SYMBIOS
LOGIC

Date: 22 November, 1996

To: X3T10 Membership

From: Ralph Weber, X3T10 Alternate member from Symbios Logic

Subject: Proposed Change in QErr=1 Behavior for SPC-2

Problem

In a multi-initiator environment, the current definition of QErr=1 can produce a situation where no work is getting done because the device server is continually reporting CHECK CONDITION status. Consider the following scenario:

Initiator A	Initiator B
Send Tag 1	
Send Tag 2	
	Send Tag 3
CHECK CONDITION for Tag 1	
>>Tags 2 and 3 are aborted<<	
Send REQUEST SENSE	
Send Tag 4	
	Send Tag 5
	CHECK CONDITION for Tag 5
>>Tag 4 is aborted.<<	
	Send REQUEST SENSE
	(UNIT ATTENTION for aborted tasks)
	Send Tag 6
Send Tag 7	
CHECK CONDITION for Tag 7	
>>Tag 6 is aborted.<<	
Send REQUEST SENSE	
(UNIT ATTENTION for aborted tasks)	

I could go on and on, but you get the picture. If the hosts keep the device server busy enough, there is a chance that the only work that will ever get done (after the first error is encountered) is moving UNIT ATTENTION sense data around.

Proposal

To resolve this problem, Symbios proposes that the definition of QErr=1 be changed from:

> A QErr bit of one specifies all the blocked tasks in the task set shall be aborted when the COMMAND TERMINATED or CHECK CONDITION status is sent. A unit attention condition (see SAM) shall be generated for each initiator that had blocked tasks aborted except for the initiator to which the COMMAND TERMINATED or CHECK CONDITION status was sent. The device server shall set the additional sense code to COMMANDS CLEARED BY ANOTHER INITIATOR.

to:

> A QErr bit of one specifies all the blocked tasks in the task set belonging to the initiator to which a COMMAND TERMINATED or CHECK CONDITION status is sent shall be aborted when the status is sent.

This proposal solves the problem by eliminating the interaction between a CHECK CONDITION status sent to one initiator and the handling of tasks belonging to other initiators. This proposal also can be viewed as reducing implementation complexity in the device server.