PCI Bus Contention Solution

Please note that VLSI Technology Inc. (a Subsidiary of Philips Semiconductors) has filed a patent application for the specific implementation detailed within this appendix. Also note, however, that there are plenty of other implementations that can be dreamed up which would not infringe on this patent application. One example might be to have external pins decide which PCI device drives the bus at a given time rather than the internal PCI arbiter. Therefore, use this appendix as one example of how to avoid bus contention on internal busses during scan testing.

The internal PCI bus is a source of problems when it comes to most ATPG tools and ATPG Pattern Generation. ATPG tools have problems resolving potential bus contention on the PCI bus. They target faults for each scan cycle, generate the necessary scan data for each scan cycle to target these faults, then simulate this data to see if it causes bus contention. As a result, they end up generating lots of vectors that they can’t use due to bus contention. This results in extremely long run times and potentially reduced coverage. The following solution guarantees that the PCI bus will never have bus contention during scan test thereby reducing ATPG generation times and potentially producing much better fault coverage.

The solution is to use the PCI bus arbiter to grant the bus to one of the PCI devices. Since the flip flops which are used to generate the bus grants are on the scan chain, the ATPG tool can force scan data such that the appropriate PCI device drives the PCI bus as desired. This general idea isn’t perfect. What if we have Target only devices which don’t use bus grant? What if the ATPG tool attempts to assert multiple bus grants?

The Internal PCI Bus Arbiter acts as the central resource for enabling each internal PCI device’s tristate drivers during scan test mode. Any PCI device which has its bus grant asserted during scan test shall drive the PCI bus (AD, CBE, PAR, PERR#, SERR#). This includes PCI Target only devices. Note that bus grant is a new signal that must be added to Target only devices. Note also that these “bus grants” for Target only devices are new outputs from the arbiter that only function during scan test mode. In the event that a Target only device is selected, the CBE signals will be tristated for the duration of the scan capture cycle (one clock). This is due to the fact that a Target only device has no ability to drive the PCI CBE signals.

During scan test the PCI Bus Arbiter is responsible for asserting one and only one PCI Bus grant. The flip flops in the arbiter responsible for generating PCI bus grants are on the scan chain such that the ATPG tool can shift data into them to grant the bus to whichever PCI device it desires. Even though the ATPG tool may attempt to assert multiple bus grants, the arbiter must guarantee that only one PCI device is selected. In the event that no device is selected the arbiter must grant the bus to the “default” PCI device. This “default PCI device” can be whichever device you desire as long as it is on the PCI bus. Note: This solution assumes that the PCI control signals FRAME#, IRDY#, TRDY#, STOP#, DEVSEL#, REQ#(0:N), and INT(A:D) are not tristated. It assumes some sort of ORing logic is used for these signals.

Figure 1 shows the PCI Bus Arbiter. It only focuses on the modifications to the grant logic. The figure shows an example arbiter with 4 PCI Master devices and 2 target only devices. The "target grant" signals are shown as TGNT_N(1:0). The only additions to the logic of the normal PCI Bus
Arbiter are the addition of the flip flops to drive the "target grant" signals and the combinational logic to guarantee that only one grant is asserted during scan test. During normal operation (SCANTESTMODE=0) the PCI bus grants, GNT_N(3:0) are driven straight from the flip flops and the "target grants" are deasserted. During scan test (SCANTESTMODE=1) the GNT_N and TGNT_N outputs are driven from the flip flops (i.e., by the ATPG tool) unless multiple grants are asserted. If multiple grants are asserted by the flip flops then the combinational logic must choose one of the grants to assert while deasserting all others. If no grants are asserted by the flip flops, then the combinational logic must choose one of the grants to assert while deasserting all others.

Figure 1  
**PCI Bus Arbiter with Bus Contention Solution Additions**

Figure 2 shows the typical logic used to generate the output enable for the PCI address/data bus. It only shows one output enable being generated for the entire bus. It is common to have multiple flops generating output enables for different portions of the bus, but this is a simple extension to this example.

Figure 2  
**Typical PCI Device AD Output Enable Logic**
Figure 3 shows the logic needed by the PCI device to guarantee that we never have bus contention during scan test mode. During normal operation (SCANTESTMODE=0) the PCI devices normal output enable signal, cr_ad_oe_n, is used to enable its output drivers. But during scan test (SCANTESTMODE=1) the grant signal, GNT_N, shall be used to enable the output drivers. This logic assumes that the output enables are active low.

![Figure 3](image)

**Figure 3** *Modifications to PCI Device Output Enable Logic to Prevent Bus Contention*

Figure 4 shows the logic if the output enables are active high.

![Figure 4](image)

**Figure 4** *Logic for Active High Output Enable Circuit*