Proposal for IEEE 1394 Suspend/Resume Capability

Intel Corporation

BACKGROUND

The IEEE 1394-1995 serial bus provides some node power-management capability, in that the node's link can be turned on and off with the Link-On and Link-Off packets, respectively. However, this bus standard requires that a node's physical layer (PHY) must remain powered and active if the node is to remain capable of relaying packets which are addressed to other nodes on the cable. (Leaf-nodes on a bus are an exception to this requirement, since they have no nodes located further "downstream" on the bus. Such nodes are free to turn off their PHY, as well as their link-layer, as appropriate).

With the preceding architecture, node power consumption can be reduced to levels dictated by PHY power consumption. With today's PHY implementations, this is on the order of 1W or less per node. Since contemporary link-layer implementations consume a similar amount of power, the Link-On/Link-Off controls can be used to roughly halve the node power consumption (relative to its full operating state).

Unfortunately, this degree of power reduction is inadequate to support the needs of mobile systems; indeed, it may be inadequate for "green" desktop PCs, which must constrain their "sleeping" power-consumption to, say, 5W total (e.g. to satisfy Energy Star or Blue Angel certification requirements).

This proposal focuses on the needs of mobile 1394 implementations, since these constitute the most stringent 1394 power-management environment. While in a low-power "suspended" state, a notebook computer may consume a total of 100 mW or less. This is well below the lowest power-consumption level that is attainable with the existing 1394 specifications (due primarily to the need to keep a node's PHY powered). This proposal suggests a mechanism which should reduce 1394 subsystem power to a small fraction of what it is today.

DEFINITIONS

"Suspend" and "resume" are ill-defined terms in today's computer industry. We adopt the following usages:

- "*Suspend*" refers to a state in which a node has saved its internal state (to battery-backed or low-refresh-rate RAM, to disk or to some other non-volatile memory), and has then stopped normal operation. In this state, the node is incapable of doing application-level processing, and even most system software is not executed. Instead, the system retains just enough intelligence to recognize the occurrence of one or more types of prescribed "*wake-up*" events, whose occurrence can be used to trigger resumption of the normal operating state. Triggering of wake-up can be caused by reception of a ring-indicate signal by a telephony device; reception of a wake-up packet by a LAN adapter; a change in the charge-state of a device's battery; etc.
- The process of returning a suspended system to its normal operating state is referred to as "*resume*" processing. This entails restoration of the system's state from suspend storage, followed by resumption of execution of system software and application software where these were interrupted by entry into the suspend state.

1394 ACPI Definitions

Table 1 describes the ACPI device power states which can be supported by a 1394 devices. In this table, resume latency (i.e. time taken to return to the D0 normal operating state) increases toward the bottom of the table. Also, the amount of power consumption decreases toward the bottom of the table.

Device Power-State	Descriptor	PHY Power	Link Power	Device Context	Wake-Up Available
D0	Run	ON	ON	Saved	No
D1	Stand-By	ON	ON-LP	Saved	No
D2	Suspend	ON-LP	ON-LP	Lost	Yes
D3	Off	OFF	OFF	Lost	No

Table 1: 1394 ACPI Device Power States

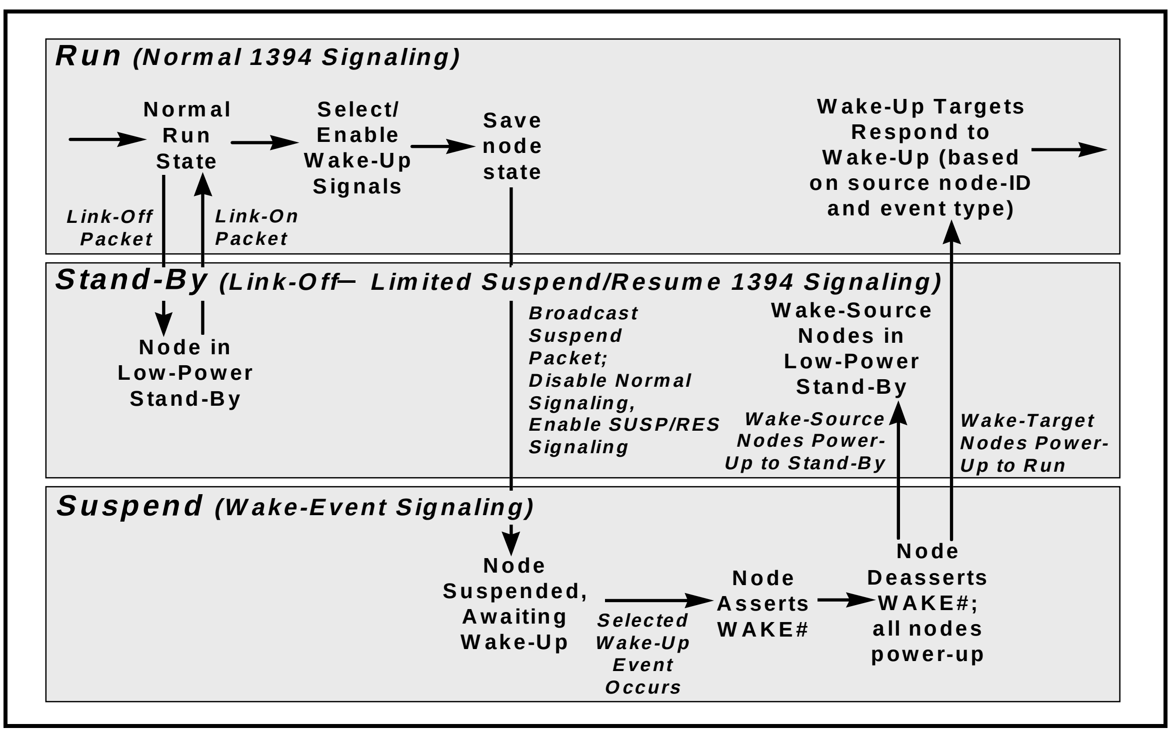
For the link-layer, the "ON-LP" state corresponds to a link-layer which has received and responded to a Link-Off packet; In that state, the link-layer conserves power while remaining able to process only a few packet types. For the PHY layer, the "ON-LP" state corresponds to the proposed new PHY signaling state, in which the PHY cannot accept or relay packets. Instead, a PHY in the ON-LP state is only capable of signaling or propagating a wake-up event indication.

In this proposal, the terms in the "Descriptor" column of the table are used to convey the operating state of devices on the bus. The ACPI device power-states could be used instead, though they are less suggestive.

SUSPEND/RESUME OVERVIEW

This proposal suggests that 1394 suspend and resume be implemented as multi-phase operations. Figure 1 illustrates the overall flow of these processes. This diagram shows that suspend/resume processing involves operation of the bus in three different operating states: normal signaling; a reduced-capability signaling state corresponding to Link-Off; and a wake-up signaling state. The first two of these levels utilize the normal 1394 packet-based communications. Wake-up signaling is based on a simpler (and less power-intensive) signal transition-detection capability which redefines use of the 1394 TPA line.

Figure 1: Suspend/Resume Processing Overview



Run State

In the "Run" state, a node's PHY and link are both fully-powered, and are capable of handling the full spectrum of 1394 packet types. Both isochronous and asynchronous processing are supported.

Stand-By State

In the "Stand-By" state, node power consumption is reduced relative to that of the Operating state. In Stand-By, a node's PHY is fully powered, but only a portion of the link-layer is powered. The state of all registers is preserved, avoiding the need for time-consuming state-restoration. Nodes in Stand-By can continue to perform essential housekeeping tasks, such as cycle-time maintenance.

From the Run state, a node enters Stand-By through reception of a Link-Off packet. Conversely, a node which is in Stand-By may be restored to normal operation by sending it a Link-On packet. In addition, a node may enter the Stand-By state from the Suspended state, through the process described below.

While in the Stand_By state, a node's power-control registers remain accessible to other nodes which may be in a higher power-state (Run). Nodes in the Run state can read from Stand-By node power registers to ascertain the nature of a wake-up event that may have left the node in Stand-By.

Suspend State

The Suspend power-state is a proposed addition to the current 1394-1995 specifications. Adding this state requires that usage of the TPA signal line be redefined when the bus is in the Suspended state.

While in Suspend, a node's link-layer is not powered, and only a small portion of its PHY is powered. Specifically, only the PHY's ability to source cable power (at a greatly-reduced level) and its ability to propagate a wake-up event from the node's device and onto the bus is preserved. Note that wake-up events do not pass through the link-layer at all, allowing the link to be turned off. This is consistent with the role of the link-layer as a provider of packetizing services, and with the fact that the wake-up mechanism is not based on packets.

WAKE-UP

The wake-up model advocated here allows for wake-up events to be generated by any of a set of suspended nodes (referred to as "wake-up source" nodes). In addition, each such node may be capable of generating a wake-up event in response to any of a set of distinct device events. For example, a 1394-based LAN adapter may generate a wake-up in response to receiving a wake-up packet from the network, or it may generate a wake-up due to low charge in a configuration back-up battery.

In order to support this model, a 1394 device must include two "event" registers: a register to capture (latch) wake-up events, and a register to allow masking of unwanted types of wake-up events. Wake-up event types are selected by writing to the mask register. Wake-up events cannot be stacked; that is, only one wake-up event of a given type can be active at any particular time.

As in handling of PCI interrupts, a wake-up recipient can clear wake-up events by writing to the device circuitry which generates the event. Both the repertoire of wake-up event types and the manner of clearing wake-up events vary by device class (e.g. LAN adapter, telephony device, etc.). These characteristics should constitute part of the class definition for each device type.

The event registers are not readable over the bus during suspend (since normal signaling is not then available); however, these must be readable in both the Stand-by and Run states. This allows a wake-up event recipient to determine the nature of a wake-up event.

A node's event-mask register must be writable in the Stand-by and Run states. The node's event (capture) register is automatically cleared on entry into suspend. This means that all nodes' asserted wake-up event-register bits must be examined and acted upon (or ignored) before the bus re-enters the suspended state.

Wake-Up Architecture

The Wake-Up signal is global (i.e. bus-wide), and is shared by all nodes which are capable of producing wake-up events. The likeliest implementation of this signal is as an active-low "dot-OR" arrangement of the various Wake-Up sources. Similarly, TPA is the most attractive existing signal line for propagating Wake-Up, since TPA already has the role of providing a common-mode bias voltage to the bus. A large-swing signal is appropriate for Wake-Up signaling.

Figure 2 shows the proposed arrangement of the Wake-Up mechanism. As shown in the diagram, while a bus is in Suspend, power is only supplied to the wake-up logic on both Wake-Up event source-nodes and Wake-Up target nodes. This is likely to be a small amount of power, which can be conveniently supplied from the power-pair in the 1394 cable. However, it is also possible for nodes to (self-)power their own wake-up logic.

A Wake-Up source-node's Wake-Up generation logic must be capable of filtering wake-up events in a programmable manner. That is, a Wake-Up target must me able to specify what types of wake-up events are of interest to it. The nature of the wake-up event must also be accessible to the wake-up source-node's link-layer, so that a wake-up recipient can poll for this information (in packetized form). This is necessary because the wake-up mechanism is global, and does not identify the source of a wake-up indication or the nature of the wake-up event.

A node's wake-up event registers must be kept powered while the node is suspended, as must whatever circuitry generates the wake-up events. The node's WAKE# buffer must also be kept powered. All other portions of the node may be powered down in suspend.

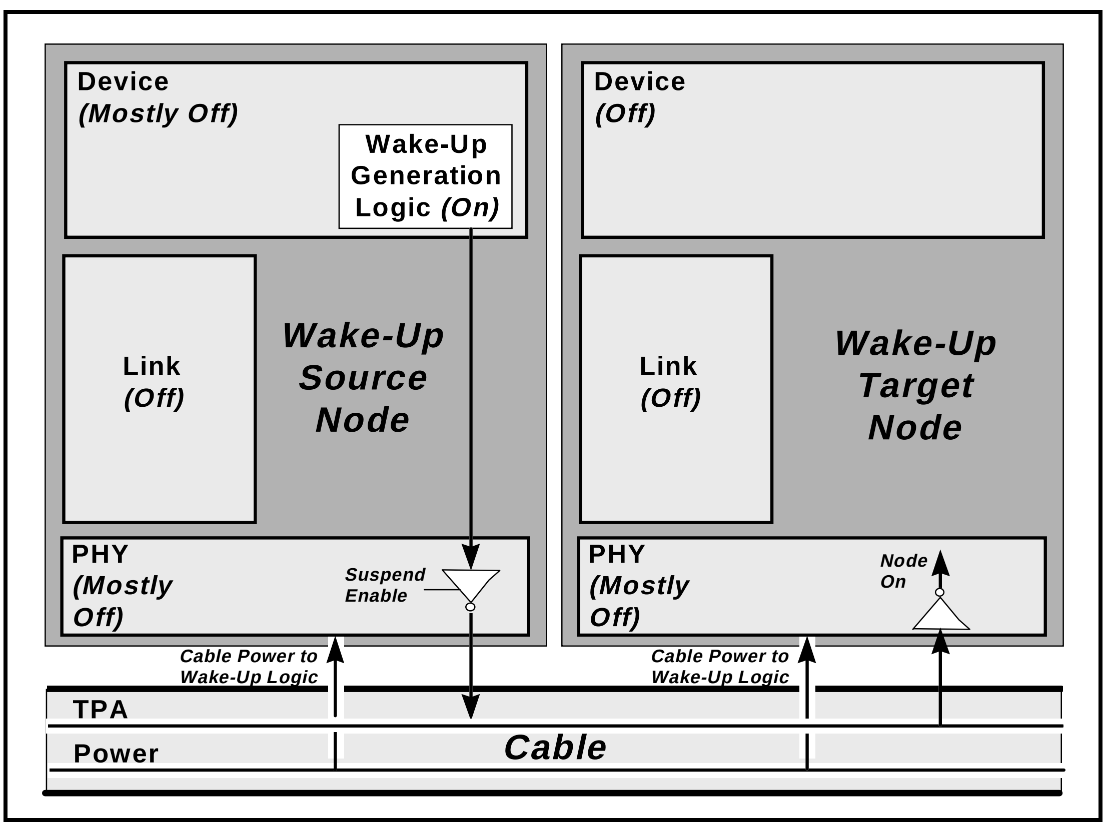


Figure 2: Wake-Up Mechanism

Suspend Process

Because the suspend state applies to a 1394 bus as a whole, the nodes on the bus must be prepared for suspension. This involves two different activities:
1) Selection and enabling of wake-up events in nodes which can produce such events; and
2) Saving of device context.

These activities are performed at appropriate points in the suspend process; they are reversed as part of the resume process. The programming of wake-up events has already been described in the section on Wake-Up. The state-save process is briefly described below.

State-Save

Per the proposed 1394 ACPI definitions, device context is lost when a device is suspended. This means that the device state (configuration and perhaps application state) must be saved in non-volatile storage prior to entry into suspend. Then, when that device is prepared for resumption of normal activity, its state can be restored by copying it back from non-volatile storage into its pre-suspend locations.

State-saving and restoration is probably best done by a bus's current suspend-manager node. That node knows the power-state of each node on the bus at a given moment, and it also controls the power-state transitions of each node. Saving and restoring node state is a logical extension of these control responsibilities.

The specific way in which device state is saved depends upon which ACPI S-state (system "sleep" state) is to be entered. Device state is preserved in states S1 and S2, and state is lost in S3, S4 and S5. The contents of system memory are preserved in S1-S3, but are lost in S4 and S5. This means that 1394 device state is saved in system memory ("save to RAM", or "STR") when the system enters S3; but device state is saved to disk ("STD") if the system is to enter state S4.

Entry into Suspend

To begin the suspend process, the bus power-state manager node broadcasts a "Suspend Request" packet. This causes all nodes to check whether they are in a state which can tolerate suspension. (This may not be the case if, for example, a node is engaged in a time-critical data transfer or control operation). If a node is able to enter suspend at present, it returns a "Suspend Gnt" (Grant) packet to the bus power-state manager; otherwise, it returns a "Suspend Deny" packet. In either case, the node then enters a wait state, pending further instructions from the manager.

The bus power-state manager processes the suspend-acknowledges as they arrive from the other nodes. This occurs during a suspend timeout period which is controlled by the manager. As Suspend Gnts arrive, they are counted. If a Suspend Deny packet is received at any time during the timeout period, the manager immediately records a "Suspend Failed", and then broadcasts a "Suspend Failed" packet. Reception of that packet enables each non-manager node to exit its wait-loop and resume normal operations. The manager can retry a suspend request later.

If the timeout elapses without reception of a Suspend Deny, then the manager compares its Suspend Gnt tally against (number of nodes - 1); that is, against the number of non-manager nodes on the bus. If these numbers are equal, then all nodes are ready for suspend, and the suspend process can proceed. At this point, the bus power-state manager prepares the bus for suspension. It programs the wake-up generation circuitry of all wake-up source nodes. The manager then saves node state-information in non-volatile memory, for later use in restoring the bus to normal operations.

To actually enter suspend, the manager broadcasts a "Suspend" packet. This causes each receiving node to configure its bus interface for wake-up event propagation (i.e. WAKE# buffer enabled to drive TPA as the shared wake-event line). Each node then powers down into the suspend state. The manager node likewise configures its interface and powers down. From that point until a wake-up event occurs, no further normal 1394 signaling occurs on the bus. The WAKE# line is kept at its inactive high state by a high-value pull-up “keeper” resistor.

If the comparison of number of Suspend Gnts fails, then the manager proceeds as in the case of receiving a Suspend Deny. The manager broadcasts a Suspend Failed packet, and all nodes resume normal operations. Note that the timeout period is necessary to deal with the case of a malfunctioning or slow-responding node.

This interlocked suspend-entry sequence is diagrammed in Figure 3.

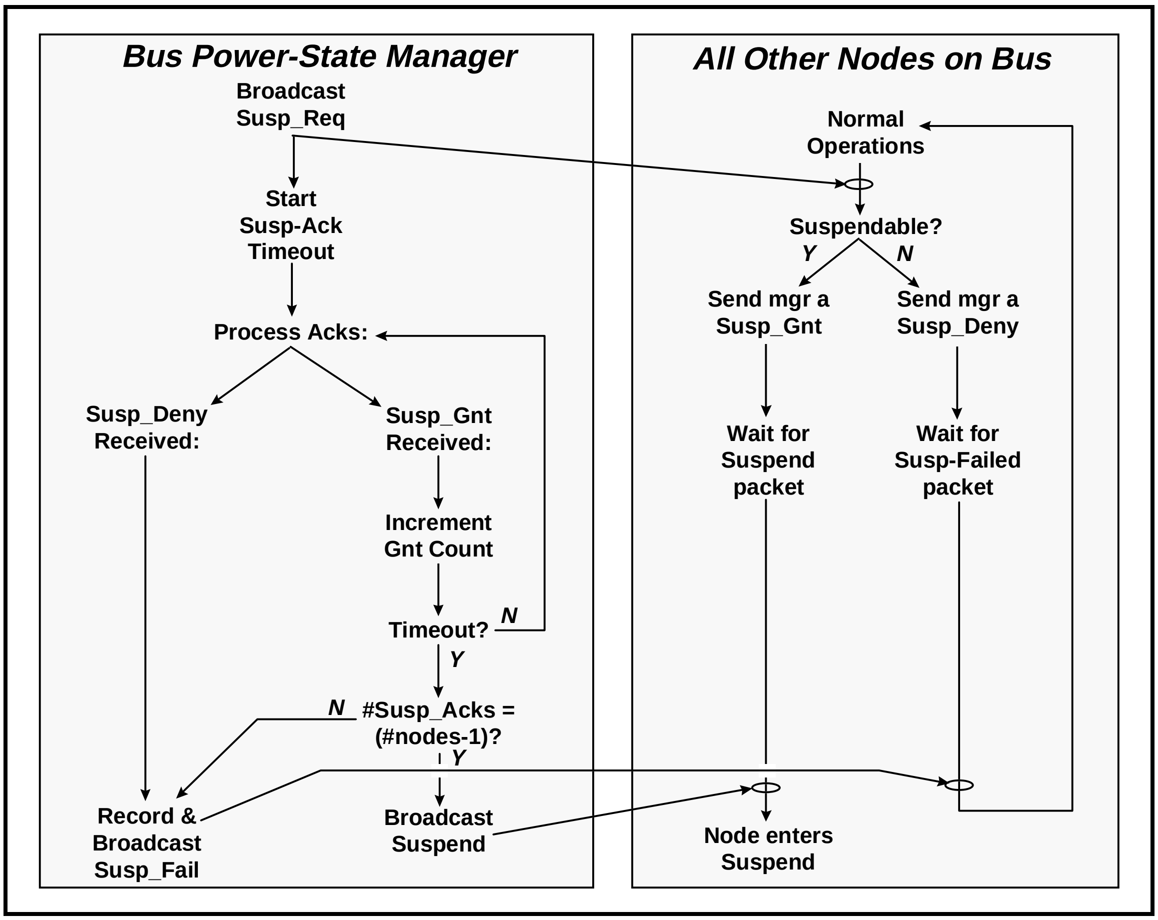


Figure 3: Suspend Entry Process

RESUME PROCESS (EXIT FROM SUSPEND)

When any suspended node receives an unmasked local wake-event, the node drives WAKE# active (low) on the bus. The wake-up initiator then deasserts WAKE# (restores it to its inactive high level). The high-to-low transition on WAKE# is seen by all nodes on the bus, which respond by powering up and reconfiguring their bus interface for normal 1394 signaling.

The bus power-state manager resumes in its full-power Run state. All other nodes resume in their Stand-by state, to minimize the size of the current-draw step experienced by the power bus on resume. The manager then polls the other nodes’ event registers to determine the source and nature of the wake-up event. The manager restores the pre-suspend state of the event-source node, and of the node(s) which had requested notification of that event source and type. The manager then restores those same nodes to their

Run state, so that they can interact to respond to the wake-up event. Note that other nodes are kept in their Stand-by state, to avoid powering up devices which are not relevant to the observed wake-up event.

Once the source of an active wake-up has been identified and serviced, the bus power-state manager can set new power-states for all nodes. The manager can potentially put the bus back into the suspend state, or it can bring nodes back to their full-power Run state. The manager can also elect to leave specific nodes in Stand-by. The bus power-state policy is fully software-programmable.

If a node is to be returned to either the Stand-By or Run state, the node's state must be restored. This is done by the bus power-state manager, which saved the node's state on entry into Suspend.

Node Isolation from Bus

In certain circumstances, it is desirable to put a device into a sleeping state, while allowing normal signaling to continue on the 1394 bus. This case is not addressed through the proposed suspend/resume mechanism, since that requires cessation of normal bus signaling. For completeness, though, we consider how this situation is handled.

A 1394 device may include wake-up signal sources which do not require signaling via the 1394 bus. For example, a 1394-equipped notebook computer can receive wake-up events from its on-board real-time clock, or from a lid-opening sense switch. When such a 1394 device is put into a low-power state, it should effectively remove itself from the 1394 bus. That is, it should allow bus signaling and cable power to pass unimpeded through the (dormant) node.

When such a node receives a wake-up event, it should re-attach itself to the 1394 bus. This is done as a "virtual device attach", which triggers a bus reset and device re-enumeration. Normal 1394 signaling can resume after completion of this bus re-initialization.

A mechanism for performing this node isolation and re-connection is described in a companion Mobile Cable-Power Distribution proposal from Intel.

Node Power-Control Architecture

In order to provide the needed degree of control over PHY and link-layer power, the architecture shown in Figure 4 is recommended. In this diagram, existing node power-control capabilities appear in ***bold italic*** font, while proposed new power-control capabilities appear in *normal-weight italic* font.

Power to the node is provided on two separate power rails. One of these, the "suspend power" bus, is always energized, so that it can support circuits which must always be functional. The other "bus operating" rail supplies the bulk of the power to both the node PHY and link layers; this bus is energized whenever the node is not in its low-power suspended state.

The PHY-layer power0control block gates power to the link-layer's packet transmit and receive blocks. In the existing specifications, the Link-On packet is used to enable power to the link layer during node configuration. A proposed new Link-Off packet allows most of the link-layer's packet-handling logic to be turned off when the node is to be put into a "Stand-by" power state.

The proposed new PHY-layer suspend capability requires that the PHY power controls be modified to allow control over the power to most of the PHY. The "Suspend" packet turns off power to most of the PHY, and enables the node's WAKE# bus-interface buffer. The node's power-control circuitry must also include the capability to detect a rising edge on the (shared) WAKE# line, so that power to the PHY can be restored when another node signals a wake-up event on WAKE#.

The PHY's power-control block is powered from the always-on suspend-power rail, so that it can always stably control node power. The node's wake-up path and cycle control block are also powered from the suspend rail, so that these functions can operate at all times.

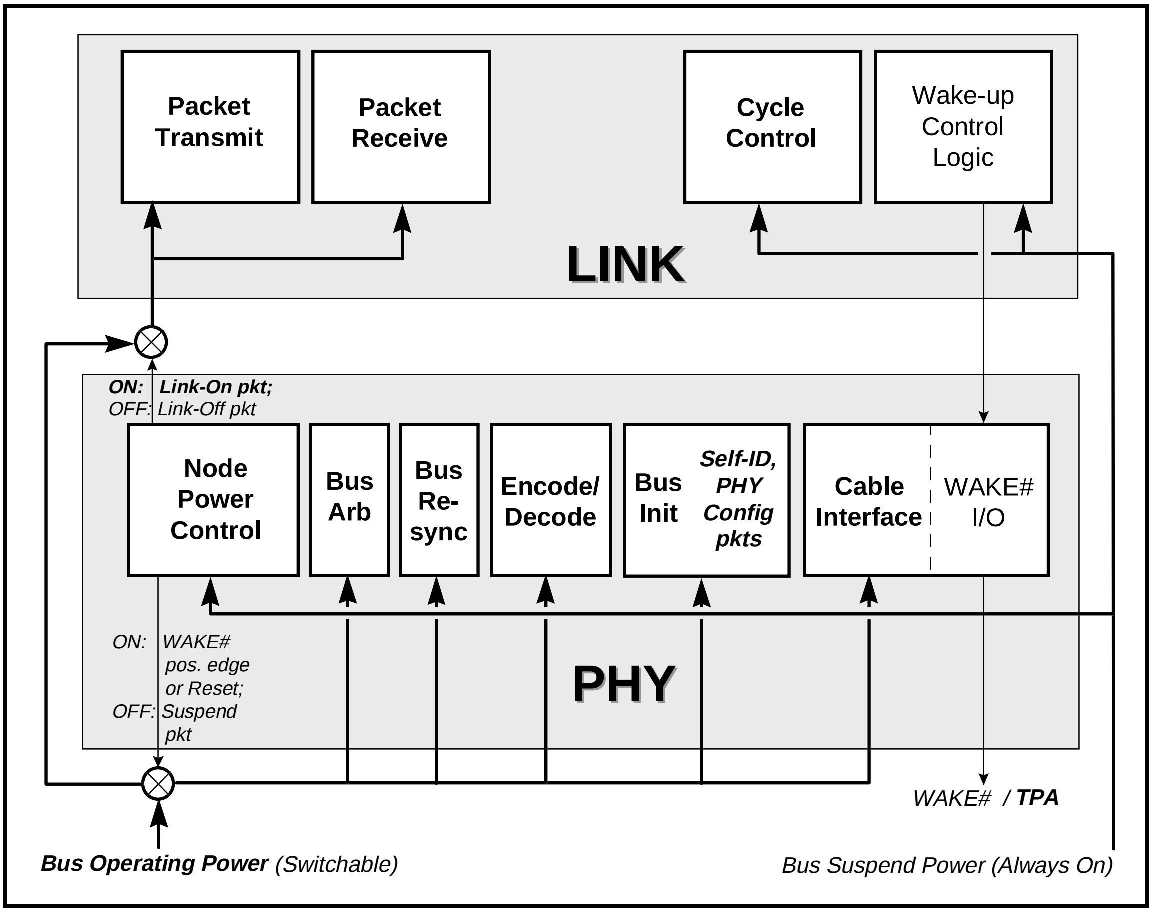


Figure 4: Node Power-Control Architecture

BACKWARD COMPATIBILITY

The proposed suspend/resume mechanism builds on the existing 1394a mechanisms. Suspend-capable nodes can coexist with "legacy" nodes which lack this capability.

Nodes which are not suspend-capable do not respond to Suspend Req, Suspend Failed or Suspend packets. They are also incapable of replying to these with Suspend Gnt or Suspend Deny packets. For such nodes, the suspend-acknowledge timeout mechanism will result in issuance of a Suspend Failed on any attempt to put the bus into a suspended state.

Since suspend/resume relies on use of the 1394 cable signals in a new operating mode, suspend is only effective for systems which contain only suspend-capable nodes. That is, all nodes on the cable must be able to enter the suspend state, so that the cable signaling mode can change to support suspend/resume. If a bus contains a mix of suspend-capable and non-suspendable nodes, the cable signals must retain their normal packet-propagation capabilities. In that case, nodes can be put into their reduced-power Stand-By state, but they cannot be put into a Suspended state.

Implementation of the proposed suspend logic implies partitioning of a node's PHY, as well as of its attached device. This is necessary so that only a small portion of the node can be kept powered during suspend. It may be desirable to provide a suspend power rail which is separate from the normal operating power rail, though this is an implementation decision.

Expected Impact on 1394a

The proposed suspend/resume mechanism requires that several new packet types be defined (Suspend Req, Suspend Gnt, Suspend Deny, Suspend and Suspend Failed). These can most likely be incorporated into one new "Suspend control" packet, which would contain a bit-field which could be encoded to convey the various stages and test results of the suspend process. The details of this have yet to be worked out.

Similarly, this proposal describes the sequence of operations which support the suspend and resume mechanisms. However, the detailed contents, location and organization of needed supporting registers have not yet been established.

In addition, the suspend/resume capability redefines the usage of the bus signal lines while the bus is in the suspended state. This is, however, an additional capability which extends the existing 1394a specifications. Use of this signaling mode should not affect the operation of the bus in its normal packet-based signaling mode.

Definition of the proposed new signaling mode is necessitated by the need to reduce suspend-state power to well below what can be achieved using a packet-based mechanism (since that requires that the PHY and substantial portions of the link-layer be kept powered).