Suspend/Resume Variables
Draft 97-067R1 - Oct. 11, 1997

The unrevised text is directly from P1394a, draft 1.0.

7.5 Node variables

Each node's PHY has a set of variables that are referenced in the C code and state machines in clause 7.7. The values of these variables may be affected by writes to PHY registers, the transmission or reception of PHY configuration packets or by arbitration state actions--including bus reset. The definitions in table 7-8 entirely replace clause 4.3.8 of IEEE Std 1394-1995, "Node variables."

Table 7-8 — Node variables

Variable name	Power reset value	Comment
accelerating	TRUE	Set TRUE or FALSE by accelerate or decelerate requests issued by the link *via* Lreq (see clause 5.2) and used by the arbitration state machines. See also enab_accel below.
arb_enable	—	TRUE when the PHY may arbitrate as soon as the next subaction gap is observed.
cable_power_active	—	TRUE is cable power is within normal operating range (see clause 7.1).
enab_accel	FALSE	Globally enables or disables all PHY accelerations specified by clause 7.7. This vari-able is visible as the PHY register bit Enab_accel.
force_root	FALSE	When TRUE, this modifies the PHY's tree identification behavior and increases the likelihood that the node becomes root (see clause 4.4.2.2 of IEEE Std 1394-1995). If only one node on a bus has force_root set TRUE, that node is guaranteed to become the root.
gap_count	63	This value determines the length of arbitration reset and subaction gaps and may be used to optimize bus performance. All nodes on the bus should have the same gap_count value else unpredictable arbitration behavior may occur.
initiated_reset	TRUE	TRUE if this node initiated the bus reset in progress. Cleared to FALSE upon comple-tion of the self-identify process.
link_active	TRUE	TRUE if the node's link is present and enabled.
more_packets	—	Flag which indicates whether or not additional self-ID packets are to be sent.
parent_port	—	The port number that is connected to the parent node; this variable is meaningless if the node is root.
physical_ID	—	The node's 6-bit physical ID established by the self-identify process.
receive_port	—	The port number that is receiving encoded data (determined by the arbitration states).
root	—	TRUE if the node is the root, as determined by tree-ID.
core_power	-	When FALSE the core's clock is not running and the circuits are consuming minimal power. When FALSE the core will only respond to sleep being FALSE.
core_functional	-	When TRUE the core's clock is running and all circuits, except ports, are funtional.
resume_all	FALSE	When TRUE a suspended or resume failed port has detected a resume event (its peer port has driven TpBias high) and the node is to resume all suspended ports.

7.6 Port variables

In addition to the variables described in the preceding clause, each node’s PHY has a set of variables replicated for each port. The definitions in table 7-9 entirely replace clause 4.3.9 of IEEE Std 1394-1995, “Port variables.”

Table 7-9 — Port variables

Variable name	Power reset value	Comment
child	-	TRUE if this port is connected to a child node.
connected	FALSE	TRUE if there is a peer PHY connected to this port.
child_ID_complete	-	TRUE when the child node connected to this port has finished its self-ID.
max_peer_speed	-	Maximum speed capability of the peer PHY connected to this port.
port_status	-	TRUE if TP bias is present. This is not filtered by any hysteresis circuitry.
speed_OK	-	The connected port can accept a packet at the requested speed.
suspend	-	TRUE if TP bias is not present.
suspend_initiator	FALSE	If TRUE the port is to initiate a suspend.
initiate_suspend	FALSE	TRUE if core is to transmit TX_SUSPEND on this port and BUS_RESET on all other active ports.
suspend_target	FALSE	TRUE if this node has detected RX_SUSPEND on this port. This port is to attempt a suspend handshake with its peer port.
port_power	TRUE	FALSE when port clock is not running, port is consuming minimal power and the port will only respond to a change in con_status or port_status. When FALSE the status and control registers can be read.
port_functional	-	TRUE when port clock is running and all circuits are functional.
con_status	-	TRUE when connected and this port is not driving TP bias. This is the output of the schmidt trigger buffer located at the top of Fig. O-4, page 9, in the Suspend/Resume draft 0.08. This variable is also qualified by con_disable.
initiate_resume	FALSE	TRUE on a write to the suspend bit of the Control Clear register.
resume_target	FALSE	TRUE when port_status goes TRUE while the port is in a Suspended or Resume Failed state.
initiate_disable	FALSE	TRUE if core is to transmit TX_DISABLE on this port and TX_DATA_PREFIX followed by BUS_RESET on all other active ports.
disable	FALSE	TRUE when this port is disabled.
fault	FALSE	TRUE when this port attempted to suspend or resume and the process failed.
active	FALSE	TRUE when this port is connected, not resuming, not disabled, not suspended, and not faulted.
active_en	FALSE	TRUE when the bus is reset and the port is resuming. This allows the port to enter the active state.
active_domain	FALSE	TRUE when a resuming port receives BUS_RESET within 5 * RESET_DETECT, indicating there is a boundary node connected to this port.

Table 7-10 — Cable PHY code definitions

```
const int FIFO_DEPTH = ?;             // IMPLEMENTATION-DEPENDENT!
enum PHY_state {R0, R1,               // Tracks the PHY state (names per state diagrams)
               S0, S1, S2, S3, S4,
               A0, A1, A2, RX, TX};
enum speedCode {S100, S200, S400};    // Speed codes
enum tpSig {L, H, Z};                 // Differential signal on twisted pair
struct portData {tpSig TpA; tpSig TpB;};     // Port data structure
enum phyData(portData signals);// Encoded types DATA_ZERO, DATA_ONE, DATA_PREFIX or DATA_END
boolean ack;                          // Set if last packet observed was exactly 8 bits
boolean arb_enable;                   // Set if a node may arbitrate upon detection of a subaction gap
timer arb_timer();                    // Timer for arbitration state machines
boolean bus_initialize_active;        // Set while the PHY is initializing the bus
int child_count;                      // Number of child ports
int contend_time;                     // Amount of time to wait during root contention
boolean DS_clock;                     // FALSE unless encoded DS clock available on the receive port
                                      // (data or strobe transition observed within the last 20 ns)
boolean end_of_reception;             // Set when reception of packet is complete
boolean force_root;                   // Set to delay start of tree-ID process for this node
dataBit fifo[FIFO_DEPTH];             // Data resynch buffer
unsigned fifo_rd_ptr, fifo_wr_ptr;    // Data resynch buffer pointers
boolean gap_count_reset_disable;// If set, a bus reset will not force the gap_count to the maximum
boolean ibr;                          // Set when a long bus reset is needed
boolean isbr;                         // Set when an arbitrated (short) bus reset should be attempted
boolean isolated_node;                // Set if no ports connected
boolean own_request;                  // Latch the value of arb_OK() at the time it is evaluated
boolean ping_response;                // Set if self-ID packet(s) needed in response to a ping
boolean phy_access_response;          // Set on port register read or write
portData portR(int port_number);      // Return current rxData signal from indicated port
speedCode portRspeed(int port_number);       // Return current speed from indicated port
void portT(int port_number, portData txData);        // Transmit txData on indicated port
void portTspeed (int port_number, speedCode speed);         // Set transmit speed on indicated port
boolean random_bool();                // Returns a random TRUE or FALSE value
int reset_time;                       // Duration to assert bus reset signal
boolean root_test;                    // Flag that is randomly set during root contention
int rx_dribble_bits;                  // Keep track of dribble bits in FIFO
speedCode rx_speed, tx_speed;         // Current packet speeds
boolean waiting_for_data_start;       // First data bit not yet received
enum resume_status (BOUNDARY_INITIATOR, // Status of node while resuming
                BOUNDARY_TARGET,
                RESUME_INITIATOR,
                RESUME_TARGET, NULL);
boolean resume_event;                 // Flag indicating that the node is to resume its suspended ports
timer resume_timer;                   // Timer for resuming events
boolean one_domain;                   // Set if at least one port has its active_domain flag set.
boolean resume_packet;                // Set if a node resume packet was received.
int addr;                             // register number
boolean rsv_port;                     // Set if phy register access is to an unused port or reserved register
timer connect_timer();                // Timer for port connect, suspend and resume - One timer for all ports
```