CONGRUENT SOFTWARE, INC.
3998 Whittle Avenue
Oakland, CA 94602
(510) 531-5472
(510) 531-2942 FAX

FROM: Peter Johansson

TO: IEEE P1394a Working Group

DATE: January 29, 1998

RE: Connect detect mode for P1394a PHYs

There is an asymmetry in the current suspend / resume proposal in P1394a Draft 1.4. A remote PHY requested to suspend a port may refuse the request by the return of a confirmation packet whose `ok` bit is zero. No similar mechanism exists to refuse the request presented by receipt of an RX_SUSPEND or RX_DISABLE_NOTIFY signal.

These signals are ignored by IEEE Std 1394-1995 PHYs—they are not cognizant of these newly defined arbitration line states. Hence a P1394a PHY that transmits either one of these signals has to be prepared if the receiving PHY ignores them (the state diagrams and C code in clause 7.10.4 describe these cases) and does not propagate them. No error condition exists from the point of view from the P1394a PHY.

This proposal defines a means by which a P1394a PHY may refuse the command implicit in RX_SUSPEND or RX_DISABLE_NOTIFY. The mechanism is controlled and explicit; it does not rely upon any vague assertion that "... the P1394a PHY behaves *like* an IEEE Std 1394-1995 PHY."

The central issue is the manner in which PHYs detect their connection status on each port. Existing PHYs use the presence or absence or TpBias. The P1394a suspend / resume facilities enhance this by means of a separate connect detect circuit activated when the PHY does not generate TpBias. I suggest that P1394a requires a method to select between these two connect detect modes. A consequence of one of the connect detect modes is the ability to refuse RX_SUSPEND and RX_DISABLE_NOTIFY requests. The details follow and reference the affected sections of P1394a Draft 1.4.

6. PHY register map

Table 6-1 — PHY register fields for the cable environment

Field	Size	Type	Power reset value	Description
Con_mode	1	rw[a]	vendor-dependent	Connection detect mode. When zero, the presence or absence of bias determines port connection status. Otherwise, the connect detect circuit specified in clause 7.2 determines port connection status.

[a] This bit may be implemented as read-only.

7.10.3.1.2 Bus reset actions and conditions

Table 7-25 — Bus reset actions and conditions (Sheet 1 of 2)

```
boolean connection_in_progress[NPORT];    // Not referenced outside of the reset state machines

void connection_status() {          // Continuously monitor port status in all states
   timer bias_timer;                // Timer for bias filter
   boolean filter_bias[NPORT];      // TRUE when applying hysteresis to bias_detect crivuit
   int active_ports = 0, i, suspended_ports = 0;

   isolated_node = TRUE;            // Remains TRUE if no active port(s) found
   for (i = 0; i < NPORT; i++) {
      if (active[i]) {
         active_ports++;            // Necessary to deduce boundary node status
         isolated_node = FALSE;     // ALL ports must be inactive at an isolated node
      } else if (connected[i] && !disabled[i])
         suspended_ports++;         // Other part of boundary node definition
      boundary_node = (active_ports > 0 && suspended_ports > 0);
   }
   for (i = 0; i < NPORT; i++) {
      if (bias_filter[i]) {
         if (bias_detect[i] == bias[i])   // Has bias detect changed since timer started?
            bias_timer = 0;            // If so, restart the filtering timer
         else if (bias_timer >= BIAS_FILTER_TIME)
            bias_filter[i] = FALSE;    // Done filtering
            bias[i] = bias_detect[i];  // Confirm new value in PHY register bit
         }
      } else if (bias_detect[i] != bias[i]) {   // Detected bias differs from reported bias?
         bias_filter[i] = TRUE;        // Yes, start a filtering period
         bias_timer = 0;
      }
      if (connection_in_progress[i]) {
         if (con_mode) {               // Use connect detect circuitry?
            if (!connect_detect[i])
               connection_in_progress[i] = FALSE;  // Lost attempted connection
            else if (connect_timer >= CONNECT_TIMEOUT) {
               connection_in_progress[i] = FALSE;
               connected[i] = TRUE;    // Confirmed connection
            }
         } else {                      // Else use TpBias á la 1394-1995
            if (!bias[i])
               connection_in_progress[i] = FALSE;  // Lost attempted connection
            else if (connect_timer >= (isolated_node) ? 2 * CONNECT_TIMEOUT : CONNECT_TIMEOUT) {
               connection_in_progress[i] = FALSE;
               connected[i] = TRUE;    // Confirmed connection
               if (isolated_node)      // Can we arbitrate?
                  ibr = TRUE;          // No, transition to R0 for reset
               else
                  isbr = TRUE;         // Yes, arbitrate for short reset
            }
         }
      } else if (!connected[i]) {
```

Table 7-25 — Bus reset actions and conditions (Sheet 2 of 2)

```
         if ((con_mode) ? connect_detect[i] : bias[i]) { // Possible new connection?
            connect_timer = 0;      // Start connect timer
            connection_in_progress[i] = TRUE;
         }
      } else if ((con_mode) ? connect_detect_valid[i] && !connect_detect[i] : !bias[i]) {
         connected[i] = FALSE;      // Detect disconnect instantaneously
         if (!con_mode)
            if (child[i])           // Parent still connected?
               isbr = TRUE;         // Yes, arbitrate for short reset
            else
               ibr = TRUE;          // No, transition to R0 for reset
         if (int_enable[i] && !port_event) {
            port_event = TRUE;
            if (link_active && LPS)
               PH_EVENT.indication(INTERRUPT);
            else
               PH_EVENT.indication(LINK_ON);
         }
      }
   }
}
```

7.10.4 Port connection

The port connection state machines operate independently for each port, *i*, where *i* is a positive integer less than NPORT. While a port is in the active state its arbitration, data transmission, reception and repeat behaviors are specified by the state machines in clause 7.10.3. When a PHY port is in any state other than active it is permissible for it to lower its power consumption; the only functional component of a PHY that shall be active in all states is the physical connection detect circuitry.

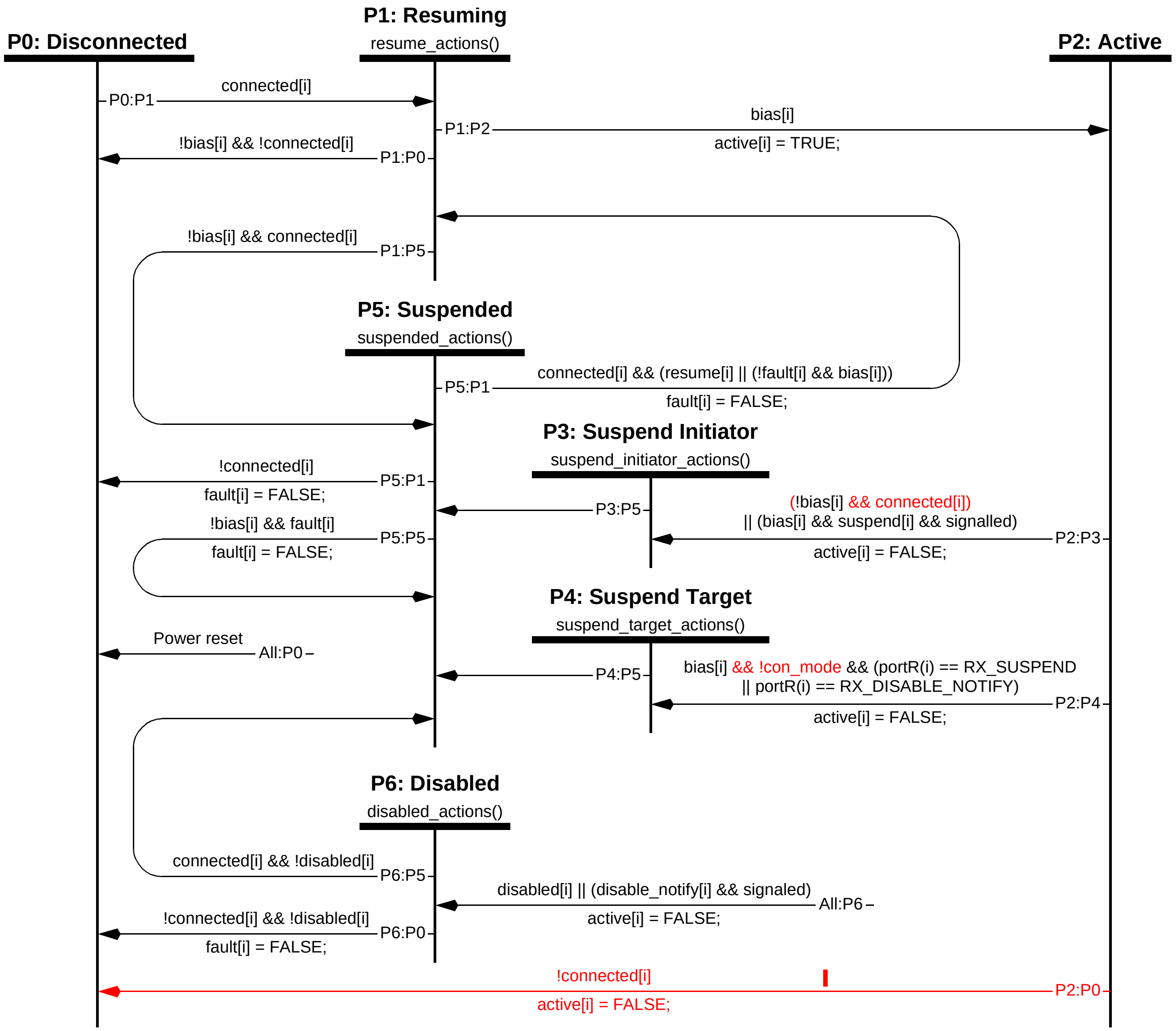

Figure 7-20 — Port connection state machine

7.10.4.1 Port connection state machine notes

Transition P2:P0. When Con_mode is zero, *connected* changes to false simultaneously with the loss of *bias* and the port transitions to the disconnected state. Note that this transition is not possible when Con_mode is one, since the loss of *bias* cannot immediately affect *connected*; transition P2:P3 is taken instead.

Transition P2:P3. Upon the loss of *bias* (if the port is still connected) or the receipt of a PHY remote command packet that sets the *suspend* variable to one, the PHY port leaves the active state to start functioning as a suspend initiator. A loss of *bias* is usually the result of a physical disconnection or the loss of power to the connected peer PHY port. If the transition is the result of a remote command packet, the PHY transmits a remote confirmation packet with the *ok* bit set to one. In the meantime, the suspend initiator has signaled TX_SUSPEND to its connected peer PHY.

Transition P2:P4. If Con_mode is one, an active port that observes an RX_DISABLE_NOTIFY or RX_SUSPEND signal becomes a suspend target and leaves the active state.