21554 PCI-to-PCI Bridge for Embedded Applications

Product Preview Datasheet

Product Features

- Full compliance with the *PCI Local Bus Specification, Revision 2.1*
- 3.3 V operation with 5 V-tolerant I/O
- Selectable asynchronous or synchronous primary and secondary interface clocks
- Concurrent primary and secondary bus operation
- Queuing of multiple transactions in either direction
- 256 bytes of posted write (data and address) buffering in each direction
- 256 bytes of read data buffering in each direction
- Four delayed transaction entries in each direction
- Two dedicated I_2O delayed transaction entries
- Two sets of standard PCI configuration registers corresponding to the primary and secondary interface; each set is accessible from either the primary or secondary interface
- Memory and I/O mapping of 21554 CSRs on both the primary and secondary interface
- Four downstream and three upstream address ranges, with programmable size and prefetchability
- Look-up table based address translation for one upstream range; direct offset address translation for all other forwarding ranges
- Inverse decoding above the 4 GB address boundary for upstream DACs
- Ability to generate Type 0 and Type 1 configuration transactions on the primary or secondary interface via configuration or I/O CSR accesses
- Ability to generate I/O transactions on the primary or secondary interface via I/O CSR
- I_2O message unit
- Doorbell registers for software generation of primary and secondary bus interrupts, 16 bits per interface
- Eight Dwords of scratchpad registers
- Parallel ROM interface with primary bus expansion ROM base address register
- Serial ROM interface
- Secondary bus arbiter support for up to nine devices (in addition to the 21554)
- CompactPCI® Hot Swap Controller
- Configurable PCI power management support
- IEEE Standard 1149.1 boundary-scan JTAG interface

Notice: This document contains information on products in the design phase of development. Do not finalize a design with this information. Revised information will be published when the product is available. Verify with your local Intel sales office that you have the latest datasheet before finalizing a design.

Order Number: 278089-001
Dec 1998

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 21554 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 data sheet describes the 21554 PCI-to-PCI bridge chip for embedded applications. For a detailed functional and register description, see the *21554 PCI-to-PCI Bridge for Embedded Applications Hardware Reference Manual*.

1.1 General Information

The 21554 is a PCI peripheral chip that performs PCI bridging functions for embedded and intelligent I/O applications. The 21554 is a “non-transparent” PCI-to-PCI bridge that acts as a gateway to an intelligent subsystem. It allows a local processor to independently configure and control the local subsystem. The 21554 implements an I_2O message unit that enables any local processor to function as an intelligent I/O processor (IOP) in an I_2O-capable system. Because the 21554 is architecture independent, it works with any host and local processors that support a PCI bus. This architecture independence enables vendors to leverage existing investments while moving products to PCI technology.

Unlike a transparent PCI-to-PCI bridge, the 21554 is specifically designed to bridge between two processor domains. The processor domain on the primary interface of the 21554 is also referred to as the host domain, and its processor is the host processor. The secondary bus interfaces to the local domain and the local processor. Special features include support of independent primary and secondary PCI clocks, independent primary and secondary address spaces, and address translation between the primary (host) and secondary (local) domains.

The 21554 enables add-in card vendors to present to the host system a higher level of abstraction than is possible with a transparent PCI-to-PCI bridge. The 21554 uses a Type 0 configuration header, which presents the entire subsystem as a single “device” to the host processor. This allows loading of a single device driver for the entire subsystem, and independent local processor initialization and control of the subsystem devices. Because the 21554 uses a Type 0 configuration header, it does not require hierarchical PCI-to-PCI bridge configuration code.

The 21554 forwards transactions between the primary and secondary PCI buses as does a transparent PCI-to-PCI bridge. In contrast to a transparent PCI-to-PCI bridge, however, the 21554 can translate the address of a forwarded transaction from a system address to a local address, or vice versa. This mechanism allows the 21554 to hide subsystem resources from the host processor and to resolve any resource conflicts that may exist between the host and local subsystems.

The 21554 operates at 3.3 V, but is also 5.0 V I/O tolerant. Adapter cards designed using the 21554 can be keyed as universal, thus permitting use in either a 5 V or 3 V slot.

The 21554 supports a 64-bit primary PCI bus, a 64-bit secondary PCI bus, and a maximum operating frequency of 33 MHz.

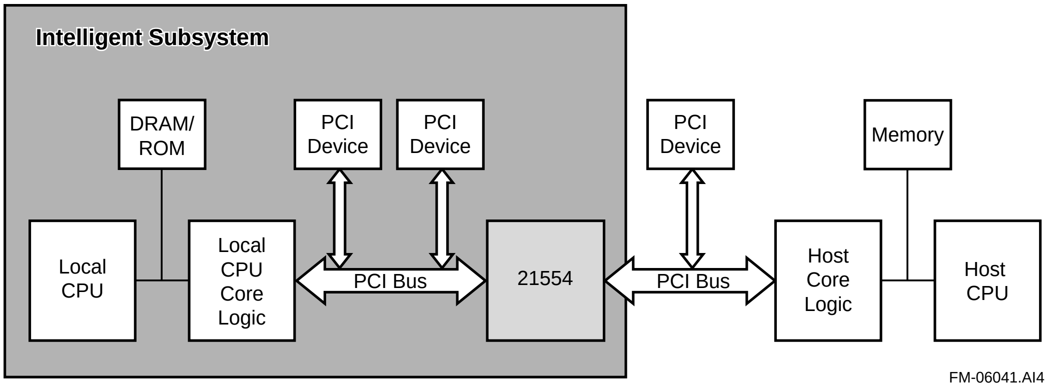
1.2 Comparison of 21554 and Standard PCI-to-PCI Bridge

The 21554 is functionally similar to a standard PCI-to-PCI bridge (PPB) in that both provide a connection path between devices attached to two independent PCI buses. A 21554 and a PPB allow the electrical loading of devices on one PCI bus to be isolated from the other bus while permitting concurrent operation on both buses. Because the *PCI Local Bus Specification* restricts PCI option cards to a single electrical load, the ability of PPBs and the 21554 to spawn PCI buses enables the design of multidevice PCI option cards. The key difference between a PPB and the 21554 is that the presence of a PPB in a connection path between the host processor and a device is transparent to devices and device drivers, while the presence of the 21554 is not. This difference enables the 21554 to provide features that better support the use of intelligent controllers in the subsystem.

It was a primary goal of the PCI-to-PCI bridge architecture that a PPB be transparent to devices and device drivers. For example, no changes are needed to a device driver when a PCI peripheral is located behind a PPB. Once configured during system initialization, a PPB operates without the aid of a device driver. A PPB does not require a device driver of its own since it does not have any resources that must be managed by software during run-time. This requirement for transparency forced the usage of a flat addressing model across PCI-to-PCI bridges. This means that a given physical address exists at only one location in the PCI bus hierarchy and that this location may be accessed by any device attached at any point in the PCI bus hierarchy. As a consequence, it is not possible for a PPB to isolate devices or address ranges from access by devices on the opposite interface of a PPB. The PPB architecture assumes that the resources of any device in a PCI system are configured and managed by the host processor.

However, there are applications where the transparency of a PCI-to-PCI bridge is not desired. For example, Figure 1 shows a hypothetical PCI add-in card used for an intelligent subsystem application.

Figure 1. 21554 Intelligent Controller Application



Assume that the local processor on the add-in card is used to manage the resources of the devices attached to the add-in card’s local PCI bus. Assume also that it is desirable to restrict access to these same resources from other PCI bus masters in the system and from the host processor. In addition, there is a need to resolve address conflicts that may exist between the host system and the local processor. The nontransparency of the 21554 is perfectly suited to this kind of configuration, where a transparent PCI-to-PCI bridge would be problematic.

Because the 21554 is not transparent, the device driver for the add-in card must be aware of the presence of the 21554 and manage its resources appropriately. The 21554 allows the entire subsystem to appear as a single virtual device to the host. This enables configuration software to identify the appropriate driver for the subsystem.

With a transparent PCI-to-PCI bridge, a driver does not need to know about the presence of the bridge and manage its resources. The subsystem appears to the host system as individual PCI devices on a secondary PCI bus, not as a single virtual device.

Table 1 shows a comparison between a 21554 and a standard transparent PCI-to-PCI bridge.

Table 1. 21554 and PPB Feature Comparison

Feature	21554	PCI-to-PCI Bridge
Transaction forwarding	Adheres to PPB ordering rules. Uses posted writes and delayed transactions. Adheres to PPB transaction error and parity error guidelines, although some errors may be reported differently.	Adheres to PPB ordering rules. Uses posted writes and delayed transactions. Adheres to PPB transaction error and parity error guidelines.
Address decoding	Base address registers are used to define both downstream and upstream forwarding windows. Inverse decoding for upstream transactions is only used above the 4GB boundary.	PPB base and limit address registers are used to define downstream forwarding windows. Inverse decoding for all I/O and memory upstream forwarding.
Address translation	Supported for both memory and I/O transactions.	None. Flat address model is assumed.
Configuration	Downstream devices are not visible to host. Does not require hierarchical configuration code (Type 0 configuration header). Does not respond to Type 1 configuration transactions. Supports configuration access from the secondary bus. Implements separate set of configuration registers for the secondary interface.	Downstream devices are visible to host. Requires hierarchical configuration code (Type 1 configuration header). Forwards and converts Type 1 configuration transactions. Does not support configuration access from the secondary bus. Same set of configuration registers is used to control both primary and secondary interfaces.
Run-time resources	Includes features such as doorbell interrupts, I_2O message unit, and so on, that must be managed by the device driver.	Typically has only configuration registers; no device driver is required.
Clocks	Generates secondary bus clock output. Asynchronous secondary clock input is also supported.	Generates one or more secondary bus clock outputs.
Secondary bus central functions	Implements secondary bus arbiter. This function can be disabled. Drives secondary bus AD, C/BE#, and PAR during reset. This function can be disabled.	Implements secondary bus arbiter. This function can be disabled. Drives secondary bus AD, C/BE#, and PAR during reset.

1.3 Architectural Overview

The 21554 consists of the following function blocks:

1.3.1 Data Buffers

Data buffers include the buffers along with the associated data path control logic. Delayed transaction buffers contain the compare functionality for completing delayed transactions. The blocks also contain the watchdog timers associated with the buffers. The data buffers are as follows:

- Four-entry downstream delayed transaction buffer
- Four-entry upstream delayed transaction buffer
- 256-byte downstream posted write buffer
- 256-byte upstream posted write buffer
- 256-byte downstream read data buffer
- 256-byte upstream read data buffer
- Two downstream I_2O delayed transaction entries

1.3.1.1 Registers

The following register blocks also contain address decode and translation logic, I_2O message unit, and interrupt control logic:

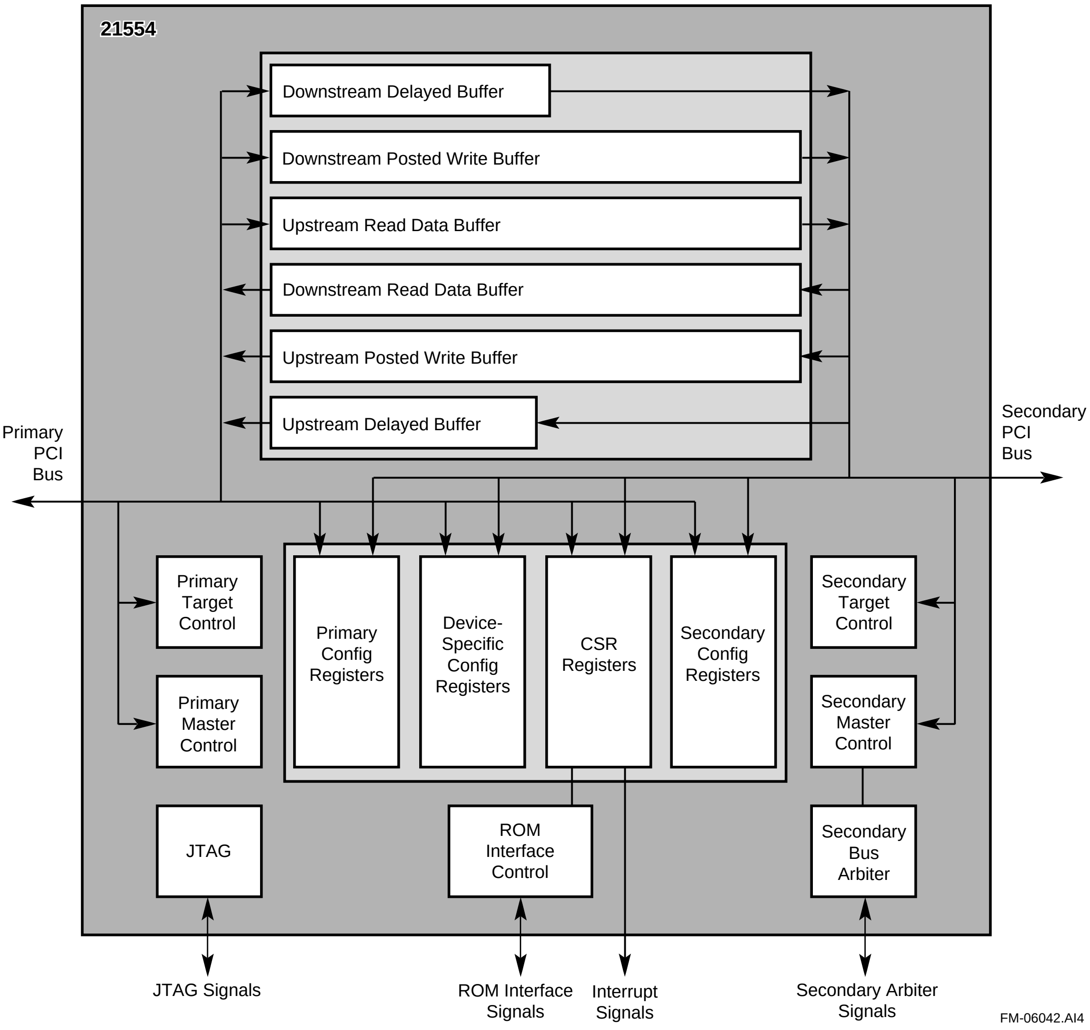
- Primary interface header Type 0 configuration registers
- Secondary interface header Type 0 configuration registers
- Device-specific configuration registers
- Memory and I/O mapped control and status registers

1.3.2 Control Logic

- Primary PCI target control logic
- Primary PCI master control logic
- Secondary PCI target control logic
- Secondary PCI master control logic
- ROM interface control logic for both serial and parallel ROM connections (interfaces between the ROM registers and ROM signals)
- Secondary PCI bus arbiter interface to secondary bus device request and grant lines, as well as the 21554 secondary master control logic
- JTAG control logic

Figure 2 shows the 21554 microarchitecture.

Figure 2. 21554 Microarchitecture



1.4 Conventions and Terminology

This section describes the terminology and conventions used in this datasheet.

1.4.1 Caution

Cautions provide information to prevent damage to equipment or loss of data.

1.4.2 Data Units

This manual uses the following data-unit terminology.

Table 2. Data-unit Terminology

Term	Words	Bytes	Bits
Byte	½	1	8
Word	1	2	16
Dword	2	4	32
Quadword	4	8	64

1.4.3 Note

Notes emphasize particularly important information.

1.4.4 Numbering

All numbers are decimal or hexadecimal unless otherwise indicated. In cases of ambiguity, a subscript indicates the radix of nondecimal numbers. For example, 19 is decimal, but 19h and 19A are hexadecimal.

1.4.5 Signal Names

Signal names are printed in lowercase, boldface type.

Signal names indicate whether a signal is low-asserted (the signal is active, or asserted, when it is at a low voltage level) or high-asserted (the signal is asserted when it is at a high voltage level). The names of low-asserted signals carry the suffix _l; the names of high-asserted signals have no suffix. For example, p_idsel is a high-asserted signal, and p_frame_l is a low-asserted signal.

The prefix p_ denotes a primary bus signal; the prefix s_ denotes a secondary bus signal. For example, p_ad is primary interface address/data bus, and s_ad is secondary interface address/data bus.

1.4.6 SIGNAME#

PCI signals that can be on either the primary interface or the secondary interface are printed in uppercase, normal type. The names of low-asserted signals are followed by #. For example, “asserting FRAME#” can refer to the assertion of the p_frame_l signal if the transaction is occurring on the primary bus, or the assertion of the s_frame_l signal if the transaction is occurring on the secondary bus.

1.5 Manual Organization

This datasheet contains the following chapters and an appendix:

- Chapter 1.0, “Introduction”, overview of the 21554 functionality and architecture.
- Chapter 2.0, “Signal Pins”, describes the signal pins.
- Chapter 3.0, “Pin Assignment”, describes the pin assignments by location and by signal name.
- Chapter 4.0, “JTAG Boundary Scan”, explains the 21554’s JTAG test port.
- Chapter 5.0, “Electrical Specifications”, describes 21554 AC and DC electrical specifications.
- Chapter 6.0, “Mechanical Specifications”, describes the 21554 mechanical specifications.
- “”, contains technical support and ordering information.

2.0 Signal Pins

This chapter provides detailed descriptions of the 21554 signal pins, grouped by function. Table 3 describes these signal pin functional groups.

Table 3. Signal Pin Function Groups

Function	Description
Primary PCI bus interface signal pins	All PCI pins required by the *PCI Local Bus Specification*, Revision 2.1.
Primary PCI bus interface 64-bit extension signal pins	All PCI 64-bit extension pins required by the *PCI Local Bus Specification*, Revision 2.1.
Secondary PCI bus interface signal pins	All PCI pins required by the *PCI Local Bus Specification*, Revision 2.1.
Secondary PCI bus interface 64-bit extension signal pins	All PCI 64-bit extension pins required by the *PCI Local Bus Specification*, Revision 2.1.
Secondary PCI bus arbitration signal pins	Nine request/grant pairs of pins for the secondary PCI bus.
Clock signal pins	Two clock inputs (one for each PCI interface). One secondary bus clock output.
Power Management, Hot Swap, and Reset signal pins	Power management and hot-swap status and events. A primary interface reset input. A secondary interface reset output.
ROM interface signal pins	8-bit multiplexed address/data bus plus control for parallel and serial ROM connection.
Miscellaneous signal pins	Two input voltage signaling level pins.
Diagnostic signal pins	IEEE Standard 1149.1 boundary-scan JTAG interface. Scan enable.

Table 4 defines the signal type abbreviations used in the signal tables.

Table 4. Signal Type Abbreviations

Signal Type	Description
I	Standard input only.
O	Standard output only.
TS	Tristate bidirectional.
STS	Sustained tristate. Active low signal must be pulled high for one clock cycle when deasserting.
OD	Standard open drain.

Note: The _l signal name suffix indicates that the signal is asserted when it is at a low voltage level and corresponds to the # suffix in the *PCI Local Bus Specification*, Revision 2.1. If this suffix is not present, the signal is asserted when it is at a high voltage level.

Table 5 describes the primary PCI bus interface signals.

Table 5. Primary PCI Bus Interface Signals (Sheet 1 of 2)

Signal Name	Type	Description
p_ad[31:0]	TS	**Primary PCI interface address/data.** These signals are a 32-bit multiplexed address and data bus. During the address phase or phases of a transaction, the initiator drives a physical address on p_ad[31:0]. During the data phases of a transaction, the initiator drives write data, or the target drives read data, on p_ad[31:0]. When the primary PCI bus is idle, the 21554 drives p_ad to a valid logic level when p_gnt_l is asserted.
p_cbe_l[3:0]	TS	**Primary PCI interface command/byte enables.** These signals are a multiplexed command field and byte enable field. During the address phase or phases of a transaction, the initiator drives the transaction type on p_cbe_l[3:0]. When there are two address phases, the first address phase carries the dual-address command and the second address phase carries the transaction type. For both read and write transactions, the initiator drives byte enables on p_cbe_l[3:0] during the data phases. When the primary PCI bus is idle, the 21554 drives p_cbe_l to a valid logic level when p_gnt_l is asserted.
p_devsel_l	STS	**Primary PCI interface DEVSEL#**. Signal p_devsel_l is asserted by the target, indicating that the device is responding to the transaction. As a target, the 21554 decodes the address of a transaction initiated on the primary bus to determine whether to assert p_devsel_l. As an initiator of a transaction on the primary bus, the 21554 looks for the assertion of p_devsel_l within five clock cycles of p_frame_l assertion; otherwise, the 21554 terminates the transaction with a master abort. Upon completion of a transaction, p_devsel_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.
p_frame_l	STS	**Primary PCI interface FRAME#.** Signal p_frame_l is driven by the initiator of a transaction to indicate the beginning and duration of an access on the primary PCI bus. Signal p_frame_l assertion (falling edge) indicates the beginning of a PCI transaction. While p_frame_l remains asserted, data transfers can continue. The deassertion of p_frame_l indicates the final data phase requested by the initiator. Upon completion of a transaction, p_frame_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.
p_gnt_l	I	**Primary PCI bus GNT#**. When asserted, p_gnt_l indicates to the 21554 that access to the primary bus is granted. The 21554 can start a transaction on the primary bus when the bus is idle and p_gnt_l is asserted. When the 21554 has not requested use of the bus and p_gnt_l is asserted, the 21554 drives p_ad, p_cbe_l, and p_par to valid logic levels.
p_idsel	I	**Primary PCI interface IDSEL**. Signal p_idsel is used as the chip select line for Type 0 configuration accesses to 21554 configuration space from the primary bus. When p_idsel is asserted during the address phase of a Type 0 configuration transaction, the 21554 responds to the transaction by asserting p_devsel_l.
p_inta_l	OD	**Primary PCI bus interrupt**. Signal p_inta_l is asserted by the 21554 when: • A primary doorbell register bit is set. • The I_2O outbound queue is not empty. • The subsystem event bit is set. All of these conditions are individually maskable. When the corresponding event bit is cleared or the outbound queue is emptied, p_inta_l is deasserted. Signal p_inta_l is pulled up through an external resistor.
p_irdy_l	STS	**Primary PCI interface IRDY#**. Signal p_irdy_l is driven by the initiator of a transaction to indicate the initiator's ability to complete the current data phase on the primary PCI bus. During a write transaction, assertion of p_irdy_l indicates that valid write data is being driven on the p_ad bus. During a read transaction, assertion of p_irdy_l indicates that the initiator is able to accept read data for the current data phase. Once asserted during a given data phase, p_irdy_l is not deasserted until the data phase completes. Upon completion of a transaction, p_irdy_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.

Table 5. Primary PCI Bus Interface Signals (Sheet 2 of 2)

Signal Name	Type	Description
p_par	TS	**Primary PCI interface parity.** Signal p_par carries the even parity of the 36 bits of p_ad[31:0] and p_cbe_l[3:0] for both address and data phases. Signal p_par is driven by the same agent that drives the address (for address parity) or the data (for data parity). Signal p_par contains valid parity one clock cycle after the address is valid (indicated by assertion of p_frame_l), or one clock cycle after the data is valid (indicated by assertion of p_irdy_l for write transactions and p_trdy_l for read transactions). Signal p_par is tristated one clock cycle after the p_ad lines are tristated. The device receiving data samples p_par as an input to check for possible parity errors. When the primary PCI bus is idle, the 21554 drives p_par to a valid logic level when p_gnt_l is asserted (one clock cycle after the p_ad bus is parked).
p_perr_l	STS	**Primary PCI interface PERR#**. Signal p_perr_l is asserted when a data parity error is detected for data received on the primary interface. The timing of p_perr_l corresponds to p_par driven one clock cycle earlier, and p_ad and p_cbe_l driven two clock cycles earlier. Signal p_perr_l is asserted by the target during write transactions, and by the initiator during read transactions. Upon completion of a transaction, p_perr_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.
p_req_l	TS	**Primary PCI bus REQ#**. Signal p_req_l is asserted by the 21554 to indicate to the primary bus arbiter that it wants to start a transaction on the primary bus.
p_serr_l	OD	**Primary PCI interface SERR#**. Signal p_serr_l can be driven low by any device on the primary bus to indicate a system error condition. The 21554 can conditionally assert p_serr_l for the following reasons: • Primary bus address parity error • Downstream posted write data parity error on secondary bus • Master abort during downstream posted write transaction • Target abort during downstream posted write transaction • Downstream posted write transaction discarded • Downstream delayed write request discarded • Downstream delayed read request discarded • Downstream delayed transaction master timeout • Secondary bus s_serr_l assertion Signal p_serr_l is pulled up through an external resistor.
p_stop_l	STS	**Primary PCI interface STOP#**. Signal p_stop_l is driven by the target of a transaction, indicating that the target is requesting the initiator to stop the transaction on the primary bus. • When p_stop_l is asserted in conjunction with p_trdy_l and p_devsel_l assertion, a disconnect with data transfer is being signaled. • When p_stop_l and p_devsel_l are asserted, but p_trdy_l is deasserted, a target disconnect without data transfer is being signaled. When this occurs on the first data phase, that is, no data is transferred during the transaction, this is referred to as a target retry. • When p_stop_l is asserted and p_devsel_l is deasserted, the target is signaling a target abort. Upon completion of a transaction, p_stop_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.
p_trdy_l	STS	**Primary PCI interface TRDY#.** Signal p_trdy_l is driven by the target of a transaction to indicate the target's ability to complete the current data phase on the primary PCI bus. During a write transaction, assertion of p_trdy_l indicates that the target is able to accept write data for the current data phase. During a read transaction, assertion of p_trdy_l indicates that the target is driving valid read data on the p_ad bus. Once asserted during a given data phase, p_trdy_l is not deasserted until the data phase completes. Upon completion of a transaction, p_trdy_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.

2.1 Primary PCI Bus Interface 64-Bit Extension Signals

Table 6 describes the primary PCI bus interface 64-bit extension signals.

Table 6. Primary PCI Bus Interface 64-Bit Extension Signals

Signal Name	Type	Description
p_ack64_l	STS	**Primary PCI interface acknowledge 64-bit transfer.** Signal p_ack64_l is asserted by the target only when p_req64_l is asserted by the initiator, to indicate the target's ability to transfer data using 64 bits. Signal p_ack64_l has the same timing as p_devsel_l. When deasserting, p_ack64_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.
p_ad[63:32]	TS	**Primary PCI interface address/data upper 32 bits.** This multiplexed address and data bus provides an additional 32 bits to the primary interface. During the address phase or phases of a transaction, when the dual-address command is used and p_req64_l is asserted, the initiator drives the upper 32 bits of a 64-bit address; otherwise, these bits are undefined, and the initiator drives a valid logic level onto the pins. During the data phases of a transaction, the initiator drives the upper 32 bits of 64-bit write data, or the target drives the upper 32 bits of 64-bit read data, when p_req64_l and p_ack64_l are both asserted. When not driven, signals p_ad[63:32] are pulled up to a valid logic level through external resistors.
p_cbe_l[7:4]	TS	**Primary PCI interface command/byte enables upper 4 bits.** These signals are a multiplexed command field and byte enable field. During the address phase or phases of a transaction, when the dual-address command is used and p_req64_l is asserted, the initiator drives the transaction type on p_cbe_l[7:4]; otherwise, these bits are undefined, and the initiator drives a valid logic level onto the pins. For both read and write transactions, the initiator drives byte enables for the p_ad[63:32] data bits on p_cbe_l[7:4] during the data phases, when p_req64_l and p_ack64_l are both asserted. When not driven, signals p_cbe_l[7:4] are pulled up to a valid logic level through external resistors.
p_par64	TS	**Primary PCI interface upper 32 bits parity.** Signal p_par64 carries the even parity of the 36 bits of p_ad[63:32] and p_cbe_l[7:4] for both address and data phases. Signal p_par64 is driven by the initiator and is valid one clock cycle after the first address phase when a dual-address command is used and p_req64_l is asserted. Signal p_par64 is also valid one clock cycle after the second address phase of a dual-address transaction when p_req64_l is asserted. Signal p_par64_l is valid one clock cycle after valid data is driven (indicated by assertion of p_irdy_l for write data and p_trdy_l for read data), when both p_req64_l and p_ack64_l are asserted for that data phase. Signal p_par64 is tristated by the device driving read or write data one clock cycle after the p_ad lines are tristated. Devices receiving data sample p_par64 as an input to check for possible parity errors during 64-bit transactions. When not driven, p_par64 is pulled up to a valid logic level through external resistors.
p_req64_l	STS	**Primary PCI interface request 64-bit transfer.** Signal p_req64_l is asserted by the initiator to indicate that the initiator is requesting 64-bit data transfer. Signal p_req64_l has the same timing as p_frame_l. When deasserting, p_req64_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor. The 21554 samples p_req64_l during primary bus reset to enable the 64-bit extension signals. If p_req64_l is sampled high during reset, the primary 64-bit extension is disabled and assumed not connected. The 21554 then drives p_ad[63:32], p_cbe_l[7:4], and p_par64 to valid logic levels.

2.2 Secondary PCI Bus Interface Signals

Table 7 describes the secondary PCI bus interface signals.

Table 7. Secondary PCI Bus Interface Signals (Sheet 1 of 2)

Signal Name	Type	Description
s_ad[31:0]	TS	**Secondary PCI interface address/data**. These signals are a 32-bit multiplexed address and data bus. During the address phase or phases of a transaction, the initiator drives a physical address on s_ad[31:0]. During the data phases of a transaction, the initiator drives write data, or the target drives read data, on s_ad[31:0]. When the secondary PCI bus is idle, the 21554 drives s_ad to a valid logic level when its secondary bus grant is asserted.
s_cbe_l[3:0]	TS	**Secondary PCI interface command/byte enables**. These signals are a multiplexed command field and byte enable field. During the address phase or phases of a transaction, the initiator drives the transaction type on s_cbe_l[3:0]. When there are two address phases, the first address phase carries the dual-address command and the second address phase carries the transaction type. For both read and write transactions, the initiator drives byte enables on s_cbe_l[3:0] during the data phases. When the secondary PCI bus is idle, the 21554 drives s_cbe_l to a valid logic level when its secondary bus grant is asserted.
s_devsel_l	STS	**Secondary PCI interface DEVSEL#**. Signal s_devsel_l is asserted by the target, indicating that the device is responding to the transaction. As a target, the 21554 decodes the address of a transaction initiated on the secondary bus to determine whether to assert s_devsel_l. As an initiator of a transaction on the secondary bus, the 21554 looks for the assertion of s_devsel_l within five clock cycles of s_frame_l assertion; otherwise, the 21554 terminates the transaction with a master abort. Upon completion of a transaction, s_devsel_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.
s_frame_l	STS	**Secondary PCI interface FRAME#**. Signal s_frame_l is driven by the initiator of a transaction to indicate the beginning and duration of an access on the secondary PCI bus. Signal s_frame_l assertion (falling edge) indicates the beginning of a PCI transaction. While s_frame_l remains asserted, data transfers can continue. The deassertion of s_frame_l indicates the final data phase requested by the initiator. Upon completion of a transaction, s_frame_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.
s_idsel	I	**Secondary PCI interface IDSEL**. Signal s_idsel is used as the chip select line for Type 0 configuration accesses to 21554 configuration space from the secondary bus. When s_idsel is asserted during the address phase of a Type 0 configuration transaction, the 21554 responds to the transaction by asserting s_devsel_l.
s_inta_l	OD	**Secondary PCI bus interrupt**. Signal s_inta_l is asserted by the 21554 when: • A secondary doorbell register bit is set. • The I_2O inbound queue is not empty. • A page boundary is reached when performing lookup table address translation. • The 21554 transitions from a D1 or D2 power state to a D0 power state. All of these conditions are individually maskable. Signal s_inta_l is deasserted when the corresponding event bit is cleared, or when the inbound queue is empty. Signal s_inta_l is pulled up through an external resistor.
s_irdy_l	STS	**Secondary PCI interface IRDY#**. Signal s_irdy_l is driven by the initiator of a transaction to indicate the initiator's ability to complete the current data phase on the secondary PCI bus. During a write transaction, assertion of s_irdy_l indicates that valid write data is being driven on the s_ad bus. During a read transaction, assertion of s_irdy_l indicates that the initiator is able to accept read data for the current data phase. Once asserted during a given data phase, s_irdy_l is not deasserted until the data phase completes. Upon completion of a transaction, s_irdy_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.

Table 7. Secondary PCI Bus Interface Signals (Sheet 2 of 2)

Signal Name	Type	Description
s_par	TS	**Secondary PCI interface parity**. Signal s_par carries the even parity of the 36 bits of s_ad[31:0] and s_cbe_l[3:0] for both address and data phases. Signal s_par is driven by the same agent that drives the address (for address parity) or the data (for data parity). Signal s_par contains valid parity one clock cycle after the address is valid (indicated by assertion of s_frame_l), or one clock cycle after the data is valid (indicated by assertion of s_irdy_l for write transactions and s_trdy_l for read transactions). Signal s_par is tristated one clock cycle after the s_ad lines are tristated. The device receiving data samples s_par as an input to check for possible parity errors. When the secondary PCI bus is idle, the 21554 drives s_par to a valid logic level when its secondary bus grant is asserted (one clock cycle after the s_ad bus is parked).
s_perr_l	STS	**Secondary PCI interface PERR#**. Signal s_perr_l is asserted when a data parity error is detected for data received on the secondary interface. The timing of s_perr_l corresponds to s_par driven one clock cycle earlier, and s_ad driven two clock cycles earlier. Signal s_perr_l is asserted by the target during write transactions, and by the initiator during read transactions. Upon completion of a transaction, s_perr_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.
s_serr_l	OD	**Secondary PCI interface SERR#**. Signal s_serr_l can be driven low by any device on the secondary bus to indicate a system error condition. The 21554 also samples s_serr_l as an input and conditionally forwards it to the primary bus on p_serr_l. The 21554 can conditionally assert s_serr_l for the following reasons: • Secondary bus address parity error • Upstream posted write data parity error on primary bus • Master abort during upstream posted write transaction • Target abort during upstream posted write transaction • Upstream posted write transaction discarded • Upstream delayed write request discarded • Upstream delayed read request discarded • Upstream delayed transaction master timeout Signal s_serr_l is pulled up through an external resistor.
s_stop_l	STS	**Secondary PCI interface STOP#**. Signal s_stop_l is driven by the target of a transaction, indicating that the target is requesting the initiator to stop the transaction on the secondary bus. • When s_stop_l is asserted in conjunction with s_trdy_l and s_devsel_l assertion, a disconnect with data transfer is being signaled. • When s_stop_l and s_devsel_l are asserted, but s_trdy_l is deasserted, a target disconnect without data transfer is being signaled. When this occurs on the first data phase, that is, no data is transferred during the transaction, this is referred to as a target retry. • When s_stop_l is asserted and s_devsel_l is deasserted, the target is signaling a target abort. Upon completion of a transaction, s_stop_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.
s_trdy_l	STS	**Secondary PCI interface TRDY#**. Signal s_trdy_l is driven by the target of a transaction to indicate the target's ability to complete the current data phase on the secondary PCI bus. During a write transaction, assertion of s_trdy_l indicates that the target is able to accept write data for the current data phase. During a read transaction, assertion of s_trdy_l indicates that the target is driving valid read data on the s_ad bus. Once asserted during a given data phase, s_trdy_l is not deasserted until the data phase completes. Upon completion of a transaction, s_trdy_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.

2.3 Secondary PCI Bus Interface 64-Bit Extension Signals

Table 8 describes the secondary PCI bus interface 64-bit extension signals.

Table 8. Secondary PCI Bus Interface 64-Bit Extension Signals

Signal Name	Type	Description
s_ack64_l	STS	**Secondary PCI interface acknowledge 64-bit transfer.** Signal s_ack64_l is asserted by the target only when s_req64_l is asserted by the initiator, to indicate the target's ability to transfer data using 64 bits. Signal s_ack64_l has the same timing as s_devsel_l. When deasserting, s_ack64_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.
s_ad[63:32]	TS	**Secondary PCI interface address/data upper 32 bits.** This multiplexed address and data bus provides an additional 32 bits to the secondary interface. During the address phase or phases of a transaction, when the dual-address command is used and s_req64_l is asserted, the initiator drives the upper 32 bits of a 64-bit address; otherwise, these bits are undefined, and the initiator drives a valid logic level onto the pins. During the data phases of a transaction, the initiator drives the upper 32 bits of 64-bit write data, or the target drives the upper 32 bits of 64-bit read data, when s_req64_l and s_ack64_l are both asserted. When not driven, signals s_ad[63:32] are pulled up to a valid logic level through external resistors.
s_cbe_l[7:4]	TS	**Secondary PCI interface command/byte enables upper 4 bits.** These signals are a multiplexed command field and byte enable field. During the address phase or phases of a transaction, when the dual-address command is used and s_req64_l is asserted, the initiator drives the transaction type on s_cbe_l[7:4]; otherwise, these bits are undefined, and the initiator drives a valid logic level onto the pins. For both read and write transactions, the initiator drives byte enables for the s_ad[63:32] data bits on s_cbe_l[7:4] during the data phases, when s_req64_l and s_ack64_l are both asserted. When not driven, signals s_cbe_l[7:4] are pulled up to a valid logic level through external resistors.
s_par64	TS	**Secondary PCI interface upper 32 bits parity.** Signal s_par64 carries the even parity of the 36 bits of s_ad[63:32] and s_cbe_l[7:4] for both address and data phases. Signal s_par64 is driven by the initiator and is valid one clock cycle after the first address phase when a dual-address command is used and s_req64_l is asserted. Signal s_par64 is also valid one clock cycle after the second address phase of a dual-address transaction when s_req64_l is asserted. Signal s_par64_l is valid one clock cycle after valid data is driven (indicated by assertion of s_irdy_l for write data and s_trdy_l for read data), when both s_req64_l and s_ack64_l are asserted for that data phase. Signal s_par64 is tristated by the device driving read or write data one clock cycle after the s_ad lines are tristated. Devices receiving data sample s_par64 as an input to check for possible parity errors during 64-bit transactions. When not driven, s_par64 is pulled up to a valid logic level through external resistors.
s_req64_l	STS	**Secondary PCI interface request 64-bit transfer.** Signal s_req64_l is asserted by the initiator to indicate that the initiator is requesting 64-bit data transfer. Signal s_req64_l has the same timing as s_frame_l. If the 21554 is the secondary bus central function, it will assert s_req64_l low during secondary bus reset to indicate that a 64-bit bus is supported. When deasserting, s_req64_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor. The 21554 samples s_req64_l during secondary bus reset to enable the 64-bit extension signals. If s_req64_l is sampled high during reset, the secondary 64-bit extension is disabled and assumed not connected. The 21554 then drives s_ad[63:32], s_cbe_l[7:4], and s_par64 to valid logic levels.

2.4 Secondary PCI Bus Arbitration Signals

Table 9 describes the secondary PCI bus arbitration signals.

Table 9. Secondary PCI Bus Arbitration Signals

Signal Name	Type	Description
s_gnt_l[8:0]	TS	**Secondary PCI interface GNT#s**. The 21554 secondary bus arbiter can assert one of nine secondary bus grant outputs, s_gnt_l[8:0], to indicate that an initiator can start a transaction on the secondary bus if the bus is idle. The 21554's secondary bus grant is an internal signal. A programmable two-level rotating priority algorithm is used. If the internal arbiter is disabled, s_gnt_l[0] is reconfigured to be an external secondary bus request output for the 21554. The 21554 asserts this signal whenever it wants to start a transaction on the secondary bus.
s_req_l[8:0]	I	**Secondary PCI interface REQ#s**. The 21554 accepts nine request inputs, s_req_l[8:0], into its secondary bus arbiter. The 21554 request input to the arbiter is an internal signal. Each request input can be programmed to be in either a high-priority rotating group or a low-priority rotating group. An asserted level on an s_req_l pin indicates that the corresponding master wants to initiate a transaction on the secondary PCI bus. If the internal arbiter is disabled, s_req_l[0] is reconfigured to be an external secondary grant input for the 21554. In this case, an asserted level on s_req_l[0] indicates that the 21554 can start a transaction on the secondary PCI bus if the bus is idle.

2.5 Clock Signals

Table 10 describes the clock signals.

Table 10. Clock Signals

Signal Name	Type	Description
p_clk	I	**Primary interface PCI CLK**. This signal provides timing for all transactions on the primary PCI bus. All primary PCI inputs are sampled on the rising edge of p_clk, and all primary PCI outputs are driven from the rising edge of p_clk. Frequencies supported by the 21554 range from 0 MHz to 33 MHz.
s_clk	I	**Secondary interface PCI CLK**. This signal provides timing for all transactions on the secondary PCI bus. All secondary PCI inputs are sampled on the rising edge of s_clk, and all secondary PCI outputs are driven from the rising edge of s_clk. Frequencies supported by the 21554 range from 0 MHz to 33 MHz.
s_clk_o	O	**Secondary interface PCI CLK output**. This signal is generated from the primary interface clock input, p_clk. This clock operates at the same frequency of p_clk and may be externally buffered to create secondary bus device clock signals. When buffered clocks are used, one of the clock outputs must be fed back to the secondary clock input, s_clk. This clock output can be disabled and driven low by writing the secondary clock disable bit in configuration space, or by pulling pr_ad[5] low during reset.

2.6 Power Management, Hot Swap, and Reset Signals

Table 11 describes the power management, hot swap, and reset signals.

Table 11. Power Management, Hot Swap, and Reset Signals

Signal Name	Type	Description
l_stat	TS	Compact PCI hot swap local status pin. As an input to the 21554, this signal indicates the sense of the ejector switch and therefore the state of the LED in a Compact PCI card supporting distributed hot-swap. As an output from the 21554, it controls the LED. If Compact PCI hot swap is not supported by the add-in card, this signal should be tied low through a resistor.
p_enum_l	OD	Primary bus Compact PCI hot swap event. Conditionally asserted by the 21554, this signal indicates either that the card has been inserted and is ready for configuration, or that the card is about to be removed and should be deactivated by software. This signal is deasserted when the corresponding insertion or removal event bit is cleared. This signal should be pulled up by an external resistor.
p_pme_l	OD	**Primary bus power management event**. Provides power management signaling capability on behalf of the subsystem. The 21554 asserts p_pme_l when all of the following are true: • Signal s_pme_l is asserted low. • Assertion of signal p_pme_l is supported in the current power state. • PME_EN bit is set. Once asserted, p_pme_l is deasserted when the PME status bit or the PME_EN bit is cleared.
p_rst_l	I	**Primary PCI bus RST#**. Signal p_rst_l forces the 21554 to a known state. All register state is cleared, and all PCI bus outputs are tristated, with the possible exception of s_ad, s_cbe_l, s_par, and s_req64_l. Signal p_rst_l is asynchronous to p_clk.
s_pme_l	I	**Secondary bus power management event**. The subsystem asserts this signal to the 21554 to indicate that it is requesting a power management event. The 21554 conditionally asserts p_pme_l when an asserting edge on s_pme_l is detected. If the subsystem does not generate power management events, this signal can also be used for a subsystem status signal. A deasserting edge on this signal may conditionally cause the 21554 to assert p_inta_l. If this signal is not used, it should be tied high through a resistor.
s_rst_l	O	**Secondary PCI bus RST#**. Signal s_rst_l is driven by the 21554 and acts as the PCI reset for the secondary bus. The 21554 asserts s_rst_l when any of the following conditions is met: • Signal p_rst_l is asserted. • The secondary reset bit in the reset control register in configuration space is set. • The chip reset bit in the reset control register in configuration space is set. • Power management transition from $D3_{hot}$ to D0 occurs. When the 21554 asserts s_rst_l, it tristates all secondary control signals and, if designated as the secondary bus central resource, asserts s_req64_l and drives zeros on s_ad, s_cbe_l, and s_par. Signal s_rst_l remains asserted until p_rst_l is deasserted, and the secondary reset bit is clear. Deassertion of s_rst_l occurs automatically based on internal timers when s_rst_l assertion is caused by setting the chip reset bit or a power management transition. Assertion of s_rst_l by itself does not clear register state, and configuration registers are still accessible from the primary PCI interface.

2.7 ROM Interface Signals

Table 12 describes the ROM interface signals.

Table 12. ROM Interface Signals (Sheet 1 of 2)

Signal Name	Type	Description
pr_ad[7:0]	TS	These signals interface to both the serial and parallel external ROM circuitry and have multiple functions. The signals pr_ad[7:0] serve as multiplexed address/data for the parallel ROM and are latched externally in the following sequence: • • Address [23:16] • • Address [15:8] • • Address [7:0] • • Data [7:0] The signals pr_ad[2:0] also serve as serial ROM signals, with no external logic required: • • pr_ad[2] : sr_do, the serial ROM data output • • pr_ad[1] : sr_di, the serial ROM data input • • pr_ad[0] : sr_ck, the serial ROM clock output During primary bus reset, external pull-up or pull-down resistors can be used on signals pr_ad[7:2] to specify their state during reset. The values of these signals during primary bus reset specify the following configuration options: pr_ad[7] During primary bus reset, pr_ad[7] specifies the arbiter enable configuration. If low, the secondary bus arbiter is disabled, s_gnt_l[0] is used for 21554 secondary bus request, and s_req_l[0] is used for 21554 secondary bus grant. If high, the internal arbiter is enabled for use. pr_ad[6] During primary bus reset, pr_ad[6] specifies the central function enable. If low, the 21554 asserts s_req64_l and drives s_ad, s_cbe_l, and s_par low during secondary reset. If high, the 21554 tristates s_req64_l, s_ad, s_cbe_l, and s_par during secondary reset. pr_ad[5] During primary bus reset, pr_ad[5] specifies the s_clk_o enable. If low, s_clk_o is disabled and driven low. If high, s_clk_o is enabled and is a buffered version of p_clk. pr_ad[4] During primary bus reset, pr_ad[4] specifies synchronous enable. If high, the 21554 assumes asynchronous primary and secondary interfaces. If low, the 21554 assumes synchronous primary and secondary interfaces. pr_ad[3] During primary bus reset, pr_ad[3] specifies the primary lockout bit reset value. If high, the primary lockout bit is set high upon completion of chip reset, which causes the 21554 to return target retry to primary bus configuration transactions until the bit is cleared. If low, the primary lockout bit is low upon completion of reset, which allows immediate primary bus access to configuration registers. pr_ad[2] This signal should be biased high through a pull-up resistor. If the serial ROM is not connected, the 21554 will not detect the pre-load enable sequence 10b. In this case, the serial ROM preload is terminated after the first bit is read and the 21554 registers remain at their reset values. This is not actually sampled at reset, but during the first serial ROM read.

Table 12. ROM Interface Signals (Sheet 2 of 2)

Signal Name	Type	Description
pr_ale_l	O	**Parallel ROM address latch enable/chip select decoder enable**. The signal pr_ale_l is used to enable the parallel ROM address latches. The 21554 asserts pr_ale_l when it drives the first eight bits of the 24-bit address on pr_ad[7:0], and keeps it asserted until the last eight bits of the address are driven. The upper bits of the address are shifted through octal D-registers while pr_ale_l is low. When in multiple device mode, pr_ale_l is also used for a chip select enable. When pr_ale_l is high, the upper latched address lines are decoded with external circuitry to assert device chip enables
pr_clk	O	**Parallel ROM address latch clock output**. The signal pr_clk is used to clock the address registers needed to de-multiplex the address, and is a buffered version of p_clk divided by two.
pr_cs_l	O/I	**Parallel ROM chip select or device ready**. For a single device attachment, pr_cs_l is used for the parallel ROM chip select. The 21554 asserts pr_cs_l low after the address is shifted out and de-multiplexed through the three external octal registers. The 21554 deasserts pr_cs_l according to the access time specified in the ROM control CSR. When in multiple device mode, pr_cs_l is reconfigured as a device ready (pr_rdy) input. If pr_cs_l is driven low while the read or write strobe is asserted, the assertion time of the read or write strobe is extended by the amount of time the device ready signal is held low.
pr_rd_l	O	**Parallel ROM read strobe**. This signal controls the output enable signal of the parallel ROM. The 21554 asserts pr_rd_l to enable the ROM to drive read data on pr_ad[7:0]. The 21554 samples this read data on the deasserting (rising) edge of pr_rd_l. The timing of pr_rd_l with respect to the chip select is dictated by the read strobe mask.
pr_wr_l	O	**Parallel ROM write strobe**. This signal controls the write enable signal of the parallel ROM. The 21554 asserts pr_wr_l when it drives write data to the ROM on pr_ad[7:0]. Write data is held stable until the deasserting (rising) edge of pr_wr_l. The timing of pr_wr_l with respect to the chip select is dictated by the write strobe mask.
sr_cs	O	**Serial ROM chip select**. The 21554 drives this signal high to enable the serial ROM for a read or write. The serial ROM operation uses pins pr_ad[2:0] for data in, data out, and clock.

2.8 Miscellaneous Signals

Table 13 describes the miscellaneous signals.

Table 13. Miscellaneous Signals

Signal Name	Type	Description
cfg[1:0]	I	**Reserved pins**. These signals must be tied to vss through a resistor for proper functionality.
p_vio	I	**Primary interface I/O voltage**. This signal must be tied to either 3.3 V or 5.0 V, corresponding to the signaling environment of the primary PCI bus as described in the *PCI Local Bus Specification*, *Revision 2.1*. When any device on the primary PCI bus uses 5-V signaling levels, tie p_vio to 5.0 V. Signal p_vio is tied to 3.3 V only when all the devices on the primary bus use 3.3-V signaling levels.
s_vio	I	**Secondary interface I/O voltage**. This signal must be tied to either 3.3 V or 5.0 V, corresponding to the signaling environment of the secondary PCI bus as described in the *PCI Local Bus Specification, Revision 2.1*. When any device on the secondary PCI bus uses 5-V signaling levels, tie s_vio to 5.0 V. Signal s_vio is tied to 3.3 V only when all the devices on the secondary bus use 3.3-V signaling levels.

2.9 Diagnostic Signals

Table 14 describes JTAG and other diagnostic signals.

Table 14. JTAG Signals

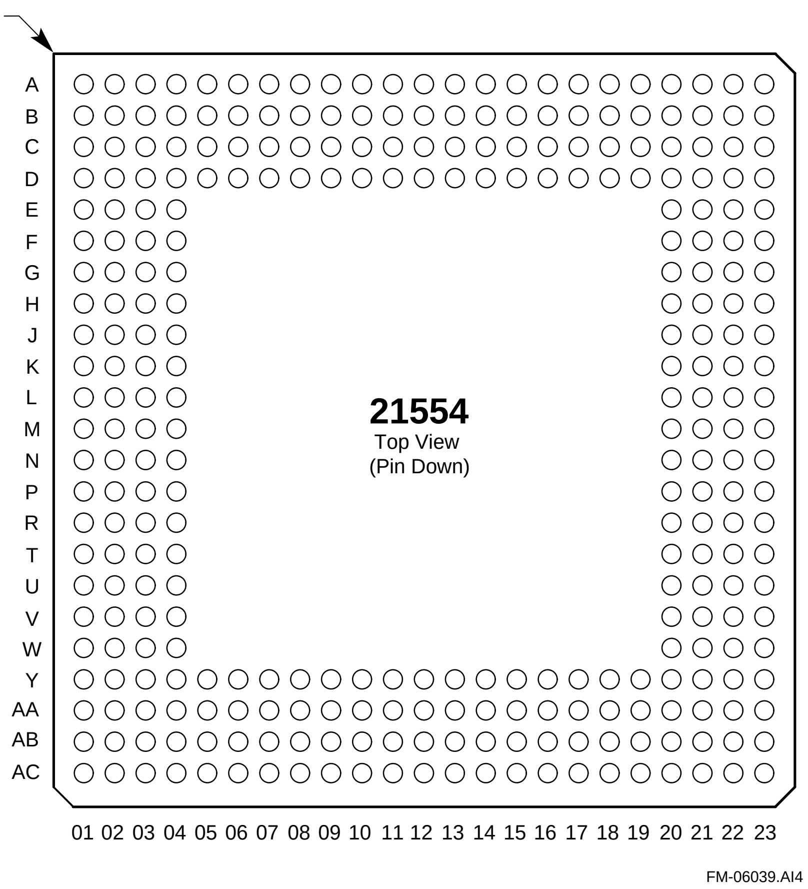
Signal Name	Type	Description
scan_ena	I	**Scan enable input**. This signal is used for chip test only.
tck	I	**JTAG boundary-scan clock.** Signal tck is the clock controlling the JTAG logic.
tdi	I	**JTAG serial data in.** Signal tdi is the serial input through which JTAG instructions and test data enter the JTAG interface. The new data on tdi is sampled on the rising edge of tck. An unterminated tdi produces the same result as if tdi were driven high.
tdo	O	**JTAG serial data out.** Signal tdo is the serial output through which test instructions and data from the test logic leave the 21554.
tms	I	**JTAG test mode select.** Signal tms causes state transitions in the test access port (TAP) controller. An undriven tms has the same result as if it were driven high.
trst_l	I	**JTAG TAP reset.** When asserted low, the TAP controller is asynchronously forced to enter a reset state, which in turn asynchronously initializes other test logic. An unterminated trst_l produces the same result as if trst_l were driven high.

3.0 Pin Assignment

This chapter describes the 21554 pin assignment and lists the pins according to location and in alphabetic order.

Figure 3 shows the 21554 304-point ball grid array (PBGA), representing the pins in vertical rows labeled numerically, and horizontal rows labeled alphabetically. Table 16 and Table 18 use these alphanumerics to identify pin assignments.

Figure 3. 21554 PBGA Cavity Down View



3.1 Pin Location List (Alphanumeric)

Table 16 lists the 21554 pins in order of location, showing the location code, signal name, and signal type of each pin.

Figure 3 provides the map for identifying the pin location codes, listed in alphanumeric order in the PBGA Location column in Table 16.

Table 15 defines the signal type abbreviations used in the Type columns in Table 16.

Table 15. Signal Type Abbreviations

Signal Type	Description
I	Standard input only.
O	Standard output only.
P	Power.
TS	Tristate bidirectional.
STS	Sustained tristate. Active low signal must be pulled high for one cycle when deasserting.
OD	Standard open drain.

Table 16. 21554 Pin Location List (Alphanumeric) (Sheet 1 of 5)

PBGA Location	Signal Name	Type	PBGA Location	Signal Name	Type
A1	s_req_l[4]	I	A20	s_ad[1]	TS
A2	s_req_l[3]	I	A21	s_req64_l	STS
A3	s_req_l[1]	I	A22	vdd	P
A4	s_ad[29]	TS	A23	s_cbe_l[6]	TS
A5	s_ad[27]	TS	AA1	p_ad[18]	TS
A6	s_ad[25]	TS	AA2	vss	P
A7	s_cbe_l[3]	TS	AA3	p_ad[17]	TS
A8	s_ad[22]	TS	AA4	vss	P
A9	s_ad[20]	TS	AA5	vdd	P
A10	s_ad[16]	TS	AA6	p_par	TS
A11	s_frame_l	STS	AA7	p_ad[12]	TS
A12	s_devsel_l	STS	AA8	p_ad[10]	TS
A13	s_par	TS	AA9	p_cbe_l[0]	TS
A14	s_ad[13]	TS	AA10	p_ad[5]	TS
A15	s_ad[10]	TS	AA11	vss	P
A16	cfg[1][1]	I	AA12	vdd	P
A17	s_cbe_l[0]	TS	AA13	vss	P
A18	s_ad[6]	TS	AA14	p_cbe_l[7]	TS
A19	s_ad[3]	TS	AA15	p_cbe_l[4]	TS

Table 16. 21554 Pin Location List (Alphanumeric) (Sheet 2 of 5)

PBGA Location	Signal Name	Type
AA16	vdd	P
AA17	p_ad[58]	TS
AA18	p_ad[54]	TS
AA19	vss	P
AA20	vdd	P
AA21	p_ad[46]	TS
AA22	p_ad[42]	TS
AA23	vdd	P
AB1	p_ad[16]	TS
AB2	vss	P
AB3	p_trdy_l	STS
AB4	p_stop_l	STS
AB5	p_serr_l	OD
AB6	p_ad[15]	TS
AB7	vss	P
AB8	vss	P
AB9	p_ad[8]	TS
AB10	p_ad[6]	TS
AB11	vdd	P
AB12	p_ad[1]	TS
AB13	p_ad[0]	TS
AB14	p_cbe_l[6]	TS
AB15	p_ad[63]	TS
AB16	p_ad[60]	TS
AB17	vss	P
AB18	p_ad[55]	TS
AB19	p_ad[53]	TS
AB20	p_ad[51]	TS
AB21	p_ad[48]	TS
AB22	vss	P
AB23	vdd	P
AC1	vdd	P
AC2	vdd	P
AC3	p_frame_l	STS
AC4	p_devsel_l	STS
AC5	p_perr_l	STS
AC6	p_cbe_l[1]	TS
AC7	p_ad[14]	TS
AC8	p_ad[11]	TS
AC9	cfg[0][1]	I
AC10	p_ad[7]	TS
AC11	p_ad[3]	TS
AC12	p_ad[2]	TS
AC13	p_ack64_l	STS
AC14	p_cbe_l[5]	TS
AC15	p_ad[61]	TS
AC16	p_ad[59]	TS
AC17	p_ad[56]	TS
AC18	vdd	P
AC19	p_ad[52]	TS
AC20	p_ad[50]	TS
AC21	p_ad[47]	TS
AC22	p_ad[45]	TS
AC23	p_ad[44]	TS
B1	vdd	P
B2	vss	P
B3	s_req_l[0]	I
B4	vdd	P
B5	s_ad[26]	TS
B6	s_ad[24]	TS
B7	s_idsel	I
B8	vss	P
B9	s_ad[18]	TS
B10	vss	P
B11	vss	P
B12	s_trdy_l	STS
B13	s_serr_l	OD
B14	s_ad[14]	TS
B15	s_ad[12]	TS
B16	vdd	P
B17	s_ad[9]	TS
B18	s_ad[7]	TS
B19	s_ad[4]	TS
B20	vdd	P

Table 16. 21554 Pin Location List (Alphanumeric) (Sheet 3 of 5)

PBGA Location	Signal Name	Type
B21	vss	P
B22	vss	P
B23	vdd	P
C1	s_req_l[6]	I
C2	s_req_l[7]	I
C3	s_req_l[2]	I
C4	s_ad[31]	TS
C5	s_ad[28]	TS
C6	vss	P
C7	s_ad[23]	TS
C8	s_ad[21]	TS
C9	s_ad[17]	TS
C10	vdd	P
C11	s_irdy_l	STS
C12	s_stop_l	STS
C13	s_perr_l	STS
C14	s_ad[15]	TS
C15	vdd	P
C16	vss	P
C17	vss	P
C18	s_ad[5]	TS
C19	s_ad[2]	TS
C20	s_ack64_l	STS
C21	s_cbe_l[5]	TS
C22	s_par64	TS
C23	s_cbe_l[4]	TS
D1	s_gnt_l[1]	TS
D2	s_gnt_l[2]	TS
D3	s_req_l[8]	I
D4	s_req_l[5]	I
D5	s_ad[30]	TS
D6	vdd	P
D7	vdd	P
D8	vss	P
D9	s_ad[19]	TS
D10	vdd	P
D11	s_cbe_l[2]	TS
D12	vss	P
D13	s_cbe_l[1]	TS
D14	vdd	P
D15	s_ad[11]	TS
D16	vss	P
D17	s_ad[8]	TS
D18	vdd	P
D19	s_ad[0]	TS
D20	s_cbe_l[7]	TS
D21	vss	P
D22	s_ad[61]	TS
D23	s_ad[62]	TS
E1	vdd	P
E2	s_gnt_l[4]	TS
E3	s_gnt_l[3]	TS
E4	s_gnt_l[0]	TS
E20	s_ad[63]	TS
E21	s_ad[60]	TS
E22	s_ad[58]	TS
E23	s_ad[59]	TS
F1	s_gnt_l[6]	TS
F2	s_gnt_l[7]	TS
F3	s_gnt_l[5]	TS
F4	vss	P
F20	vss	P
F21	vss	P
F22	s_ad[56]	TS
F23	s_ad[57]	TS
G1	s_gnt_l[8]	TS
G2	vss	P
G3	s_clk	I
G4	s_clk_o	O
G20	vdd	P
G21	s_ad[53]	TS
G22	s_ad[54]	TS
G23	s_ad[55]	TS
H1	s_rst_l	O

Table 16. 21554 Pin Location List (Alphanumeric) (Sheet 4 of 5)

PBGA Location	Signal Name	Type
H2	s_inta_l	OD
H3	tdi	I
H4	vdd	P
H20	vdd	P
H21	s_ad[50]	TS
H22	s_ad[51]	TS
H23	s_ad[52]	TS
J1	tdo	O
J2	tck	I
J3	trst_l	I
J4	tms	I
J20	vdd	P
J21	s_ad[47]	TS
J22	s_ad[48]	TS
J23	s_ad[49]	TS
K1	sr_cs	O
K2	pr_ad[7]	TS
K3	pr_ad[6]	TS
K4	vss	P
K20	vss	P
K21	s_ad[45]	TS
K22	vss	P
K23	s_ad[46]	TS
L1	pr_ad[4]	TS
L2	pr_ad[3]	TS
L3	pr_ad[2]	TS
L4	pr_ad[5]	TS
L20	s_ad[44]	TS
L21	s_ad[42]	TS
L22	s_ad[41]	TS
L23	s_ad[43]	TS
M1	pr_ad[0]	TS
M2	pr_rd_l	O
M3	pr_ad[1]	TS
M4	vdd	P
M20	vdd	P
M21	vdd	P
M22	s_ad[40]	TS
M23	vss	P
N1	pr_wr_l	O
N2	pr_ale_l	O
N3	pr_cs_l / pr_rdy	O/I
N4	pr_clk	O
N20	s_ad[36]	TS
N21	s_ad[39]	TS
N22	s_ad[38]	TS
N23	s_ad[37]	TS
P1	p_rst_l	I
P2	p_inta_l	OD
P3	scan_ena	I
P4	vss	P
P20	vss	P
P21	s_ad[35]	TS
P22	s_ad[34]	TS
P23	vss	P
R1	p_gnt_l	I
R2	vss	P
R3	vdd	P
R4	p_clk	I
R20	l_stat	TS
R21	s_ad[33]	TS
R22	s_ad[32]	TS
R23	s_pme_l	I
T1	p_ad[30]	TS
T2	p_ad[31]	TS
T3	p_req_l	TS
T4	vdd	P
T20	vdd	P
T21	s_vio	I
T22	p_enum_l	OD
T23	p_pme_l	OD
U1	p_ad[27]	TS
U2	p_ad[29]	TS
U3	vss	P

Table 16. 21554 Pin Location List (Alphanumeric) (Sheet 5 of 5)

PBGA Location	Signal Name	Type
U4	p_ad[28]	TS
U20	p_par64	TS
U21	p_vio	I
U22	vdd	P
U23	p_ad[32]	TS
V1	p_ad[25]	TS
V2	p_ad[26]	TS
V3	p_ad[24]	TS
V4	vss	P
V20	vss	P
V21	p_ad[35]	TS
V22	p_ad[33]	TS
V23	p_ad[34]	TS
W1	p_idsel	I
W2	p_cbe_l[3]	TS
W3	p_ad[23]	TS
W4	p_ad[20]	TS
W20	p_ad[40]	TS
W21	p_ad[38]	TS
W22	p_ad[36]	TS
W23	p_ad[37]	TS
Y1	p_ad[21]	TS
Y2	p_ad[22]	TS
Y3	p_ad[19]	TS
Y4	p_cbe_l[2]	TS
Y5	p_irdy_l	STS
Y6	vdd	P
Y7	p_ad[13]	TS
Y8	vss	P
Y9	p_ad[9]	TS
Y10	vdd	P
Y11	p_ad[4]	TS
Y12	vss	P
Y13	p_req64_l	STS
Y14	vdd	P
Y15	p_ad[62]	TS
Y16	vss	P
Y17	p_ad[57]	TS
Y18	vdd	P
Y19	p_ad[49]	TS
Y20	p_ad[43]	TS
Y21	p_ad[41]	TS
Y22	p_ad[39]	TS
Y23	vss	P

[1] Reserved pin. Tie low through an external resistor.

3.2 Pin Signal List (Alphanumeric)

Table 18 lists the 21554 signals in alphanumeric order, showing the name, location code, and type of each signal.

Figure 3 provides the map for identifying the pin location codes that are listed under PBGA Location in Table 18.

Table 17 defines the signal type abbreviations used in the Type columns in Table 18.

Table 17. Signal Type Abbreviations

Signal Type	Description
I	Standard input only.
O	Standard output only.
P	Power.
TS	Tristate bidirectional.
STS	Sustained tristate. Active low signal must be pulled high for one cycle when deasserting.
OD	Standard open drain.

Table 18. 21554 Pin Signal List (Alphanumeric) (Sheet 1 of 5)

Signal Name	PBGA Location	Type
cfg[0][1]	AC9	I
cfg[1][1]	A16	I
l_stat	R20	TS
p_ack64_l	AC13	STS
p_ad[0]	AB13	TS
p_ad[1]	AB12	TS
p_ad[2]	AC12	TS
p_ad[3]	AC11	TS
p_ad[4]	Y11	TS
p_ad[5]	AA10	TS
p_ad[6]	AB10	TS
p_ad[7]	AC10	TS
p_ad[8]	AB9	TS
p_ad[9]	Y9	TS
p_ad[10]	AA8	TS
p_ad[11]	AC8	TS
p_ad[12]	AA7	TS
p_ad[13]	Y7	TS
p_ad[14]	AC7	TS
p_ad[15]	AB6	TS
p_ad[16]	AB1	TS
p_ad[17]	AA3	TS
p_ad[18]	AA1	TS
p_ad[19]	Y3	TS
p_ad[20]	W4	TS
p_ad[21]	Y1	TS
p_ad[22]	Y2	TS
p_ad[23]	W3	TS
p_ad[24]	V3	TS
p_ad[25]	V1	TS
p_ad[26]	V2	TS
p_ad[27]	U1	TS
p_ad[28]	U4	TS
p_ad[29]	U2	TS
p_ad[30]	T1	TS
p_ad[31]	T2	TS
p_ad[32]	U23	TS
p_ad[33]	V22	TS
p_ad[34]	V23	TS
p_ad[35]	V21	TS
p_ad[36]	W22	TS
p_ad[37]	W23	TS
p_ad[38]	W21	TS
p_ad[39]	Y22	TS
p_ad[40]	W20	TS
p_ad[41]	Y21	TS
p_ad[42]	AA22	TS
p_ad[43]	Y20	TS
p_ad[44]	AC23	TS
p_ad[45]	AC22	TS

Table 18. 21554 Pin Signal List (Alphanumeric) (Sheet 2 of 5)

Signal Name	PBGA Location	Type
p_ad[46]	AA21	TS
p_ad[47]	AC21	TS
p_ad[48]	AB21	TS
p_ad[49]	Y19	TS
p_ad[50]	AC20	TS
p_ad[51]	AB20	TS
p_ad[52]	AC19	TS
p_ad[53]	AB19	TS
p_ad[54]	AA18	TS
p_ad[55]	AB18	TS
p_ad[56]	AC17	TS
p_ad[57]	Y17	TS
p_ad[58]	AA17	TS
p_ad[59]	AC16	TS
p_ad[60]	AB16	TS
p_ad[61]	AC15	TS
p_ad[62]	Y15	TS
p_ad[63]	AB15	TS
p_cbe_l[0]	AA9	TS
p_cbe_l[1]	AC6	TS
p_cbe_l[2]	Y4	TS
p_cbe_l[3]	W2	TS
p_cbe_l[4]	AA15	TS
p_cbe_l[5]	AC14	TS
p_cbe_l[6]	AB14	TS
p_cbe_l[7]	AA14	TS
p_clk	R4	I
p_devsel_l	AC4	STS
p_enum_l	T22	OD
p_frame_l	AC3	STS
p_gnt_l	R1	I
p_idsel	W1	I
p_inta_l	P2	OD
p_irdy_l	Y5	STS
p_par	AA6	TS
p_par64	U20	TS
p_perr_l	AC5	STS

Signal Name	PBGA Location	Type
p_pme_l	T23	OD
p_req_l	T3	TS
p_req64_l	Y13	STS
p_rst_l	P1	I
p_serr_l	AB5	OD
p_stop_l	AB4	STS
p_trdy_l	AB3	STS
pr_ad[0]	M1	TS
pr_ad[1]	M3	TS
pr_ad[2]	L3	TS
pr_ad[3]	L2	TS
pr_ad[4]	L1	TS
pr_ad[5]	L4	TS
pr_ad[6]	K3	TS
pr_ad[7]	K2	TS
pr_ale_l	N2	O
pr_clk	N4	O
pr_cs_l / pr_rdy	N3	O/I
pr_rd_l	M2	O
pr_wr_l	N1	O
p_vio	U21	I
s_ack64_l	C20	STS
s_ad[0]	D19	TS
s_ad[1]	A20	TS
s_ad[2]	C19	TS
s_ad[3]	A19	TS
s_ad[4]	B19	TS
s_ad[5]	C18	TS
s_ad[6]	A18	TS
s_ad[7]	B18	TS
s_ad[8]	D17	TS
s_ad[9]	B17	TS
s_ad[10]	A15	TS
s_ad[11]	D15	TS
s_ad[12]	B15	TS
s_ad[13]	A14	TS
s_ad[14]	B14	TS

Table 18. 21554 Pin Signal List (Alphanumeric) (Sheet 3 of 5)

Signal Name	PBGA Location	Type
s_ad[15]	C14	TS
s_ad[16]	A10	TS
s_ad[17]	C9	TS
s_ad[18]	B9	TS
s_ad[19]	D9	TS
s_ad[20]	A9	TS
s_ad[21]	C8	TS
s_ad[22]	A8	TS
s_ad[23]	C7	TS
s_ad[24]	B6	TS
s_ad[25]	A6	TS
s_ad[26]	B5	TS
s_ad[27]	A5	TS
s_ad[28]	C5	TS
s_ad[29]	A4	TS
s_ad[30]	D5	TS
s_ad[31]	C4	TS
s_ad[32]	R22	TS
s_ad[33]	R21	TS
s_ad[34]	P22	TS
s_ad[35]	P21	TS
s_ad[36]	N20	TS
s_ad[37]	N23	TS
s_ad[38]	N22	TS
s_ad[39]	N21	TS
s_ad[40]	M22	TS
s_ad[41]	L22	TS
s_ad[42]	L21	TS
s_ad[43]	L23	TS
s_ad[44]	L20	TS
s_ad[45]	K21	TS
s_ad[46]	K23	TS
s_ad[47]	J21	TS
s_ad[48]	J22	TS
s_ad[49]	J23	TS
s_ad[50]	H21	TS
s_ad[51]	H22	TS
s_ad[52]	H23	TS
s_ad[53]	G21	TS
s_ad[54]	G22	TS
s_ad[55]	G23	TS
s_ad[56]	F22	TS
s_ad[57]	F23	TS
s_ad[58]	E22	TS
s_ad[59]	E23	TS
s_ad[60]	E21	TS
s_ad[61]	D22	TS
s_ad[62]	D23	TS
s_ad[63]	E20	TS
s_cbe_l[0]	A17	TS
s_cbe_l[1]	D13	TS
s_cbe_l[2]	D11	TS
s_cbe_l[3]	A7	TS
s_cbe_l[4]	C23	TS
s_cbe_l[5]	C21	TS
s_cbe_l[6]	A23	TS
s_cbe_l[7]	D20	TS
s_clk	G3	I
s_clk_o	G4	O
s_devsel_l	A12	STS
s_frame_l	A11	STS
s_gnt_l[0]	E4	TS
s_gnt_l[1]	D1	TS
s_gnt_l[2]	D2	TS
s_gnt_l[3]	E3	TS
s_gnt_l[4]	E2	TS
s_gnt_l[5]	F3	TS
s_gnt_l[6]	F1	TS
s_gnt_l[7]	F2	TS
s_gnt_l[8]	G1	TS
s_idsel	B7	I
s_inta_l	H2	OD
s_irdy_l	C11	STS
s_par	A13	TS

Table 18. 21554 Pin Signal List (Alphanumeric) (Sheet 4 of 5)

Signal Name	PBGA Location	Type
s_par64	C22	TS
s_perr_l	C13	STS
s_pme_l	R23	I
s_req_l[0]	B3	I
s_req_l[1]	A3	I
s_req_l[2]	C3	I
s_req_l[3]	A2	I
s_req_l[4]	A1	I
s_req_l[5]	D4	I
s_req_l[6]	C1	I
s_req_l[7]	C2	I
s_req_l[8]	D3	I
s_req64_l	A21	STS
s_rst_l	H1	O
s_serr_l	B13	OD
s_stop_l	C12	STS
s_trdy_l	B12	STS
scan_ena	P3	I
sr_cs	K1	O
s_vio	T21	I
tck	J2	I
tdi	H3	I
tdo	J1	O
tms	J4	I
trst_l	J3	I
vdd	A22	P
vdd	AA5	P
vdd	AA12	P
vdd	AA16	P
vdd	AA20	P
vdd	AA23	P
vdd	AB11	P
vdd	AB23	P
vdd	AC1	P
vdd	AC2	P
vdd	AC18	P
vdd	B1	P
vdd	B4	P
vdd	B16	P
vdd	B20	P
vdd	B23	P
vdd	C10	P
vdd	C15	P
vdd	D6	P
vdd	D7	P
vdd	D10	P
vdd	D14	P
vdd	D18	P
vdd	E1	P
vdd	G20	P
vdd	H4	P
vdd	H20	P
vdd	J20	P
vdd	M4	P
vdd	M20	P
vdd	M21	P
vdd	R3	P
vdd	T4	P
vdd	T20	P
vdd	U22	P
vdd	Y6	P
vdd	Y10	P
vdd	Y14	P
vdd	Y18	P
vss	AA2	P
vss	AA4	P
vss	AA11	P
vss	AA13	P
vss	AA19	P
vss	AB2	P
vss	AB7	P
vss	AB8	P
vss	AB17	P
vss	AB22	P

Table 18. 21554 Pin Signal List (Alphanumeric) (Sheet 5 of 5)

Signal Name	PBGA Location	Type	Signal Name	PBGA Location	Type
vss	B2	P	vss	G2	P
vss	B8	P	vss	K4	P
vss	B10	P	vss	K20	P
vss	B11	P	vss	K22	P
vss	B21	P	vss	M23	P
vss	B22	P	vss	P4	P
vss	C6	P	vss	P20	P
vss	C16	P	vss	P23	P
vss	C17	P	vss	R2	P
vss	D8	P	vss	U3	P
vss	D12	P	vss	V4	P
vss	D16	P	vss	V20	P
vss	D21	P	vss	Y8	P
vss	F4	P	vss	Y12	P
vss	F20	P	vss	Y16	P
vss	F21	P	vss	Y23	P

1. Reserved pin. Tie low through an external resistor.

4.0 JTAG Boundary Scan

This chapter describes the implementation of the 21554's JTAG test port according to IEEE Standard 1149.1, IEEE Standard Test Access Port and Boundary-Scan Architecture.

The 21554's JTAG test port consists of the following:

- A 5-signal test port interface
- A test access port controller
- An instruction register
- A bypass register
- A boundary-scan register

The JTAG test access port is to be used only while the 21554 is not operating. Table 19 lists the JTAG signal pins with a brief description of each pin.

Table 19. JTAG Signal Pins

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

4.1 Test Access Port Controller

The test access port controller is a finite state machine that interprets IEEE 1149.1 protocols received through the tms signal. The state transitions in the controller are caused by the tms signal on the rising edge of tck. In each state, the controller generates appropriate clock and control signals that control the operation of the test features. After entry into a state, test feature operations are initiated on the rising edge of tck.

4.1.1 Initialization

The test access port controller and the instruction register 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 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: Signal trst_l must either be asserted or have been asserted after power-up in order for the chip to function.

4.2 Instruction Register

The 5-bit instruction register selects the test mode and features. The instruction codes are shown in Table 20. These instructions select and control the operation of the boundary-scan and bypass registers. The instruction register is loaded through the tdi pin. The instruction register has a serial shift-in stage from which the instruction is then loaded in parallel.

Table 20. JTAG Instruction Register Options

Instruction Register Contents	Instruction Name	Test Register Selected	Operation
0000	EXTEST	Boundary-scan	External test (drives pins from boundary-scan register)
0001	SAMPLE	Boundary-scan	Samples inputs
0010	HIGHZ	Bypass	Tristates all output and I/O pins except the tdo pin
0011	CLAMP	Bypass	Drives pins from the boundary-scan register and selects the bypass register for shifts
0100	IDCODE	Idcode	Reads the manufacturer’s identification number, the design part number, and the design verification number
0101-1111	BYPASS	Bypass	Selects the bypass register for shifts

4.3 Bypass Register

The bypass register is a 1-bit shift register that provides a means for effectively bypassing the JTAG test logic through a single-bit serial connection through the chip from tdi to tdo. At board-level testing, this helps reduce the overall length of the scan ring.

4.4 Boundary-Scan Register

The boundary-scan register (BSR) is a single-shift register-based path formed by boundary-scan cells placed at the chip’s signal pins. The register is accessed through the JTAG port’s tdi and tdo pins.

Each boundary-scan cell operates in conjunction with the current instruction and the current state in the test access port controller state machine. The function of the BSR cells is determined by the associated pins, as follows:

- Input-only pins: the boundary-scan cell is basically a 1-bit shift register. The cell supports sample and shift operations.
- Output-only pins: the boundary-scan cell comprises a 1-bit shift register and an output multiplexer. The cell supports the sample, shift, and drive output functions.
- Bidirectional pins: the boundary-scan cell is identical to the output-only pin cell, but it captures test data from the incoming data line. The cell supports sample, shift, drive output, and hold output functions.

4.5 Boundary-Scan Order

The boundary-scan register is a shift register path formed by boundary-scan cells placed at the signal pins. The register is accessed through the JTAG tdi and tdo pins.

Table 21 provides the boundary-scan index for each I/O cell in the 21554 design. The list is ordered from tdo to tdi; the index starts at 1 and ends at 452. tdo is applied on a rising edge of tck. tdo is valid on the falling edge.

Note: The tdo pin is enabled by the JTAG specification. All other tristate pins are controlled by the boundary-scan register scan cell indicated in Table 21, as well as the JTAG HIGHZ instruction.

All input ports are monitored by BC_4 type cells. EXTEST overrides the clock. All output ports are controlled by a BC_1 cell.

All bidirectional ports contain a BC_1 for the output path and a BC_4 for the input path. The enable is controlled by a BC_2 cell.

Table 21. Boundary-Scan Order (Sheet 1 of 6)

Port	Input Monitor Cell	Output Control Cell	Group Enable Number
sr_cs	1	2	30
pr_ad[7]	3	4	13
pr_ad[6]	5	6	13
pr_ad[5]	7	8	13
pr_ad[4]	9	10	13
pr_ad[3]	11	12	13
pr_ad[2]	14	15	16
pr_ad[1]	17	18	13
pr_ad[0]	19	20	13
pr_rd_l	21	22	30
pr_ale_l	23	24	—
pr_rdy/pr_cs_l	25	26	27
pr_wr_l	28	29	30
pr_clk	31	32	30
scan_ena	—	—	—
p_inta_l	33	226	34
p_rst_l	35	—	—
p_clk	36	—	—
p_gnt_l	37	—	—
p_req_l	38	39	40
p_ad[31]	41	42	49
p_ad[30]	43	44	49
p_ad[29]	45	46	49
p_ad[28]	47	48	49
p_ad[27]	50	51	49
p_ad[26]	52	53	49

Table 21. Boundary-Scan Order (Sheet 2 of 6)

Port	Input Monitor Cell	Output Control Cell	Group Enable Number
p_ad[25]	54	55	49
p_ad[24]	56	57	49
p_cbe_l[3]	58	59	103
p_idsel	60	—	—
p_ad[23]	61	62	69
p_ad[22]	63	64	69
p_ad[21]	65	66	69
p_ad[20]	67	68	69
p_ad[19]	70	71	69
p_ad[18]	72	73	69
p_ad[17]	74	75	69
p_ad[16]	76	77	69
p_cbe_l[2]	78	79	103
p_frame_l	80	81	82
p_trdy_l	83	84	92
p_irdy_l	85	86	87
p_devsel_l	88	89	92
p_stop_l	90	91	92
p_perr_l	93	94	95
p_serr_l	96	226	97
p_par	98	99	100
p_cbe_l[1]	101	102	103
p_ad[15]	104	105	112
p_ad[14]	106	107	112
p_ad[13]	108	109	112
p_ad[12]	110	111	112
p_ad[11]	113	114	112
p_ad[10]	115	116	112
cfg[0]	117	—	—
p_ad[9]	118	119	112
p_ad[8]	120	121	112
p_cbe_l[0]	122	123	103
p_ad[7]	124	125	132
p_ad[6]	126	127	132
p_ad[5]	128	129	132
p_ad[4]	130	131	132
p_ad[3]	133	134	132
p_ad[2]	135	136	132
p_ad[1]	137	138	132
p_ad[0]	139	140	132
p_ack64_l	141	142	92

Table 21. Boundary-Scan Order (Sheet 3 of 6)

Port	Input Monitor Cell	Output Control Cell	Group Enable Number
p_req64_l	143	144	82
p_cbe_l[7]	145	146	149
p_cbe_l[6]	147	148	149
p_cbe_l[5]	150	151	149
p_cbe_l[4]	152	153	149
p_ad[63]	154	155	162
p_ad[62]	156	157	162
p_ad[61]	158	159	162
p_ad[60]	160	161	162
p_ad[59]	163	164	162
p_ad[58]	165	166	162
p_ad[57]	167	168	162
p_ad[56]	169	170	162
p_ad[55]	171	172	179
p_ad[54]	173	174	179
p_ad[53]	175	176	179
p_ad[52]	177	178	179
p_ad[51]	180	181	179
p_ad[50]	182	183	179
p_ad[49]	184	185	179
p_ad[48]	186	187	179
p_ad[47]	188	189	196
p_ad[46]	190	191	196
p_ad[45]	192	193	196
p_ad[44]	194	195	196
p_ad[43]	197	198	196
p_ad[42]	199	200	196
p_ad[41]	201	202	196
p_ad[40]	203	204	196
p_ad[39]	205	206	213
p_ad[38]	207	208	213
p_ad[37]	209	210	213
p_ad[36]	211	212	213
p_ad[35]	214	215	213
p_ad[34]	216	217	213
p_ad[33]	218	219	213
p_ad[32]	220	221	213
p_par64	222	223	224
p_pme_l	225	226	227
p_enum_l	228	226	229
s_pme_l	230	—	—

Table 21. Boundary-Scan Order (Sheet 4 of 6)

Port	Input Monitor Cell	Output Control Cell	Group Enable Number
l_stat	231	232	233
s_ad[32]	234	235	242
s_ad[33]	236	237	242
s_ad[34]	238	239	242
s_ad[35]	240	241	242
s_ad[36]	243	244	242
s_ad[37]	245	246	242
s_ad[38]	247	248	242
s_ad[39]	249	250	242
s_ad[40]	251	252	259
s_ad[41]	253	254	259
s_ad[42]	255	256	259
s_ad[43]	257	258	259
s_ad[44]	260	261	259
s_ad[45]	262	263	259
s_ad[46]	264	265	259
s_ad[47]	266	267	259
s_ad[48]	268	269	276
s_ad[49]	270	271	276
s_ad[50]	272	273	276
s_ad[51]	274	275	276
s_ad[52]	277	278	276
s_ad[53]	279	280	276
s_ad[54]	281	282	276
s_ad[55]	283	284	276
s_ad[56]	285	286	293
s_ad[57]	287	288	293
s_ad[58]	289	290	293
s_ad[59]	291	292	293
s_ad[60]	294	295	293
s_ad[61]	296	297	293
s_ad[62]	298	299	293
s_ad[63]	300	301	293
s_par64	302	303	304
s_cbe_l[4]	305	306	307
s_cbe_l[5]	308	309	307
s_cbe_l[6]	310	311	307
s_cbe_l[7]	312	313	307
s_req64_l	314	315	316
s_ack64_l	317	318	368
s_ad[0]	319	320	329

Table 21. Boundary-Scan Order (Sheet 5 of 6)

Port	Input Monitor Cell	Output Control Cell	Group Enable Number
s_ad[1]	321	322	329
s_ad[2]	323	324	329
s_ad[3]	325	326	329
s_ad[4]	327	328	329
s_ad[5]	330	331	329
s_ad[6]	332	333	329
s_ad[7]	334	335	329
s_cbe_l[0]	336	337	381
s_ad[8]	338	339	347
s_ad[9]	340	341	347
cfg[1]	342	—	—
s_ad[10]	343	344	347
s_ad[11]	345	346	347
s_ad[12]	348	349	347
s_ad[13]	350	351	347
s_ad[14]	352	353	347
s_ad[15]	354	355	347
s_cbe_l[1]	356	357	381
s_par	358	359	360
s_serr_l	361	226	362
s_perr_l	363	364	365
s_stop_l	366	367	368
s_devsel_l	369	370	368
s_trdy_l	371	372	368
s_irdy_l	373	374	375
s_frame_l	376	377	378
s_cbe_l[2]	379	380	381
s_ad[16]	382	383	390
s_ad[17]	384	385	390
s_ad[18]	386	387	390
s_ad[19]	388	389	390
s_ad[20]	391	392	390
s_ad[21]	393	394	390
s_ad[22]	395	396	390
s_ad[23]	397	398	390
s_idsel	399	—	—
s_cbe_l[3]	400	401	381
s_ad[24]	402	403	410
s_ad[25]	404	405	410
s_ad[26]	406	407	410
s_ad[27]	408	409	410

Table 21. Boundary-Scan Order (Sheet 6 of 6)

Port	Input Monitor Cell	Output Control Cell	Group Enable Number
s_ad[28]	411	412	410
s_ad[29]	413	414	410
s_ad[30]	415	416	410
s_ad[31]	417	418	410
s_req_l[0]	419	—	—
s_req_l[1]	420	—	—
s_req_l[2]	421	—	—
s_req_l[3]	422	—	—
s_req_l[4]	423	—	—
s_req_l[5]	424	—	—
s_req_l[6]	425	—	—
s_req_l[7]	426	—	—
s_req_l[8]	427	—	—
s_gnt_l[0]	428	429	438
s_gnt_l[1]	430	431	438
s_gnt_l[2]	432	433	438
s_gnt_l[3]	434	435	438
s_gnt_l[4]	436	437	438
s_gnt_l[5]	439	440	438
s_gnt_l[6]	441	442	438
s_gnt_l[7]	443	444	438
s_gnt_l[8]	445	446	438
s_clk_o	—	447	—
s_clk	448	—	—
s_rst_l	449	450	30
s_inta_l	451	226	452

5.0 Electrical Specifications

This chapter specifies the following electrical behavior of the 21554:

- PCI electrical conformance
- Absolute maximum ratings
- DC specifications
- AC timing specifications

5.1 PCI Electrical Specification Conformance

The 21554 PCI pins conform to the basic set of PCI electrical specifications in the *PCI Local Bus Specification*, Revision 2.1. See that document for a complete description of the PCI I/O protocol and pin ac specifications.

5.2 Absolute Maximum Ratings

The 21554 is specified to operate at a maximum frequency of 33 MHz at a junction temperature (T_j) not to exceed 125°C. Table 22 lists the absolute maximum ratings for the 21554. Stressing the device beyond the absolute maximum ratings may cause permanent damage. These are stress ratings only. Operating beyond the functional operating range is not recommended and extended exposure beyond the functional operating range may affect reliability.

Table 22. Absolute Maximum Ratings

Parameter	Minimum	Maximum
Junction temperature, T_j	—	125°C
Supply voltage V_{cc}	—	3.9 V
Maximum voltage applied to signal pins	—	5.5 V
Maximum power, P_{WC}	—	2.1 W
Storage temperature range, T_{stg}	–55°C	125°C

Table 23. Functional Operating Range

Parameter	Minimum	Maximum
Supply voltage, V_{cc}	3.0 V	3.6 V
Operating ambient temperature, T_a	0°C	70°C

5.3 DC Specifications

Table 24 defines the DC parameters met by all 21554 signals under the conditions of the functional operating range.

Table 24. DC Parameters

Symbol	Parameter	Condition	Minimum	Maximum	Unit
V_{cc}	Supply voltage	—	3.0	3.6	V
V_{il}	Low-level input voltage[1]	—	–0.5	0.3 V_{cc}	V
V_{ih}	High-level input voltage[1]	—	0.5 V_{cc}	V_{IO} + 0.5 V	V
V_{ol}	Low-level output voltage[2]	I_{out} = 1500 µA	—	0.1 V_{cc}	V
V_{ol5V}	Low-level output voltage[3]	I_{out} = 6 mA	—	0.55	V
V_{oh}	High-level output voltage[2]	I_{out} = –500 µA	0.9 V_{cc}	—	V
V_{oh5V}	High-level output voltage[3]	I_{out} = –2 mA	2.4	—	V
I_{il}	Low-level input leakage current[1,4]	0 <V_{in} <V_{cc}	—	±10	µA
C_{in}	Input pin capacitance	—	—	10.0	pF
C_{IDSEL}	p_idsel pin capacitance	—	—	8.0	pF
C_{clk}	p_clk, s_clk pin capacitance	—	5.0	12.0	pF

1. Guarantees meeting the specification for the 5-V signaling environment.
2. For 3.3-V signaling environment.
3. For 5-V signaling environment.
4. Input leakage currents include high-Z output leakage for all bidirectional buffers with tristate outputs.

Note: In Table 24, currents into the chip (chip sinking) are denoted as positive (+) current. Currents from the chip (chip sourcing) are denoted as negative (–) current.

5.4 AC Timing Specifications

The next sections specify the ac characteristics met by all 21554 signals under the conditions of the functional operating range:

- Clock timing
- PCI signal timing
- Reset timing
- Serial ROM timing
- Parallel ROM timing
- JTAG timing

5.4.1 Clock Timing Specifications

The ac specifications consist of input requirements and output responses. The input requirements consist of setup and hold times, pulse widths, and high and low times. The output responses are delays from clock to signal. The ac specifications are defined separately for each clock domain within the 21554.

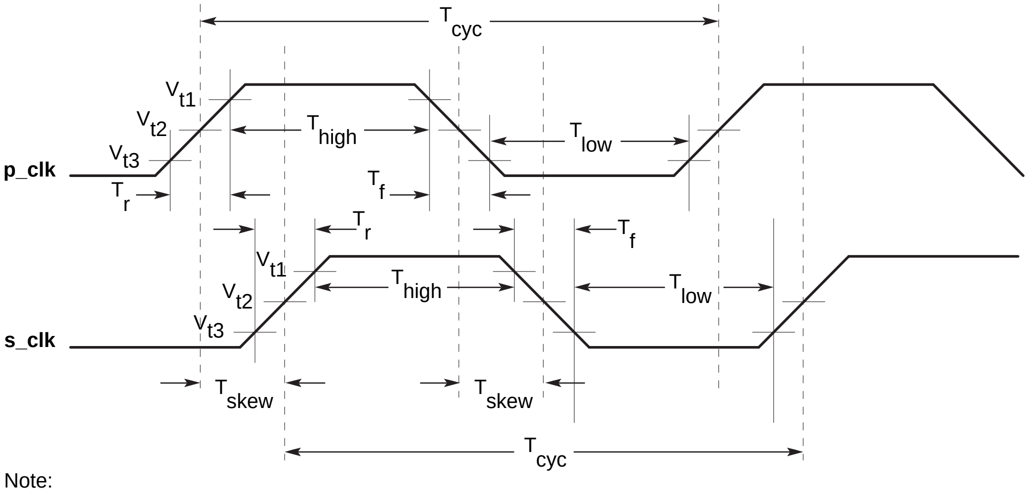
Table 25 specifies p_clk and s_clk parameter values for clock signal ac timing, and Figure 4 shows the ac parameter measurements for the p_clk and s_clk signals. See also Figure 5 for a further illustration of signal timing. Unless otherwise indicated, all ac parameters are guaranteed when tested within the functional operating range of Table 23.

Table 25. PCI Clock Signal AC Parameters

Symbol	Parameter	Minimum	Maximum	Unit
T_{cyc}	p_clk,s_clk cycle time	30	∞	ns
T_{high}	p_clk, s_clk high time	11	—	ns
T_{low}	p_clk, s_clk low time	11	—	ns
	p_clk, s_clk slew rate[1]	1	4	V/ns
T_{sclk}	Delay from p_clk to s_clk[2]	3	15	ns
T_{sclkr}	p_clk rising to s_clk_o rising	0	8	ns
T_{sclkf}	p_clk falling to s_clk_o falling[3]	0	8	ns
T_{dskew}	s_clk_o duty cycle skew from p_clk duty cycle[3]	—	.75	ns

1. 0.2 V_{CC} to 0.6 V_{CC}.
2. Required when the 21554 is operating in synchronous mode.
3. Measured with 30 pF lumped load.

Figure 4. PCI Clock Signal AC Parameter Measurements



Note:

V_{t1} – 2.0 V for 5-V clocks; 0.5 V_{cc} for 3.3-V clocks

V_{t2} – 1.5 V for 5-V clocks; 0.4 V_{cc} for 3.3-V clocks

V_{t3} – 0.8 V for 5-V clocks; 0.3 V_{cc} for 3.3-V clocks

LJ-04738.AI4

5.4.2 PCI Signal Timing Specifications

Figure 5 and Table 26 show the PCI signal timing specifications.

Figure 5. PCI Signal Timing Measurement Conditions

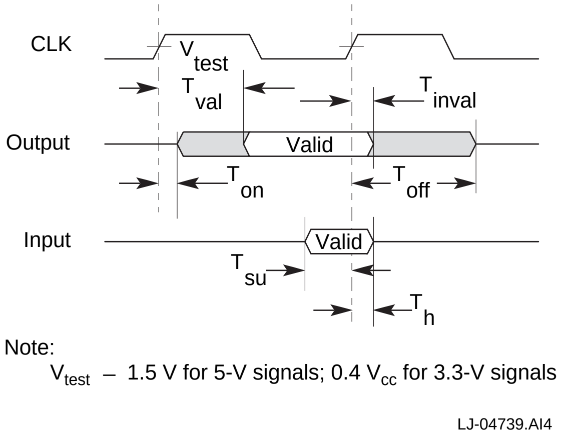


Table 26. PCI Signal Timing Specifications

Symbol	Parameter	Minimum	Maximum	Unit
T_{val}	CLK to signal valid delay — bused signals[1,2,3]	2	11	ns
$T_{val(ptp)}$	CLK to signal valid delay — point-to-point[1,2,3]	2	12	ns
T_{on}	Float to active delay[1,2]	2	—	ns
T_{off}	Active to float delay[1,2]	—	28	ns
T_{su}	Input setup time to CLK — bused signals[1,2,3]	7	—	ns
$T_{su(ptp)}$	Input setup time to CLK—point-to-point[1,2,3]	10, 12	—	ns
T_h	Input signal hold time from CLK[1,2]	0	—	ns

1. See Figure 5.
2. All primary interface signals are synchronized to p_clk. All secondary interface signals are synchronized to s_clk.
3. Point-to-point signals are p_req_l, s_req_l[8:0], p_gnt_l, and s_gnt_l[8:0]. Bused signals are p_ad, p_cbe_l, p_par, p_par64, p_perr_l, p_serr_l, p_frame_l, p_irdy_l, p_trdy_l, p_devsel_l, p_stop_l, p_idsel, p_req64_l, p_ack64_l, s_ad, s_cbe_l, s_par, s_par64, s_perr_l, s_serr_l, s_frame_l, s_irdy_l, s_trdy_l, s_devsel_l, s_stop_l, s_req64_l, s_ack64_l, and s_idsel.

5.4.3 Reset Timing Specifications

Table 27 shows the reset timing specifications for p_rst_l and s_rst_l.

Table 27. Reset Timing Specifications

Symbol	Parameter	Minimum	Maximum	Unit
T_{rst}	p_rst_l active time after power stable	1	—	µs
$T_{rst\text{-}clk}$	p_rst_l active time after p_clk stable	100	—	µs
$T_{rst\text{-}off}$	p_rst_l active-to-output float delay	—	40	ns
T_{srst}	s_rst_l active after p_rst_l assertion	—	40	ns
$T_{srst\text{-}on}$	s_rst_l active time after s_clk stable	100	—	µs
T_{dsrst}	s_rst_l deassertion after p_rst_l deassertion	0	25	Cycles
	p_rst_l slew rate[1]	50	—	mV/ns
T_{rrsus}	s_req64_l asserted to s_rst_l deasserted	10*T_{cyc}	—	ns[1]
T_{rrval}	s_rst_l to s_req64_l deasserted delay time	0 T_{cyc}	—	ns[1]
T_{rrsu}	REQ64# to RST# deasserting setup time	T_{cyc}	—	ns
T_{rrh}	REQ64# from RST# deasserting hold time	0	50	ns

[1] Applies to rising (deasserting) edge only.

5.4.4 Serial ROM Timing Specifications

Table 28 shows the serial ROM timing specifications.

Table 28. Serial ROM Timing Specifications

Symbol	Parameter	Minimum	Maximum	Unit
T_{scval}	pclk to pr_ad[0] serial ROM clock valid	—	14	ns
T_{son}	pr_ad float to active delay	2	—	ns
T_{soff}	pr_ad active to float delay	—	28	ns
T_{ssu}	pr_ad[1] di to pr_ad[0] serial ROM clock setup time	400	—	ns
T_{sh}	pr_ad[1] to pr_ad[0] serial ROM clock hold time	20	—	ns
T_{smcs}	sr_cs minimum low time	400	—	ns
T_{scyc}	pr_ad[0] serial ROM clock cycle time	1000	—	ns

5.4.5 Parallel ROM Timing Specifications

Table 29 shows the parallel ROM timing specifications.

Table 29. Parallel ROM Timing Specifications

Symbol	Parameter	Minimum	Maximum	Unit
T_{pas}	pr_ale_l setup to pr_clk rising	40	—	ns
T_{pcc}	pr_clk cycle time	60	—	ns
T_{pacs}	pr_ale_l rising to pr_cs_l falling	25	—	ns
T_{pcsl}	pr_cs_l low	200	—	ns
T_{pcrw}	pr_cs_l falling to pr_rd_l or pr_wr_l falling	25	—	ns
T_{prs}	pr_ad setup time to pr_rd_l rising	180	—	ns
T_{prh}	pr_ad hold time from pr_rd_l rising	0	—	ns
T_{prv}	pr_clk rising to pr_ad valid	0	15	ns

5.4.6 JTAG Timing Specifications

Table 30 shows the JTAG timing specifications.

Table 30. JTAG Timing Specifications

Symbol	Parameter	Minimum	Maximum	Unit
T_{jr}	tck frequency	0	10	MHz
T_{jp}	tck period	100	∞	ns
T_{ght}	tck high time	45	—	ns
T_{glt}	tck low time	45	—	ns
T_{jrt}	tck rise time[1]	—	10	ns
T_{gft}	tck fall time[2]	—	10	ns
T_{js}	tdi, tms setup time to tck rising edge	10	—	ns
T_{jh}	tdi, tms hold time from tck rising edge	25	—	ns
T_{jd}	tdo valid delay from tck falling edge[3]	—	30	ns
T_{jfd}	tdo float delay from tck falling edge	—	30	ns

1. Measured between 0.8 V and 2.0 V.
2. Measured between 2.0 V and 0.8 V.
3. C_1=50 pF.

6.0 Mechanical Specifications

The 21554 is contained in an industry-standard 304 PBGA, a two-layer plastic ball grid array package, as shown in Figure 6.

Figure 6. 304 PBGA (Two-Layer) Package

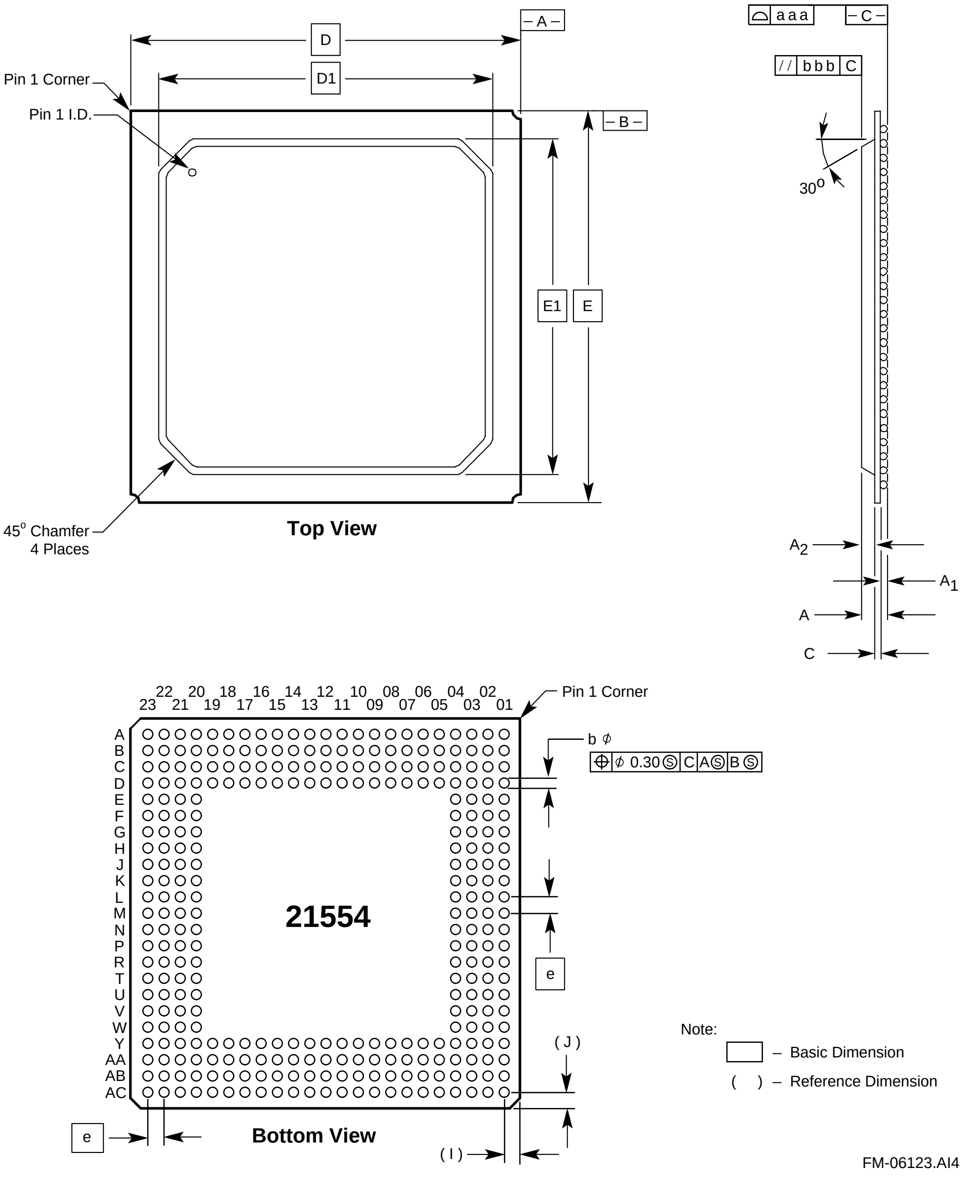


Table 31 lists the package dimensions in millimeters.

Table 31. 304-Point 2-Layer PBGA Package Dimensions

Symbol	Dimension	Minimum Value	Nominal Value	Maximum Value
e	Ball pitch	—	1.27 BSC[1]	—
A	Overall package height	2.12	2.33	2.54
A_1	Package standoff height	0.50	0.60	0.70
A_2	Encapsulation thickness	1.12	1.17	1.22
b	Ball diameter	0.60	0.76	0.90
C	Substrate thickness	0.50	0.56	0.62
aaa	Coplanarity	—	—	0.15
bbb	Overall package planarity	—	—	0.15
D	Overall package width	30.80	31.00	31.20
D_1	Overall encapsulation width	—	26.00	26.70
E	Overall package width	30.80	31.00	31.20
E_1	Overall encapsulation width	—	26.00	26.70
I	Location of first row (*x*-direction)	—	1.53 reference[2]	—
J	Location of first row (*y*-direction)	—	1.53 reference[2]	—

[1] ANSI Y14.5M-1982 American National Standard Dimensioning and Tolerancing, Section 1.3.2, defines Basic Dimension (BSC) as: A numerical value used to describe the theoretically exact size, profile, orientation, or location of a feature or datum target. It is the basis from which permissible variations are established by tolerances on other dimensions, in notes, or in feature control frames.

[2] The value for this measurement is for reference only.

Support, Products, and Documentation

If you need technical support, a *Product Catalog*, or help deciding which documentation best meets your needs, visit the Intel World Wide Web Internet site:

http://www.intel.com

Copies of documents that have an ordering number and are referenced in this document, or other Intel literature may be obtained by calling **1-800-332-2717** or by visiting Intel's website for developers at:

http://developer.intel.com

You can also contact the Intel Massachusetts Information Line or the Intel Massachusetts Customer Technology Center. Please use the following information lines for support:

For documentation and general information:	
Intel Massachusetts Information Line	
United States:	1–800–332–2717
Outside United States:	1–303-675-2148
Electronic mail address:	techdoc@intel.com

For technical support:	
Intel Massachusetts Customer Technology Center	
Phone (U.S. and international):	1–978–568–7474
Fax:	1–978–568–6698
Electronic mail address:	techsup@intel.com