Letter ballot comments on SPI-3 r10

Bill Galloway
BREA Technologies, Inc.
BillG@breatech.com
(281) 530- 3063

KEY: (E) = Editorial; (T) = Technical;

1) (T) Page 87, section 9.2.20:

I believe the next to last sentence in 99-169r1 was meant to apply to phases in both directions. “A QAS capable initiator shall assert ACK for a minimum 50ns period.” The description of output phases does not agree with 99-169r1. Replace entire section with following:

The minimum time a QAS-capable initiator shall insure that REQ, ACK are asserted, and that DATA is valid during command, message and status phases.

Other changes are in individual phase descriptions.

2) (T) Page 101, section 10.2.2.1 clause c):

Move “and if there is no attention condition” from ACK true to ACK false.

3) (T) Page 102, section 10.3.1.1.2, paragraph 1:

With SCSI today, all hot plug or independent reset events are handled gracefully by negotiation. If either the initiator or target underwent a reset event without the other one knowing it, the negotiated parameters were inconsistent. This inconsistency was handled gracefully on the first command because the device that was reset initiates a negotiation and that first command goes to completion. No commands were allowed to progress far enough without the negotiation that they had to be aborted.

The current paragraph does not allow for the graceful negotiation. The non-PPR message should not be rejected. The target should initiate a PPR negotiation if the initiator does not. If after the PPR, information units are turned off, then the command can continue normally with the current I_T_L or I_T_L_Q nexus. In addition, the first message should not be rejected if it is a TARGET RESET.

4) (T) Page 103, section 10.3.1.2, paragraph 9:

Same general comment as above.

Why can’t the target try to initiate a PPR negotiation before going to command phase. The initiator can always reject the PPR and let the target continue with the command if that is what it wants. If the PPR is successful then there would be no aborted commands.

5) (E) Page 108, section 10.5.2.2.1.1, paragraph 1:
Sentence is confusing, replace with:

The initiator shall not negate the ACK for the last byte of the last iuCRC in an information unit until the entire information unit has been verified and any required attention condition has been established.

6) (T) Page 108, section 10.5.2.2.1.1, paragraph 2:
Allow the initiator to send either ABORT TASK or INITIATOR DETECTED ERROR. The initiator knows what error it got; it does not need the target to tell it in a subsequent status (which may have an error also). A target cannot assume (and log) a CRC error when it receives an INITIATOR DETECTED MESSAGE because the definition of the message says that an initiator could send it for reasons unrelated to any previous SCSI activity.

7) (T) Page 109, section 10.5.2.2.1.1, paragraph 3:
There is no need to tie an error to a specific iuCRC interval in an information unit. Change “before the iuCRC is acknowledged” to before the last iuCRC of the information unit is acknowledged”.

8) (E) Page 109, section 10.5.2.2.2, paragraph 1:
Sentence is confusing, replace with:

When the target is transferring consecutive data groups, it shall not transition the REQ signal while the P_CRCA signal is asserted for the current data group until the initiator has acknowledged the entire previous data group.

9) (E) Page 111, section 10.5.2.2.2.1, paragraph 8:
Since this could be an improperly formatted data group change “before the last transfer of the pCRC field” to “before the last transfer of the data group”.

10) (T) Page 111, section 10.5.2.2.2.1, paragraph 8:
Allow the initiator to send either ABORT TASK or INITIATOR DETECTED ERROR. The initiator knows what error it got; it does not need the target to tell it in a subsequent status (which may have an error also). A target cannot assume (and log) a CRC error when it receives an INITIATOR DETECTED MESSAGE because the definition of the message says that an initiator could send it for reasons unrelated to any previous SCSI activity.

11) (T) Page 114, section 10.6, paragraph 3:
Replace paragraph with following:

A QAS-capable initiator shall wait a minimum of a QAS non-DATA phase REQ (ACK) period to assert ACK after detecting the assertion of REQ.

A QAS-capable initiator shall assert ACK for a minimum of a QAS non-DATA phase REQ (ACK) period and shall keep the command data valid until the negation of ACK.

12) (T) Page 115, section 10.8, paragraph 3:
Replace paragraph with following:

A QAS-capable initiator shall wait a minimum of a QAS non-DATA phase REQ (ACK) period to assert ACK after detecting the assertion of REQ.

A QAS-capable initiator shall assert ACK for a minimum of a QAS non-DATA phase REQ (ACK) period.

13) (T) Page 115, section 10.8.1, paragraph 1:
Allow the initiator to send either ABORT TASK or INITIATOR DETECTED ERROR. The initiator knows what error it got; it does not need the target to tell it in a subsequent status (which may have an error also). A target cannot assume (and log) a parity error when it receives an INITIATOR DETECTED MESSAGE because the definition of the message says that an initiator could send it for reasons unrelated to any previous SCSI activity.

14) (T) Page 115, section 10.9.1, paragraph 3:
Same as 10.8 above

15) (T) Page 115, section 10.9.2, paragraph 3 and 4:
Replace paragraph with following:

A QAS-capable initiator shall wait a minimum of a QAS non-DATA phase REQ (ACK) period to assert ACK after detecting the assertion of REQ.

A QAS-capable initiator shall assert ACK for a minimum of a QAS non-DATA phase REQ (ACK) period and shall keep the message data valid until the negation of ACK.

16) (E) Page 116, section 10.10 clause b):
I/O is not the only signal that causes a direction change. The C/D line can cause the direction of the P_CRCA line to change.

17) (E) Page 124, figure 52:
Since QAS was removed from this figure there should be no arrow from arbitration to bus free.

18) (T) Page 132, section 14.2.1, paragraph 3:
Change “TASK SET FULL” to “TASK SET FULL or BUSY as specified in SAM-2”

19) (T) Page 132, section 14.2.1:
Require TASK MANAGEMENT FLAGS to be zero for Multiple Command type.

20) (E) Page 140, section 14.2.5, paragraph 8:
Status will be good for packetized failures.

21) (E) Page 146, section 16.2.2, paragraph 5, clause b)
Grammar, change “is completed” to “can be completed”.

22) (E) Page 147, section 16.2.3, paragraph 2:
32-bit obsolete, Change “residual byte(s) contain(s)” to “residual byte contains”.

23) (E) Page 154, table 57, row 3:
32-bit obsolete, no REJECT, Replace with:
TRANSFER WIDTH equal to one | Sixteen-bit data

24) (E) Page 161, table 62, row 1:
32-bit obsolete, no REJECT, Replace with:
TRANSFER WIDTH equal to one | Sixteen-bit data

25) (T) Page 192, Annex B:
Replace entire annex with:

SCSI bus fairness
B.1 Model

Implementation of the SCSI bus fairness is optional, however, if implemented, the SCSI bus fairness protocol shall conform to this annex.

An SCSI device determines “fairness” by monitoring prior arbitration attempts by other SCSI devices. It shall postpone arbitration for itself until all lower priority SCSI devices that previously lost arbitration either win a subsequent arbitration or discontinue their arbitration attempts (as in the case where the initiator aborted an outstanding command thus removing the need to re-arbitrate).

When an SCSI device does not need to arbitrate for the SCSI bus, it shall monitor the arbitration attempts of the other SCSI devices and refresh the fairness register with the SCSI IDs of any lower priority SCSI devices that lost arbitration.

Whenever a requirement for arbitration arises, an SCSI device shall first check to see if its fairness register is clear. If it is clear, then no lower priority SCSI device has attempted and lost the previous arbitration and therefore, this SCSI device may now participate in arbitration. If the fairness register is not clear, the SCSI device shall postpone arbitration until all lower priority SCSI IDs have been cleared from the fairness register. Lower SCSI IDs are cleared as those SCSI devices win arbitration. SCSI IDs shall also be cleared if a SCSI device discontinues arbitration (e.g., as a result of an ABORT TASK message, ABORT TAST SET message, CLEAR TASK SET message, TARGET RESET message, hard reset).

The fairness register may be refreshed, updated or cleared. The fairness register is refreshed by copying the SCSI IDs of any lower priority SCSI devices that lost arbitration into the fairness register. A refresh of the fairness register completely replaces the previous contents of the fairness register. The fairness register is updated by removing the SCSI IDs of any lower priority devices that win arbitration or discontinue arbitration. The fairness register is cleared by setting all of its bits to zero. SCSI IDs may only be added to the fairness register by a refresh but may be subtracted by a refresh, update or clear.

Since the fairness register is only refreshed when an SCSI device is not arbitrating for itself, the fairness register is effectively frozen by an SCSI device prior to a requirement for its own arbitration arising. Therefore, only those lower priority SCSI devices latched into the fairness register at that time arbitrate ahead of this SCSI device. Other lower priority SCSI devices that were not latched shall not be added to the fairness register until this SCSI device has successfully arbitrated.

Arbitration fairness in targets is controlled with the disconnect-reconnect mode page (see 18.1.1).

B.2 Determining fairness by monitoring prior bus activity

B.2.1 Fairness for normal arbitration method

For the normal arbitration method this standard requires that within a bus set delay of when BSY was first asserted, the SCSI ID of all arbitrating SCSI devices shall appear on the bus. An SCSI device shall sample the bus after this time, to determine which SCSI devices are attempting arbitration, which SCSI device won, and which SCSI devices lost. Since the lower priority SCSI IDs begin to disappear after an arbitration delay from the assertion of BSY, the data bus must be sampled after a bus set delay but before an arbitration delay.

> NOTE 38 - For ease of implementation, the sample window may begin when BSY is asserted to begin arbitration and extending until SEL is asserted. Sampling of the bus during this time should occur at a high enough rate to ensure multiple samples within this window.

B.2.2 Fairness for QAS

For QAS, after detection of a valid QAS REQUEST message, this standard requires that between 1000 nsec (QAS arbitration delay) and 1490 nsec (QAS arbitration delay+bus settle delay+2 deskew delays) after detection of the MSG, C/D, and I/O signals being false the SCSI ID for all arbitrating SCSI devices shall appear on the bus. The SCSI device shall sample the bus during this time, to determine which SCSI devices are attempting arbitration, which SCSI device won, and which SCSI devices lost. Since the lower priority SCSI IDs begin to disappear at 1490 nsec, a continuous sampling of the data bus between 1000 nsec and 1490 nsec is required.

B.3 Fairness algorithm

An SCSI device shall be in one of three fairness modes. An SCSI device shall be in Fair Wait mode if it is waiting for a clear fairness register to participate in arbitration. An SCSI device shall be in the Fair Participate mode if it is participating in arbitration. An SCSI device shall be in the Fair Idle mode for all other conditions. An SCSI device shall enter the Fair Idle mode after any reset event. An SCSI device shall implement a lockout delay to prevent devices that stop arbitrating from causing deadlock.

B.3.1 Fairness Idle mode

An SCSI device shall be in Fair Idle mode if it does not need to arbitrate for the SCSI bus. The fairness register shall be refreshed during each arbitration while in the Fair Idle mode. The fairness register will then contain the SCSI IDs of any lower priority arbitration losers. An SCSI device shall transition to the Fair Wait mode if it needs to arbitrate and the fairness register is not clear. An SCSI device shall transition to the Fair Participate mode if it needs to arbitrate and the fairness register is clear.

B.3.2 Fair Wait mode

An SCSI device shall be in Fair Wait mode if it needs to arbitrate for the SCSI bus, but the fairness register is not clear. The fairness register shall be updated during each arbitration while in the Fair Wait mode. The updates subtract out arbitration winners and non-participants. The updates shall not add any new arbitration participants. An SCSI device transitions to the Fair Participate mode once all of the bits in the fairness register have been cleared.

B.3.3 Fair Participate mode

An SCSI device shall be in Fair Participate mode if it is participating in arbitration. An SCSI device can only be in Fair Participate mode if the fairness register is clear. If the SCSI device wins the arbitration it shall refresh the fairness register. If the SCSI device loses arbitration it shall keep the fairness register clear and remain in the Fair Participate mode.

B.3.3 Lockout delay

An SCSI device shall implement a Lockout delay to prevent deadlock during normal arbitration. The lockout delay shall be at least 2us and shall start once bus free has been detected. Each SCSI device shall clear its fairness register if no SCSI device starts arbitration before the lockout delay expires.

> NOTE 39 - The lockout delay is required because all the SCSI devices may be waiting for other SCSI devices to start the arbitration process. Although rare, the following example is valid and may occur. Assume an initiator at SCSI ID 7 that starts tasks in SCSI devices at SCSI IDs 0, 2, and 4. After a while, SCSI devices 0 and 2 begin arbitration, 2 wins and device 0 is recorded in the fairness register of all SCSI devices. Assume at the next arbitration, SCSI device 4 would like to arbitrate but does not because of fairness to SCSI device 0. However, this second arbitration is won by the initiator at SCSI device address 7 for purposes of ABORTING the task in SCSI device address 0. The result is that the initiator is waiting for SCSI device 4, SCSI device 4 is waiting in fairness for SCSI device 0 and SCSI device 0 no longer needs to arbitrate since its task has been aborted. When the lockout delay expires, SCSI devices 7 or 4 may begin arbitration.
>
> NOTE 40 – The lockout delay is not required for QAS arbitration because the bus will go bus free after a QAS arbitration delay. The fairness register will be cleared because of mixed arbitration (B.3.4).

B.3.4 Mixed arbitration

The fairness register shall be cleared by all SCSI devices that detect a change in arbitration mode. If the previous arbitration was an QAS arbitration then the fairness register shall be cleared if bus free is detected. If the previous arbitration was a normal arbitration then all devices that detect the QAS arbitration shall clear the fairness register.

> NOTE 41 – Some legacy devices may not detect the QAS arbitration and may not clear their fairness register.

B.4 Initiator fairness recommendations

It is generally desirable for the initiator to be the highest priority SCSI device on the bus. In this way, the initiator is guaranteed to win arbitration and may overlap commands to multiple SCSI devices. To maintain this capability, the initiator should not implement fairness towards lower level targets.

In the case of a multi-initiator system, it would again be desirable for the initiators to be the highest priority SCSI devices. However, in order to implement fairness between them, the higher priority initiator could implement fairness with the lower priority initiators only. This would require a second register in which a bit is enabled for each lower priority SCSI device to which a higher priority SCSI device shall be fair.