Hardware Implementation Guide for the PI7C8154

By Glenn Sanders

Introduction

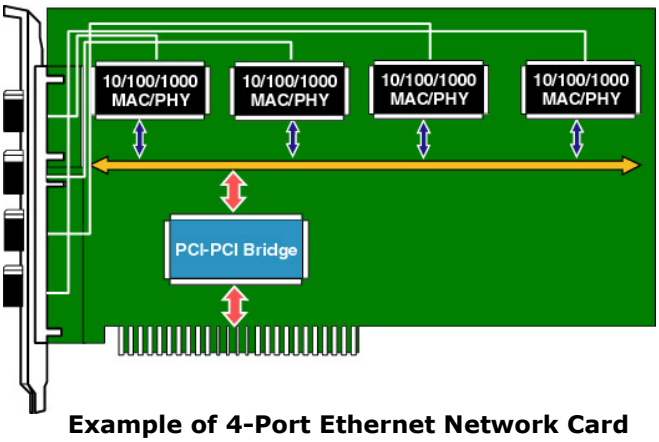
The PCI interface was originally created for the personal computing industry. These days it has been adopted by system designers who incorporate it into Datacom, Telecom, PCs, Servers, and many other systems. Currently the PCI interface is used mainly as an expansion bus to add PCI slots on the system 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.

As bandwidth requirements increased, the PCI bus was extended to support 64-bit transfers at up to 66MHz clock speed, giving approximately 4 times greater bandwidth over the earlier 32-bit/33MHz PCI bus implementations. Three factors complicated this approach:

- Increasing clock speed decreased the number of available loads per PCI bus (2 PCI connectors or 4 embedded devices at 66MHz)
- The trace length of the bus is decreased
- A 66MHz PCI bus will drop to 33MHz speed when an older style 33MHz add-in card is inserted.

A PCI-to-PCI bridge overcomes these limitations by adding a new "secondary" bus, which is electrically and physically distinct from the previous or "primary" PCI bus. Now this bus can have loading, layout distances, and potentially differing bus clock frequencies.

The PI7C8154 2-Port PCI-to-PCI Bridge is a 64-bit, 66MHz chip that can be used either within a motherboard, Backplane, or Add-In Card.



Example of 4-Port Ethernet Network Card

Schematic and Layout Guidelines

This section has guidelines for hardware implementation of the PI7C8154 PCI-to-PCI Bridge into a motherboard or add-in card.

Power

The 8154 bridge supports both 3.3V and 5V signaling environments. The chip core is powered by 3.3V V_{DD} and signaling on either bus is at the voltage level of the respective P_VIO (pin R20) or S_VIO (pin N22) inputs.

Clock Frequency

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:

Primary bus clock	S_M66EN	Secondary bus speed
66 MHz	High	66 MHz
66 MHz	Low	33 MHz
50 MHz	High	50 MHz
50 MHz	Low	25 MHz
33 MHz	Low	33 MHz
33 MHz *	[externally pulled High by 5K-ohm or greater resistor]	33 MHz; + S_M66EN will be driven low by the bridge

*The last case is where an expansion card designed for 66 MHz operation is placed into a 33 MHz slot or the normally 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*** 3.3 V.

Output Clocks:

Each secondary clock output is limited to one load. One secondary clock output is used to feedback into S_CLKIN, with the remaining 8 clocks driving embedded PCI devices/slots.

All secondary clock traces including feedback should have the same length so as to deliver the clock at the same time at their respective destinations. This means that the furthest secondary bus device from the bridge governs the effective secondary bus clock trace lengths. Unused clock outputs can be disabled by writing to the bridge configuration register at offset 68h, or terminated electrically.

Clock lines are best terminated with a series termination resistor. The value to use depends on the impedance of your transmission lines. For example, our 65-ohm trace impedance reference board uses 22-ohm resistors placed close to the bridge.

Programming clock outputs

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

The 8154 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 deasserts. S_RESET_L delays deassertion until the 8154 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 (enabling all clocks) and eliminate the need for the external shift register circuit at several dollars savings.

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.

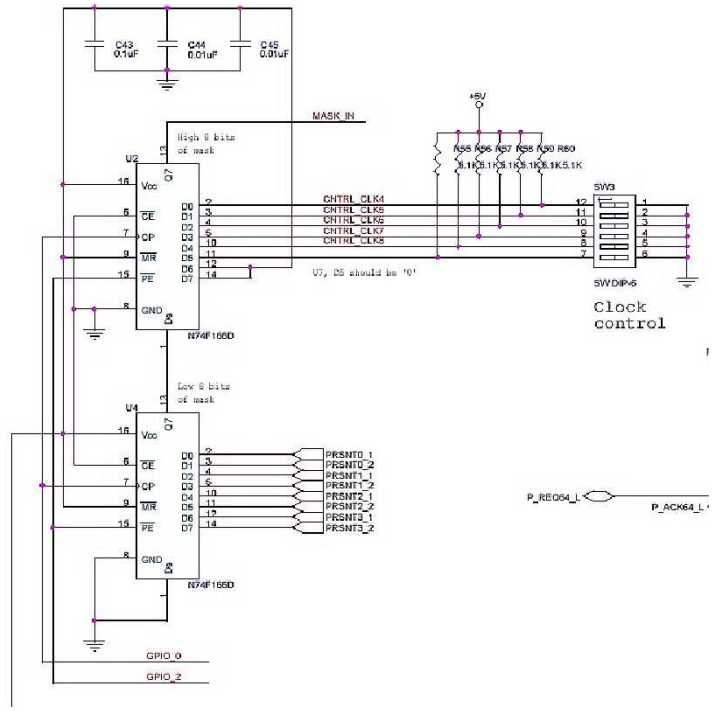


Figure 2: Schematics section: Programmng secondary clocks

Pericom Semiconductor ▪ 2380 Bering Drive ▪ San Jose, CA 95131
PH: 800-435-2336 FX: 408-435-1100
www.pericom.com

Miscellaneous Signal Connections
Unused INPUT signals should not be left floating.
Where "pull up" is used, use a 5-10K-ohm resistor to V_{CC}/V_{DD}.

For 8154 IC:		
Pin name	**Pin**	**Requested value**
MSK_IN	R21	***NOT*** pulled internally; do not float this signal. Tie low to enable all secondary bus clocks, or connect to the circuit shown in figure 2.
CONFIG66	R22	Used to prevent the bridge from 66MHz operation when tied low; this pin should be tied high to allow 66MHz operation.
TCK	N20	Pull up if JTAG port not used
TMS	P21	Pulled Up internally so can be NC
TDO	P22	Output so can be no connect
TDI	P23	Pulled Up internally so can be NC
TRST_L	N23	Pulled Up internally so can be NC
GPIO[3:0]	K2,K3, L4,L1	Pull Up through 10K-ohm resistor
P_M66EN	AB10	Pull up through 10K-ohm resistor if capable of 66MHz
S_M66EN	A14	Pull up through 10K-ohm resistor if capable of 66MHz; else tie low for 33MHz only secondary bus
S_CFN_L	K1	Tie **low** for secondary bus internal arbiter; pull high for external arbiter. For more details see page 4. Do not float
BPCCE	R4	Tie low if not using power management; pull up to enable power management
PMEENA_L	D11	Tie this signal low to indicate some downstream devices are capable of asserting PME#, else tie high to disable power management.
P_AD[63:32]		For add-in cards, the motherboard is assumed to have pull-up resistors here already, in order to support 32-bit cards placed into 64-bit slots. If this bridge is a motherboard or backplane, the primary PCI bus willl need to have pullups in the 5-10K-ohm range.**
P_CBE[7:4]		Pulled high by motherboard**
P_PAR64	T21	Pulled high by motherboard**
P_REQ64_L	AC14	Pulled high by motherboard**
P_ACK64_L	AB14	Pulled high by motherboard**
S_SERR_L	B11	Pull up
S_PERR_L	C11	Pull up
S_LOCK_L	A11	Pull up
S_STOP_L	C10	Pull up
S_DEVSEL_L	B10	Pull up
S_TRDY_L	A10	Pull up
S_IRDY_L	C9	Pull up
S_FRAME_L	B9	Pull up
S_RESET_L	H2	Pull up
S_AD[63:32]		Pull high individually
S_CBE[7:4]		Pull high individually
S_PAR64	N21	Pull up
S_ACK64_L	C18	Pull up
S_REQ64_L	B19	Pull up
For **each PCI** slot:		
Pin name	**location**	**Requested value**
REQ#		Pull high through external resistor
SDONE	(A40)	Pull up
SBO#	(A41)	Pull up
PRSNT1#	(B9)	Pull up and decouple with a 0.01uF capacitor
PRSNT2#	(B11)	Pull up and decouple with a 0.01uF capacitor

Additional PCI signals per PCI specification 2.2, section 2.2.7:

PRSNT[1:2] Normally these are pulled high with a decoupling capacitor to ground on the secondary bus.
PME# Power Management Event signal, an optional signal. *PMEENA_L. When low, 5 bits [31:27] are set at offset DEh to indicate the corresponding secondary bus devices support the PME# pin. The bridge does not have a PME# input; but it does have this as a method to notify device drivers that downstream devices may support PME# assertion, and to scan them to determine actual PME# pin support. The actual PCI connector PME# signal is then bused from the secondary PCI bus, around the bridge and out onto the primary PCI bus edge connector (for an add-in card) or onto the primary PCI bus for a motherboard implementation.

3.3Vaux: This power source, if implemented on your design, should be bused from the primary PCI connector around the bridge to the secondary bus connectors.

Power Decoupling

In order to reduce noise at Vdd or ground from impacting the bridge, place 4 sets of decoupling capacitors top and bottom 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} 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"***.

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}. Use high quality, low ESR surface mounted ceramic capacitors.

PCI INTERRUPTS

PCI interrupts are processed at the motherboard south bridge, which sits on the primary PCI bus (thus upstream from the 8154). Thus there aren't signals at the bridge for interrupt processing; rather 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., per PCI 2.2 spec 2.2.6 (page 14).

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	3.3 V
Internal plane 3	5 V (with 12V islands)
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, and no signals routed between ground and power planes. Use FR-4 material for board fabrication.

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 impedance traces. Raising the impedance will also increase the bandwidth. Per PCI specification 2.2 section 4.4.3.3 trace impedance should be controlled to be within 60 to 100 ohms range.
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 Vcc 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 (PI7C8154).
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 impedance 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 PI7C8154.
13. Remember that per PCI spec 2.2 sec 4.4.3.1, the PI7C8154 should have a primary PCI edge connector to BGA 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.

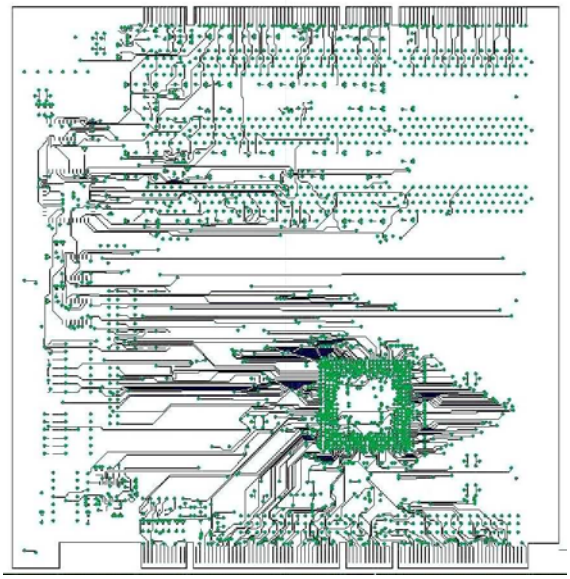


Figure 3: Top Layer Overview

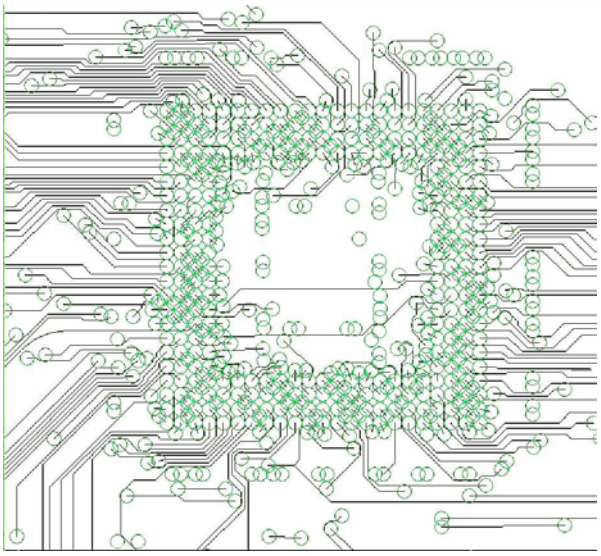


Figure 4: Top layer at bridge, showing vias and pads at surface mount

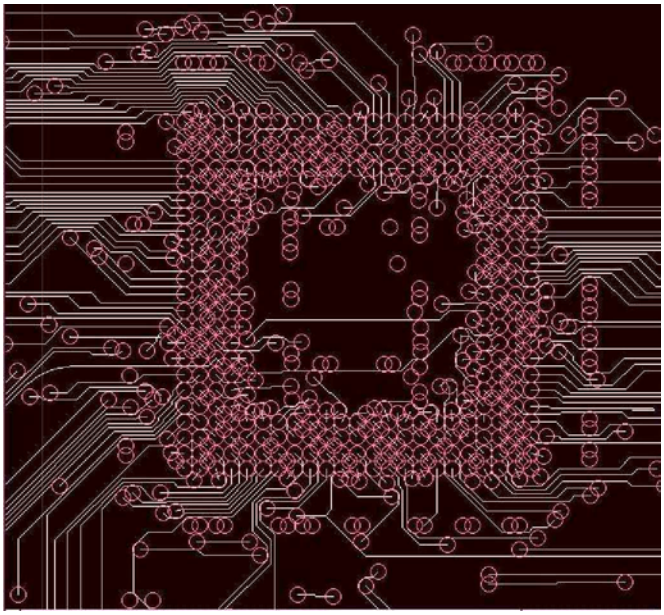


Figure 5: Same view as Figure 4, Negative film

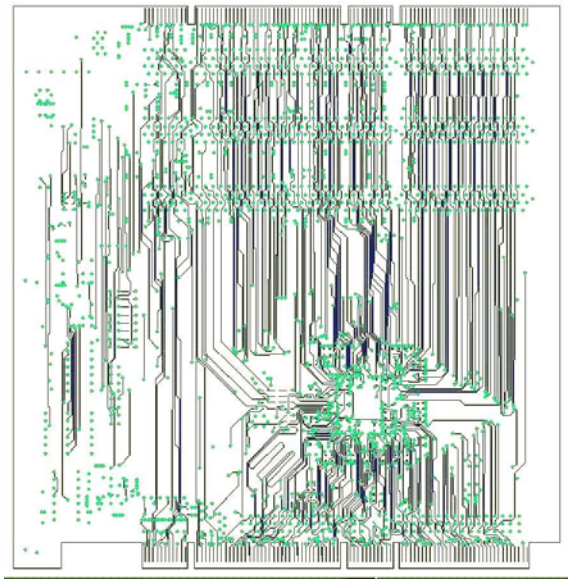


Figure 6: Bottom Layer

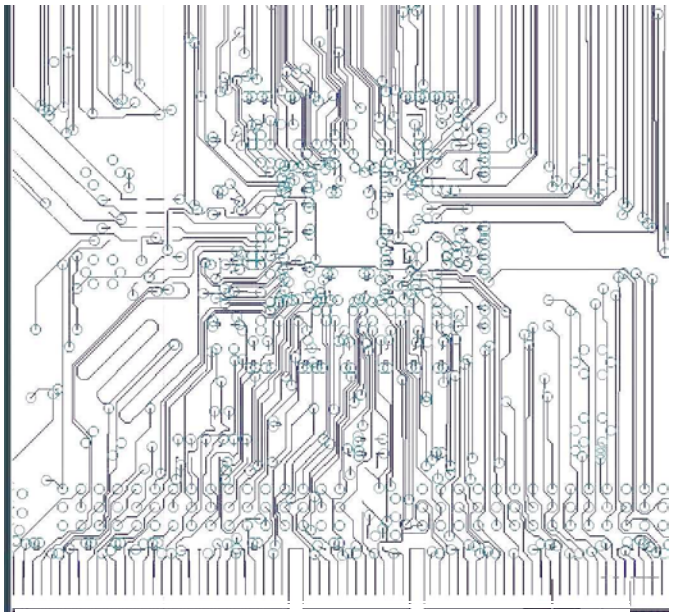


Figure 7: Bottom layer zoom near bridge

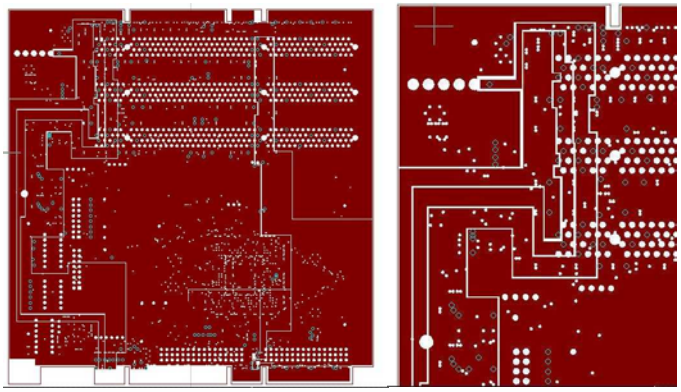


Figure 8: 5V layer with 12V islands / detail of 12V/-12V islands

Pericom Semiconductor ▪ 2380 Bering Drive ▪ San Jose, CA 95131
PH: 800-435-2336 FX: 408-435-1100
www.pericom.com

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 8154 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 K1) 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 in the internal arbiter mode, 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 E2) 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 D4), as it is an input, will be the bridge GNT# on the secondary bus when in external arbiter mode.

Finally, route the S_REQ_L[3:0] traces from the PCI slot connectors/embedded devices to the external arbiter. Route the S_GNT_L[3:0] devices likewise. At the 8154 bridge, the inputs S_REQ_L[3:0] do not have any internal pullups, so the signals S_REQ_L[3:0] need pull ups at the inputs to the bridge. As GNT#s are normally output by the bridge, the S_GNT#[3:0] can also be left as no connect in external arbiter mode.

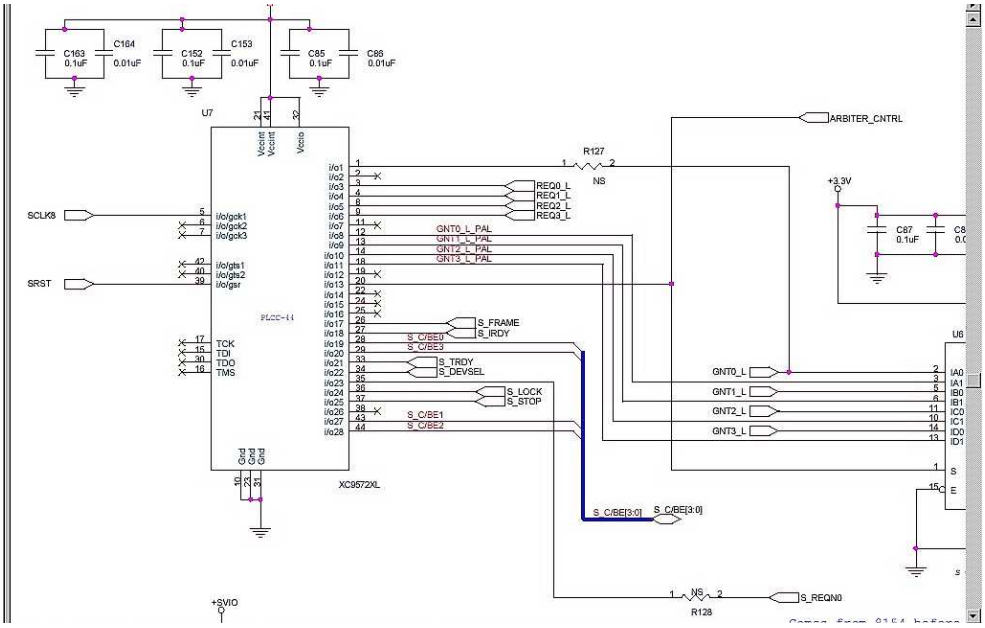


Figure 10. Example of connections to an external arbiter

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 impedance 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”
6) Pericom semiconductor App Note #31 “Zero-Delay Clock Buffer Layout and Schematic Guidelines” p1.

Reference board schematics and gerber files available on request.

Pericom Semiconductor ▪ 2380 Bering Drive ▪ San Jose, CA 95131
PH: 800-435-2336 FX: 408-435-1100
www.pericom.com