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
		•

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;
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 connected and 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;
$\boldsymbol{Notify_{hold}}$: A period of time defined as 8,192 SCLKs - approximately 166.67 µs;
$\boldsymbol{Bias_{hold}}$; A period of time defined as 8,192 SCLKs - approximately 166.67 µs;
$\boldsymbol{Detect_{min}}$: A period of time defined as 8,192 SCLKs - approximately 166.67 µs;
$\boldsymbol{time_{res\_notify}}$: A period of time defined as 16,384 SCLKs - approximately 333.33 µs;
RESET_DETECT interval: 80.0 milliseconds, maximum 85.3 milliseconds
TX_DISABLE**:** Arb_A_Tx = 0, Arb_B_Tx = 0
TX_SUSPEND: Arb_A_Tx = ‘Z,’ Arb_B_Tx = ‘1’

6.7.1 Port Register

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***, ***Suspend***, ***Disable***, and ***Fault***.

	Contents							
Address*	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
(1)000b	Astat			BStat		Ch	Con	Bias	rsrvd
(1)001b	Negotiated_speed					rsrvd		
(1)010b	Chg_int_en	Suspend	Disable	Enab_token	Fault		rsrvd	
(1)011b	Chg_int_en	Suspend	Disable	Enab_token	Fault		rsrvd	
(1)100b	rsrvd							
(1)101b	rsrvd							
(1)110b	rsrvd							
(1)111b	rsrvd							

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

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

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

*Note: The most significant bit of the address (parenthesized) is used only by the link during a PHY register 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 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 010b) 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 011b) 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.

Note: The ***Suspend* bit in the control register does not become set or cleared during a PHY register write transaction. The ***Suspend*** bit is set or cleared by the internal state machine based upon the presence of incoming TpBias. The ***Suspend*** bit is set when incoming TpBias is NOT present. The ***Suspend*** bit is clear when incoming TpBias is present.

Figure 0-3 – Port Register Map Bit Field Definitions

Register (1)000 - ***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.
Rsrvd	1	ro	reserved - must not be used or interpreted as having a meaningful value

Register (1) 001b - ***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) 010b and (1) 011b - ***Control Set*** and ***Control Clear*** (respectively)

Field	Size	Type	Description

Chg_int_en	1	rw	***change interrupt enable*** - if set, the PHY will interrupt the link with a 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.
Suspend	1	ro	When writing this bit at register ***Control Set***, the bit does **not** set, however the write operation will select the port as a suspend initiator; writing this bit at register ***Control Clear*** will **not** clear this bit, however the write operation will select the port as a resume initiator; this bit is set subsequent to a power reset.
Disable	1	rw	when set, a port is disabled (placing the port in a low power state similar to suspend); a disabled port will not respond to or generate bus resets, suspend notification, or resume events; this bit is clear subsequent to a power reset.
Enab_token	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; 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)100b, (1)101b, (1)110b, (1)111b

Field	Size	Type	Description
rsrvd	8	ro	reserved - must not be used or interpreted as having a meaningful value

1
2
3

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 the port's ***Control Set*** register is written with ***Suspend*** by a PHY register write packet. The port thus selected becomes the suspend initiator. The 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 in which both ports are 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.

Suspended ports connected within a suspend domain can be restored to normal operation via a "resume" event (within the constraints outlined in proceeding sections). A resume event may be initiated by any port in a suspend state.

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: a) not connected, or b) has it's ***Disable*** bit set; or c) has it's ***Suspend*** bit set will report a disconnected status in it's node self-ID.

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 another node's 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 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 it's output.

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

Current is supplied through a cable connection to a destination 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 ***Suspend*** bit is set (i.e. port is suspended);

Figure 0-4 shows the cable media signal interface configuration - including circuitry required to be active in a suspended and connected state.

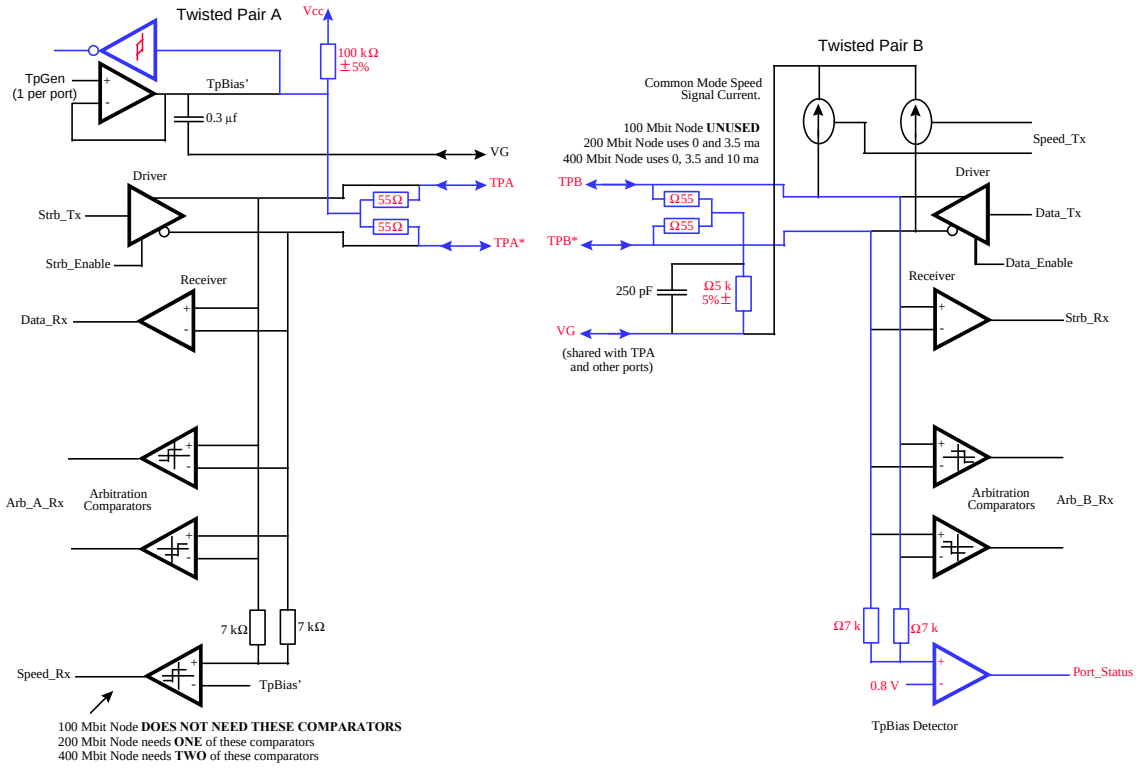


Figure 0-4 – Cable media signal interface configuration with Suspend Connection Circuitry

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.

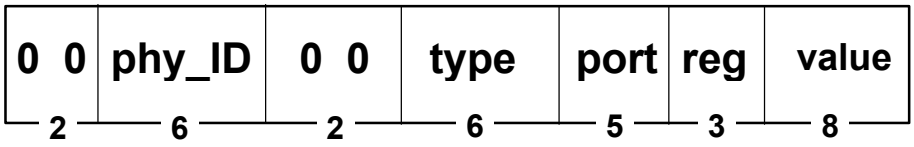
Connect and disconnect detection while a port is in a suspended state is obtained via the output of voltage sense circuitry whose input is connected to the TpBias generator output. A leakage current source (attached to the TpBias generator output) is delivered into a current sink on the TPB port connection of a suspended port on another node 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 and the port's ***Con*** bit is set - among other things (outlined in proceeding sections). Disconnect notification results in the port's ***Con*** bit being cleared - a resume event is **not** generated.

6.7.3 PHY Register Addressing

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

Figure 0-5 – PHY Control Packet

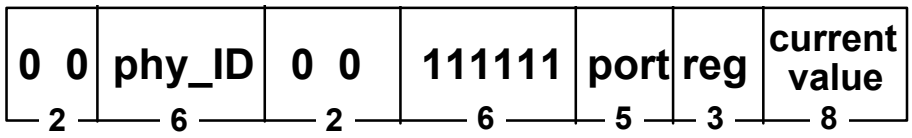


The fields in this packet are defined as follows:

phy_ID: 6-bit node address of the port's 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 register select field used to address a particular port or node register
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.

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 0-6

Figure 0-6 – PHY Access Response Packet



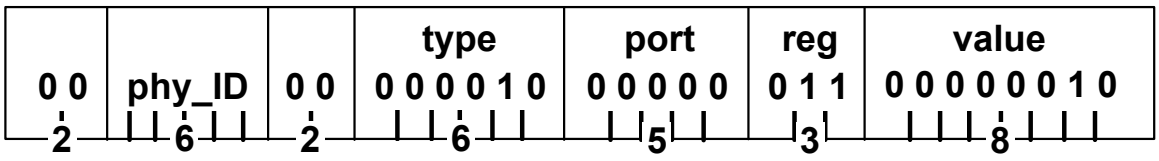
The fields in this packet are defined as follows:

phy_id: node ID of node generating the response packet;
type: 111111b;
port: port register value (same as the port register value provided in the extended PHY packet for which the response is being generated);
reg: register value (same as 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, the value returned will be zero for all bits.

A Bus Manager may send an extended PHY packet of type "***Resume***."

An extended PHY packet of type ***Resume*** takes the following format:

Figure 0-7 – Extended PHY Packet of type "*Resume*"



The fields in this packet are defined as follows:

phy_id: node ID of node generating packet;
type: 000010b (***Resume***);

port: port register value = 00000b;
reg: port ***Control Clear*** register value = 011b;
value: 00000010b (***Suspend*** bit position set);

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 ***Suspend*** written to it's ***Control Clear*** register.

Each port selected as a resume initiator will generate a resume event to it's connected port.

The node which receives the ***Resume*** PHY packet will wait five **RESET_DETECT** times before asserting a reset on it's resume initiators and connected and active ports (following the constraints outlined in section 6.7.5).

6.7.3.1 *REG* Value Usage

For port values of $0\text{-}26_{10}$ inclusive: register 000b (***Status A***) and 001b (***Status B***) are read only for link access or an extended PHY packet command.

A read from 010b (***Control Set***) or 011b (***Control Clear***) returns the current value of the port's ***Control*** register - the contents of the ***Control Set*** and ***Control Clear*** registers are identical.

Writes to 010b (***Control Set***) set corresponding bits (as specified in the 'value' field) in the port's ***Control*** register - with the exception of the ***Suspend*** bit which is set only when the incoming TpBias.to a port is not present

Writes to 011b (***Control Clear***) clears the corresponding bits (as specified in the 'value' field) in the port's ***Control*** register - with the exception of the ***Suspend*** bit which is cleared only when the incoming TpBias.to a port is present.

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});

6.7.4 Entry into Suspend

A port will initiate suspend notification when it's port ***Control Set*** register is written with ***Suspend*** from either a Bus Manager or the link associated with the node in which the port resides. Each method is described in the following sections.

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.

While in a suspend state, a port should not consume more than 1.65 milliwatts.

When a port detects it's incoming TpBias decrease to below 0.4 volts, it will set it's own ***Suspend*** 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 in which the port resides that detected a loss of incoming TpBias will generate a short reset on all of it's connected and active ports.
A port which has it's ***Suspend*** bit set 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 Suspend Initiator selected via an Extended PHY Packet

An extended PHY packet of type 'PHY Register Write" to a node's port ***Control Set*** register with the value ***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 a PHY response packet which includes the value contained in the initiator's ***Control*** register.

Upon completion of sending the PHY response packet, the initiator asserts **TX_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 connected and active ports except the target, which is sent a **TX_SUSPEND** (Arb_A_Tx = 'Z,' Arb_B_Tx = '1').

The initiator holds **TX_SUSPEND** to the target for a time of $\textbf{\textit{Notify}}_{\textbf{\textit{hold}}}$, waiting for the target to respond by driving it's TpBias to the initiator to below 0.4 volts. Upon seeing the low voltage TpBias from the target, the initiator completes the suspend connection protocol process by setting it's ***Suspend*** bit and driving TpBias below 0.4 volts to the target for a time of $\textbf{\textit{Bias}}_{\textbf{\textit{hold}}}$. After driving a low voltage TpBias to the target for the specified time, the initiator disables the output of it's TpBias generator (generator output goes to a high impedance 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 during when a port is in a suspend state).

If the initiator does not detect an incoming low voltage TpBias during $\textbf{\textit{Notify}}_{\textbf{\textit{hold}}}$ time while asserting **TX_SUSPEND** to the target, 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 enters into a suspend state. If the voltage sense circuitry output is in a logic low state (indicating no connection to another port) the initiator enters into a disconnected suspend state (i.e. ***Suspend*** bit set, ***Fault*** bit clear, ***Disable*** bit clear, ***Con*** bit clear, ***Bias*** bit clear) until a connection is detected at which time a resume event will be generated.

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 which has ***Fault*** set will not respond to or generate bus resets, suspend initiator selection, resume initiator notification, suspend notification or resume events.

If a connected port with it's ***Fault*** bit set becomes disconnected, the ***Fault*** bit will clear and the port will enter into a disconnected suspend state. A resume event or a bus reset will not occur when the disconnect event takes place.

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 Selected from the Link

A link selects a port to become a suspend initiator by generating an extended PHY packet of type "PHY Register Write" to it's own PHY. The PHY packet will have the node ID of the PHY associated with the link. The PHY packet will address the port ***Control Set*** register of the port to be selected as a suspend initiator with the data field valude of ***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 "PHY register write" with it's own node ID, port ID, and register value of ***Suspend***. This notifies all other nodes on the bus (at least those that are interested) of the event. The Bus Manager (if not the node that generated the PHY packet) understands that it did not send the PHY packet, therefore, it knows the only other condition which would result in such a packet is when the suspend initiator request came from a nodes own link. The Bus Manager performs the same internal processing (if any) as it would have if it had sent the PHY packet.

The suspend initiator asserts **TX_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 connected and active ports except the target, which is sent a **TX_SUSPEND** (Arb_A_Tx = 'Z,' Arb_B_Tx = '1').

The suspend connection state process now continues as previously outlined in clause 6.7.4.1.

6.7.4.3 Target Response to Suspend Request

When an initiator generates **TX_SUSPEND** (Arb_A_Tx = 'Z,' Arb_B_Tx = '1') to the target, it will be received as an **RX_SUSPEND** (Arb_A = '1,' Arb_B = 'Z'). Under normal operating circumstances, the target would interpret **RX_SUSPEND** as **RX_CHILD_HANDSHAKE** - an acknowledgment by the attached peer PHY of TX_CHILD_NOTIFY (the peer PHY is a child of this PHY). In this circumstance, an event has **not** occurred in which the target would be expecting to receive an **RX_CHILD_HANDSHAKE** from an attached peer PHY, therefore, the target interprets the *out-of-context* **RX_CHILD_HANDSHAKE** as a suspend notification event from an initiator.

The target, upon seeing **RX_SUSPEND**, responds by repeating the **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 $\boldsymbol{Bias_{hold}}$ (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 $\boldsymbol{Bias_{hold}}$ 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 during 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 setting it's ***Suspend*** bit.

If the target continues to detect TpBias from the initiator (***Bias*** bit of the suspend target is set) after the $\boldsymbol{Bias_{hold}}$ time of asserting of it's low voltage TpBias to the initiator, the target sets it's ***Fault*** bit. If, however, the output of the voltage sense circuitry is a logic low and the ***Bias bit is clear*** (indicating no port connection) the target will enter into a disconnected suspend state (i.e. ***Suspend*** bit set, ***Fault*** bit clear, ***Disable*** bit clear, ***Con*** bit clear, ***Bias*** bit clear) until a connection is detected at which time a resume event will be generated.

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 which has ***Fault*** set will not respond to or generate bus resets, suspend initiator selection, resume initiator notification, suspend notification or resume events.

If a connected port with it's ***Fault*** bit set becomes disconnected, the ***Fault*** bit will clear and the port will enter into a disconnected suspend state. A resume event or a bus reset will not occur when the disconnect event takes place.

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 connected and 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:

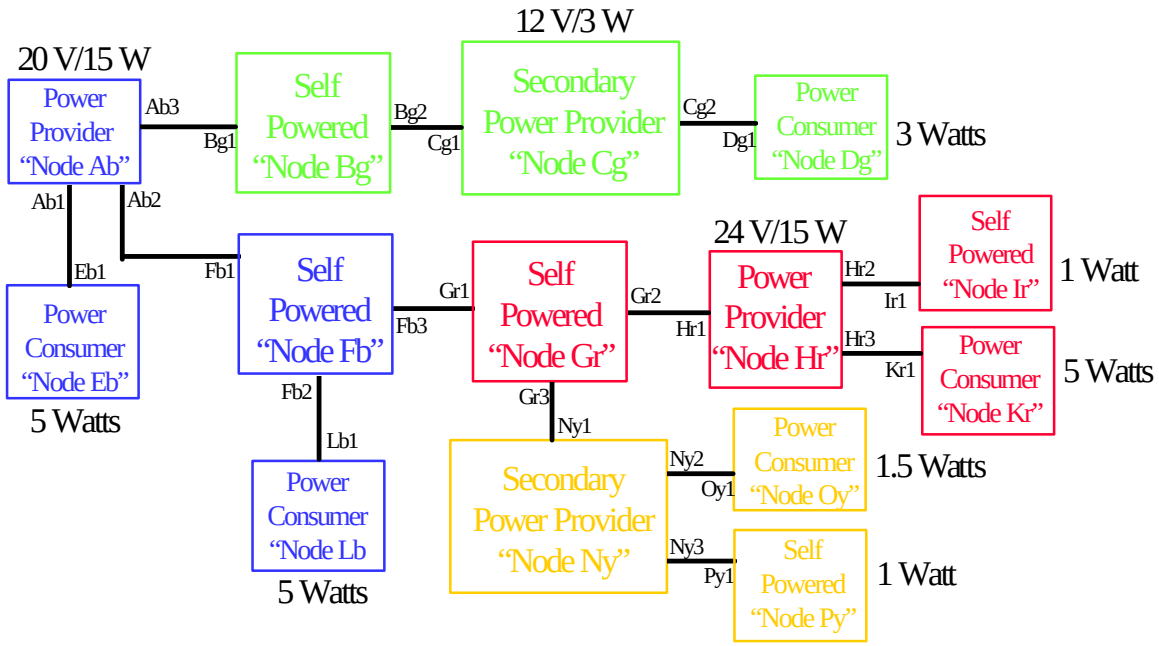


Figure 0-8 – Example Suspend/Resume Domain Topology

Further, assume node Bg is the Bus 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 a PHY register write packet to node Gr, port Gr3, setting port Gr3's ***Disable*** bit. Port Gr3 immediately drops it's TpBias to port Ny1, setting the ***Suspend*** bit in Ny1. Port Ny1 responds by dropping it's TpBias to port Gr3, setting the ***Suspend*** 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.

6.7.5 Resume (Exit from a Suspended-Connected State)

A connected port in a suspend state will generate a resume event when the port's ***Control Clear*** register is written with the vaulue ***Suspend*** or when a node receives an extended PHY packet of type '***RESUME***' (000010b - see section 6.7.3).

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.

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 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 upon which it then writes the value ***Suspend*** to the ***Control Clear*** registers of all other suspended and connected ports in it's node which do not have their ***Disable*** 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 (i.e. it will re-enter the suspend state).

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

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

A connected port with it's ***Fault*** and ***Suspend*** bits set 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, the PHY in which the resume target resides will not propagate any activity from the active bus segment to any resume initiator. The PHY in which the resume target exists asserts **TX_REQUEST** on all of it's ports in the process of resuming (resume initiators). **TX_REQUEST** is asserted within one **RESET_DETECT** interval from when the resume target first began asserting TpBias to it's resume initiator. The PHY 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 connected and active ports as well as to all of it's resume targets). If the PHY 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 connected and active ports as well as all of it's resume initiators.

A resume initiator which detects a **TX_REQUEST** from it's resume target will propagate the **TX_REQUEST** to all other ports in it's node which may be in the process of resuming.
A resume initiator will wait five **RESET_DETECT** intervals (measured from the time the resume target cleared it's ***Suspend*** bit in it's own ***Control*** register) after which it will generate a long reset - providing it has not seen detected a reset it's resume target during the delay period..

A five **RESET_DETECT** delay interval provides an opportunity for a resume target (which may be connected to a boundary node as far 16 hops away from the original resume initiator) to request an opportunity to arbitrate for control of an active bus (for the purpose of generating an arbitrated short reset) to which it may have an active port connection.

The farthest resume target node from a resume initiator (16 hops) will have TpBias restored to all of it's suspended ports in approximately 170 milliseconds (assuming resume event propagation takes as long as 10 milliseconds per node and a single **RESET_DETECT** interval of approximately 85 milliseconds, the farthest resume target will see TpBias restored in about two **RESET_DETECT** intervals).

The farthest resume target node from a resume initiator will (as previously described), assuming it has one or more active port connections to an active bus, will 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 there is no connection to an active bus segment, therefore, the resume initiator will asserts a long reset.

When multiple resume target nodes are asserting **TX_REQUEST**, the node which first wins bus arbitration with it's active bus segment will generate a short reset to it's resume initiator.

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

When a bus reset is detected by another resume target node which also had asserted **TX_REQUEST**, that node will generate a long reset on both the active segment to which it is attached and it's resume initiator.

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 ***Disable*** 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 ***Suspend*** or when a node receives an extended PHY packet of type '***RESUME***' (000010b - see section 6.7.3).

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 upon which it then writes the value ***Suspend*** to the ***Control Clear*** registers of all other suspended ports in it's node which do not have their ***Disable*** 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 (i.e. it will re-enter the suspend state).

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

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

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

A resume initiator which detects a **TX_REQUEST** from it's resume target will propagate the **TX_REQUEST** to all other ports in it's node which may be in the process of resuming.

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

6.7.6 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 New Connect Detection During Suspend

A node port with no connection, when it powers up, will have all ports configured as shown in figure 0-9:

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

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

When a port has a new connection made to it, the output of it's voltage sense circuit (see figure 0-4) will transition from a low state to a high state. When the port sees this transition, it will sample it's TPB pair for the presence of TpBias.

If the newly connected port detects an incoming TpBias on it's TPB pairthe port will assert TpBias on it's TPA pair, clear it's ***Suspend*** bit and generate a resume event to all other ports in it's node which have a suspend connection (but not disabled) by writing to their ***Control Clear*** registers the value ***Suspend*** (selecting them as resume initiators). Resume events are propagated through the node's resume initiator's as previously outlined.

If an incoming TpBias is not detected on it's TPB pair, the newly connected port will assert TpBias on it's TPA pair to it's connected port for a time period of $\boldsymbol{Bias_{hold}}$ 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 ***Con*** bit will still be set. When TpBias is detected, the ***Fault*** bit will clear and the port will participate as a resume target.

If, during the $\boldsymbol{Bias_{hold}}$ time, TpBias is detected on it's TPB pair, it's ***Suspend*** bit will clear and it will generate a resume event to all other ports in it's node which have a suspend connection (but not disabled) by writing to their ***Control Clear*** registers with the value ***Suspend*** (selecting them as resume initiators). Resume events are propagated through the node's resume initiator's as previously outlined.

6.7.7.2 Existing Connect Detection During Suspend

When a port in a node (with a connection to a port on another node) powers up, the output of it's voltage sense circuit will be high (indicating a connection condition).

The port will sample it's TPB pair for the presence of a TpBias. If it detects an incoming TpBias on it's TPB pairthe port will assert TpBias on it's TPA pair, clear it's ***Suspend*** bit and generate a resume event to all other ports in it's node which have a suspend connection (but not disabled) by writing to their ***Control Clear*** registers with the value ***Suspend*** (selecting them as resume initiators). Resume events are propagated through the node's resume initiator's as previously outlined.

If an incoming TpBias is not detected on it's TPB pair, the newly connected port will assert TpBias on it's TPA pair to it's connected port for a time period of $\boldsymbol{Bias_{hold}}$, 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 $\boldsymbol{Bias_{hold}}$ time, TpBias is detected on it's TPB pair, it's ***Suspend*** bit will clear and it will generate a resume event to all other ports in it's node which have a suspend connection (but not disabled) by writing to their ***Control Clear*** registers with the value ***Suspend*** (selecting them as resume initiators). Resume events are propagated through the node's resume initiator's as previously outlined.

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 ***Suspend***, ***Fault*** or ***Disable*** 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.

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 0101b) 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 a bus extended PHY register write packet.

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 setting the ***Disable*** bit in all of the ports in the PHY. When the last port has entered into a suspend state, the link may then remove LPS from the PHY and the PHY will enter into 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 entering a suspend state or have been successfully disabled).

If the link uses the ***Disable*** bit, once all ports have been disabled, the link could then (after some TBD delay) make certain all ***Suspend*** bits in all ports of the PHY are set, after which the link clears all ***Disable*** bits in all ports of the PHY, followed by setting the PHY ***Resume*** bit, and then remove LPS 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).

6.7.11 Resume/Resume Collisions

When a port in a node is in the process of participating as a resume target as a result of detecting a resume event from a connected resume initiator and a suspended port in the same node as the resume target detects a resume event from a second connected resume initiator (as opposed to a resume event generated from the current resume target in it's own node), the second suspended port in the node will begin a resume target response to it's resume initiator **and** assert **TX_REQUEST** on both resume initiators. When four **RESET_DETECT** intervals have elapsed since the detection of the second resume event, the node will generate a reset (providing no other node has generated a reset). If the node in which the resume targets reside receives a reset from

the first resume initiator before the second resume event has been active for two **RESET_DETECT** intervals, the reset is not propagated to the second resume initiator. If a reset is received from the second resume initiator before a reset is seen from the first resume initiator, the reset is 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 recently selected as a suspend initiator or suspend target, the resume event will not be propagated into the resume target for a delay of $\boldsymbol{time_{res\_notify}}$. If the newly selected suspend initiator (or target) in the node happens to be the last port in the node to suspend, the nodes clocks will not enter into a suspend state for 1mS after detecting all ports in the node have their ***Suspend*** bit set.

6.7.13 Port Disable

A port will be disabled when it's port ***Control Set*** register is written to with the ***Disable*** bit set via an extended PHY packet of type 'PHY register write.'

When a port has it's ***Disable*** bit set the port's node immediately responds with a PHY response packet which includes the value contained in the ***Control*** register of the port to be disabled.

Upon completion of sending the PHY response packet, the node in which the port to be disabled resides asserts **TX_DATA_END** to all of it's active ports followed by a short reset to all but the port to be disabled. The disabled port asserts **TX_DISABLE** and ceases to drive TpBias (i.e. it's TpBias generator is turned off - output goes to a high impedance). The port being disabled asserts **TX_DISABLE** for a time period of $\boldsymbol{Notify_{hold}}$.

The port on the far end (connected to the newly disabled port) receives **TX_DISABLE** as **RX_DISABLE** and will respond as if it had detected an **RX_ROOT_CONTENTION** arbitration value. When the port on the far end detects a loss of TpBias from the disabled port to which it is connected, it will responds as previously outlined.

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 "PHY register write" with it's own node ID, port ID, and control register value with bit 2 of the data field set - as if it were going to set it's own Disable bit. This notifies all other nodes on the bus (at least those that are interested) of the event. The Bus Manager (if not the node that generated the PHY packet) understands that it did not send the PHY packet, therefore, it knows the only other condition which would result in such a packet is when the port disable command came from a nodes own link. The Bus Manager performs the same internal processing as it would have if it had sent the PHY packet.

At the conclusion of the PHY packet, the node in which the port to be disabled resides asserts **TX_DATA_END** to all of it's active ports followed by a short reset to all but the port to be disabled. The disabled port asserts **TX_DISABLE** and ceases to drive TpBias (i.e. it's TpBias generator is turned off - output goes to a high impedance). The port being disabled asserts **TX_DISABLE** for a time period of $\boldsymbol{Notify_{hold}}$.

The port on the far end (connected to the newly disabled port) receives **TX_DISABLE** as **RX_ROOT_CONTENTION**. When the port on the far end detects a loss of TpBias from the disabled port to which it is connected, it will responds as previously outlined.

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 asserts **TX_DISABLE** (Arb_A_Tx = 0, Arb_B_Tx = 0) and ceases to drive TpBias (i.e. it's TpBias generator is turned off - output goes to a high impedance). The port being disabled asserts **TX_DISABLE** for a time interval of $\boldsymbol{Notify_{hold}}$. The port on the far end

(connected to the newly disabled port) receives **TX_DISABLE** as **RX_ROOT_CONTENTION**. When the port on the far end detects a loss of TpBias from the disabled port to which it is connected, it will responds as previously outlined.

A port which has been disabled as the result of a PHY packet (either from it's associated link or from a Bus Manager) cannot have it's ***Disable*** bit cleared as the result of a disconnect or a connect event. The ***Disable*** bit can only be cleared by a PHY packet command (from either it's associated link or from a Bus Manager via a connected and active port in the node which the disabled port resides).

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

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

Figure 0-10 – **Configuration for a connected and disabled port**

*Note: If the port to which the disabled port is connected is an IEEE 1394-1995 PHY, the disabled port's ***Suspend*** bit will be clear and it'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 ***Suspend*** bit will be set and it'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 status packet to the link when a port's status change occurs (***Suspend***, ***Disable***, or ***Fault*** bits change state), it's ***Chg_int_en*** bit 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 (***Suspend***, ***Disable***, or ***Fault*** bits change state), it's ***Chg_int_en*** bit 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 0-11 –**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 ***Suspend***, ***Disable***, 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.