Accredited Standards Committee
X3, Information Processing Systems

Doc: X3T10.1/95a201R2X3T10.1/95a201R3
Date: December 4, 1995
Project:
Ref Doc.:
Reply to: Charles Monia

To: X3T10.1 Membership
From: Charles Monia

Subject: S2P Exception Condition Interlock (Preliminary)

BACKGROUND

The intent of this proposal is to replace theBufferQueue Full and SCA interlocks with a generic mechanism that covers all command exception conditions includingQUEUE FULL, CHECK CONDITION, RESERVATION CONFLICT and BUSY.

The original BufferQueue Full interlock was added to address a flow control problem which occurs when there are no buffers to store incoming SCSI Command SMSs. When congestion is detected, the interlock discards the incoming SMS and returns a SCSI STATUS SMS with a status value of QUEUE FULL. Even after the congestion goes away, the logical unit must continue to discard SCSI COMMAND SMSs until the pipeline of in-transit commands has been purged. If the logical unit did not do so, a subsequent in-flight command could be received and executed out of order, possibly resulting in data corruption.

This proposal is based on the observation that SCSI command exceptions of any type have the same potential for out-of-order command execution and data corruption and therefore require similar initiator responses. For that reason, a common mechanism ought to be specified which handles these in the same manner. SSA currently defines two such mechanisms -- SCA and the buffer full interlock. The goal of this proposal is to replace these by a common interlock.

In addition, this mechanism can provide a way foran existing SCSI-2 host software layered along the lines of a CAM implementationapplication to be easily 'ported' to a full-duplex, non-interlocked bus (such as SSA) without significant code changes. If implemented as described, it would supersede the use of SCA for Contingent Allegiance emulation. In an SCSI-3 environment, the use of ACA for this purposewould be unneccessary.

To undestand why, a recap of past historyis needed. The problem is that host code written for SCSI-2, implicitly relies on the half-duplex, interlocked protocolused by the parallel bus to prevent out-of-order command execution and loss of sense data.Recall that in SCSI-2, a Contingent Allegiance condition is automatically cleared (with loss of sense data) upon receipt of the next command for that logical unit from the faulted initiator (the initiator receiving the CHECK CONDITION status). In parallel SCSI, since all data transfers use the REQ/ACK handshake, the transfer of status indicating CHECK CONDITION (or any other exception) is automatically interlocked during its delivery to the initiator. Under these circumstances, there are no commands in-transit when the condition is reported. In a non-interlocked, full duplex bus, on the other hand, there might be several commands in flight when a CHECK CONDITION occurs. Without an interlock, there'd be no way to prevent these commands from being received and executed.

The standard way of dealing with this in SCSI-3 is to define a new condition, called Auto Contingent Allegiance, which effectively transfers the site of the interlock from the initiator to the logical unit. Once set, further commands are rejected with ACA ACTIVE status until the host issues a special task management request (CLEAR ACA) to remove the condition. The problem with this mechanism istwofold. First it's optional. Devices don't have to implement it.This happened becausethe SCSI community mistakenly assumed that

autosense eliminates the need for this interlock.Nothing could be farther from the truth.Any command failure represents a break in thepipeline. Without some kind of in-flight command interlock, there's no way that the host intelligence can intervene to prevent data integrity from being compromised.Secondly, it's not transparent. A CHECK CONDITION or COMMAND TERMINATED status won't set the ACA interlock unless a special bit is set in the SCSI CDB. For SCSI-2 compatibility, an intelligent adapter that wants to hide this can do so, but it's not easy.

PROPOSAL

The following is a preliminary proposal. If approved by the committee, a detailed technical proposal will be developed which contains the S2P text changes.

This proposal replaces the Buffer Full mechanism with a single generic interlock that:

1) Handles the 'buffer full' condition;
2) Provides a way to easily emulate the SCSI-2 Contingent Allegiance condition transparently to existing SCSI-2 device drivers. SCA would not be used for Contingent Allegiance emulation.
3) Can coexist with be easily extended to handleSCSI-3 Auto Contingent Allegiance;
4) Prevents out-of-order command execution whenever a command is terminated with CHECK CONDITION, TASK SET FULL, BUSY or RESERVATION CONFLICT status.

The advantage of the proposed interlock is that it can be implemented entirely within the protocol where it can be concealed from higher layers.In particular, the interconnect independent parts of an application would not have to use Auto Contingent Allegiance to operate properly with a non-interlocked busConsequentlyIn addition, an intelligent adapter could be designed with a common host interface having almost no interconnect dependencies.

In this approach, the generic interlock emulates the behavior of an interlocked bus during an exception condition. It does so through protocol extensions which allow interlock control to be transferred from the target to the initiator and automatically purge all in-flight SCSI COMMAND SMSs from the faulted initiator to the logical unit.

The new interlock is called the task set lock. Its operation is as follows:

1) Upon detecting an exception condition, the target's S2P layer sets the task set lock for the logical unit and notifies the initiator via the SCSI STATUS SMS. A new single-bit field in the SCSI STATUS SMS, the TASK SET LOCKED field, is set to indicate that the interlock is in effect. The initiator's S2P layer shall determine whether or not the interlock has been set by checking the state of this field.
2) While the lock is set, the target shall reject subsequent commands from the faulted initiator as described below. In addition, the target shall not send further SCSI STATUS SMSs from the affected logical unit to the faulted initiator. The task set lock does not affect statuses sent by other logical units and commands received from other initiators.
3) As soon as the SCSI STATUS SMS is received, the initiator's S2P layer shall immediately respond by returning a CLEAR TASK SET LOCK SMS (a new SMS). Upon receipt of this sms, the target shall unconditionally clear the task set lock and return a SCSI RESPONSE SMS with RETURN CODE set to REQUESTED FUNCTION COMPLETED SUCCESSFULLY. At this point, control of the interlock now belongs to the initiator. Clearing the lock shall not affect pending sense data or a Contingent Allegiance condition if in effect.
4) While the interlock is set, The first SCSI COMMAND SMS received from the faulted initiator after the interlock is set shall be rejected with a SCSI RESPONSE SMS having the RETURN CODE field set to a value of TASK SET LOCKED (a new value). Subsequent SCSI COMMAND SMSs from the faulted initiator shall be discarded without returning a response or status SMS.

The task set lock shall be spontaneously cleared in response to a TARGET RESET, Quiesce, or other hard reset condition.

Other Issues Design Considerations:

Controls for Setting the Task Set Lock

Intelligent adapters provide a way for the host to specify on a command by command basis whether or not to 'freeze' the command queue on a CHECK CONDITION. We may want to reflect this capability in the protocol. One way would be to define a field in the SCSI COMMAND SMS which instrcts the logical unit to not set the task set lock if the command completes with a CHECK CONDITION status.

Contingent Allegiance and Autosense

Autosense is defined in SCSI-3 as the automatic return of sense data when a command completes with a CHECK CONDITION status. In certain cases, such as a recovered error or the report of a short record by a sequential device, the CHECK CONDITION status does not mean the command failed. When autosense is used under these circumstances, the device need not set the TASK SET LOCKED field in the SCSI STATUS SMS.

Avoiding Additional SMS Traffic

One potential problem is that, compared to the current method for handling the buffer full condition, this new scheme generates one additional response SMS when the condition a buffer full condition occurs. In the above proposal, the SCSI STATUS SMS may be followed by one SCSI RESPONSE SMS to mark the first discarded command. Since lack of buffers often goes hand-in-hand with communications system congestion, this added traffic may be undesirable. For the buffer full case, this can be optimized by allowing the target to send the SCSI RESPONSE SMS described in step 4 without first sending the SCSI STATUS SMS. The down side is that, since command status is never passed to higher layers, the initiator's S2P layer would own the responsibility for retrying the command. Of course, the adapter could solve this by returning a QUEUE FULL status to the host on receiving the SCSI RESPONSE SMS.

Use of a Separate Field to Report Interlock State

In step 1, using a separate field to record the state of the interlock makes the interlock independent of STATUS contents. As shown in the autosense case and the ACA example below, decoupling the task set lock from the status value is useful. Eliminating the need for the initiator's S2P layer to check status is probably a good idea.

Inhibiting the Return of Status

Inhibiting the return of status in step 2 is necessary to insure that the state of the interlock remains synchronized between the initiator and target. If this is not done, the interlock for a new exception condition could be inadvertently lost should the CLEAR TASK SET LOCK for the existing exception and a new exception status happen to "cross in the mail".

Eliminating the RESUME bit

The CLEAR TASK SET LOCK task management function described in step 3 replaces the use of the RESUME field in the SCSI COMMAND SMS for this purpose. As noted above, since the return of status is suspended while the interlock is in effect controlled by the logical unit, it seemed like a good idea to provide the initiator's S2P layer with a way to clear the interlock as soon as possible without waiting for the a higher layer to issue another SCSI command response.

Use of Buffering Resources

Since CLEAR TASK SET LOCK is a task management function, its use of buffering resources is governed by the S2P policy for task management SMSs. That is, it doesn't aggravate the buffer full condition. Also, if a single path is used, this function and its response SMS automatically sweep the path clear of in-flight commands.

Identifying Discarded In-Flight Commands

In this proposal, the means for identifying in-flight commands discarded by the interlock are decoupled from SCSI status. There are a couple of reasons for this. First, according to theexistingcurrent version of S2Pthe draft standard, a buffer full condition is reported by returning a status value of QUEUE FULL. From that information, the initiator is supposed to infer that all subsequent commands have been discarded. That assumption is valid only when the QUEUE FULL status is caused by the scenario envisioned in the current standard (i.e., a temporary shortage of SCSI COMMAND SMSs). It doesn't work when the QUEUE FULL status is reported by some entity above the protocol layer. Such a condition can occur when some other resource unknown to the protocol layer is exhausted. Under such circumstances, the initiator can't draw any inference about discarded commands. Secondly. other exception conditions don't fit the buffer full model. For example, the reporting of RESERVATION CONFLICT, BUSY or CHECK CONDITION can happen at any time. Without a separate way to identify discarded commands, the initiator would not know which to reissue in these cases.

Scenarios:

The following examples describes how the interlock would be used.

Contingent Allegiance

A Contingent Allegiance condition exists when a command terminates with a CHECK CONDITION or COMMAND TERMINATED status. In SSA, the SCSI STATUS SMS used to report the condition is returned with TASK SET LOCKED asserted. The initiator's S2P layer immediately responds by issuing a CLEAR TASK SET LOCK SMS. On receiving the SMS, the target clears the lock. The Contingent Allegiance condition and any pending sense data are unaffected.

While the lock is set, the target discards all in-flight commands received from the faulting initiator. A SCSI RESPONSE SMS with a value of TASK SET LOCKED is returned for the first discarded command. Subsequent in-flight commands are discarded without returning status.

Upon receiving the CLEAR TASK SET LOCK, the target resets the lock and returns the SCSI RESPONSE SMS. In addition to indicating function completion, the response allows the initiator to determine which in-flight commands have been purged.

Note that throughout, all protocol transactions have occurred without the involvement of upper layers.

Auto Contingent Allegiance (ACA)

This example shows the interaction between the task set lock and ACA.

ACA is an optional 'latching' condition that occurs whenever a command completes with a status of CHECK CONDITION or COMMAND TERMINATED. It is new toSCSI-3 and replaces SCSI-2 Extended Contingent Allegiance. Unlike Contingent Allegiance, an ACA condition is not implicitely reset whenever another command is received. As a result, it accomplishes the following:

1) Provides a mechanism for multi-step error recovery, like SCSI-2 ECA;
2) Prevents in-flight commands from being executed;
3) Prevents in-flight commands from causing loss of sense data.

Except as described below,commands received while an ACA condition is in effect are terminated with a status of ACA ACTIVE.While an ACA is in effect, a single command may be executed only if it has the ACA attribute. To perform multi-step error recovery, the initiatormay issue a sequence of ACA commands. The ACA condition is cleared by means of the CLEAR ACA task management function.

Support for ACA is normally a device option. When supported by a device, the initiator must specify for each command whether or not the ACA condition shall be set ifthe command completes with a CHECK CONDITION or COMMAND TERMINATED status. The initiator does so by setting the appropriate bit in the CDB control byte for that command. If this bit is not set, command termination as described above results in a Contingent Allegiance condition.

Unlike the previous example,an ACA condition is enforced by the logical unit. Therefore,the return of a CHECK CONDITION or COMMAND TERMINATED statusdoes not cause the task set lock to be set.(i.e., the TASK SET LOCKED field is not set in the SCSI STATUS SMS). Subsequent commands are received by the target's S3P layer and passed to the logical unit.Commands from the faulted initiator are executed if they

have the ACA attribute. All other commands will be rejected with a status of ACA ACTIVEAs for the ACA condition, the TASK SET LOCKED field in the SCSI STATUS SMSfor such commands is not set.

Sincerely,

Charles Monia
Voice: (508) 841-6757
FAX: (508) 841-6100
Email: monia@shr.dec.com