** register.

6.7.3.7 Indexed Register PHY Response Packet

The PHY response packet for an indexed register read or write will contain the same data for field ***page***, ***port***, and ***reg*** as the read or write packet. The ***value field*** will contain the contents of the register being read or written.

In the instance of a write packet, the value returned will include a state value for ***Initiate Suspend***, ***Initiate Resume***, ***Initiate Disable***, or ***Initiate Enable*** ONLY when the write packet included a set value for the corresponding bit. This is the only time set values for these bits will be returned.

Returning the value written for ***Initiate Suspend***, ***Initiate Resume***, ***Initiate Disable***, or ***Initiate Enable*** in the response packet provides acknowledgement of the requested action to the originator of the PHY register write packet.

6.7.4 Entry into Suspend

A port will initiate suspend notification when it's port ***Control Set*** register is written with the value ***Initiate Suspend***.

Once a suspend connection state has been established between a suspend initiator and a suspend target, both the initiator and target nodes are required to have only their connect/disconnect voltage sense circuitry and **Port_Status** comparator active.

When a port detects it's incoming TpBias decrease to below 0.4 volts, it will clear it's own ***Bias*** bit and drive it's outgoing TpBias to below 0.4 volts. After driving a low voltage TpBias for a time interval of $\boldsymbol{Bias_{hold}}$, the port will configure it's TpBias generator output to a high impedance state, thereby allowing the voltage-sense circuitry input connected to the output of the TpBias generator to detect a port connection. A connection will exist if the input to the voltage sense circuitry is held low - a result of a cable connection to a TPB port's current sink path on a port of another node.

The node containing a suspend initiator will generate a short reset on all of it's active ports when the suspend target acknowledges receipt of the suspend request (driving TpBias to the suspend initiator to a low state).

A port which has it's ***Bias*** bit clear will not respond to or generate a bus reset or subsequent suspend notifications. A port in a suspend state will not drive it's TpBias.

6.7.4.1 Remote Suspend Initiator Selection

An extended PHY packet of type 'Indexed Register Write" to a node's port ***Control Set*** register with the value ***Initiate Suspend*** will select that port as a suspend initiator. The port connected to the suspend initiator becomes the suspend target.

Immediately upon receiving the extended PHY packet, the suspend initiator responds with an indexed register PHY response packet which includes the value contained in the initiator's ***Control*** register (with the ***Initiate Suspend*** bit set – acknowledging receipt of the suspend initiator selection).

The figures below illustrates observable signal transactions on the bus during a suspend request sequence as seen by the suspend target and by the active ports in the node containing the suspend initiator.

Figure 0-6 – Bus Signals Received by Suspend Target

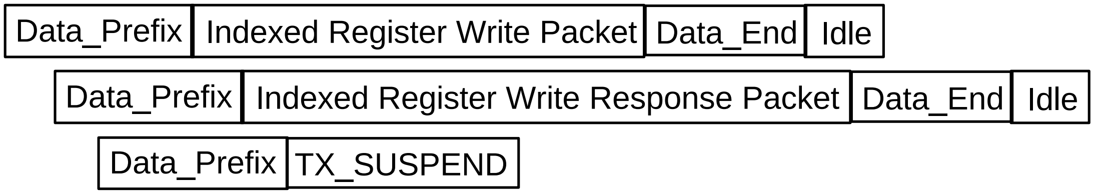
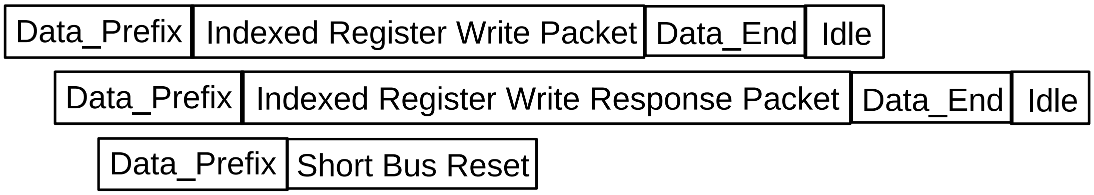

Data_Prefix
Indexed Register Write Packet
Data_End
Idle
Data_Prefix
Indexed Register Write Response Packet
Data_End
Idle
Data_Prefix
TX_SUSPEND

Figure 0-7 – Bus Signals Received by active ports in a Node containing a Suspend Initiator



The suspend initiator asserts **TX_SUSPEND** to the target for a ***Short_Reset*** time interval. Upon terminating **TX_SUSPEND** the suspend initiator monitors incoming TpBias (on its TPB pair) from the suspend target for a time period of ***Notify***$_{hold}$. The suspend initiator monitors for acknowledgement of the suspend request by the target when the target drives it's TpBias to the initiator to below 0.4 volts. Upon detecting a low voltage TpBias from the target the initiator clears it's ***Bias*** bit and drives TpBias to the target to below 0.4 volts for a time interval of ***Bias***$_{hold}$. Upon expiration of ***Bias***$_{hold}$, the initiator places the output of its TpBias generator into a high impedance state (the TpBias generator is disabled) and the initiator transitions into a suspend state. All non-essential circuitry for the initiator is placed into the lowest power state possible (only connect/disconnect voltage sense circuitry and the **Port_Status** bias detector are required to remain active).

If the initiator does not detect an incoming low voltage TpBias during ***Notify***$_{hold}$, the initiator will, after placing the output of it's TpBias generator in a high impedance state, sample the output of it's voltage-sense circuitry. If the voltage-sense circuitry output is in a logic high state (indicating a port connection), the initiator sets it's ***Fault*** bit and transitions into a suspend state (the port's ***Bias*** bit will remain set as long as the port's incoming TpBias is present). If, however, the output of the voltage-sense circuitry is low (indicating that a disconnect occurred during the suspend process) the suspend initiator will transition into a low-power disconnected state.

A set ***Fault*** bit in a port's ***Control*** register indicates a condition in which suspend or resume notification between an initiator and a target did not complete properly.

A port with ***Fault*** set will not respond to or generate bus resets, suspend initiator selection, suspend notification or resume events. The port may, however, be selected as a resume initiator when the ***Fault*** bit is cleared via a register write to the port's ***Control Clear*** register with a value of ***Fault***.

If a connected port with it's ***Fault*** bit set becomes disconnected, the ***Fault*** bit will clear and the port will transition to a disconnected suspend state.

A connected port with it's ***Fault*** bit set and ***Bias*** bit set will clear the ***Fault*** bit when the ***Bias*** bit clears (e.g. when the port's incoming TpBias is removed). A resume event or a bus reset will not occur when the TpBias loss is detected.

6.7.4.2 Suspend Initiator Selection from the Link

A link selects a port to become a suspend initiator by generating an extended PHY packet of type indexed register write to it's own PHY. The PHY packet will contain the node ID of the PHY associated with the link. The PHY packet will write to the port ***Control Set*** register of the port to be selected as a suspend initiator with the data field value of ***Initiate Suspend***.

The suspend initiator node will arbitrate the bus. When the node gains control of the bus, it generates an extended PHY packet of type indexed register write with it's own node ID and page, port, and reg fields set to address the ***Control Set*** register of the port to be selected as the suspend initiator. A value of ***Initiate Suspend*** is written to the ***Control Set*** register. This notifies all other nodes on the bus (at least those that are interested) of the event.

An indexed register PHY response packet will be generated by a node which receives an indexed register write PHY packet from the PHY's own link.

The suspend initiator asserts **DATA_PREFIX** (Arb_A_TX = '0,' Arb_B_TX = '1') to all of it's connected and active ports followed by a short reset to all active ports except the target. The target is sent a **TX_SUSPEND** arbitration state.

The suspend connection state process is the same as that outlined in clause 6.7.4.1.

6.7.4.3 Target Response to Suspend Request

When an initiator generates **TX_SUSPEND** to the target, it will be received as an **RX_SUSPEND**

The target, upon seeing **RX_SUSPEND**, responds by repeating **TX_SUSPEND** out through all of it's other active ports (selecting them as suspend initiators). The target will drive it's TpBias to the initiator to below 0.4 volts for a time of $\boldsymbol{Bias}_{\boldsymbol{hold}}$ (waiting for the initiator to respond by driving it's TpBias into the target to a voltage level below 0.4 volts). After driving a low voltage TpBias to the initiator for the $\boldsymbol{Bias}_{\boldsymbol{hold}}$ time, the target disables the output of it's TpBias generator (generator output goes to a high impedance state). All non-essential circuitry for the target is placed into the lowest power state possible (only connect/disconnect voltage sense circuitry and the **Port_Status** bias detector are required to remain active when a port is in a suspend state).

As soon as the target sees it's incoming TpBias drop to below 0.4 volts, the target completes it's portion of the suspend connection protocol by clearing it's ***Bias*** bit and then transitions to a suspend state.

If the target continues to detect TpBias from the initiator (***Bias*** bit of the suspend target is set) after the $\boldsymbol{Bias}_{\boldsymbol{hold}}$ time of asserting of it's low voltage TpBias to the initiator, the target sets it's ***Fault*** bit and transitions to a suspend state (the port's ***Bias*** bit will remain set as long as the initiator continues to assert TpBias to the target).

A set ***Fault*** bit in a port's ***Control*** register indicates a condition in which suspend or resume notification between an initiator and a target did not complete properly.

A port with ***Fault*** set will not respond to or generate bus resets, suspend initiator selection, suspend notification or resume events. The port may, however, be selected as a resume initiator when the ***Fault*** bit is cleared via a register write to the port's ***Control Clear*** register with a value of ***Fault***.

If a connected port with it's ***Fault*** bit set becomes disconnected, the ***Fault*** bit will clear and the port will transition to a disconnected suspend state.

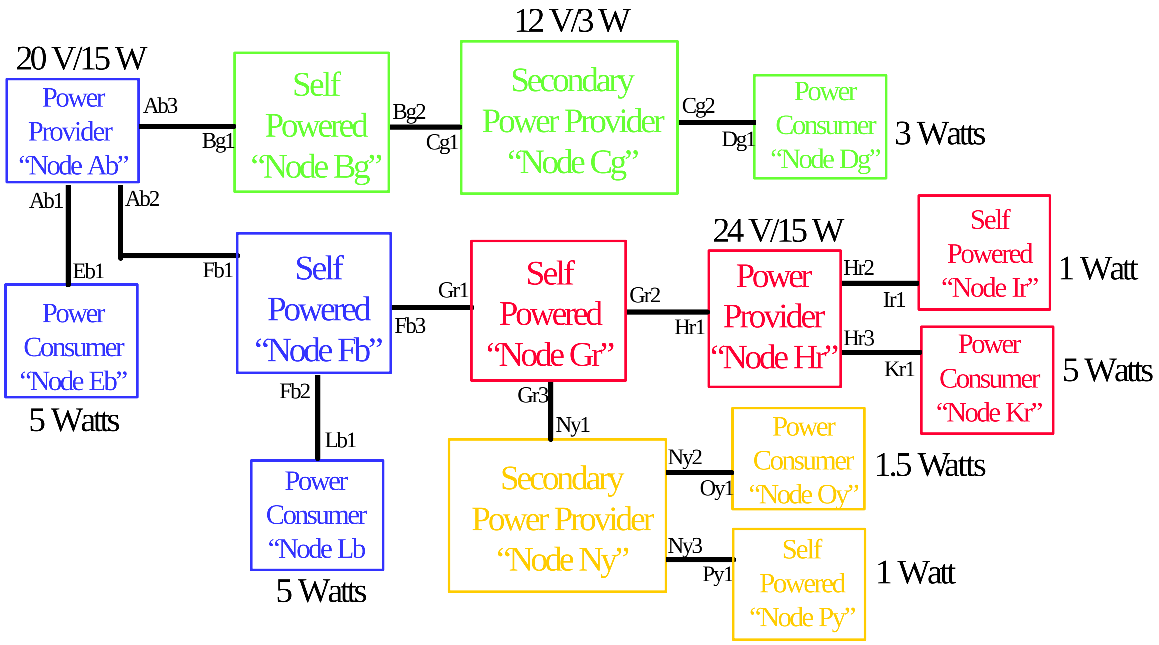
A connected port with it's ***Fault*** bit set and ***Bias*** bit set will clear the ***Fault*** bit when the ***Bias*** bit clears (e.g. when the port's incoming TpBias is removed). A resume event or a bus reset will not occur when the TpBias loss is detected.

6.7.4.3.1 Suspend Blocking

A port can be pre-configured to not respond to suspend notification (thereby preventing it from propagating suspend notification into other active ports in it's node). The method described is the recommended suspend propagation block mechanism:

Assume a bus topology as in Figure 0-8:

Figure 0-8 – Example Suspend/Resume Domain Topology



Further, assume node Bg is the Bus/Power Manager and desires to create a suspend domain consisting of nodes Gr, Hr, Ir, and Kr. Nodes Ny, Oy and Py are not to be suspended as are nodes Lb, Fb, Eb, Ab, Bg, Cg, and Dg.

The Bus Manager begins by sending an index register write PHY packet to node Gr, port Gr3, setting port Gr3's ***Initiate Disable*** bit. Port Gr3 drops it's TpBias* to port Ny1, clearing the ***Bias*** bit in Ny1. Port Ny1 responds by dropping it's TpBias to port Gr3, clearing the ***Bias*** bit in Gr3. Both nodes Gr and Ny assert a short reset to the other connected and active ports in their domains (Gr2, Gr1, and Ny3, Ny2 respectively). Ports Ny1 and Gr3 (of nodes Ny and Gr, respectively) are in a suspended and maintain a suspend connection between an active domain consisting of nodes NY, Oy, and Py and a second active domain consisting of all other nodes.

The Bus Manager node (Bg) sends another PHY register write packet to Node Gr, port Gr3, clearing Gr3's ***Disable*** bit (port Gr3's ***Suspend*** bit remains set). This enables port Gr3 to recognize a resume event from port Ny1 if port Ny1 ever generates a resume event. If the Bus Manager never wants Gr3 to respond to a resume event from port Ny1, the Bus Manager never clears port Gr3's ***Disable*** bit.

The Bus Manager node (Bg) sends a PHY register write packet to node Fb, port Fb3, selecting it as a suspend initiator. The suspend propagates into and through node Gr into nodes Hr, Ir, and Kr.

If the Bus Manager wanted to configure ALL remaining nodes to a suspend state (since domain 'y' is active and port Gr3 will now respond to a resume event from port Ny1) the Bus Manager could generate a PHY register write packet to node Kr, port Kr1, selecting it as the suspend initiator. The suspend event would propagate through all of the 'r' domain and on into the 'b' and 'g' domains.

*There additional events and signals on the bus generated as a result of setting a port's ***Initiate Disable*** bit. Refer to clause 6.7.13 for specific details. The process of port disable has been paraphrased in this clause for the purpose of being able to focus on the use of the disable process to create a single port to port suspend connection without creating an entire suspend domain.

6.7.5 Resume (Exit from a Suspended-Connected State)

A connected port in a suspend state will generate a resume event when one of the following events occur:

a) The port's ***Control Clear*** register is written with the value ***Initiate Resume***;
b) A node receives an extended PHY packet of type ***RESUME***;
c) The port detects a state change on its TPB pair from no TpBias to TpBias present (e.g. a port's ***Bias*** bit is set).

A port may participate in the resume process as both a resume initiator and a resume target. A port assumes the role of a resume initiator when it asserts TpBias to a connected and suspended port. A port assumes the role of a resume target when it is in a suspend state and becomes the recipient of TpBias from a resume initiator.

A unique condition exists when a port is in a ***Suspended Fault*** or a ***Resume Fault*** state.

A port in a ***Suspended Fault*** state will assume the role of a resume initiator when it's ***Control Clear*** register is written with a value if ***Fault Clear*** + ***Initiate Resume***. If the port's ***Control Clear*** register is written with the value ***Fault Clear*** no action will be taken.

A port in a ***Resume Fault*** state will assume the role of a resume target when it's connected port asserts TpBias. Assertion of TpBias by a port to a port with it's ***Fault*** bit set will clear the ***Fault*** bit.

When a resume initiator resides in a node with one or more active ports, transactions from the active port(s) will not be repeated into the resume initiator or from the resume initiator to the resume target.

A resume target responds to a resume initiator by asserting it's TpBias to the resume initiator as well as causing all other connected and suspended ports in it's node to assert TpBias to the port they are connected to (as if the value ***Initiate Suspend*** to the ***Control Clear*** registers of all other suspended and connected ports in it's node which do not have their ***Disabled*** bit set – selecting them as resume initiators).

If a resume initiator does not detect TpBias from it's resume target within a time interval of ***Detect***$_{min}$ (measured from the time the resume initiator's clocks become stable), the resume initiator will set it's own port ***Fault*** bit, no longer drive TpBias to the resume target, and transition to a suspend state.

A port which has it's ***Fault*** bit set and it's ***Bias*** bit clear is an indication of a failed resume response from a resume target. A port in this state will remain in this state until the resume target asserts TpBias (i.e. ***Bias*** sets) at which time ***Fault*** will clear and the port will participate as a resume target.

A connected port with it's ***Fault*** bit set and it's ***Bias*** bit clear will clear it's ***Fault*** bit when it becomes disconnected (i.e. ***Con*** clears).

A connected port with it's ***Fault*** bit set and it's ***Bias*** bit clear will not respond to or generate bus resets or suspend notification.

When a resume target exists in a node which has one or more active ports (i.e. a boundary node), the PHY in which the resume target resides will not propagate any activity from the active bus segment to any port participating in the resume process. The boundary node containing a resume target will wait two **RESET_DETECT** intervals (measured from the time the resume target detected TpBias from it's resume initiator) after which it begins arbitration for control of the active bus (for the purpose of generating an arbitrated short reset to all of it's active ports as well as to all of it's resume targets). If the boundary node detects a short reset on it's resume target before the expiration of the two **RESET_DETECT** delay, it will generate a long reset on all of it's active ports as well as all of it's resuming ports.

A resume initiator will wait five **RESET_DETECT** intervals (measured from the time the resume target begins to assert TpBias) after which it will generate a long reset. The long reset will not be asserted if a short reset is detected prior to the expiration of the five **RESET_DETECT** delay interval.

A five **RESET_DETECT** delay interval provides an opportunity for a resume target in a boundary node to win arbitration for control of it's node's active bus (for the purpose of generating an arbitrated short reset).

The farthest resume target node from a resume initiator will, assuming it resides in a boundary node, begin arbitrating for control of the active bus for the purpose of generating an arbitrated short reset. Arbitration may take as long as two **RESET_DETECT** intervals, therefore, the farthest resume target node may not be able to generate a short reset until the resume initiator has entered into it's fifth **RESET_DETECT** interval. If a bus reset does not occur after five **RESET_DETECT** intervals, the resume initiator will assume a boundary node does not exist and will, therefore, assert a long reset to begin active operation of the bus.

When multiple boundary nodes exist in a resuming suspend domain, they all will arbitrate for control of their respective active domains. The boundary the node which first wins bus arbitration will generate a short reset to it's resuming ports. All other boundary nodes will detect the short reset and will generate a long reset to facilitate the connection of one active domain to another active domain.

A resume initiator will always propagate a bus reset into any other resuming ports in it's node.

When a bus reset is detected by resume target in a boundary node in process of arbitrating for it's active bus, that node will generate a long reset on both the active segment and all of it's resuming ports.

6.7.5.1 Resume Propagation

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

A port which has it's ***Disabled*** bit set will not respond to or generate bus resets, suspend notification or resume events. A "*disabled*" port will not drive it's TpBias.

A connected port in a suspend state will generate a resume event when it ***Control Clear*** *register is written to with* the value ***Initiate Resume*** or when a node receives an indexed register write PHY packet of type ***RESUME***.

A port which generates a resume event becomes a resume initiator. A resume event is generated when the resume initiator asserts TpBias to the suspend port to which it is connected.

A connected port in a suspend state which detects TpBias asserted on it's TPB pair, becomes a resume target.

A resume target responds to a resume initiator by asserting it's TpBias to the resume initiator as well as causing all other connected and suspended ports in it's node to assert TpBias to the port they are connected to (as if the value ***Initiate Suspend*** to the ***Control Clear*** registers of all other suspended and connected ports in it's node which do not have their ***Disabled*** bit set – selecting them as resume initiators).

If a resume initiator does not detect TpBias from it's resume target within a time interval of $\boldsymbol{Detect_{min}}$ (measured from the time the resume initiator's clocks become stable), the resume initiator will set it's own port ***Fault*** bit and no longer drive TpBias to the resume target – it will transition to a low-power consumption state.

A port which has it's ***Fault*** bit set and it's ***Bias*** bit clear is an indication of a failed resume response from a resume target and is defined to be in a ***Resume Fault*** state. A port in this state will remain in a low power state until the resume target asserts TpBias at which time ***Fault*** will clear and ***Bias*** will set - the port will respond as a resume target. A port in a ***Resume Fault*** state may be selected as resume initiator if it's ***Control Clear*** register is written with the value ***Fault Clear*** + ***Initiate Resume***.

A connected port with it’s ***Fault*** bit set and it’s ***Bias*** bits clear will clear it’s ***Fault*** bit when it is disconnected and the port will transition to the disconnected state.

A connected port with it’s ***Fault*** bit set and it’s ***Bias*** bit clear will not respond to or generate bus resets or suspend notification.

A resume initiator will always propagate a bus reset into any other port in it’s node which is in the process of resuming.

6.7.5.2 Resume via Extended PHY Packet

A node’s suspended ports may be selected as resume initiators via an extended PHY packet of type ***Resume***. The extended PHY packet may be generated remotely and enter a node through an active port or it may be generated by the node’s link through the PHY-Link interface.

A node which receives a ***Resume*** PHY packet configures all its connected and suspended ports to be resume initiators - as if each port had the value ***Resume Initiate*** written to it’s ***Control Clear*** register.

Each resuming port in the node will generate a resume event to it’s connected port.

A PHY response packet is not returned by a node receiving an extended PHY packet of type ***Resume***.

6.7.6 Detach Detection During Suspend

A disconnect between two connected ports that are in a suspend state is detected when the TPA port current source current sink through the cable connection is lost as a result of the cable being disconnected. When the cable connection is removed from one port or the other, a rise in voltage on the input to a port’s voltage sense circuitry occurs. The voltage sensing circuit generates connect and disconnect notification. Connect notification results in a resume event on both ports and both port’s ***Con*** bits are set. Disconnect notification results in both port’s ***Con*** bit clearing - a resume event is **not** generated as a result of a disconnect while a port is in the suspend state.

6.7.7 Connect Detection During Suspend

A node will power up with all of it’s ports in a suspend state. When a port powers up it may or may not have a connection to a port on another node. This section describes the behavior for both instances.

6.7.7.1 Connect Detection During Suspend/Power-Up

A node, when it powers up, will have all ports configured as shown in the figure below:

Figure 0-9 – **Port Power-up Configuration**

Bit name & State

Register	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
000b	Astat		Bstat		Ch	Con	Bias	Disabled
(Status)	X	X	X	X	X	0	0	Y*
010b	Chg_int_en	Suspend	Disable	Enab_token	Fault	rsrvd		
(Control)	0	0	0	0	0			

*Y=Power-up state is dependent upon power-up strapping option.

During the power-up process, it is determined whether a port has a connection to another port or not.

When a port has a to another port, the output of it’s voltage sense circuit (see figure 0-4) will be in a high state. When in this state, the port will sample it’s TPB pair for the presence of TpBias.

If the port detects an incoming TpBias on it’s TPB pair the port will assert TpBias on it’s TPA pair, set it’s ***Bias*** bit and cause all other connected and suspended ports in it’s node to assert TpBias to the port they are connected to (as if the value ***Initiate Suspend*** to the ***Control Clear*** registers of all other suspended and connected ports in it's node which do not have their ***Disabled*** bit set – selecting them as resume initiators).

If an incoming TpBias is not detected on it's TPB pair, the port will assert TpBias on it's TPA pair to it's connected port for a time period of ***Bias***$_{hold}$ after which it will sample for TpBias on it's TPB pair. If it does not detect TpBias, it will set it's ***Fault*** bit and no longer drive TpBias - it's ***Bias*** bit will be clear and it's ***Con*** bit will be set. When TpBias is detected, the ***Fault*** bit will clear and the port will participate as a resume target.

If, during the ***Bias***$_{hold}$ time, TpBias is detected on it's TPB pair, it's ***Bias*** bit will set and cause all other connected and suspended ports in it's node to assert TpBias to the port they are connected to (as if the value ***Initiate Suspend*** to the ***Control Clear*** registers of all other suspended and connected ports in it's node which do not have their ***Disabled*** bit set – selecting them as resume initiators).

6.7.8 Port Power Loss During Suspend

When power is lost, all connected ports continue to provide a current sink to their connected ports that are in a suspend state and no event notification will be provided to the suspended port connection.

A port which had an active connection to a port on another node will no longer assert TpBias to it's connected port. If the connected port is an IEEE 1394-1995 PHY it will see the loss of TpBias as a disconnect event and respond accordingly.

If the connected port is a suspend/resume capable PHY, the connected port will see a drop in it's incoming TpBias and will respond as outlined in section 6.7.4.

6.7.9 Reset Propagation

A reset will not propagate into and/or through a port with either it's ***Bias*** bit clear or it's ***Fault*** or ***Disabled*** bits set.

6.7.10 PHY-Core Suspend/Resume Control

A PHY core will not enter into a suspend state as long as LPS is asserted by the link.

If a port's ***Chg_int_en*** bit is set, an interrupt event to the link will occur with a 4-bit status packet each time a port's ***Control*** port state changes.

In this manner, a link can be notified each time a PHY port enters into a suspend, fault, or disabled state.

When all ports on a PHY are inactive (i.e. suspended, disabled, or not connected) a link can, at it's option, remove LPS from the PHY. It should be noted, however, the Link may remove LPS (at its option) regardless of the state of any or all ports in the PHY.

When all ports on a PHY are inactive (i.e. suspended, disabled, or not connected) and LPS is removed by the link, the PHY will enter into a suspend state.

A suspended PHY maintains an ability to generate a LinkOn when one of it's port's ***Control*** register bits changes state.

However, if a PHY's ***Resume*** bit (bit 0 of PHY node register 0101_b) is clear, the PHY will not generate a LinkOn to the link.

The link can set the PHY ***Resume*** bit through the PHY-link interface, however, ***Resume*** cannot be set or cleared via an indexed register write extended PHY packet.

If, while the PHY is in a suspend state, the Link asserts LPS, the PHY core will transition to an active state. If the node ***Resume*** bit is set, all suspended and connected ports in the PHY will transition into a resume state.

If, however, the node ***Resume*** bit is clear, none of the suspended and connected ports in the PHY will transition to the resume event until selected as a resume initiator.

6.7.10.1 PHY-Core Control from the Link

A link can instruct a PHY to enter a suspend state by selecting each of the PHY ports to become suspend initiators or by writing to the ***Control Set*** register of each port with the value ***Initiate Disable***. When the last port has entered into a suspend or disabled state, the link may then remove LPS from the PHY and the PHY will transition to a suspend state.

The software stack controlling the link must be cognizant of the timing constraints involved (i.e. LPS should not be removed before all ports in the PHY have completed transitioning to a suspend or disabled state.

If the link disables all ports, and, once all ports have transitioned to the disabled state, the link could then (after some ____TBD____ delay) enable all ports by writing the value ***Initiate Enable*** to the ***Control Clear*** register of each port which has it's ***Bias*** bit clear. The link would then set the node's ***Resume*** bit followed by a removal of the LPS signal to the PHY. This would precondition all ports in the suspended PHY to respond to a resume event - resuming the PHY and asserting LinkON to the link (notifying the link of a resume event on the PHY).

Note: The point to note here is that the PHY core may resume from a suspend state when LPS is not asserted but the PHY's ***Resume*** bit is set and it's ports are configured to respond to a resume event.

6.7.11 Resume/Resume Collisions

When a port is involved in the process of performing the function of a resume target when a suspended port in the same node detects a resume event, the suspended port in the node will begin the process of responding as a resume target to it's resume initiator **and** will delay four **RESET_DETECT** intervals before generating a long reset (providing a reset was not detected during the four **RESET_DETECT** delay). If the node detects a reset from the first resume initiator before the second resume event has been active for two **RESET_DETECT** intervals, the reset will not be propagated to the second resume initiator. If a reset is detected from the second resume initiator before a reset is detected from the first resume initiator, the reset will be propagated to the first resume initiator.

6.7.12 Suspend/Resume Collisions

If a resume event is detected by a resume target in a node containing a port involved in the process of performing the function of a suspend initiator or suspend target, the resume event will not be propagated into the resume target for a delay of $\boldsymbol{time}_{res\_notify}$, thus providing sufficient time for completion of the process currently underway.

6.7.13 Port Disable

A port will transition to a disabled state when it's port ***Control Set*** register is written to with the value ***Initiate Disable***.

The target node of an indexed register write extended PHY packet generates a response packet.

The figure below illustrates the signals which could be observed on the bus by both the port connected to a port in the process of transitioning to a disabled state and the ports connected to the other ports in a node containing the port transitioning to the disabled state.

Figure 0-10 – Bus Signals Received by active ports in a Node containing a Disabling Port

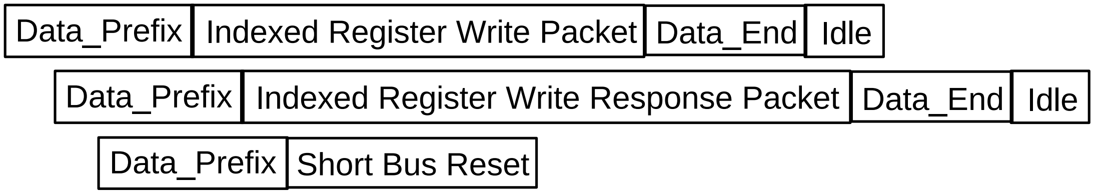
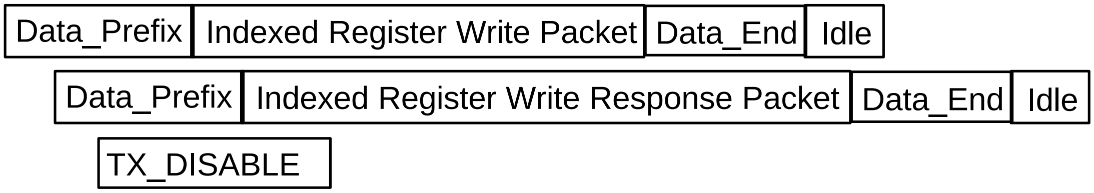


Figure 0-11 – Bus Signals Received by a port connected to a disabling port


Data_Prefix
Indexed Register Write Packet
Data_End
Idle
Data_Prefix
Indexed Register Write Response Packet
Data_End
Idle
TX_DISABLE

A port concludes the process of transitioning to a disabled state by asserting a short reset to all of it's active ports and an arbitration state of **TX_DISABLE** to the port it is connected to for a time period of ***Notify***$_{hold}$. Upon removing **TX_DISABLE**, it then disables it's TpBias generator (turning it off, causing it's output to go to a high impedance condition).

The port on the far end (connected to the newly disabled port) receives **TX_DISABLE** as **RX_DISABLE**. A p1394a PHY will arbitrate its active bus for the purpose of asserting a short reset. A 1394-1995 PHY will interpret the **RX_*DISABLE*** *as* **ROOT_CONTENTION**. It has been determined that a 1394-1995 PHY will assert a long bus reset when ***ROOT*_CONTENTION** is detected.

When the port connected to a disabling port detects a loss of TpBias from the disabled port to which it is connected, it will responds to the event as outlined previously.

When a port is selected by it's associated link to be disabled, the node in which the port to be disabled resides will arbitrate the bus. When the node gains control of the bus, it generates an extended PHY packet of type "indexed register write" with it's own node ID, page, port and reg values to access the ***Control Set*** register of the port to be disabled with a value field value of ***Initiate Disable***. This notifies all other nodes on the bus of the event.

If the node containing the port to be disabled is not able to gain control of the bus (via arbitration) for three **RESET_DETECT** intervals, the node will assert a long reset to all connected and active ports in it's node and will assert **TX_DISABLE** to the port it is connected to for a time period of ***Notify***$_{hold}$. Upon removing **TX_DISABLE**, it then disables it's TpBias generator (turning it off, causing it's output to go to a high impedance condition).

A port which has been disabled as the result of a PHY packet cannot have it's ***Disabled*** bit cleared as the result of a disconnect or a connect event. A port's ***Disabled*** bit can only be cleared by a indexed register write extended PHY packet.

A disabled port with a connection to another port will have the following ***Control*** and ***Status*** register configuration:

Figure 0-12 – **Configuration for a connected and disabled port**

	Bit name & State							
Register	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
000b	Astat		Bstat		Ch	Con	Bias	Disabled
(Status)	X	X	X	X	X	1	Y*	1
010b	Chg_int_en	Suspend	Disable	Enab_token	Fault	rsrvd		
(Control)	X	0	0	X	0			

*Note: If the port to which the disabled port is connected is an IEEE 1394-1995 PHY, the disabled port's ***Bias*** bit will be set. If the port to which the disabled port is connected is a suspend/resume capable PHY, the disabled port's ***Bias*** bit will be clear.

6.7.14 Link "*Wake*" Notification

The link is notified of suspend/resume activity via the LinkOn signal line from the PHY and/or via status packets from it's PHY. The link will only be notified of suspend/resume activity under certain and specific conditions.

The link maintains control of it's own wake notification via the port ***Chg_int_en*** bit and the node ***Resume*** bit.

The PHY will generate a 4-bit PHY interrupt status packet to the link when a port's status change occurs (***Bias***, ***Disabled***, ***Con***, or ***Fault*** bits change state), if it's ***Chg_int_en*** bit is set, and LPS is active (indicating an active and powered-on link) - regardless of the state of the node's ***Resume*** bit.

The PHY will generate a LinkOn when a port's status change occurs (***Bias***, ***Disabled***, ***Con***, or ***Fault*** bits change state), if it's ***Chg_int_en*** bit is set, and LPS is inactive (indicating an inactive and/or powered-off link) only when the node's ***Resume*** bit is set. When the node's ***Resume*** bit is set, the PHY will always generate a LinkOn when a resume event occurs in a node - regardless of the state of any port's ***Chg_int_en*** bit.

The figure below provides further clarification of link wake notification:

Figure 0-13 –**Link *Wake* Notification**

State Condition:				**Event Generated:**	
LPS Signal	***Resume* bit Set**	***Chg_int_en* bit Set**	**Change in Port Status***	**LinkOn**	**Status Packet**
Yes	X**	Yes	Yes	No	Yes
Yes	Yes	Yes	No	No	No
Yes	X	No	X	No	No
Yes	No	Yes	No	No	No
No	Yes	Yes	Yes	Yes	No
No	Yes	X	No	No	No
No	Yes	No	Yes	Yes***	No
No	No	X	X	No	No

*Note: Port status change occurs when ***Bias***, ***Disabled***, ***Con***, or ***Fault*** bits change state.
**Note: X state is a *Don't Care* state.
***Note: LinkOn is only generated as the result of a resume event.