Hardware Implementation Guide for PI7C8150 PCI-PCI Bridge

By Glenn Sanders

Introduction

The PCI interface was originally created for the personal computing industry. These days it has been adopted by system designers who incorporate it into Datacom, Telecom, PCs, Servers, and many other systems.

Currently the PCI interface is used mainly as an expansion bus to add PCI slots on the system motherboards that have wide ranging applications. It is also used in add-in cards since most systems have PCI slots available to insert the PCI add-in cards.

The PI7C8150 PCI-to-PCI Bridge is a 32-bit, 66 MHz chip that is required on a system motherboard or add-in card. The following block diagrams illustrate various applications of the PCI-to-PCI Bridge.

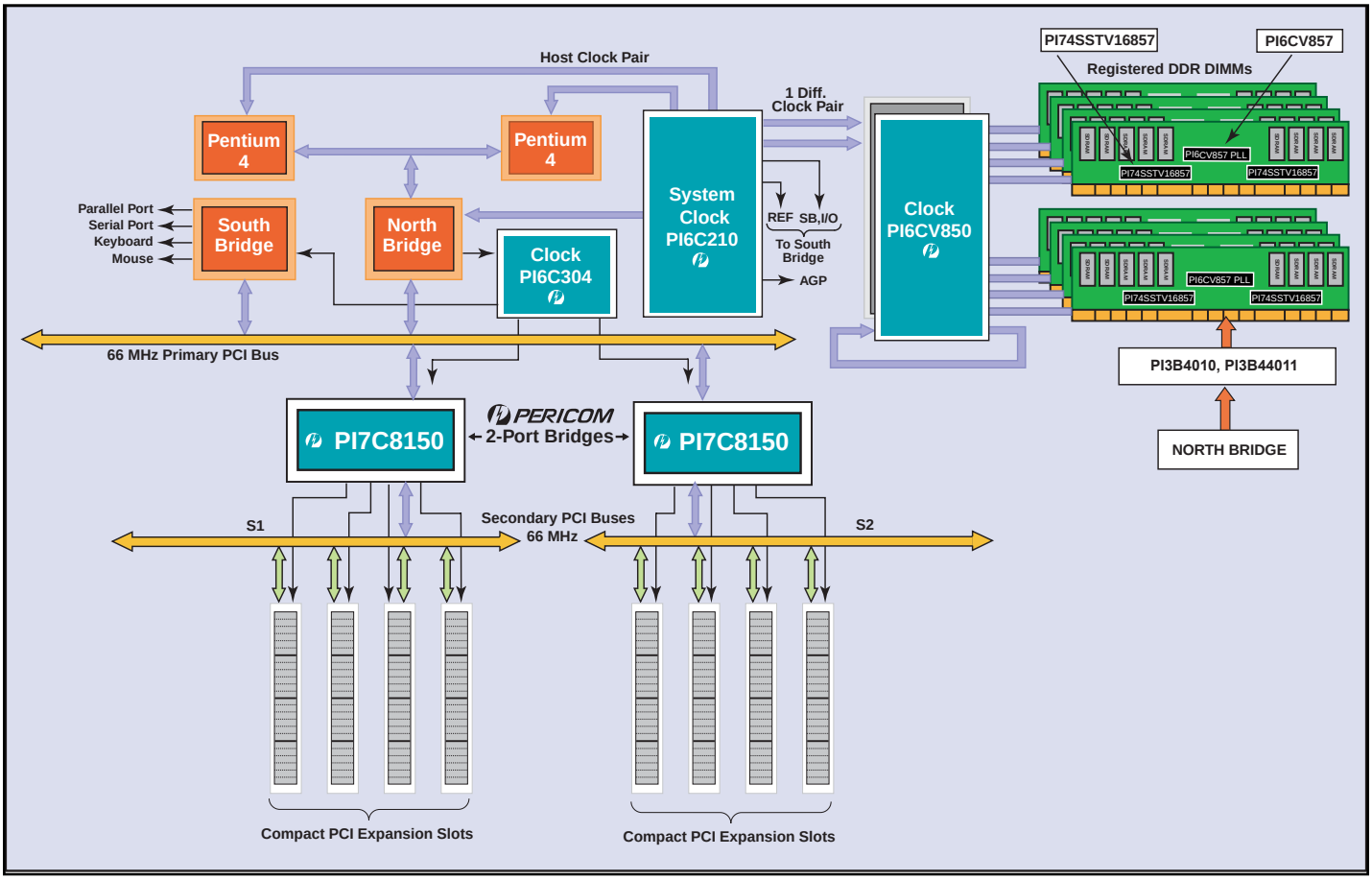


Figure 1. Server Motherboard Block Diagram

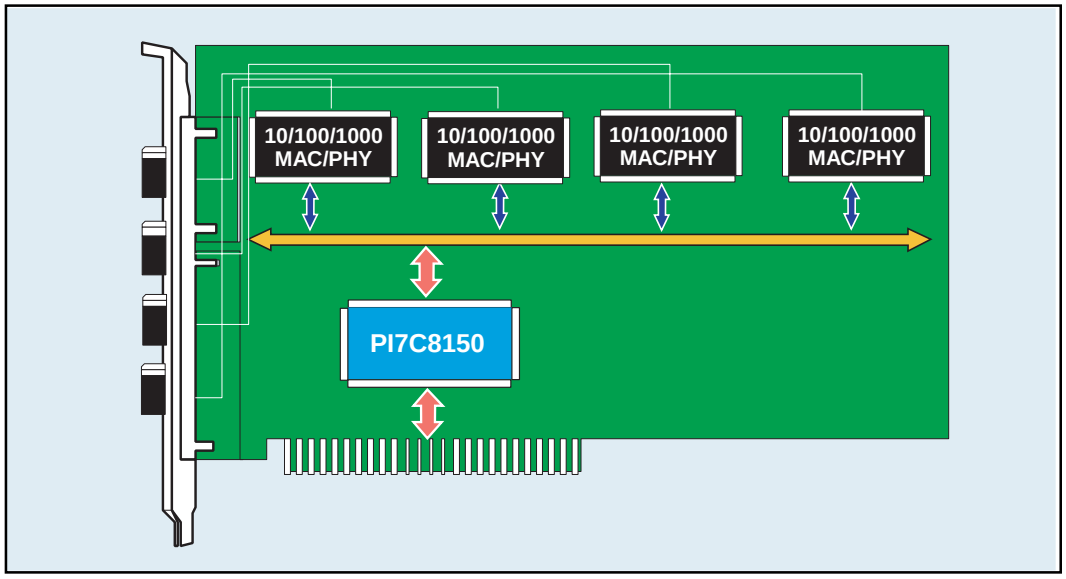


Figure 2. Four Port Ethernet Network Card

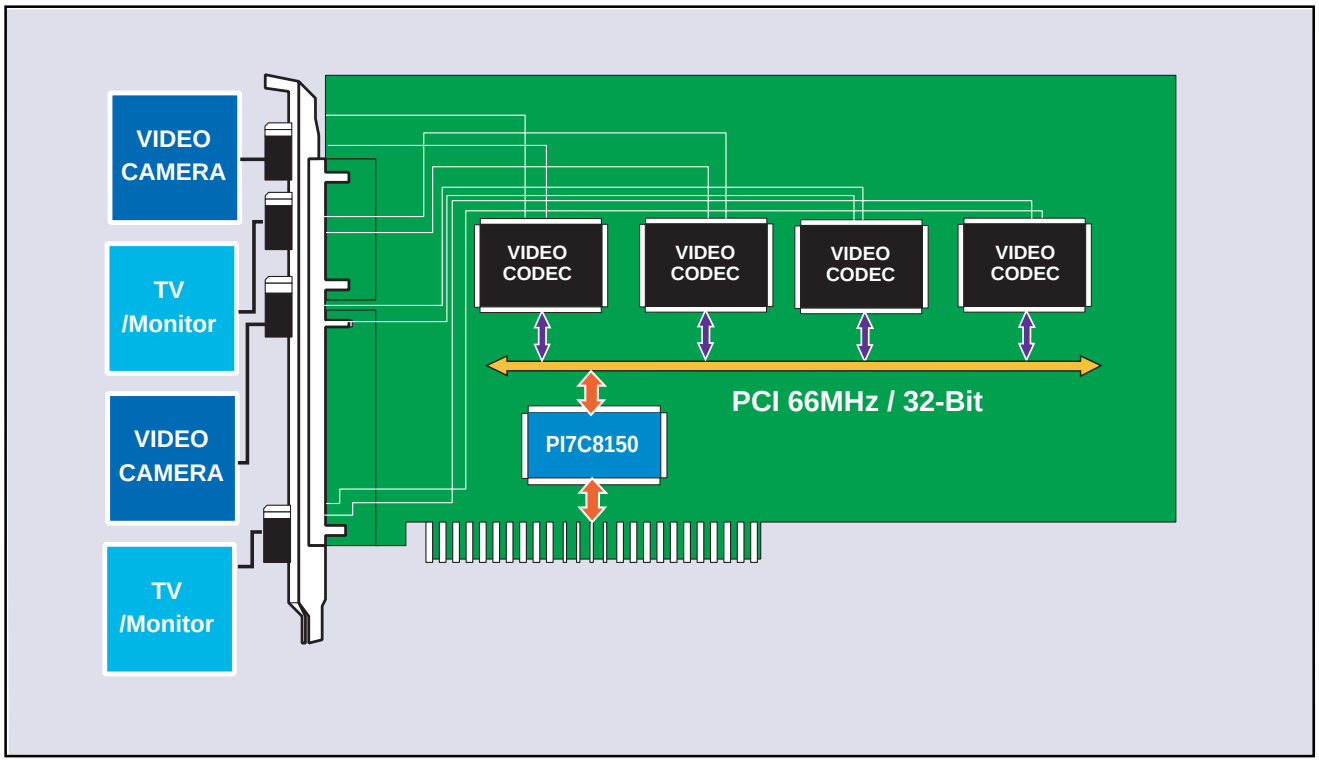


Figure 3. Video Surveillance Application

Schematic & Layout Guidelines

This section has guidelines for hardware implementation of the Pericom PI7C8150 PCI-to-PCI Bridge (8150) in a system motherboard or add-in card.

The data book shows one pin with dual definitions; (pin 125) is normally the config66 signal. This should be set to the same value as S_M66EN (high allows 66 MHz operation on the secondary bus and low disables it). However during JTAG boundary scan test this pin determines whether full-scan is in shift operation or parallel operation. Most designs leave the bridge JTAG pins unconnected.

Power

The 8150 supports both 5V and 3.3V signaling environments. The chip core is driven by 3.3V V_{DD} and signalling on either bus is at the voltage level of the respective p_vio (pin 24) or s_vio (pin 35) inputs.

p_vio and S_vio Connections: desired signal environment

Primary Bus	Secondary Bus	p_vio	s_vio
5V	5V	5V	5V
3.3V	5V	3.3V	5V
5V	3.3V	5V	3.3V
3.3V	3.3V	3.3V	3.3V

Clock guidelines

Input clocks:

The input clock frequency comes through signal P_CLK. This signal can be up to 66 MHz; the secondary bus clock will be internally derived from this clock and output at 1x or 1/2x primary input clock frequency according to:

P_M66EN	Primary bus clock	S_M66EN	Secondary Clock	CFG66
High	66 MHz	High	66 MHz	High
High	66 MHz	Low	33 MHz	Low
High	50 MHz	High	50 MHz	High
High	50 MHz	Low	25 MHz	Low
Low	33 MHz	Low	33 MHz	Low
Low	33 MHz *	Driven Low by the Bridge	33 MHz	Low

* An expansion card designed for 66 MHz operation is placed into a 33 MHz slot, or the 66 MHz primary bus has some other add-in card with M66EN tied low (thus forcing that bus to 33 MHz operation).

The input clock can be either 3.3V or 5V logic levels at 33 MHz; per PCI spec 66 MHz clocks should be 3.3V.

Output clocks

Each secondary clock output is limited to one load. One secondary clock output is used to feedback into S_CLKIN. The clock line used for S_CLKIN must match the length of the longest secondary output trace.

All secondary clock traces should have the same length so as to deliver the clock at the same time at their respective destination. Should this be impractical, a zero-delay programmable clock buffer can be used to skew, or de-skew individual traces.

Clock lines are best terminated with a series termination resistor. The value to use depends on the impedance of your transmission lines. For our reference board using 55 Ohm +/- 5% impedance we chose 22 Ohm resistors; whereas at 75 Ohm transmission lines at 50 Ohm series resistor are appropriate.

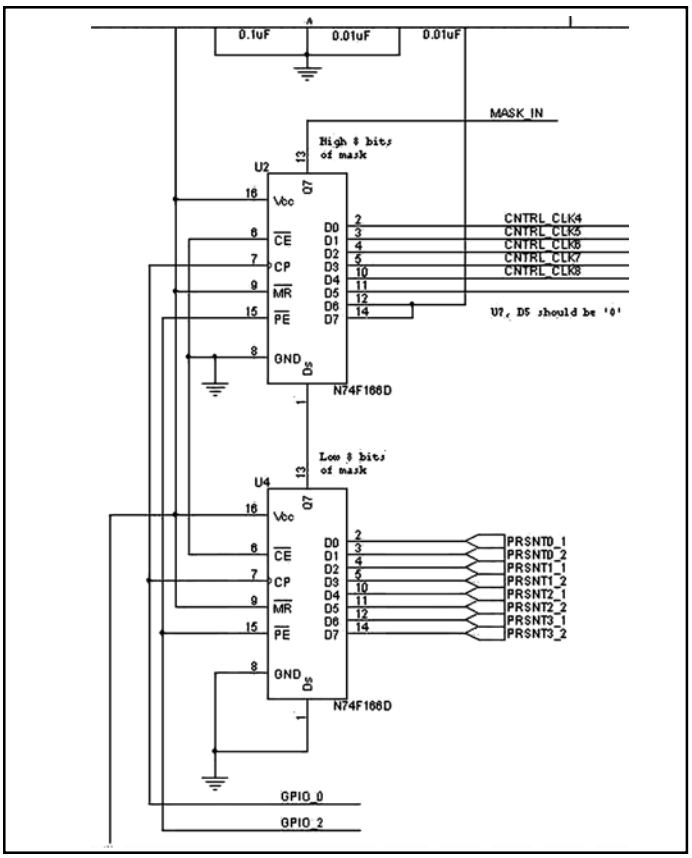
Programming clock outputs

Unused clock outputs can be disabled by using a serial clock mask shift to selectively disable secondary clock outputs.

The 8150 uses GPIO[0] and GPIO[2] pins and the MSK_IN signal to input a 16-bit serial data stream. This data is shifted into the secondary clock control register as soon as P_RESET_L# de-asserts. S_RESET_L# delays de-assertion until the 8150 completes shifting in the clock mask data. GPIO[0] acts as the shift register clock, and GPIO[2] determines shift or load operation during that shift register clock cycle. MSK_IN is the 1-bit serial data bus for this operation; thus tying it low will bus all "0" values and eliminate the need for the external shift register circuit.

Bit	Description	S_clk_o
1:0	Device 0 / slot 0 PRSNT#<1:2>	0
3:2	Device 1 / slot 1 PRSNT#<1:2>	1
5:4	Device 2 / slot 2 PRSNT#<1:2>	2
7:6	Device 3 / slot 3 PRSNT#<1:2>	3
8	Device 4	4
9	Device 5	5
10	Device 6	6
11	Device 7	7
12	Device 8	8
13	S_CLKIN (feedback clock input)	9
14	Reserved	Ignored
15	Reserved	Ignored

For the first 4 devices, the two bits corresponding to PRSNT#[2] and PRSNT#[1] are ANDed, so that a low on either is counted as a low for that device, low being an enable for that device's PCI clock.



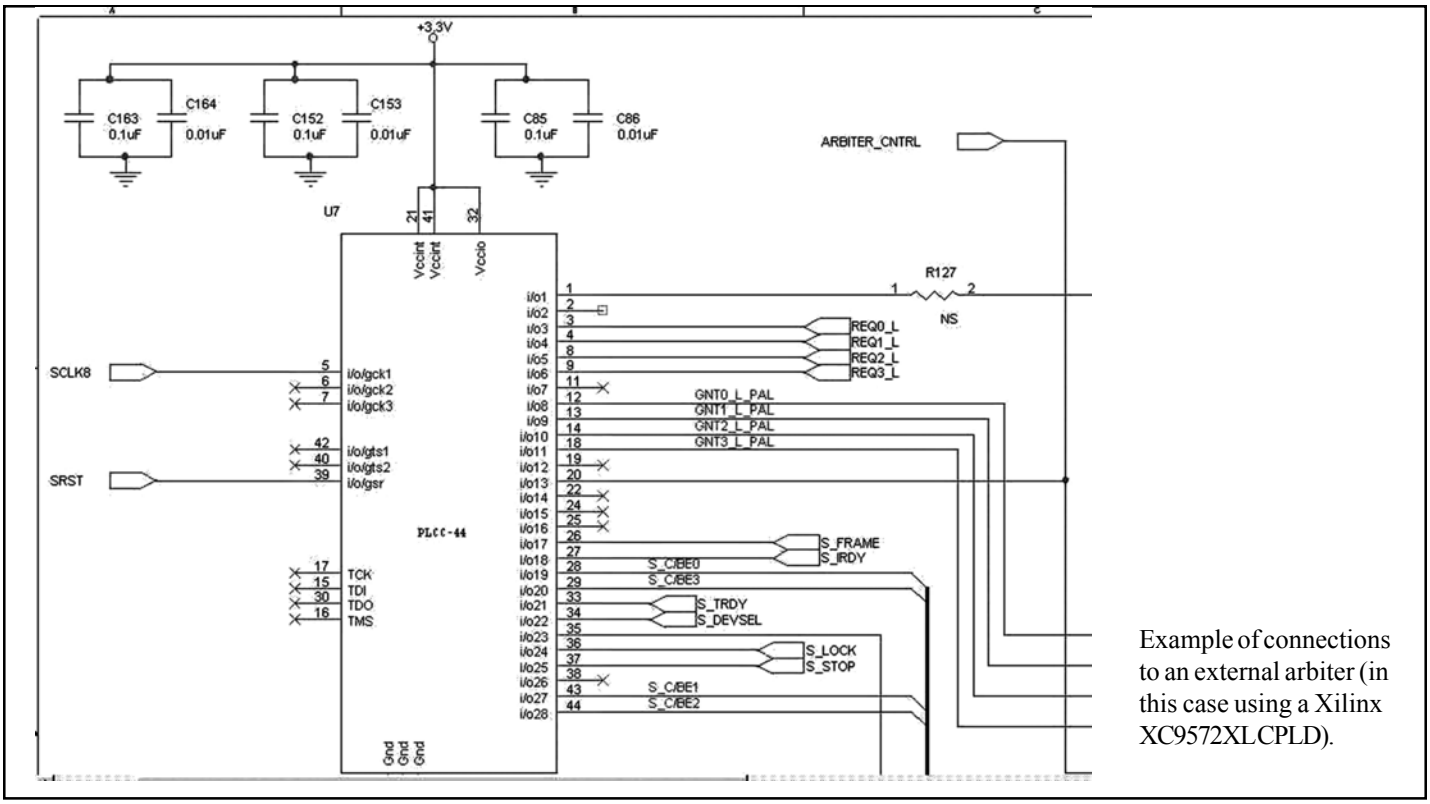
Detail of circuit used to program the enabling of secondary bus system clocks.

External Arbiter

An arbiter needs to watch the PCI bus clock, each REQ#, and the control signals RESET#, TRDY#, DEVSEL#, LOCK#, STOP#, FRAME#, CBE[3:0] and IRDY#. The 8150 normally uses an internal abiter for the secondary PCI bus. However there exists a method to use an external arbiter. In order to disable the internal arbiter, tie high (pin 23) S_CFN_L.

Next, the bridge needs to output a REQ# and wait for a GNT# input just like any other bus master device. When the bridge is the arbiter, it waits to receive as inputs REQ#s from bus master devices on the secondary bus and then issues as an output a corresponding GNT#. Now that the bridge is yet another device, it must output a REQ# to the external arbiter and wait its turn to use the bus, which will be an input GNT# signal. The signal named S_GNT_L[0] (pin 10) will become the bridge REQ# on the secondary bus, since it is an output from the bridge, in external arbiter mode. Signal name S_REQ_L[0](pin 207), as it is an input, will be the bridge GNT# on the secondary bus when in external arbiter mode.

Finally, route the S_REQ# traces from the PCI slot connectors/ embedded devices to the external arbiter. Route the S_GNT# devices likewise. At the 8150 bridge, the inputs S_REQ# already have weak pull-ups, so presuming the bridge is in reset condition after power on for many clock cycles, S_REQ#[7:1] can be left as no connect. Our data book advises to pull these signals high As GNT#s are normally output by the bridge, the S_GNT#[7:1] can also be left as no connect in external arbiter mode.



Example of connections to an external arbiter (in this case using a Xilinx XC9572XLCPLD).

Detail of circuit used to program the enabling of secondary bus system clocks.

PCI Interrupts

PCI interrupts are processed at the motherboard south bridge which sits on the primary PCI bus (thus upstream from the 8150). Thus, there are no signals at the bridge for interrupt processing; during layout the board designer routes the INTA#, INTB#, INTC#, and INTD# signals directly to the corresponding signals on the primary bus.

When the secondary bus is to have PCI connectors, the pin position of the PCI INTx# signals rotates from slot to slot.

JTAG Test Access Port Boundary Scan Signals.

When not in use, the Test Access Port (TAP) signals (TCK, TMS, TDO, TDI, and TRST_L) can be left "no connect" at the primary bus interface. These floating values are internally (weakly) pulled within the bridge, as shown by our reference board and schematic.

On the secondary bus, our reference design did not intend to perform TAP instructions, but add in cards might have those signals connected to firmware. Therefore, the reference design pulls TRST# and TCK# low at the secondary bus PCI connectors and ties TMS, TDO, and TDI to high. Thus, one add-in card will not inadvertently interfere with others.

Additional PCI signals per PCI specification 2.2, section 2.2.7: PRSNT[1:2] are pulled high with a decoupling capacitor to ground on the secondary bus. Although these are not directly connected to the bridge IC, they indirectly influence the secondary clock enable at startup when used as shown previously under "programming clock outputs".

PME# Power Management Event signal, an optional signal.

The 8150 does not have a dedicated PME# pin, so this could be bused from each PCI connector then routed around our bridge out to the primary PCI bus with some pull-up resistor. In our reference board we ran this signal as bused on the secondary side and crossing a non-stuffed zero-Ohm resistor, so that noncompliant (to the PCI Bus Power Management Interface Specification) devices inserted onto our reference board would not crash the system during power management events. An example of a PME# event might be wake-on-LAN.

3.3Vaux: This power source was bused around the bridge to the secondary bus connectors.

Miscellaneous signal connections

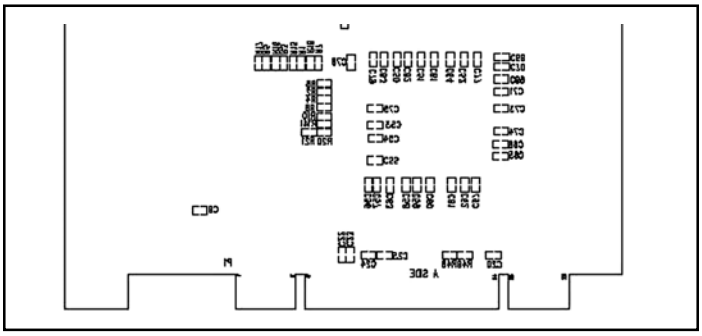
Verify that the miscellaneous signals listed below are connected to the proper value explained in the table below:

Pin name	location	Requested value
S_M66EN	Pin 153	Pull high through 5.1K Ohm resistor to allow 66 MHz secondary bus operation
S_M66EN	Pin 153	Pull low through 5.1K Ohm resistor to force 33 MHz secondary bus operation
CONFIG66	Pin 125	This should be high for 66 MHz on secondary bus
LOCK#	Pin 172	Pull up
PERR#	Pin 171	Pull up
SERR#	Pin 169	Pull up
STOP#	Pin 173	Pull up
FRAME#	Pin 179	Pull up
TRDY#	Pin 176	Pull up
IRDY#	Pin 177	Pull up
DEVSEL#	Pin 175	Pull up
for each slot		
REQ#		Pull high to Vdd through external resistor
ACK64#		Pull up
REQ64#		Pull up
SDONE	(A40)	Pull up
SBO#	(A41)	Pull up

Where "**Pull**" rather than "**tie**" is used, we intend a 5.1K-Ohm resistor to corresponding high/low.

Power Decoupling

Four sets of decoupling capacitors, top and bottom for the bridge are required, as close as possible to each corner of the bridge IC. These should be {0.1 uF, 0.01 uF, 0.001 uF} on bottom side, and be {10 uF, 0.1 uF, 0.01 uF, 0.001 uF} on top side. These are in addition to further decoupling at the PCI primary interface and secondary slots as needed per PCI spec 2.2 sec. 4.4.2.1 "***power decoupling***".



Bottom layer detail showing decoupling capacitors directly under each corner of the bridge IC.

For add-in cards, please add the following decoupling capacitors at the edge connector, for 3.3V and 5V pins, with values {0.1 uF, 0.01 uF, 0.001 uF}.

Regarding AV_{DD} and AGND: for an add-in card, these signals may be tied to digital V_{DD} and GND.

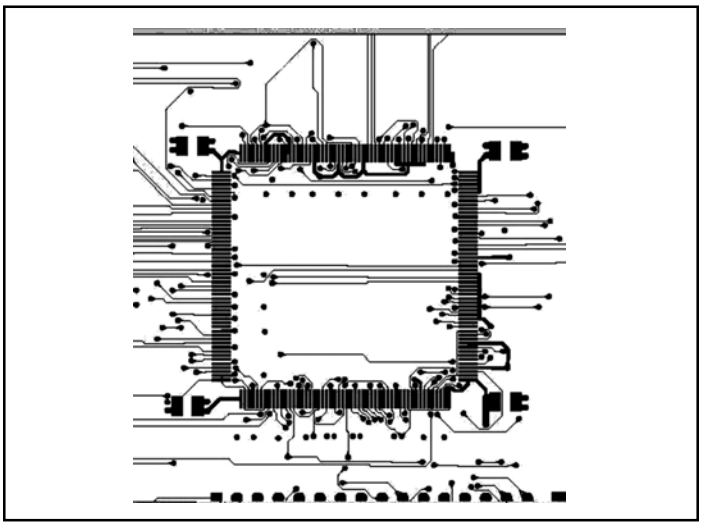
Six layer board stacking recommendation

For 5V or mixed signaling environments, we recommend a 6 layer board arranged as follows:

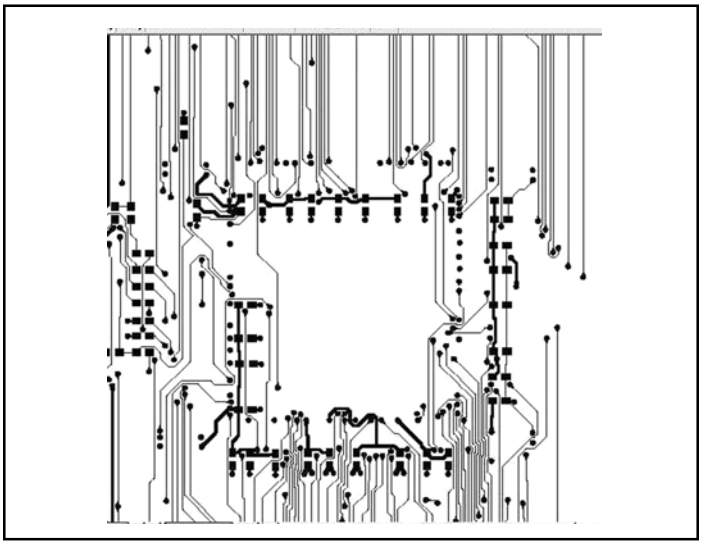
Top	route clock and other critical signals on top
Internal plane 1	Ground
Internal plane 2	3.3V
Internal plane 3	5 V (only needed for pass through power to 5V PCI slots)
Internal plane 4	Ground
Bottom	signal connections

Do NOT route high frequency bus signals under the bridge.

Signal layers should be separated by ground planes, with no signals routed between ground and power planes. We prefer FR-4 material for board fabrication.



Top layer detail:
CLKx, GNT#x, REQ#x, B side primary PCI bus



Bottom layer showing traces to PCI slots: AD bus, INTx, A side primary bus

General layout guidelines:

1. Limit your trace lengths. Longer traces display more resistance and induction and introduce more delays. It also limits the bandwidth which varies inversely with the square of trace length.
2. Use higher impedence traces. Raising the impedence will also increase the bandwidth. We advise 65 Ohm impedence with +/- 10% tolerance.
3. Do not use any clock signal loops. Keep clock lines straight when possible.
4. For related clock signals that have skew specifications, match the clock trace lengths.
5. Do not route signals in the ground and V_{CC} planes.
6. Do not route signals close to the edge of the PCB board.
7. Make sure there is a solid ground plane beneath the bridge IC (PI7C8150).
8. The power plane should face the return ground plane. No signals should be routed between power and ground.
9. Route clock signals on the top layer and avoid vias for these signals. Vias change the impedence and introduce more skew and reflections.
10. Do not use any connectors on clock traces.
11. Use wide traces for power and ground.
12. Keep high speed noise sources away from the PI7C8150.
13. Remember that per PCI spec 2.2 sec 4.4.3.1, the PI7C8150 should have a primary PCI edge connector to PQFP pad trace distance of not more than 1.5 inches (37.5 mm) for signals coming from the primary PCI interface. Secondary interface signals would then be limited as in PCI motherboard layout rules.

References:

1) Pericom Semiconductor App Note #22 "Solutions to Current High-Speed Board Design"
2) PCI Local Bus specification 2.2 section 4.4 "Expansion Board Specification" [decoupling through routing recommendations and impedence sections] p150-152.
3) PCI Local Bus specification 2.2 section 4.2.6 Pinout recommendation p131.
4) PCI Local Bus specification 2.2 section 4.3.3 Pull-ups p136.
5) Compact PCI PICMG 2.0 R3.0 p17-20 "Electrical Requirements"