ECR #: 24
Title: Potential deadlock on A.G.P.
Release Date: Feb. 17, 1997
Impact: Clarification
Spec Version: A.G.P. 1.0

Summary: Since the A.G.P. interface does not support LOCK# (PCI Signal) there is a condition in which a deadlock can occur.

Background: On PCI transaction where a deadlock can occur because of write posting, LOCK# can be used to prevent it from occurring. Since A.G.P. does not support locked operations on the interface (no LOCK# signal) a potential deadlock can occur. There are a couple of choices for the A.G.P. compliant master interface. The corelogic can not prevent the deadlock from occurring when accesses on A.G.P. must be snooped before they are allowed to complete. See section 3.11 item 5 in the PCI 2.1 Local Bus specification for a description of the deadlock condition.

Change Current Specification as shown:

Add a new section called: "3.8. Special Design Considerations"

Potential Deadlock with Misaligned read access to an A.G.P. Compliant Master

When data generated by the A.G.P. compliant master is being written into main memory, the potential deadlock can occur when the A.G.P. compliant master requires the data to be flushed before completing the read initiated by the corelogic (PCI transaction). The read must be a misaligned memory access that straddles an odd DWORD boundary. The deadlock occurs when the CPU initiates a read reference that the corelogic must split into 2 accesses on the A.G.P. interface. After the initial read completes but before the 2nd access is attempted, the A.G.P. compliant master posts write data into the interface. When the corelogic attempts the 2nd part of the read, the A.G.P. compliant master terminates with Retry forcing the access to stall in the corelogic. Since the first half of the access has completed, the corelogic is not allowed to discard the data since a side affect may occur. The corelogic may not allow the write to occur until the read can complete. This is caused by the condition where the write may be required to be coherent with main memory and the CPUs cache. Since the processor bus is stalled pending the read request, the sequence ends in a livelock or deadlock condition. In either case, no agent can make forward progress and neither can be backed up to allow the other to progress.

There are two proposed solutions that can be implemented in the PCI compliant target interface of an A.G.P. compliant master to prevent this from occurring.

1) The A.G.P. compliant master can require its driver[1] to never access its internal register with a read operation that is misaligned when it has posted write data.
2) Not require that posted write data be flushed before completing a PCI read transaction to the PCI compliant target interface of an A.G.P. compliant master.

Option 1 is a reasonable choice if the device never writes data to main memory or posts data internally. Posting write data internally occurs when one state machine believes the access has completed at the final destination and changes status or causes a new process to occur that depends on the write being at the final destination. This sequence assumes that when a read of the status register occurs that the data will be pushed or pulled to the final destination.

[1] This may not be possible when support of legacy software is required.

Option 2 is reasonable when the read of status register has no effect on the posted write operation. In this case, the status register does not get updated until the write data leaves the A.G.P. compliant master.