Reissued under new document number because it is part of current discussion.

Date: May 5, 1989 X3T9.2/89-061R0

To: X3T9.2 Committee (SCSI)

From: George Penokie / Greg Florance (IBM)

Subject: SCSI Bus Fairness Technique

BACKGROUND INFORMATION

The Small Computer Systems Interface, commonly referred to as SCSI, advertises a peer to peer relationship among initiator (host) and target (slave) devices on the bus. However, this is not true for the case when multiple devices are simultaneously attempting to arbitrate for the bus. Each SCSI bus device is assigned a unique device ID in the range of 0 to 7 in which 7 has the highest priority. During arbitration, each device drives it's corresponding device ID data bit. The device with the highest device ID wins the arbitration and becomes the bus owner while those devices which lost arbitration remove there device ID bit from the bus and wait for the next opportunity to arbitrate. The effect is that although SCSI is advertised as a peer to peer bus, the arbitration process favors the device with the higher priority device ID.

PROBLEM SOLVED

In it's infancy, the SCSI bus was primarily used by small systems typically having only one or two devices on the bus. For such a system, bus fairness was of little concern since there was sufficient bandwidth available to support the limited number of devices. However, as systems have evolved, the expansion of additional device types (tape, DASD, printer, optical, scanner, communications, etc.) with relatively higher performance capabilities have revealed a problem with the lack of SCSI bus fairness among the devices on the bus.

The nature of the SCSI bus allows an initiator to begin multiple overlapped operations to the devices on the bus. To explain how this happens, a read command is offered as an illustration. After transferring the command block from the initiator, the device will disconnect from the bus, verify the command parameters, and seek to the target track. Following disk latency, the data is transferred from the media into the device buffer. The device then reconnects to the SCSI bus and transfers the data to the initiator completing command execution. The problem experienced on a heavily utilized SCSI bus is that when it is time to reconnect to the initiator to transfer the data, a low priority devices ID may repeatedly loose arbitration to higher priority devices with similar commands. In such a situation, the delay in obtaining the data from the low priority device would create a delay in system response time to the user.

Another example of a problem arising from the lack of fairness on the SCSI bus is in the possibility of a system simultaneously mixing both batch and interactive applications which share resources on the same bus. Even if the system attempted to execute the batch applications in the background, the possibility that the batch application may utilize data stored on a higher priority SCSI device than the interactive application would result in SCSI arbitration circumventing the desired priorities in favor of the batch job.

Since SCSI devices do not incorporate fairness today, the only solution available to system designers is to add additional SCSI busses. This solution is both costly and sometimes impractical for small systems which are limited to a single bus because of power, packaging and cooling constraints.

The flexibility and utility of the SCSI bus to support many device types will make it an ideal peripheral device attachment bus for small and medium size computer systems. With the current architected limit of 8 devices per bus, the problem of SCSI bus fairness is just becoming evident today. The evolution of SCSI architecture to support 16 devices per bus will raise the problem of bus fairness to a higher level in the future.

In addition to addressing the system problem of device fairness, this invention also provides a side benefit for those devices which implement FAIRNESS. The virtually transparent implementation method proposed by this invention should add little or no cost to a product. However, the product value add should provide a marketing advantage which is especially applicable to IBM Low End Storage Products.

OVERVIEW

Simply stated, a device determines "FAIRNESS" by monitoring prior arbitration attempts by other devices. It then postpones arbitration for itself until all lower priority SCSI devices which 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 a device does not need to arbitrate for the SCSI bus, it monitors the arbitration attempts of the other devices and updates a fairness register with the device IDs of any lower priority devices which lost arbitration.

Whenever a requirement for arbitration arises, the device first checks to see of it's fairness register is clear. If it is clear, then no lower priority device had attempted and lost the previous arbitration and therefore, this device may now participate in arbitration. If on the other hand, the fairness register is not clear, the device postpones arbitration until all lower priority device IDs have been cleared from the fairness register. Lower device IDs are cleared as those lower level devices win arbitration. Device IDs can also be cleared if a device discontinues arbitration (as a result of an internal RESET or initiator directed ABORT).

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

FEATURES

The "FAIRNESS" technique defined by this invention provides the following features:

- Minimizes impact to existing software and hardware.
- Allows co-existence with non-FAIRNESS devices on the same SCSI bus.
- Maintains conformance to the evolving SCSI-2 Rev 8 standard.

DETAILED DESCRIPTION OF THE ARBITRATION PROCESS

Figure 1 illustrates the timing and signal level details of the SCSI arbitration process in which device ID 7 is arbitrating against lower priority devices. An 800 nsec Bus Free phase begins 400 nsec after BSY=SEL=0. Following the establishment of Bus Free, any device can begin Arbitration by asserting it's SCSI data bus ID bit and BSY. (BSY is a dot or'd line which can be simultaneously driven by multiple devices.) Other devices participate in arbitration by asserting both their SCSI data bus ID bit and BSY within 1800 nsec of BSY initially becoming active. A device must then wait an arbitration delay time of 2400 nsec following it's activation of BSY after which time it may sample the SCSI bus. If upon sampling the bus, the device has the highest SCSI bus device ID asserted, it then asserts SEL indicating it has won arbitration. All other devices must get off the bus within 800 nsec of the assertion of SEL. (Note that the 1800 nsec window following initial BSY=1 implies that for other than the initial arbitrating device, the assertion of SEL may occur within 600 nsec of it's activation of BSY thus effectively reducing the 2400 nsec timeout delay to 600 nsec.) 1200 nsec after SEL=1, the winning device can begin the selection process by driving both it's own device ID bit as well as the ID bit of the device to be selected. 90 nsec later, the selecting device degates BSY and continues the selection process.

DETERMINING FAIRNESS BY MONITORING PRIOR BUS ACTIVITY

It can be observed in Figure 1 that during the time between 3000 nsec and 3600 nsec, the device ID for all arbitrating devices must appear on the bus. If the initiator were to sample the bus during this time, the initiator could determine which devices were attempting arbitration, which device won and which devices lost. Since the lower priority device addresses will begin to disappear at t=3600, a continuous sampling of the data bus during this time frame is required.

For ease of implementation, the sample window can be considered to begin when BSY=1 following BUS FREE and extending until SEL=1. Sampling of the SCSI bus during this time should occur at a high enough rate to insure multiple samples within the 600 nsec window.

THE FAIRNESS ALGORITHM

Figure 2 on page 7 summarizes the operation of a FAIRNESS algorithm which is described in detail by the following line by line discussion.

- Lines 1-4 describe the circuit operation if the device is not required to participate in arbitration for itself at this time. This circuitry refreshes the FAIRNESS register F-REG each time the other devices arbitrate. The result is that F-REG contains the device address bits of lower priority devices (if any) which have attempted and lost arbitration. Note that the F-REG is refreshed after every non-participating arbitration so that devices which have discontinued arbitration are automatically removed. Thus, the contents of F-REG only reflect the participants of the arbitration process which could immediately precede a subsequent arbitration which this device may participate in.
- Line 2 latches all arbitration participants into F-REG during the sample window.
- Line 3 removes the arbitration winner from F-REG.
- Line 4 removes device IDs greater than or equal to the devices own address from F-REG. The need to remove the devices own address arises from line 8 which repeats these steps if the device wins arbitration.
- Line 4 removes device IDs greater than the devices own address from F-REG.
- Lines 5-9 describe the circuit operation if the device is required to participate in arbitration for itself and F-REG = 0 indicating that there is no lower priority device to be fair too.
- Lines 7-9 describe the normal arbitration process.
- Line 8 describes how, if the device wins arbitration, the lower priority devices IDs which lost must be saved in order to determine fairness during the next arbitration cycle. This insures that this device does unfairly participate in consecutive arbitrations, (as the case for a multi-LUN device or queued command implementation).
- Line 9 describes how, if the device lost arbitration to a higher priority device, the F-REG should remain zero so that the device will participate in the next arbitration cycle. This insures that a lower priority device will not now preempt this device from the next arbitration because a higher priority device won this arbitration.
- Lines 10-18 describe the circuit operation if the device is required to participate in arbitration for itself and F-REG <> 0 indicating that arbitration should be postponed because a lower priority device had attempted and lost arbitration earlier.
- Line 11 starts a lockout timer of greater than 2.4 usec. The SCSI standard requires that all devices which wish to participate shall begin arbitration within 1.8 usec of initial BSY=1 and must activate SEL 2.4 usec later.
- Line 12 simply waits to see if bus lockout occurs (almost never) or if another device will begin arbitration.
- Line 13 simply indicates that the fairness logic is waiting for either another device to arbitrate or for the bus lockout timeout to occur.
- Lines 14-16 describe the circuit operation if another device does begin arbitration within the lockout timeout.
- Line 14 latches the arbitration participants into T-REG so that the fairness logic can begin the process of eliminating those device ID(s) from F-REG.
- Line 15 strips removes the winning arbitration device ID from T-REG.
- Line 16 modifies the old F-REG by removing any device IDs in F-REG for which fairness is no longer required. Note that this also eliminates devices from the F-REG which discontinue arbitration prior to ever having won.
- Lines 17-18 are included to handle the case were no other device participated in arbitration within the bus lockout timeout. Lockout can occur as a result of all devices waiting for someone else to start the arbitration process. Although rare, the following example is valid and can occur. Given an initiator at address 7 which starts tasks in devices at addresses 0, 2 and 4. After a while, devices 0 and 2 begin arbitration, 2 wins and device 0 is recorded in the fairness register of device. Assume at the next arbitration, device 4 would like to

arbitrate but does not because of fairness to device 0. However, this second arbitration is won by the initiator at device address 7 for purposes of ABORTING the task in device address 0. The result is that the initiator is waiting for device 4, device 4 is waiting in fairness for device 0 and device 0 no longer needs to arbitrate since it’s task has been aborted. This lockout of the bus can be prevented by clearing the fairness register F-REG if the lockout timeout completes and returning to the beginning of the algorithm.

ADDITIONAL COMMENTS

It is generally desirable for the initiator to be the highest priority device on the bus. In this way, the initiator is guaranteed to win arbitration and can quickly and easily overlap commands to multiple devices. In the case of DASD, this can minimize the seek start delay for read and write commands. To maintain this capability, the initiator should not implement fairness towards lower level TARGET devices.

In the case of a multi-initiator system, it would again be desirable for the initiators to be the highest priority 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 mask register in which a bit is enabled for each lower priority device for which a higher priority device would be fair too.

```
===================================================================================
                Bus
 Phase ->    | Free -+--------  Arbitration -------------+ Selection

 x100    |---+-------+-----------------+----+--------+---++---+-
 nsec    0   4      12                30   36       44  48   51
        -+           +------------------------------------+
 BSY     +-----------+                                    +-----
                                            +-------------------
 SEL    ------------------------------------+
        -+           +------------------------------------------
 D7     -------------+
        -+           +-------------------------------+   +------
 D0-D6  --------------------------------------------------------

      Window during which arbitrating
      device ID's are required by the  |----|
      SCSI standard to be valid.

         |---+-------+-----------------+----+--------+---++---+-
         0   4      12                30   36       44  48   51

 Note: For convenient implementation, the sample window can be
       extended from BSY=1 following BUS FREE until SEL=1.
===================================================================================
```

Figure 1: SCSI BUS ARBITRATION TIMING

```
===================================================================================
   1    If others arbitrating but own arbitration not required then
   2      F-REG = latched arbitration participants
   3      Mask off winner ID bit in F-REG (Most Significant bit)
   4      Mask off all ID bits >= own ID in F-REG
   5    else (* own arbitration is required *)
   6      if F-REG = 0 then (* participate in arbitration *)
   7        perform normal arbitration with own ID
   8        if arbitration won, execute lines 1-4 above
   9        else re-arbitrate at next opportunity
  10      else (* F-REG <> 0 so perform fairness *)
  11        Start lockout timer of > 2.4 usec
  12        Wait for either lockout timeout or SEL=1
  13        if SEL = 1 then (* another device started arbitration *)
  14          latch arbitrating participants into T-REG
  15          Mask off winner ID bit in T-REG
  16          F-REG = F-REG and T-REG
  17        else  (* lockout timeout occurred *)
  18          F-REG = 0
  19          goto line 5
===================================================================================
```

Figure 2: FAIRNESS CIRCUIT PSEUDO CODE ALGORITHM