21153 PCI-to-PCI Bridge Hardware Implementation

Application Note

February 1999

Order Number: 278069-002

Information in this document is provided in connection with Intel products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not intended for use in medical, life saving, or life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The 21153 may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature may be obtained by calling 1-800-548-4725 or by visiting Intel's website at http://www.intel.com.

Contents

Figures

Tables

1.0 Introduction

This document presents guidelines for hardware implementation of the 21153 PCI-to-PCI Bridge chip (21153) in a system. This application note is limited to hardware implementation of the 21153 only and does not cover any devices that might be behind the 21153 or any initialization code needed to configure the 21153 chip.

This application note includes implementation notes on layout, clocking, secondary bus IDSEL mapping, interrupt routing, and secondary bus arbitration. Implementation on both a motherboard and an option card is covered. For most situations, hardware implementation issues are the same. In addition, guidelines for interrupt routing on option cards are provided in the interrupt discussion.

The following related Intel documents may be useful:

- *21153 PCI-to-PCI Bridge Data Sheet*
- *21153 PCI-to-PCI Bridge Configuration Application Note*
- *21153 PCI-to-PCI Bridge Evaluation Board User's Guide*

The following PCI Special Interest Group documents may also be useful:

- *PCI Local Bus Specification, Revision 2.1*
- *PCI-to-PCI Bridge Architecture Specification, Revision 1.0*

See the Support, Products, and Documentation section at the end of this document for ordering information.

2.0 Functional Overview

The 21153 fully complies with the *PCI Local Bus Specification, Revision 2.1*. The 21153 has full support for delayed transactions, which enables the buffering of memory read, I/O, and configuration transactions. The 21153 supports buffering of simultaneous multiple posted write and delayed transactions in both directions.

The 21153 provides a connection between two independent PCI buses. The two PCI buses are referred to as the primary PCI bus, which is the PCI bus closest to the host CPU, and the secondary PCI bus, which is the PCI bus farthest from the host CPU.

The 21153 allows the two PCI buses to operate concurrently. A master and target on the same PCI bus can communicate while the other PCI bus is busy.

The 21153 has two common applications. System designers can use the 21153 on a motherboard to add more devices or add-in card slots than a single PCI bus can support. Add-in card designers can use the 21153 to enable multiple devices on a single option card. Up to nine devices can be attached to the secondary bus of the 21153.

The 21153 supports both 5-V and 3.3-V signaling environments.

The 21153 PCI-to-PCI Bridge has the following interfaces:

- Primary and secondary PCI bus interfaces

  Provide control and data to two PCI buses, including LOCK#, PERR#, and SERR#. To implement these interfaces, follow the guidelines in the *PCI Local Bus Specification, Revision 2.1*. The 21153 primary interface implements *PCI Local Bus Specification, Revision 2.1*, 3.3-V drivers.
- Secondary bus arbiter

  Provides arbitration support for nine secondary bus devices, including:
  - A programmable 2-level arbiter
  - Hardware disable control, **s_cfn_l**, to permit use of an external arbiter
- Clocks

  There is a separate clock input for each PCI interface. The 10 secondary clock outputs have the following features:
  - Low skew permitting direct drive of options slots
  - Individual clock disables
  - Automatic configuration during reset using general-purpose I/O interface
- Four-pin, general-purpose I/O interface, accessible through device-specific configuration space
- Serial-scan JTAG test port that conforms to IEEE Standard 1149.1, *Standard Test Access Port and Boundary-Scan Architecture*

3.0 21153 Power Supply

The 21153 must be powered by a 3.3-V power supply. If 3.3 V is not available in the system, it can be generated from a 5-V power supply by using a voltage regulator. Figure 1 shows a recommended regulator circuit. A tantalum electrolytic capacitor of at least 10µF is required at the output of the regulator (see Figure 1).

Figure 1. 21153 Voltage Regulator Circuit

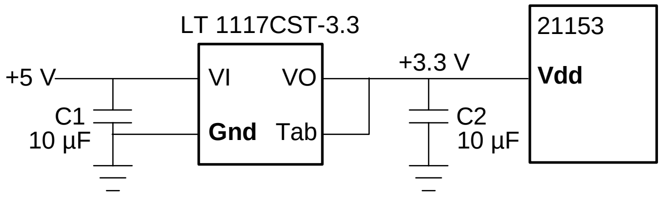


Table 1 lists the voltage regulators that can be used. To implement any of these regulators, refer to the vendor's data sheet.

Table 1. Voltage Regulator Vendors

Vendor	Part Number
Linear Technology	LT1117CST-3.3
Texas Instruments	TLV2217-33
Motorola	MC33269D
National Semiconductor	LM3940

3.1 Power Sequencing

Early PCI applications used a single 5V power supply. As 3.3V PCI devices became available, a combination of 3.3V and 5V supplies were needed for a single option. Most, if not all, of these applications generally use a voltage regulator circuit as part of application to supply 3.3V from the 5-volt source.

In the natural circuit timing of a voltage regulator, the 5V supply precedes the 3.3V supply by default. When a system provides 3.3-volts from a separate power supply the timing of the power supplies when turning on or ramping down may not be well defined. It is important that the two supplies should follow the example of the voltage regulator circuit (see Figure 1).

For the PCI-to-PCI bridge products, there is a specified sequence requirement for the 3.3V and 5V supply activation and deactivation. These power sequencing requirements for the 21150, 21152, 21153, 21154, and 21554 products is as follows (Vdd refers to 3.3V and Vcc refers to 5V):

While activating or deactivating power the 5-volt and 3.3-volt supplies should track each other within 1.8-volts:

- If Vdd < 3.0V then Vcc - Vdd < 1.8V
- If Vcc < 1.8V then Vdd can be 0
- If Vcc > 1.8V then Vdd must be > 0

The 3.3V (Vdd) supply may lag the 5V (Vcc) supply by up to 1.8 volts while the supplies are changing. When both supplies have settled to their final values, Vcc can be up to 5.25 volts if Vdd > 3.0V.

Note: Slow supply ramp rates, greater than 10 ms, could cause die heating in CMOS devices. This would depend on factors other than the I/O of the PCI-to-PCI bridge. Measurements to determine ramp rates should be taken between Vmax and Vmin. It should also be noted that a power down rate greater than 10ms will not damage the device if the Vcc and Vdd requirements that follow are met.

There is no ramp rate (timing to voltage stable) limitation. Only the difference in voltages is important. This restriction applies both to powering up and down. There are no restrictions if the 3.3V supply comes up before 5V, or shuts off last.

Note: To prevent possible damage when using separate 3.3 and 5V power supplies a 1K ohm resistor should be placed between p_vio and s_vio and the 5V supply when 5-volt options are used.

3.2 5-V and 3.3-V Signaling

The 21153 I/O pads are 5-V tolerant and will operate under both 3.3-V and 5-V signaling environments. The primary and secondary PCI buses can be independently interfaced to either a 3.3-V or the 5-V signaling environment by connecting the **p_vio** and **s_vio** pins to the appropriate voltages (refer to Table 2). These pins are connected to a clamp circuit in the pad driver that turns on when the output voltage matches the voltage on the **p_vio** and **s_vio** pins.

Pay special attention to board layout and signal environments when mixing 3.3-V and 5-V devices on the same bus. It is guaranteed by design that the 21153 is reliable in both 3.3-V and 5-V environments when **p_vio** and **s_vio** are connected in accordance with. In all applications, follow the *PCI Local Bus Specification, Revision 2.1* concerning layout and signal integrity issues.

Table 2. p_vio and s_vio Connections

Signaling Environment Primary Bus	Secondary Bus	p_vio (Pin 24)	s_vio (Pin 35)
5 V	5 V	5 V	5 V
5 V	3.3 V	5 V	3.3 V
3.3 V	5 V	3.3 V	5 V
3.3 V	3.3 V	3.3 V	3.3 V

4.0 Clocking

The following sections provide an overview of 21153 clocking requirements, and explain how to implement clocks when using the 21153 on a motherboard and on an option card.

4.1 21153 Clocking Domains

The 21153 has two clocking domains: one for the primary PCI interface and one for the secondary PCI interface. Each PCI interface has a separate clock input. The primary interface is controlled by the primary clock input, **p_clk**, and the secondary interface and arbiter are controlled by the secondary clock input, **s_clk**.

Both interfaces must operate at the same frequency and must be synchronous to each other. Their edge relationships to each other are well-defined. The relationship between the **p_clk** and **s_clk** inputs has the following restrictions:

- The 21153 operates at a maximum frequency of 33 MHz, and **s_clk** always operates at the same frequency as **p_clk**.
- The maximum delay between **p_clk** and **s_clk** is 7 ns for both rising and falling edges.
- The minimum delay between **p_clk** and **s_clk** is 0 ns, that is, **s_clk** cannot precede **p_clk** for both rising and falling edges

- To compensate for the 21153 clock buffer characteristics to the secondary bus logic, the rise and fall times of **p_clk** must be constrained so that the clock provided to a slot has a duty cycle derived from maximum rise and fall times minus the clock skew.

  No duty cycle compensation is needed if :

  Tcycle ≥ Tcycle_min + 2 x Tskew

  if Tskew = 750 ps and Tcycle ≥ 31.5 ns (or 31.75 MHz and below)

  Motherboard designers must compensate for duty cycle skew of the 21153 clock buffer by constraining Tr and Tf (rise and fall time) when:

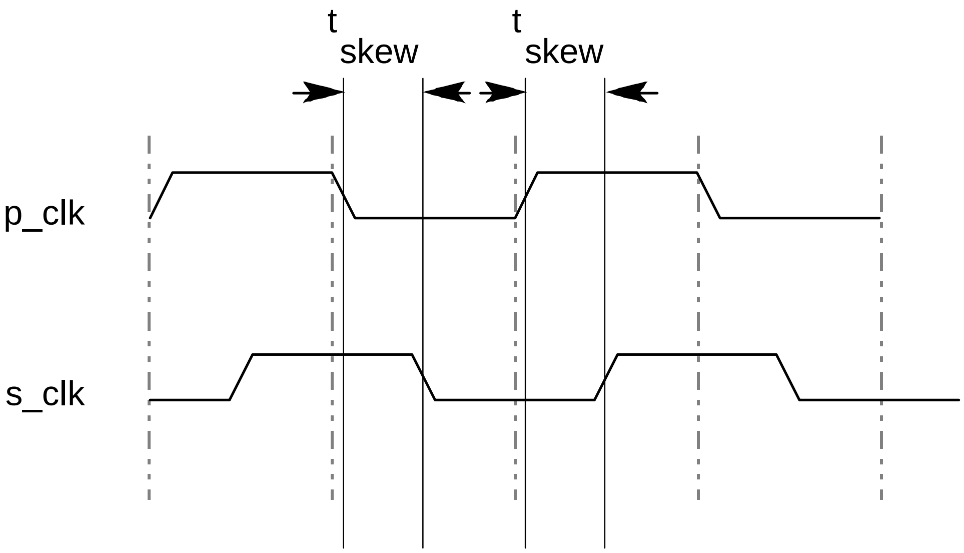
  Tcycle ≤ Tcycle_min + 2 x Tskew

  if Tskew = 750 ps and Tcycle ≤ 31.5 ns (or 31.75 MHz and above)

  Therefore, a Tr and Tf maximum = 3 ns – skew of 750 ps = 2.25 ns will give a Thigh or Tlow of 12.75 ns for a Tcycle minimum of 30 ns.

Figure 2 shows the timing relationship between the primary and the secondary clock inputs.

Figure 2. p_clk and s_clk Relative Timing



LJ-04646.AI5

4.2 21153 Output Clocks

The 21153 generates 10 secondary bus clock outputs, **s_clk_o<9:0>**. These clocks are derived from **p_clk**. Use one of the secondary clock outputs as the secondary clock input to the 21153; use the other nine outputs for devices on the secondary bus.

The **s_clk_o<9:0>** clock outputs are generated directly from **p_clk** by internal buffers, and their relationships to **p_clk** have the following properties:

- All clock outputs operate at the same frequency as **p_clk**.
- The maximum delay between **p_clk** and **s_clk_o** is 5 ns for both rising and falling edges. Therefore, to meet the **p_clk** and **s_clk** requirements, no more than 2 ns of delay is allowed for secondary clock etch returning to the device secondary clock input.
- The minimum delay between **p_clk** and **s_clk_o** is 0 ns for both rising and falling edges.
- The maximum skew between **s_clk_o** edges is 500 ps.

4.3 Guidelines for Using Secondary Clock Outputs

The guidelines for using secondary clocks are:

- Each secondary clock output is limited to one load.
- One of the secondary clock outputs must be used for the 21153 **s_clk** input.
- Intel recommends using an equivalent amount of etch on the board for all secondary clocks to minimize skew between them.

  Route **s_clk_o<9:0>** using the same length of etch for each output including that used as **s_clk** input. Intel recommends that the amount of delay due to etch be limited to less than 2 ns (10 inches if etch has a delay of 200 ps/in).
- Intel recommends that you use a series-termination resistor to terminate the **s_clk_o<9:0>** outputs to limit reflections on the clock lines. The value of the resistor depends on the transmission line impedance of the etch, which depends on the board layout. As a result, it is difficult to recommend a precise value. Based on a 75 Ω transmission line, a 50 Ω series resistor has been seen to provide acceptable clock waveforms.

4.4 Disabling Unused Secondary Clock Outputs

When secondary clock outputs are not used, both **gpio<3:0>** and **msk_in** can be used to clock in a serial mask that selectively tristates secondary clock outputs.

4.5 Serial Clock Mask Shift

The 21153 uses the **gpio** pins and the **msk_in** signal to input a 16-bit serial data stream. This data stream is shifted into the secondary clock control register and is used for selectively disabling secondary clock outputs.

The serial data stream is shifted in as soon as **p_rst_l** is detected deasserted and the secondary reset signal, **s_rst_l**, is detected asserted. The deassertion of **s_rst_l** is delayed until the 21153 completes shifting in the clock mask data, which takes 23 clock cycles.

An external shift register is used in this example to load and shift the data. The **gpio** pins are used for shift register control and serial data input.

Table 3 shows the operation of the **gpio** pins.

Table 3. gpio Operation

gpio Pin	Operation
gpio<0>	Shift register clock output at 33 MHz maximum frequency.
gpio<1>	Not used.
gpio<2>	Shift register control: • 0—Load • 1—Shift
gpio<3>	Not used.

The data is input through the dedicated input signal, **msk_in**.

The shift register circuitry is not necessary for correct operation of the 21153. The shift registers can be eliminated, and **msk_in** can be tied low to enable all secondary clock outputs or tied high to force all secondary clock outputs high.

Table 4 shows **gpio** pin serial data format.

Table 4. gpio Serial Data Format

Bits	Description	s_clk_o Output
<1:0>	Slot 0 PRSNT#<1:0> or device 0	0
<3:2>	Slot 1 PRSNT#<1:0> or device 1	1
<5:4>	Slot 2 PRSNT#<1:0> or device 2	2
<7:6>	Slot 3 PRSNT#<1:0> or device 3	3
<8>	Device 4	4
<9>	Device 5	5
<10>	Device 6	6
<11>	Device 7	7
<12>	Device 8	8
<13>	21153 **s_clk** input	9
<14>	Reserved	Not applicable
<15>	Reserved	Not applicable

The first eight bits contain the PRSNT#<1:0> signal values for four slots; these bits control the **s_clk_o<3:0>** outputs. If one or both of the PRSNT#<1:0> signals are 0, a card is present in the slot and the secondary clock for that slot is not masked. If these clocks are connected to devices and not to slots, one or both of the bits should be tied low to enable the clock.

The next five bits are the clock mask for devices. Each bit enables or disables the clock for one device. These bits control the **s_clk_o<8:4>** outputs: 0 enables the clock and 1 disables the clock.

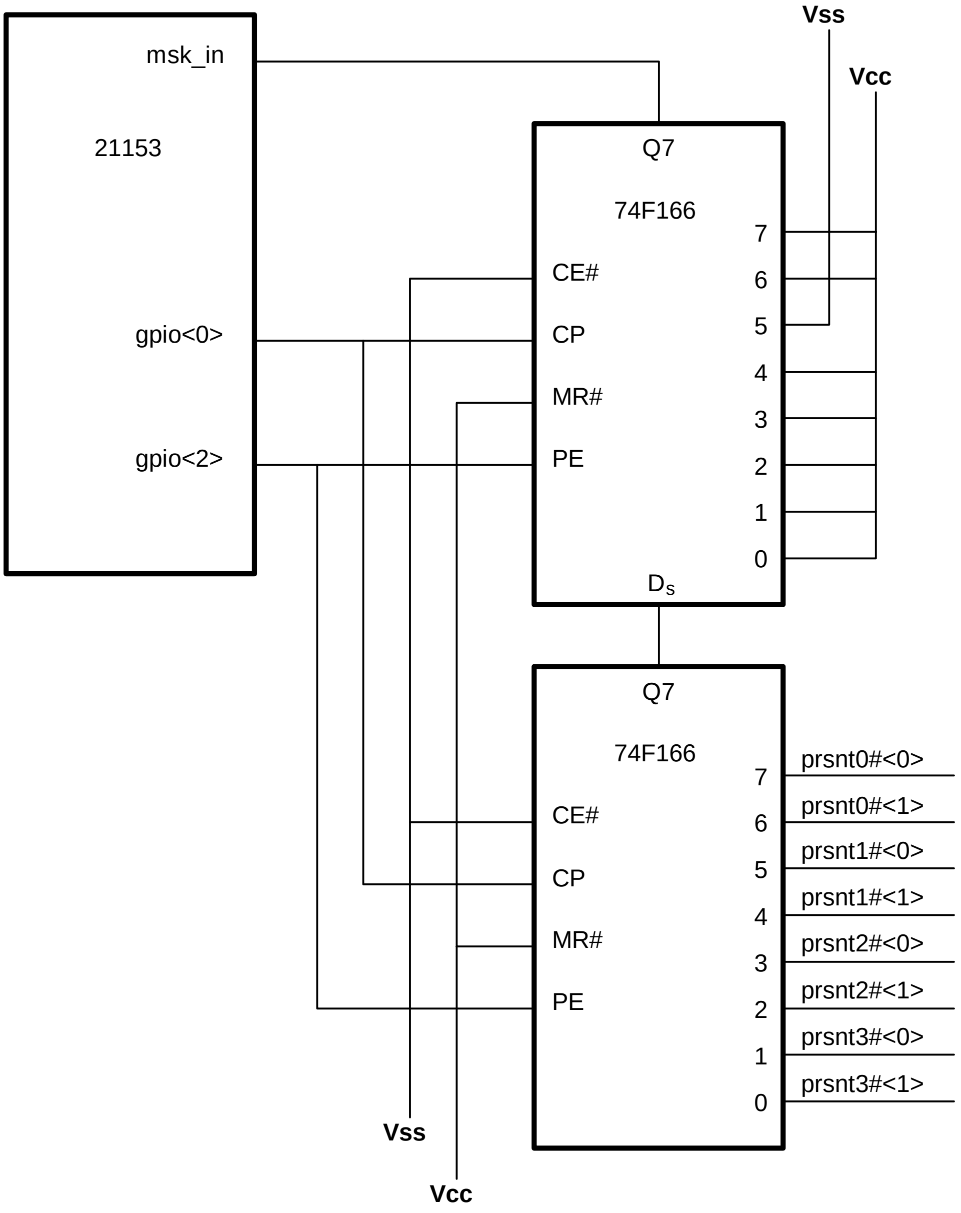
Bit 13 is the clock enable bit for **s_clk_o<9>**, which is connected to the 21153 **s_clk** input.

If desired, the assignment of **s_clk_o** clock outputs to slots or devices, and the 21153’s **s_clk** input can be rearranged from the assignment shown in . However, the serial data stream format must match the assignment of **s_clk_o** outputs.

The **gpio** pin serial protocol is designed to work with two 74F166 8-bit shift registers.

Figure 3 shows how the serial mask circuitry may be implemented for a motherboard with four PCI connector slots.

Figure 3. Example of gpio Clock Mask Implementation on the System Board

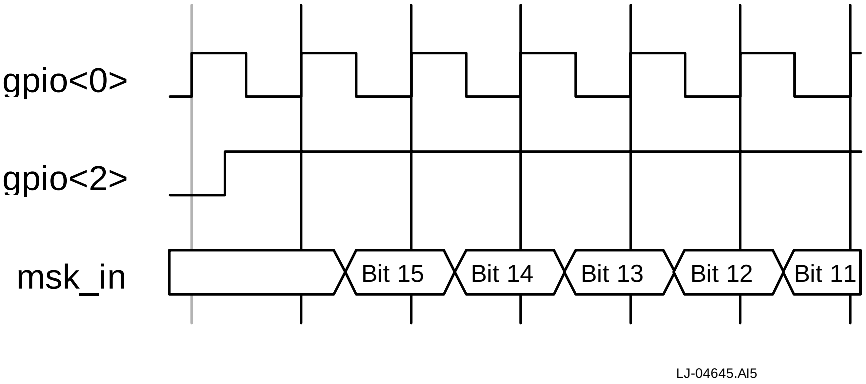


The eight least significant bits are connected to the PRSNT# pins for the slots. The next five bits are tied high to disable their secondary clocks because those clocks are not connected to anything. The next bit is tied low because its secondary clock output is connected to the 21153 **s_clk** input.

When the secondary reset signal, **s_rst_l**, is detected asserted and the primary reset signal, **p_rst_l**, is detected deasserted, the 21153 drives **gpio<2>** low for one cycle to load the clock mask inputs into the shift register. On the next cycle, the 21153 drives **gpio<2>** high to perform a shift operation. This shifts the clock mask into **msk_in**. The most significant bit is shifted in first, and the least significant bit is shifted in last.

Figure 4 shows a timing diagram for the load and for the beginning of the shift operation.

Figure 4. Clock Mask Load and Shift Timing



After the shift operation is complete, the 21153 tristates the **gpio** signals and can deassert **s_rst_l** if the secondary reset bit is clear. The 21153 then ignores **msk_in**. Control of the **gpio** signal now reverts to the 21153 **gpio** control registers. The clock disable mask can be modified subsequently through a configuration write command to the secondary clock control register in device-specific configuration space.

The 21153 delays deasserting the secondary reset signal, **s_rst_l**, until the serial clock mask has been completely shifted in and the secondary clocks have been disabled or enabled, according to the mask. The delay between **p_rst_l** deassertion and **s_rst_l** deassertion is approximately 23 cycles.

4.6 Clock Implementation on the Motherboard

When the 21153 is implemented on a motherboard and the secondary bus is routed to one or more PCI expansion slots or to another 21153, you can use the 21153 secondary clock outputs **s_clk_o<9:0>**, provided you guarantee that the output clocks meet PCI duty cycle specifications. You must include the 750-ps skew added by the 21153 to the duty cycle in the analysis of the clocks’ circuits.

If you choose not to use the secondary clock outputs, you can substitute a low-skew clock buffer to generate secondary clocks.

Table 5 lists some of the buffers that can be used.

Table 5. Low-Skew Clock Buffers

Vendor	Part Number
Texas Instruments	CDC328A
National Semiconductor	CGS74B2525
IDC	1DT74FCT805CT

One of the secondary clock signals must be connected to the 21153 secondary clock input, **s_clk**. Plan the layout of clocks to minimize skew between the 21153 and other PCI devices.

5.0 JTAG Overview

The 21153 contains a Joint Test Action Group (JTAG) serial-scan test port that conforms to IEEE standard 1149.1, *IEEE Standard Test Access Port and Boundary-Scan Architecture*. Refer to the *21153 PCI-to-PCI Bridge Data Sheet* for detailed operation.

Table 6. JTAG Pins

Signal Name	Type	Description
tdi	Input	Serial boundary-scan data in
tdo	Output	Serial boundary-scan data out
tms	Input	JTAG test mode select
tck	Input	Boundary-scan clock
trst_l	Input	JTAG test access port reset

5.1 Initialization

The test access port controller and the instruction register output latches are initialized when the **trst_l** input is asserted. The test access port controller enters the test-logic reset state. The instruction register is reset to hold the bypass register instruction. During test-logic reset state, all JTAG test logic is disabled, and the chip performs normal functions. The test access port controller leaves this state only when an appropriate JTAG test operation sequence is sent on the **tms** and **tck** pins.

Note: The JTAG test access port is to be used only while the 21153 is not operating. During normal chip operation, the JTAG logic must be disabled. To accomplish this, the **trst_l** pin must be pulled low or grounded.

6.0 General Layout Guidelines

When using the 21153, you need to consult the general layout guidelines provided in the *PCI Local Bus Specification, Revision 2.1*.

Clock routing has some special requirements. Guidelines and requirements for clock routing are discussed in the Clocking section of this application note. Guidelines for routing of secondary IDSEL signals are given in the Secondary IDSEL Mapping section of this application note.

This section discusses pull-ups and expansion card routing.

6.1 Motherboard Requirements

Consult the physical requirements in the *PCI Local Bus Specification, Revision 2.1* for the layout of the PCI on a system motherboard. The system timing requirements, clock skew, signal velocity, and round-trip propagation delay of 10 ns should be the goal for operation at 33 MHz.

6.2 Expansion Card Routing

Follow the guidelines and requirements for routing on expansion cards in the *PCI Local Bus Specification, Revision 2.1*. This section highlights some important requirements.

PCI signals coming from the motherboard onto the expansion card must be limited to only one load. This includes the primary clock. These are the signals on the primary interface of the 21153. These signals also have trace length limitations, which are 1.5 inches for PCI signals and 2.5 inches for the primary clock.

PCI signals on the secondary side of the 21153 do not need to adhere to these restrictive loading and trace length requirements. The secondary PCI bus can support the full 10 loads (including the 21153) and can be treated like a motherboard PCI bus.

6.3 Pull-Ups

Pull-up resistors are required on the following shared PCI control signals: FRAME#, IRDY#, TRDY#, STOP#, DEVSEL#, PERR#, SERR#, and LOCK#. These signals must have pull-ups on both the primary and secondary PCI buses, and they should be pulled up to **p_vio** or **s_vio** voltages. When optional shared signals such as LOCK# are not used, they must be pulled up for proper 21153 operation.

For resistor values, use the formula specified in Section 4.3.3 of the *PCI Local Bus Specification, Revision 2.1*. In this formula, use a value of 1.5 mA for Iol (per the specifications shown in the *21153 PCI-to-PCI Bridge Data Sheet*) and the appropriate value for **Vcc**.

The 21153-AB has some power management features. If power management is required, the signal **bpcc** should be pulled high. If power management is not used this pin can be tied to **Vss**. In the 21153-AA the **bpcc** pin was Vss. The 21153-AB can be used in applications calling for the 21153-AA without changes to the application hardware.

When an external arbiter is used, or if all of the **s_req_l** lines are not used, all of the unused **s_req_l** lines should have pull-up resistors.

If the **gpio** interface is not being used, **gpio<3:0>** must have pull-up resistors.

If boundary scan is not implemented on the motherboard, **tds** and **tdi** should be independently bused and individually pulled up with ~5-KΩ resistors. The **trst_l** and **tck** signals should be independently bused and pulled down, each with ~5-KΩ resistors. The **tdo** signal should be left open.

See special requirements for boundary scan on option cards in the *PCI Local Bus Specification, Revision 2.1*.

If the 21153 is implemented on a motherboard, both primary and secondary bus pull-ups should be located on the motherboard. When the 21153 is being used on an option card with the primary bus interfacing to the card edge, primary bus pull-ups must be located on the motherboard and secondary bus pull-ups must be located on the option card. In addition, **s_req_l<3:0>** must be pulled up to the **s_vio** voltage. The recommended value for the pull-up resistors is 5 KΩ.

6.4 Decoupling

According to Section 4.4.2.1 of the *PCI Local Bus Specification, Revision 2.1*:

“Under typical conditions, the **Vcc** plane to ground plane capacitance will provide adequate decoupling for the **Vcc** connector pins. The maximum trace length from a connector pad to the **Vcc**/**Gnd** plane is 0.25 inches, assuming a 20 mil trace width.

However, on the Universal board, it is likely that the I/O buffer power rail will not provide adequate capacitance to the ground plane to provide the necessary decoupling. Pins labeled “+V i/o” should be decoupled to ground with an average of 0.047 µF per pin.

Additionally, all 3.3 V pins (even if they are not actually delivering power), and any unused 5 V and V I/O pins on the PCI edge connector provide an ac return path. These pins must have plated edge fingers and be decoupled to the ground plane on the add-in board to ensure they continue to function as efficient ac reference points:

- The decoupling must average at least 0.0 µF (high-speed) per **Vcc** pin.
- The trace length from pin pad to capacitor pad shall be no greater than 0.25 inches using a trace width of at least 0.02 inches.
- There is no limit to the number of pins that can share the same capacitor provided that the previous two requirements are met.”

6.5 FCC Considerations

To minimize electromagnetic interference (EMI) and to control signal integrity characteristics, follow the suggestions in Section 4.4.3 of the *PCI Local Bus Specification, Revision 2.1*. This section covers trace lengths and routing of signals. Although not an optimal solution where EMI and FCC compliance are concerned, four-layer boards are recommended as a midrange solution. When four-layer boards are used, Intel recommends that you use the following layout guidelines, as per the *PCI Local Bus Specification, Revision 2.1*:

“…arrange the signal level layouts so that no high speed signal (e.g., 33 MHz) is referenced to both planes. Signal traces should either remain entirely over the 3.3 V plane or entirely over the 5 V plane. Signals that must cross from one domain to the other should be routed on the opposite side of the board so that they are referenced to the ground plane which is not split. If this is not possible, and signals must be routed over the plane split, the two planes should be capacitively tied together (5 V plane decoupled directly to 3.3 V plane) with 0.01 µF high-speed capacitors for each four signals crossing the split and the capacitor should be placed not more than 0.25 inches from the point the signals cross the split."

6.6 Additional Board Layout Guidelines

The following list contains some additional board layout guidelines:

1. Avoid signals crossing over a split in the ground, power, or both because they will contribute to noise problems.
2. Decoupling for high frequencies:
    a. Add one high frequency decoupling capacitor per power pin where possible. To minimize inductance, Intel recommends using 0805 or 1206 style surface-mount 0.001 µF capacitors.

    b. The maximum trace length from a connect pad to the **Vcc**/**Gnd** plane is 0.25 inches, assuming a 0.02 inch trace width.

    There is no limit to the number of pins that can share the same capacitor provided that the previous two requirements are met.

    Locate each capacitor as close to the pin as possible. Any etch length that is added at this path is inductive and will cause oscillations. A preferred method of adding decoupling capacitors when the board is crowded is to extend and merge the **Vdd** pads of the device with the capacitor pad and put a power via in the pad. Form a similar connection for ground.

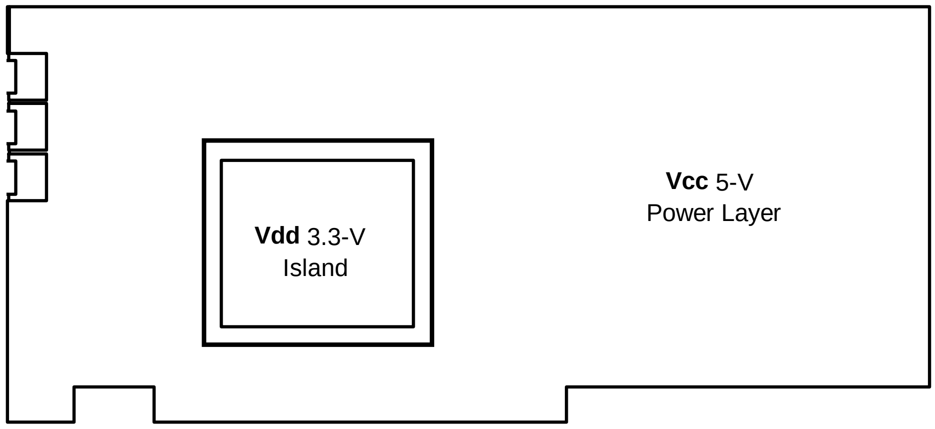
    For effective decoupling, it may be necessary to place components on the reverse side of the board.

    On the Universal board, it is likely that the I/O buffer power will not provide adequate capacitance to the ground plane to provide the necessary decoupling. Pins labeled "+V i/o" (**v_pio** and **v_sio**) should be decoupled to ground with an average of 0.047 µF per pin.

    Additionally, all 3.3 V pins (even if they are not actually delivering power) and any unused 5 V and V I/O pins on the PCI edge connector provide an ac return path. These pins must have plated edge fingers and must be decoupled to the ground plane on the add-in board to ensure that they continue to function as efficient ac reference points.
3. Add one bulk decoupling capacitor per device. The size of the bulk decoupling capacitor should be greater than the total capacitance being charged and discharged. Intel recommends using 22 µF 1812 style surface-mount capacitor.
4. On six-layer boards and higher, assign power and ground layer as close as possible. This will provide a large decoupling capacitance and will have greater power filtering effect.
5. Bury clocks on internal signal layers. Add a guard signal adjacent to each clock line and terminate the guard signal at both ends.
6. To prevent noise generated by the 21153 from coupling into other components, form a voltage island around the chip. This can be done easily for the chip that requires a voltage supply other than 5 V.

Figure 5 shows Intel's implementation of a voltage island.

Figure 5. 3.3-Voltage Island



7. Logic ground of the board should never be directly connected to chassis ground. Normally chassis ground will be referenced to the main power supply in the system. The bracket of the option card will be connected to that ground through chassis.

7.0 Secondary IDSEL Mapping

The *PCI Local Bus Specification*, *Revision 2.1* defines two formats for configuration transactions in hierarchical systems:

- Type 0 configuration transactions are used to configure devices on the same PCI bus.
- Type 1 configuration transactions are used to configure devices that reside on a downstream PCI bus.

When the 21153 detects a Type 1 configuration transaction for a device connected to the secondary interface of the bridge, the 21153 translates the Type 1 transaction into a Type 0 transaction on the downstream interface.

Instead of using IDSEL to identify the target of a configuration transaction, the Type 1 configuration format uses a 5-bit field at bits <15:11> in the address as a device number. A device number in the Type 1 format is translated by the 21153 into an IDSEL line for Type 0 transactions on the target interface.

As required by the *PCI-to-PCI Bridge Architecture Specification, Revision 1.0*, the 21153 uses **s_ad<31:16>** as secondary IDSEL lines. The 21153 uses the mapping shown in for translation of a device number to an **s_ad** pin.

The 21153 supports mapping for device numbers 0h through Fh. Configuration transactions with device numbers outside of this range are still forwarded, but no IDSEL mapping is performed (**s_ad<31:16>** are 00h).

Table 7. Device Number to Secondary IDSEL Mapping

Device Number	p_ad<15:11>	Secondary IDSEL s_ad<31:16>	s_ad Bit
0h	00000	0000 0000 0000 0001	16
1h	00001	0000 0000 0000 0010	17
2h	00010	0000 0000 0000 0100	18
3h	00011	0000 0000 0000 1000	19
4h	00100	0000 0000 0001 0000	20
5h	00101	0000 0000 0010 0000	21
6h	00110	0000 0000 0100 0000	22
7h	00111	0000 0000 1000 0000	23
8h	01000	0000 0001 0000 0000	24
9h	01001	0000 0010 0000 0000	25
Ah	01010	0000 0100 0000 0000	26
Bh	01011	0000 1000 0000 0000	27
Ch	01100	0001 0000 0000 0000	28
Dh	01101	0010 0000 0000 0000	29
Eh	01110	0100 0000 0000 0000	30
Fh	01111	1000 0000 0000 0000	31
10h–1Eh	10000–11110	0000 0000 0000 0000	–
1Fh	11111	Generate special cycle (**p_ad<7:2>**=00h) 0000 0000 0000 0000 (**p_ad<7:2>**(00h)	–

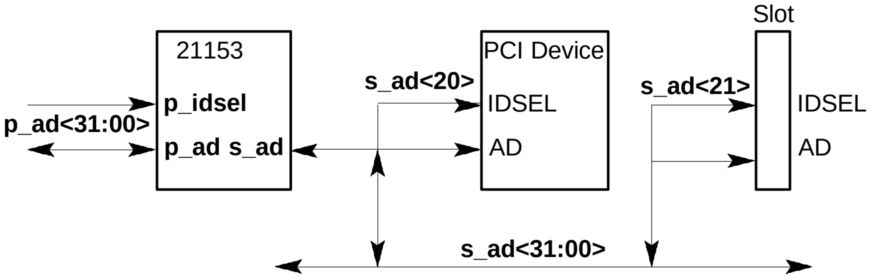
Note: Previous versions of Intel’s PCI-to-PCI bridge chips that implemented address stepping required a 1-KΩ resistor between the AD# input and the IDSEL# input of the PCI device on the secondary bus. This is no longer a requirement.

When implementing the 21153 in a system, each PCI device and PCI slot connected to the secondary PCI bus must have its IDSEL line connected to one of the **s_ad**<**31:16**> lines, as shown in.

When you assign secondary AD signals to secondary IDSEL signals on an expansion card, make sure that the device numbers are consistent with the required interrupt binding given in .

Note: Some early PCI host bridges automatically claim all configuration commands directed to device 0. Therefore, Intel recommends that you avoid using device 0.

Figure 6. Secondary IDSEL Implementation Example



8.0 Interrupts

The following section describes how interrupts should be implemented in a 21153 application.

8.1 Data Synchronization and Interrupts

The *PCI Local Bus Specification, Revision 2.1* requires that either the interrupt handler (service routine) or the device that initiates the interrupt guarantees that all buffers are flushed between the device and the final destination. To accomplish this, the interrupt service routine of the device driver can perform a read of the device, or the device itself can perform a read of the last location written by the device. In either case, the read transaction forces buffers between the device and the final destination to be flushed.

Section 6.3.4 of the *PCI Local Bus Specification, Revision 2.1* states, “Device drivers are ultimately responsible for guaranteeing consistency of interrupts and data.”

Interrupts originating from secondary bus devices are not routed through the 21153.

9.0 Interrupt Binding on Option Cards

When PCI-to-PCI bridges are used in option cards, a binding is required by the *PCI-to-PCI Bridge Architecture Specification, Revision 1.0* between the device number (as given in the Type 1 configuration address and, therefore, the IDSEL line) and the INTx# line it uses when requesting an interrupt.

The PCI connector has only four interrupt lines assigned to it: INTA#, INTB#, INTC#, and INTD#. Multiple devices might have to share these lines. Refer to Table 8.

Because only the BIOS knows how the PCI INTx# lines are routed to the system interrupt controller, a mechanism is required to inform the device driver which IRQ its device will request an interrupt on. The interrupt line register stores this information.

The BIOS code assumes the binding is as listed in behind a PCI-to-PCI bridge and writes the IRQ number in each device. The interrupt binding is mandatory for option cards using PCI-to-PCI bridges.

Table 8. Interrupt Binding on Option Cards

Device Number on Secondary Bus	Interrupt Pin on Device	Interrupt Pin on Connector
0, 4, 8, 12, 16, 20, 24, 28	INTA# INTB# INTC# INTD#	INTA# INTB# INTC# INTD#
1, 5, 9, 13, 17, 21, 25, 29	INTA# INTB# INTC# INTD#	INTB# INTC# INTD# INTA#
2, 6, 10, 14, 18, 22, 26, 30	INTA# INTB# INTC# INTD#	INTC# INTD# INTA# INTB#
3, 7, 11, 15, 19, 23, 27, 31	INTA# INTB# INTC# INTD#	INTD# INTA# INTB# INTC#

10.0 Arbitration

The 21153 must arbitrate for use of the primary bus when forwarding upstream transactions, and for use of the secondary bus when forwarding downstream transactions. The arbiter for the primary bus resides external to the 21153, typically on the motherboard.

For the secondary PCI bus, the 21153 implements an internal arbiter. This arbiter can be disabled, and an external arbiter can be used instead.

The following sections describe primary and secondary bus arbitration.

10.1 Primary Bus Arbitration

The 21153 implements a request output pin, **p_req_l**, and a grant input pin, **p_gnt_l**, for primary PCI bus arbitration. The 21153 asserts **p_req_l** when forwarding transactions upstream; that is, it acts as initiator on the primary PCI bus.

As long as at least one pending transaction resides in the queues in the upstream direction, either posted write data or delayed transaction requests, the 21153 keeps **p_req_l** asserted. However, if a target retry, target disconnect, or a target abort is received in response to a transaction initiated by the 21153 on the primary PCI bus, the 21153 deasserts **p_req_l** for two PCI clock cycles.

For additional information on primary bus cycle arbitration, refer to the *21153 PCI-to-PCI Bridge Data Sheet.*

10.2 Secondary Bus Arbitration

The 21153 implements an internal secondary PCI bus arbiter. This arbiter supports nine external masters in addition to the 21153. The internal arbiter can be disabled, and an external arbiter can be used instead for secondary bus arbitration.

10.2.1 Secondary Bus Arbitration Using the Internal Arbiter

To use the 21153 secondary bus arbiter:

- Tie the enable pin, **s_cfn_l**, to ground (low) through a 1-KΩ resistor.
- Connect the nine secondary bus request input pins, **s_req_l<8:0>**, and the nine secondary bus grant output pins, **s_gnt_l<8:0>**, to external secondary bus masters as required.

Note: The 21153 secondary bus request and grant signals are connected internally to the arbiter and are not brought out to external pins when **s_cfn_l** is low.

The internal secondary arbiter supports a programmable two-level rotating algorithm. Two groups of masters are assigned, a high-priority group and a low-priority group. The low-priority group as a whole represents one entry in the high-priority group; that is, if the high-priority group consists of *n* masters, then in at least every *n*+1 transaction the highest priority is assigned to the low-priority group. Priority rotates evenly among the low-priority group.

If the 21153 detects that an initiator has failed to assert **s_frame_l** after 16 cycles of both grant assertion and a secondary idle bus condition, the arbiter deasserts the grant. That master does not receive any more grants until it deasserts its request for a least one PCI clock cycle.

To prevent bus contention, if the secondary PCI bus is idle, the arbiter never asserts one grant signal in the same PCI cycle in which it deasserts another. It deasserts one grant, and then asserts the next grant, no earlier than one PCI clock cycle later. If the secondary PCI bus is busy, that is, if either **s_frame_l** or **s_irdy_l** is asserted, the arbiter can deassert one grant and assert another grant during the same PCI clock cycle.

10.2.2 Secondary Bus Arbitration Using an External Arbiter

To use an external arbiter, tie the central function enable pin, **s_cfn_l**, high through a 1-KΩ resistor. This disables the internal arbiter.

With **s_cfn_l** tied high, the 21153 reconfigures two pins to be external request and grant pins. The **s_gnt_l<0>** pin is reconfigured to be the 21153's external request pin because it is an output. Connect this pin to one of the request lines of the external arbiter. The **s_req_l<0>** pin is reconfigured to be the external grant pin because it is an input. Connect this pin to one of the grant lines of the external arbiter.

The unused secondary bus grant outputs, **s_gnt_l<8:1>**, are driven high. Unused secondary bus request inputs, **s_req_l<8:1>**, should be pulled high using 1-KΩ resistors.

10.2.3 Bus Parking

Bus parking refers to driving the AD, C/BE#, and PAR lines to a known value while the bus is idle. In general, the device implementing the bus arbiter is responsible for parking the bus or assigning another device to park the bus. A device parks the bus when the bus is idle, its bus grant is asserted, and the device's request is not asserted. The AD and C/BE# signals should be driven first, with the PAR signal driven one cycle later.

The 21153 parks the primary bus only when **p_gnt_l** is asserted, **p_req_l** is deasserted, and the primary PCI bus is idle. When **p_gnt_l** is deasserted, the 21153 tristates the **p_ad**, **p_cbe_l**, and **p_par** signals on the next PCI clock cycle. If the 21153 is parking the primary PCI bus and wants to initiate a transaction on that bus, then the 21153 can start the transaction on the next PCI clock cycle by asserting **p_frame_l** if **p_gnt_l** is still asserted.

If the internal secondary bus arbiter is enabled, the secondary bus is always parked at the last master that used the PCI bus. That is, the 21153 keeps the secondary bus grant asserted to a particular master until a new secondary bus request comes along. After reset, the 21153 parks the secondary bus at itself until transactions start occurring on the secondary bus. If the internal arbiter is disabled, the 21153 parks the secondary bus only when the reconfigured grant signal, **s_req_l<0>,** is asserted and the secondary bus is idle.

11.0 Live Insertion

The **gpio<3>** pin can be used with a live insertion mode bit to disable transaction forwarding.

To enable live insertion mode, the live insertion mode bit in the chip control register must be set to 1. The output enable control for **gpio<3>** must be set to input only in the **gpio** output enable control register. When live insertion mode is enabled and whenever **gpio<3>** is driven to a value of 1, the I/O enable, the memory enable, and the master enable bits are internally masked to 0. This means that the 21153 no longer accepts any I/O or memory transactions as a target on either the primary or secondary bus.

When read, the command register still reflects the value originally written by a configuration write command. When **gpio<3>** is deasserted, the internal enable bits return to their original value (as the bits appear when read from the command register). When live insertion mode is enabled and **gpio<3>** is asserted, the 21153 completes any cycles as master, such as posted write or delayed request transactions that have already been queued.

Delayed completion transactions are not returned to the master in this mode because the 21153 is not responding as a target to any I/O or any memory transactions during this time.

In this mode, the internal arbiter continues to operate and will assert GNT# in response to a REQ# on the secondary bus.

Support, Products, and Documentation

If you need general information or support, call **1-800-628-8686** or visit Intel’s website at:

http://www.intel.com

Copies of documents that have an ordering number and are referenced in this document, a product catalog, or other Intel literature may be obtained by calling **1-800-548-4725** or by visiting Intel’s website for developers at:

http://developer.intel.com