

From: Gerry Houlder, Seagate Technology <gerry.houlder@seagate.com>
Subj: SAS: Add low power transceiver options
Date: Dec. 12, 2007

Overview

This proposal adds lower power transceiver modes to SAS. The intent is to add low power modes that are similar to or compatible with the low power modes available in SATA. This should allow an expander design that can work with either SATA or SAS devices in low power mode.

This proposal focuses on target to initiator and target to expander links. These links should provide the greatest opportunity for power savings because they are likely to have more idle time and will not have performance impact on access to other targets. For these cases, only the target would need to initiate low power modes and could reasonably predict idle times based on commands in its queue and timers that allow for idle and standby modes.

Expander to expander links and initiator to expander links would affect performance to large numbers of targets if they were in a low power mode. Rules for these links could be added if the group thinks the performance impact was acceptable.

For compatibility with SATA, there needs to be 4 primitives. If SAS chooses the same primitive encoding that SATA has, expander design might be easier. The primitives are:

- (a) PMREQ_P to request partial power mode (transceiver must recover within 10 us);
- (b) PMREQ_S to request slumber power mode (transceiver must recover within 10 Msec);
- (c) PMACK to accept change into the requested power mode;
- (d) PMNAK to reject the power mode request.

When a power mode request is accepted, both ends of the link stop transmitting. Either end may revive the link by sending COMWAKE. Both ends are required to remember whether the link is SSP or STP protocol and previously negotiated settings (speed, SSC, multiplexing, etc.). The intent is that both ends can restart without needing to change receiver settings or redoing training; having to repeat training may increase the recovery time so much that partial power mode may not be feasible.

When an initiator requests an expander to open a link that is in partial power mode, recovery time should be fast enough so any delay can be covered by returning AIP(WAITING ON PARTIAL) until the link is ready. When an initiator requests to open a link that is in slumber power mode, the expander will have to return OPEN_REJECT(PATHWAY BLOCKED) while the link is revived; the initiator should repeat the request later and will be successful when the link is ready. I think it is important to NOT pick new primitives for this to maintain compatibility with older initiators that wouldn't recognize the new primitive.

The partial mode will have fast enough recovery so that it can be used by target-expander link without explicit approval by an initiator. The slumber mode will cause enough delay so that an initiator should explicitly allow/ disallow this based on its performance requirements. There are two possibilities for this:

- (a) Use SNW-3 to negotiate capability/ permission similar to speed negotiation. Expander would have to base its setting on something from an initiator. Could it be SNW-3 setting from each initiator? Could it be a setting in an SMP function?
- (b) Use a SAS specific mode page. The initiator can grant permission directly to a target without needing expander participation, but using application layer to control link layer behavior has been frowned upon in the past.

- (c) Only invoke slumber when idle, standby, or sleep is either commanded by initiator or initiated by timer expiration. Recovery from slumber could also command the drive to change to active state (from idle or standby).

There may be a desire to add information to SMP functions to indicate links that are in a low power mode, especially slumber mode. The DISCOVER function and the DISCOVER LIST function seem like obvious candidates to me. Are there others?

The recovery procedure when partial or slumber mode is active will have to describe hot plug recovery case where an expander to drive link is in low power mode, the drive is removed, and the expander tries to revive a drive that is no longer there or a newly inserted drive.

Questions have also been raised about whether slumber mode actually saves a significant amount of power compared to partial mode. If the additional savings are not significant enough then perhaps this mode is not worth defining? Of course, if it is used in conjunction with idle or other longer recovery low power modes then perhaps it is still useful. This will be a discussion point. Anyone willing to provide comparative data on this is invited to do so.

When we have agreement on the basic concepts, wording changes for SAS will be created. I expect that this proposal will be for a document beyond SAS-2.