Maxtor Corporation
500 McCarthy Boulevard
Milpitas, CA 95035 USA

To: T10 SAS Protocol Working Group
Contact: Mark Evans
Phone: 408-894-5310
Email: mark_evans@maxtor.com
Date: 02 March 2005

Subject: SAS-1.1, Responding to an OPEN address frame while in the BreakWait state

1 Related documents

05-040r0 SAS-1.1 Break_Wait handling (Tim Hoglund, LSI Logic) (incorporated in sas1r08)
sas1r08 - Serial Attached SCSI 1.1 revision 8
05-086r0 SAS-1.1 Link layer timeout race conditions

2 Introduction

There has been much discussion about various race conditions that may occur when a SAS device sends a BREAK primitive (see 05-040r0 and 05-086r0), and different solutions have been proposed each of which requires significant changes to existing hardware. This proposal focusses on a simple solution to the actual issue with the minimum impact on hardware. It is important to note the chance of any of the cases is very small making it even more important to keep the solution as simple as possible.

The salient point in the discussion is that we want to prevent a SAS device that sends an OPEN address frame from ever getting to the BreakWait state and transmitting a BREAK. This could start one of the loops described in 05-086. (It’s really a “don’t care” if a SAS device fails to discern an OPEN_REJECT primitive or a CLOSE primitive while in the BreakWait state, because the connection is being shut down or aborted anyhow.)

The way to prevent a SAS device that sends an OPEN address frame from ever getting to the BreakWait state and transmitting a BREAK is to have a SAS device in the BreakWait state discern the OPEN address frame and respond with an OPEN_REJECT (RETRY) primitive. Though the path up and back from the port layer is a little convoluted, this should result, after several “instantaneous” state transitions, in the SAS device that sent the OPEN address frame sending another OPEN address frame from the ArbSel state (where it could possibly see the original BREAK). Eventually the Break Timeout timer in the SAS device in the BreakWait state will time out, and the SAS device will go to the Idle state without the other SAS device ever transmitting a BREAK primitive. When the SAS device completes the transition from the BreakWait state to the Idle state, it can process the OPEN address frame.

The solution in this proposal does not violate the rule that a phy may send a BREAK primitive to abandon a pending connection request at any time. This rule is still in effect. The worst case would be if a SAS device sent an OPEN_REJECT primitive or a CLOSE primitive, followed by an OPEN address frame, followed by a BREAK. Either the other SAS device was in BreakWait, saw this as a response, and transitioned to the Idle state, or the other SAS device had timed out in the BreakWait state and had transitioned to the Idle state. This is no different than it is today. The only thing that would cause a problem is if the other SAS device sent an OPEN address frame while the first SAS device was still in the BreakWait state. The proposed solution that follows resolves this possible issue.

As a sidebar, there is no guarantee that a SAS device that sends an OPEN_REJECT primitive or CLOSE primitive will not respond to a received BREAK after sending the OPEN_REJECT primitive or CLOSE primitive as required in the other proposals. Specifying that a SAS device shall not send a BREAK primitive after sending an OPEN_REJECT primitive or CLOSE primitive would require even more words in the standard and more changes to hardware. So we can't use receipt of an OPEN_REJECT primitive or CLOSE primitive as a reason to transition from BreakWait to Idle.

An additional advantage to the solution proposed herein is that a SAS device that implements the ability to respond to an OPEN address frame with an OPEN_REJECT (RETRY) primitive while in the BreakWait state is that this device will keep a loop from happening when in a system with other SAS devices that DON'T implement this solution.

3 Proposal

This following is based on SAS-1.1 revision 8.

.....

7.14.4.7 SL_CC5:BreakWait state

7.14.4.7.1 State description

This state closes the connection if one is established and releases all resources associated with the connection.

This state shall:

1) send a Transmit BREAK message to the SL transmitter; and
2) initialize and start the Break Timeout timer.

If an OPEN Address Frame Received message is received in this state, then this state should send a Transmit OPEN_REJECT message to the SL transmitter.

> NOTE x - Future versions of this standard may require that SAS devices send a Transmit OPEN_REJECT message in response to an OPEN Address Frame Received message while in the BreakWait state.

[editor's note: I think that the "should" can be changed to a "shall" in SAS 2, but it's just too late for SAS 1.1.]

7.14.4.7.2 Transition SL_CC5:BreakWait to SL_CC0:Idle

This transition shall occur after:

a) receiving a BREAK Received message;
~~b) receiving an OPEN_REJECT Received message; or~~
~~c)~~ b) the Break Timeout timer expires.

[editor's note: the above changes this clause back to be exactly as it was in SAS 1.1 revision 7.]