PRELIMINARY

INTEL 380FB PCISET: 82380FB MOBILE PCI-to-PCI BRIDGE (MPCI2)

- **PCI-to-PCI Bridge**
  - **Efficient Repeater Architecture. Mirrors Most Transactions Across the Bridge**
  - **Subtractive Decoding Guarantees that All Accesses Targeted for a Down Stream ISA Bridge (such as the MISA) Arrive at Destination**
  - **Supports the PCI Bus Enumeration Mechanism for PCI-to-PCI Bridges**
  - **High Performance Bridge Supports Fast Back-to-Back agents, and Memory Prefetching**
  - **Supports a 5V Desktop PCI Interface for up to Four Bus Master PCI Add-in Card Slots on the Secondary PCI Bus**
  - **The MISA PCI-to-ISA Bridge Allows a Docking Station to have an Additional Three ISA Slots**
  - **PC/PCI DMA Protocol and PCI Docking Interface Creates a Very Low Pin Count Docking Connector**
- **Full Docking Support**
  - **Notebooks can be Docked with No Pre-conditioning: On, Off, or Suspended (powered-on, to DRAM, or to disk)**
  - **Undocking Mechanism Guarantees Uninterrupted Notebook Operation**
  - **The Same Docking Station can be used with 5V and 3.3V Notebooks**
  - **Supports Automatic Isolation of All Active Docking Connector Signals**
  - **Support for Both Desktop (A/C powered) and Mobile (Battery Powered) Docking Stations**
  - **Non-Volatile Memory Interface to Store Docking Identification, and Notebook Configuration Information**
- **Full Power Management Support for Mobile Docking Stations**
  - **Suspend (Powered-on, to DRAM, and to Disk)**
  - **Resume**
  - **PCI Clockrun Protocol**
  - **Powered-on Suspend/Resume Mode for A/C Powered Desktop Docking Stations**
  - **Low Power Mode Support for Undocked Mobile Docking Stations**
- **208-lead SQFP Package for the 82380FB MPCI2**

The Intel 380FB PCIset (380FB) consists of the 82380FB Mobile PCI-to-PCI Bridge (MPCI2) and the 82380AB Mobile PCI-to-ISA Bridge (MISA). The 380FB supports four PCI slots and three ISA slots. The MPCI2 and MISA can also be used individually to provide either PCI slot expansion or ISA slot expansion.

The 380FB supports a full Hot Docking capable docking station with 5V PCI and ISA add-in expansion slots. MPCI2 provides the docking control for hot insertion, power management, and a PCI-to-PCI bridge to a 5V PCI desktop style add-in bus. Internal arbitration supports four bus masters on the secondary PCI bus. The PC/PCI arbitration interface logic provides PC/PCI bridge support. The 380FB controls all docking, undocking and suspend/resume sequences for the docking station. The E^2PROM interface logic provides an industry standard interface to a non-volatile memory device (E^2PROM) for supporting dynamic autoconfiguration of a previously configured notebook/docking station combination. The Power management logic provides a control and status interface between the docking station and notebook that allows the docking station to control the state of the notebook. A non-volatile memory interface is used to store docking identification and notebook configuration information to speed dynamic configuration for a pre-configured notebook docking combination.

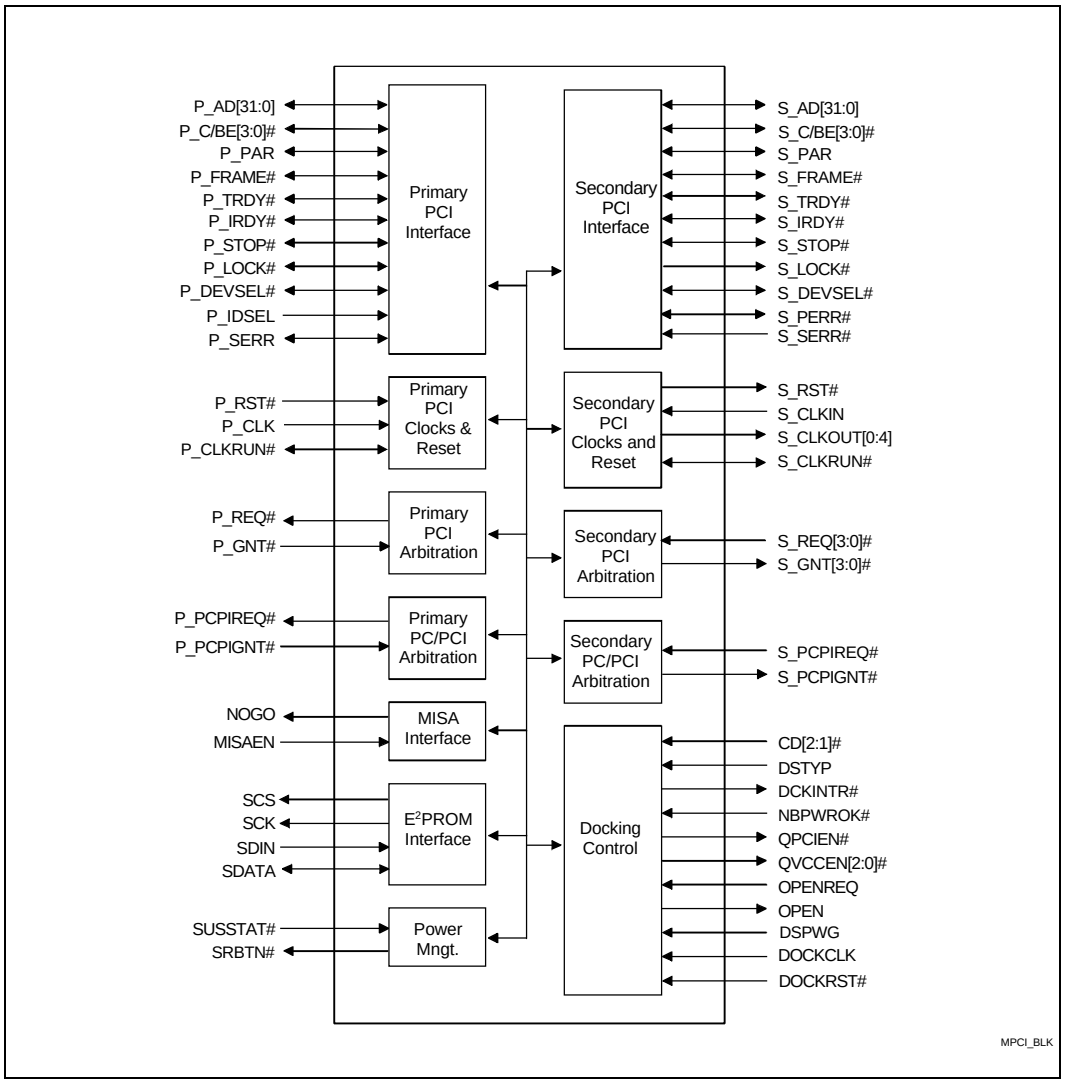
MPCI2 supports the PCI bus enumeration mechanism for PCI-to-PCI bridges. This is needed to support the Windows* 95 dynamic configuration of system resources when the system docks or undocks. Otherwise, the operating system must reset the system after reconfiguration.

February 1997

Order Number: 290567-001

The undocking mechanism of the 380FB guarantees a safe notebook removal. Event notification allows docking resources to be dynamically removed and applications gracefully shut down, if needed. A hardware mechanism is provided to indicate when the notebook is prepared to undock. This can be used to eject or unlock the notebook from the docking station.

The MPCI2's subtractive decoding guarantees that all accesses targeted for a downstream ISA bridge (such as the MISA) arrive at their destination. Software does not need to determine the devices on the ISA bridge and then program positive decode ranges (as is needed on traditional positive decode bridges).



MPCI2 Simplified Block Diagram

CONTENTS

PAGE

1.0. ARCHITECTURE OVERVIEW

The Intel 380FB PCIset consists of the 82380FB MPCI2 and the 82380AB MISA. This PCIset provides the logic necessary to build a full Hot Docking capable docking station with 5V PCI and ISA add-in expansion slots. MPCI2 provides the docking control for hot insertion, power management and a PCI-to-PCI bridge to a 5V PCI desktop style add-in bus. MISA provides a PCI-to-ISA bridge for a 5V ISA desktop style add-in bus. This document describes the MPCI2 component.

MPCI2's Primary and Secondary PCI bus units control the PCI-to-PCI bridge functions. The PCI Arbitration logic provides a secondary PCI arbiter that handles four PCI request grant pairs for docking station slots. The PC/PCI arbitration interface logic provides PC/PCI bridge support. The MISA interface logic provides a proprietary interface between the MPCI2 and the MISA (when present). The Docking Control logic controls all docking, undocking and suspend/resume sequences for the docking station. The E^2PROM interface logic provides an industry standard interface to a non-volatile memory device (E^2PROM) for supporting dynamic autoconfiguration of a previously configured notebook/docking station combination. The Power management logic provides a control and status interface between the docking station and notebook that allows the docking station to know and control the state of the notebook.

The typical 380FB-based docking system is illustrated in Figure 1. This diagram illustrates all of the major components, busses, important logic blocks and interconnects.

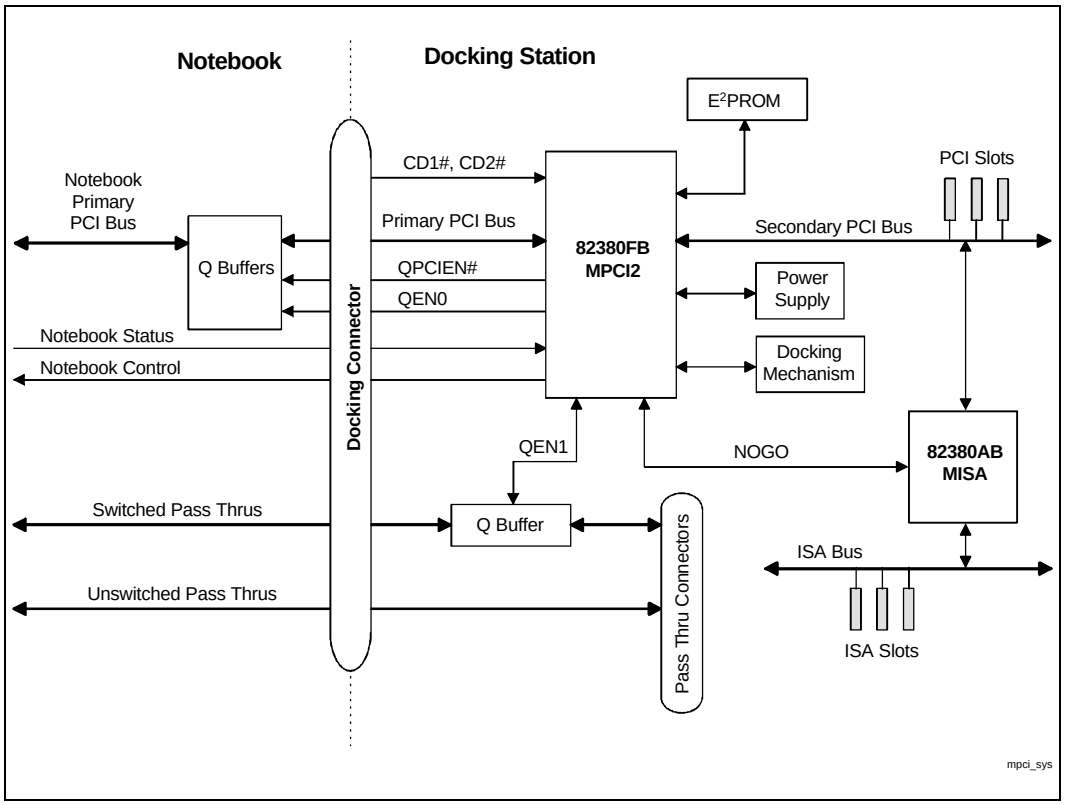


Figure 1. MPCI2 System Block Diagram

2.0. SIGNAL DESCRIPTION

This section provides a detailed description of each signal. The signals are arranged in functional groups according to their associated interface.

The "#" symbol at the end of a signal name indicates that the active, or asserted state occurs when the signal is at a low voltage level. When "#" is not present after the signal name, the signal is asserted when at the high voltage level.

The terms assertion and negation are used extensively. This is done to avoid confusion when working with a mixture of "active-low" and "active-high" signals. The term **assert**, or **assertion** indicates that a signal is active, independent of whether that level is represented by a high or low voltage. The term **negate**, or **negation** indicates that a signal is inactive.

The I/O buffer types are shown below:

Buffer Type	Description
I	input-only signal
O	totem pole output
I/O	bi-direction, tri-state input/output pin
s/t/s	sustained tri-state
od	open drain

2.1. Primary PCI Bus Signals (Docking Connector Side)

Name	Type	Description
P_CLK	I	**Primary PCI Bus System Clock.** P_CLK provides timing for all transactions on the primary PCI bus. All other PCI signals are sampled on the rising edge of PCICLK, and all timing parameters are defined with respect to this edge. PCI supports 0–33 MHz.
P_RST#	I	**Primary PCI Bus Reset.** When P_RST# is active all primary PCI bus signals are tri-stated.
P_AD[31:0]	I/O	**Primary PCI Bus Address and Data.** The standard PCI address and data lines. Address is driven with FRAME# assertion, data is driven or received in following clocks. These signals are connected to the primary PCI bus Address and Data lines from the notebook docking connector.
P_C/BE[3:0]#	I/O	**Primary PCI Bus Command and Byte Enables.** During the address phase of a transaction, P_C/BE[3:0]# define the bus command. During the data phase P_C/BE[3:0]# are used as Byte Enables.
P_PAR	I/O	**Primary PCI Bus Parity Signal.** Parity is even parity across P_AD[31:0] and P_C/BE[3:0]#.
P_FRAME#	s/t/s	**Primary PCI Bus Cycle Frame Signal.** This signal is driven by the current primary PCI bus master to indicate the beginning and duration of an access.
P_IRDY#	s/t/s	**Primary PCI Bus Initiator Ready.** This signal indicates the initiating agent's ability to complete the current data phase of the primary bus transaction.
P_TRDY#	s/t/s	**Primary PCI Bus Target Ready.** This signal indicates the target agent's ability to complete the current data phase of the primary bus transaction.

Name	Type	Description
P_STOP#	s/t/s	**Primary PCI Bus Stop.** This signal indicates the current target is requesting the master to stop the current primary bus transaction.
P_LOCK#	s/t/s	**Primary PCI Bus Lock.** This signal indicates an atomic operation that may require multiple primary PCI bus transactions to complete.
P_IDSEL	I	**Primary PCI Bus Initialization Device Select.** This signal is used as a chip select during configuration read and write transactions. This signal should be externally tied to one of upper 21 address signals.
P_DEVSEL#	s/t/s	**Primary PCI Bus Device Select.** MPCI2 drives P_DEVSEL# to indicate that it is the target of the current primary bus transaction. MPCI2 always subtractively drives this signal.
P_REQ#	I/O	**Primary PCI Bus Request.** MPCI2 drives this signal to request the primary PCI bus on behalf of a secondary PCI bus master.
P_GNT#	I	**Primary PCI Bus Grant.** P_GNT# indicates that the primary PCI bus has been granted to the MPCI, who requested the primary bus on behalf of a secondary PCI bus master or to flush its write post buffers upstream.
P_PCPCIREQ#	I/O	**Primary PCI Bus PC/PCI Request.** This pin meets the PC/PCI DMA specification.
P_PCPCIGNT#	I/O	**Primary PCI Bus PC/PCI Grant.** This pin meets the PC/PCI DMA specification.
P_SERR#	od	**Primary PCI Bus System Error.** MPCI2 pulses system error for one CLK upon detecting reported parity errors on the secondary PCI bus, or any other system error where the result will be catastrophic.
P_CLKRUN#	I/O, od	**Primary PCI Bus Clock Run.** The CLKRUN# central resource negates this signal to generate a primary PCI stop clock request. MPCI2 asserts this signal to request the primary PCI clock to be restarted.

2.2. Secondary PCI Bus Signals (Docking Station Side)

Name	Type	Description
S_CLKOUT[4:0]	O	**Secondary PCI Bus System Clocks (0–33 MHz).** These are five identical PCI clock images for the docking station PCI peripherals. These secondary PCI clocks are stopped when the primary PCI clock is stopped.
S_CLKIN	I	**Secondary PCI Bus System Clock Feedback.** This clock input should be connected to one of the output clocks (SCLKOUT[4:0]). This signal is used by MPCI2 to minimize clock skew.
S_RST#	O	**Secondary PCI Bus Reset.** At power-up S_RST# is asserted and remains asserted until the docking station state machine negates it. S_RST# is asynchronously asserted and is synchronously negated.
S_AD[31:0]	I/O	**Secondary PCI Bus Address and Data.** The standard PCI address and data lines. Address is driven with FRAME# assertion, data is driven or received in following clocks. These signals are connected to the secondary PCI bus Address and Data lines.

Name	Type	Description
S_C/BE[3:0]#	I/O	**Secondary PCI Bus Command and Byte Enables.** During the address phase of a transaction, S_C/BE[3:0]# define the bus command. During the data phase S_C/BE[3:0]# are used as Byte Enables.
S_PAR	I/O	**Secondary PCI Bus Parity.** Parity is even parity across S_AD[31:0] and S_C/BE[3:0]#.
S_FRAME#	s/t/s	**Secondary PCI Bus Cycle Frame.** This signal is driven by the current secondary PCI bus master to indicate the beginning and duration of an access.
S_IRDY#	s/t/s	**Secondary PCI Bus Initiator Ready.** This signal indicates the initiating agent's ability to complete the current data phase of the secondary bus transaction.
S_TRDY#	s/t/s	**Secondary PCI Bus Target Ready.** This signal indicates the target agent's ability to complete the current data phase of the secondary bus transaction.
S_STOP#	s/t/s	**Secondary PCI Bus Stop.** This signal indicates the current target is requesting the master to stop the current secondary bus transaction.
S_LOCK#	s/t/s	**Secondary PCI Bus Lock.** This signal indicates an atomic operation that may require multiple secondary PCI bus transactions to complete.
S_DEVSEL#	s/t/s	**Secondary PCI Bus Device Select.** MPCI2 drives S_DEVSEL# to indicate that it is the target of the current secondary PCI bus transaction. MPCI2 always subtractively drives this signal, except during configuration cycles, when MPCI2 drives S_DEVSEL# during the medium decode phase.
S_REQ[3:0]#	I	**Secondary PCI Bus Request [3:0].** Secondary PCI bus masters drive these signals to MPCI2 to indicate a need to use the bus.
S_GNT[3:0]#	O	**Secondary PCI Bus Grant [3:0].** MPCI2 drives one of these grants active to indicate to the requesting master that the master may initiate a bus cycle on the secondary PCI bus.
S_PCPCIREQ#	I/O	**Secondary PC/PCI Bus Request.** This pin meets the PC/PCI DMA specification.
S_PCPCIGNT#	I/O	**Secondary PC/PCI Bus Grant.** This pin meets the PC/PCI DMA specification.
S_PERR#	I	**Secondary PCI Bus Parity Error.** This pin is driven by a secondary PCI bus agent to indicate a parity error has occurred on the secondary PCI bus.
S_SERR#	I	**Secondary PCI Bus System Error.** This pin is driven by a secondary PCI bus agent to indicate a system error has occurred on the secondary PCI bus.
S_CLKRUN#	I/O, s/t/s pu	**Secondary PCI CLKRUN.** MPCI2 and the other secondary PCI bus agents use this pin to communicate their positions on whether the secondary clocks may currently be stopped or not.

2.3. MPCI2 To MISA Interface

Name	Type	Description
NOGO	O	**NO GO.** Signal to MISA indicating the source and destination of a PCI bus transaction. See section 3.5.
MISAEN	I	**MISA Enable.** MISAEN asserted indicates that a MISA is present in the docking station. This pin must be grounded if the docking station does not use MISA or tied to Vcc if MISA is present in the system.

2.4. Docking Station Management

Name	Type	Description
DCKINTR#	od	**DOCK Event Interrupt.** This output is asserted during any docking event.
NBPWROK	I	**Notebook Power Ok.** A high indicates the notebook power supply is valid. A low indicates the notebook power supply is not valid.
DSPWRGD	I	**Docking Station Power Good.** A high indicates the docking station power supply is valid. A low indicates the docking station power supply is not valid.
SUSSTAT#	I	**Notebook Suspend State.** Low indicates the notebook is in a suspended state. High indicates the notebook is operating.
SRBTN#	od	**Notebook Suspend/Resume Button.**
DOCKCLK	I	**Docking Clock.** 32.768 KHz Oscillator Input. This clock is used to clock the docking state machine.
CD[2:1]#	I	**Docking Station Connector Detects 1 and 2.** These two pins detect when the notebook's docking connector is fully seated with the docking station's docking connector.
OPENREQ	I	**Docking Station Door Open Request.** This signal requests removal of the notebook from the docking station. This signal should be connected to a push-button that can generate an active high pulse with a minimum 128 µs pulse.
OPENACK	O	**Docking Station Door Open Pulse.** MPCI2 uses this signal to initiate an undocking sequence. MPCI2 pulses this signal high for 256 µs after it has prepared the docking station and notebook to undock. External logic uses this signal to unlock the notebook from the docking station, and then eject it from the docking station.
DSTYP	I	**Docking Station Type.** A high indicates a type 1 docking station (docking station power is independent of the notebook power). A low indicates a type 2 docking station (docking station power follows notebook power).
DOCKRST#	I	**Dock Reset.** Resets all the logic powered by VBAT. This signal should be asserted when the backup battery (VBAT) is changed.
QPCIEN#	O	**PCI Q-Buffers Enable.** Enables/Disables the PCI Q-buffers.
QEN0	O	**PCI and Quiet Dock Q-Buffers Enable.** This signal is used to turn on the PCI and Quiet Dock Q-buffers.

Name	Type	Description
QEN[2:1]	O	**PCI and Quiet Dock Q-Buffers Enable.** QEN[2:1] are general purpose outputs controlled by CONCNTL register.
SCK	O	**Serial E²PROM clock.** This pin is driven to reflect the state of SIF_BURSTEN register bit 0.
SCS	O	**Serial E²PROM Chip Select.** This pin is driven to reflect the state of SIF_BURSTEN register bit 1.
SDIN	I	**Serial E²PROM Input Data.** The state of bit 4 of the SIF_BURSTEN register follows this input pin.
SDATA	I/O	**Serial E²PROM Data.** Data direction determined by SIF_BURSTEN register bit 10. As an output, the pin is driven to reflect the state of SIF_BURSTEN register bit 5. As an input, the state of bit 5 of the SIF_BURSTEN register follows this pin.

2.5. Power, Ground, and Test

Name	Type	Description
VBAT		**Backup Battery Power.** This power input is used to supply power to the docking state machine. Power should be supplied to this pin by the normal docking power source when on, or by a backup battery (VBAT) when off.
V_{CC}		**Power.** Powered by the 5V docking station power source.
V_{SS}		**Grounds.** These signals should be connected to the system ground.
NC		**No Connects.** These pins should be left floating.
TEST#		**Test Input.** This signal should be pulled HIGH through a pull-up resistor (10 kΩ).

3.0. REGISTER DESCRIPTION

The MPCI2 contains only PCI Configuration registers. These registers are accessible only by primary PCI masters. The registers can be accessed as Byte, Word (16-bit), or DWord (32-bit) quantities. All multi-byte numeric fields use “little-endian” ordering (i.e., lower addresses contain the least significant parts of the field). The following nomenclature is used for access attributes.

RO **Read Only**. If a register is read only, writes to this register have no effect.
R/W **Read/Write**. A register with this attribute can be read and written.
R/WC **Read/Write Clear**. A register bit with this attribute can be read and written. However, a write of 1 clears (sets to 0) the corresponding bit and a write of 0 has no effect.

Some of the MPCI2 registers described in this section contain reserved bits. Software must deal correctly with fields that are reserved. On reads, software must use appropriate masks to extract the defined bits and not rely on reserved bits being any particular value. On writes, software must ensure that the values of reserved bit positions are preserved. That is, the values of reserved bit positions must first be read, merged with the new values for other bit positions and then written back.

In addition to reserved bits within a register, MPCI2 contains address locations in the PCI configuration space that are marked “Reserved” (Table 2). MPCI2 responds to accesses to these address locations by completing the Host cycle. Software should not write to reserved MPCI2 configuration locations in the device-specific region (above address offset 3Fh).

During a hard reset (RST# asserted), MPCI2 sets its internal configuration registers to predetermined **default** states. The default state represents the minimum functionality feature set required to successfully bring up the system. Hence, it does not represent the optimal system configuration. MPCI2 configuration registers assume their default conditions when:

- A low level occurs on any of the following pins: DOCKRST#, P_RST#, DSPWRGD, NBPWROK, SUSSTAT#

  OR
- A high level on any of the following pins: CD1#, CD2#

Table 1. MPCI2 Configuration Space

Address Offset	Register Symbol	Register Name	Access
PCI Specific Registers			
01–00h	VID	Vendor Identification	RO
02–03h	DID	Device Identification	RO
04–05h	P_PCICMD	Primary bus PCI Command Register	R/W
06–07h	P_PCISTS	Primary bus PCI Status Register	R/W
08h	RID	Revision Identification	RO
09–0Bh	CCODE	Class Code	RO
0Ch	CACHE_LS	Cache Line Size	RO
0Dh	P_MLT	Primary Bus Master Latency Timer	R/W
0Eh	HEADT	Header Type	RO
0F–17h	—	Reserved	—
18h	P_BUSNUM	Primary PCI Bus Number	R/W
19h	S_BUSNUM	Secondary PCI Bus Number	R/W
1Ah	SUB_BUSNUM	Subordinate PCI Bus Number	R/W
1B–1Dh	—	Reserved	—
1E–1Fh	S_PCISTS	Secondary PCI bus Status Register	R/W
20–23h	—	Reserved	—
24–25h	PMEMBASE	Prefetchable Memory Base Register	R/W
26–27h	PMEMLMT	Prefetchable Memory Limit Register	R/W
28–3Dh	—	Reserved	—
3E–3Fh	BRDG_C	Bridge Control Register	R/W
MPCI2 Specific Registers			
40h	CONCNTL	Connector Control Register	R/W
41h	CONEVNT	Connector Event Register	R/W
42–43h	—	Reserved	—
44–45h	SIF_BURSTEN	Serial Bus Interface/Burst Enable Register	R/W
46h	MISC_STS	MPCI2 Miscellaneous Status Register	R/W
47–4Bh	—	Reserved	—
4C–4Dh	SPCYC_MSG	Special Message Encode Register	R/W
4Eh–FFh	—	Reserved	—

3.1. VID—Vendor Identification Register

Address Offset: 00–01h
Default Value: 8086h
Attribute: Read Only

The VID Register contains the Intel vendor identification number (8086h). This 16-bit register combined with the Device Identification Register (DEVID) uniquely identifies MPCI2 device. Writes to this register have no effect.

Bit	Description
15:0	**Vendor Identification Number.** This is a 16-bit value assigned to Intel (8086h).

3.2. DID—Device Identification Register

Address Offset: 02–03h
Default Value: 124Bh
Attribute: Read Only

This register contains the device identification number. This register along with the Vendor ID (VID), uniquely identifies the MPCI2. Writes to this register have no effect.

Bit	Description
15:0	**Device Identification Number.** This is a 16-bit value assigned to MPCI2.

3.3. P_PCICMD—Primary Bus PCI Command Register

Address Offset: 04–05h
Default Value: 0080h
Attribute: Read/Write

The P_PCICMD register controls cycle transfers, other than configuration cycles, between the primary PCI bus and secondary PCI bus. Configuration cycles are always transferred by MPCI2, regardless of the state of this register.

Bit	Description
15:9	**Reserved.**
8	**P_SERR# Enable.** 1=Enable P_SERR# pin. 0=Disable.
7	**Reserved.** Returns a 1 on a Read operation.
6:3	**Reserved.**
2	**Secondary-to-primary Bus Enable.** When this bit is set (and bits[1:0]=11), secondary-to-primary PCI transfers are enabled.
1:0	**Primary-to-secondary Bus Enable.** 11=Enables cycle transfer from the primary PCI bus to the secondary PCI bus. 00=Disable. 01 and 10 values are not supported.

3.4. P_PCISTS—Primary PCI Status Register

Address Offset: 06–07h
Default Value: 0400h
Attribute: Read/Read Write Clear

The P_PCISTS register provides primary PCI bus event information. Reads to this register behave normally. Status bits in this register are set by events on the MPCI2 primary PCI bus. Software sets these bits to 0 by writing a 1 to them.

Bit	Description
15	**Reserved.**
14	**Signaled System Error—R/WC.** This bit is set when MPCI2 asserts P_SERR# on the primary PCI bus.
13	**Received Master Abort—R/WC.** This bit is set when a MPCI2 initiated cycle on the primary PCI bus terminates with a master abort.
12	**Received Target Abort—R/WC.** This bit is set whenever a MPCI2 initiated cycle on the primary PCI bus terminates with a target abort.
11	**Signaled Target Abort—R/WC.** This bit is set whenever the MPCI2 as a target on the primary interface terminates a transaction with a target abort.
10:9	**DEVSEL# Timing—RO.** This 2-bit field is read only and always returns a 10b. Since MPCI2 performs positive decode on configuration cycles and subtractive decode on all others, these bits will not be used by the system.
8:0	**Reserved.**

3.5. RID—Revision Identification Register

Address Offset: 08h
Default: See latest stepping information
Access: Read Only

This read only register contains the revision number of each MPCI2 component.

Bit	Description
7:0	**Revision Identification Number.** Indicates the revision identification number for the MPCI2. (00h indicates A-0 stepping)

3.6. CCODE—Class Code Register

Address Offset: 09–0Bh
Default Value: 060480h
Attribute: Read Only

This register indicates MPCI2's class code as defined in the PCI Specification.

Bit	Description
23:0	**Base Class Code.** Value=06h indicating a Bus Bridge.
15:0	**Sub-Class Code.** Value=04h indicating a PCI-to-PCI Bridge.
7:0	**Programming Interface.** Value=80h. Bit 7 High indicates a subtractive decode bridge.

3.7. CACHE_LS—Cache Line Size Register

Address Offset: 0Ch
Default Value: 08h
Attribute: Read Only

The Cache Line Size register is an 8-bit read-only register that specifies the system cache line size in units of 32-bit words. MPCI2 only allows a cache line size of 32 bytes. This register is used when accessing a non-prefetchable area using the memory read line (MRL) or memory read multiple (MRM) commands. In these cases, MPCI2 terminates the access on any 32-byte aligned address. A non-prefetchable memory area is defined as any burstable read access (MRM, MRL, MR) that is:

- Upstream with the UP_BPF_EN bit reset low
- Downstream and does not fall within the prefetchable memory window

Bit	Description
7:0	**Cache Line Size.** 08h indicates that the MPCI2 cache line size is fixed at 32 bytes.

3.8. P_MLT—Primary Bus Master Latency Timer

Address Offset: 0Dh
Default Value: 00h
Attribute: Read/Write

This register specifies, in units of PCI bus clocks, the value of MPCI2's master latency timer when MPCI2 flushes its write post buffers on the primary bus after an upstream write has been completed by a secondary bus master. The counter is used in the following two ways:

1. During the initial write burst from a secondary PCI bus master, once the secondary PCI bus master terminates its cycle and the secondary PCI bus becomes idle, the master latency timer begins to count down. If the counter expires before the write post buffer flushing ends and MPCI2's P_GNT# signal is sampled negated at that time, MPCI2 terminates its flushing onto the primary PCI bus after one more data phases, and re-asserts P_REQ# to continue flushing at a later time. At this time, another primary bus master may get on the bus and generate transactions. While primary-to-primary PCI bus transactions

progress unimpeded, any primary-to-secondary PCI bus transaction, other than a write, are retried by MPCI2. Any write that must cross MPCI2 are disconnected at the first data phase.

If the timer does not expire before the flushing finishes, or if it does expire but P_GNT# is still asserted, MPCI2 will continue flushing the post buffers until all data is written out.

2. If an initial write burst from a secondary PCI bus master is disconnected by a primary PCI bus target, MPCI2 will request the bus via P_REQ# to flush the remaining data in its write post buffers. When P_GNT# is given and MPCI2 starts flushing, it enables the master latency timer to count down at the start of the cycle (P_FRAME# assertion). If the timer expires before the write post buffer flushing ends and MPCI2's P_GNT# signal is sampled negated at that time, MPCI2 terminates its flushing on the primary PCI bus after one more data phases, and re-asserts P_REQ# to continue flushing at a later time. At this time, another primary bus master may get on the bus and generate transactions. While primary-to-primary transactions progress unimpeded, any primary-to-secondary transaction, other than a write, will be retried by MPCI2. Any write that must cross MPCI2 will be disconnected at the first data phase.

   If the timer does not expire before the flushing finishes, or if it does expire but P_GNT# is still asserted, MPCI2 continues flushing the write post buffers until all data is written out.

   When the MPCI2 is not flushing posted data, it relies on the initiating master's latency timer. In these cases, MPCI2 is simply generating cycles on behalf of a requesting bus master, and the cycles are subject to that master's latency timer. Thus, this register has no impact on other types of read or write cycles such as read bursting or prefetching.

   When the MPCI2 is flushing posted data on a PCPCIGT# (Cascade Channel) the Master Latency Timer is disabled.

Bit	Description
7:6	**Reserved.**
5:4	**Timer Value.** This field determines the master latency timer value as follows: **Bits[5:4] Latency Time** 00 timer disabled (no time-out) 01 16 clocks 10 32 clocks 11 0 clocks (immediate time-out)
3:0	**Reserved.**

3.9. HEADT—Header Type Register

Address Offset: 0Eh
Default Value: 01h
Attribute: Read Only

This register indicates the configuration space format.

Bit	Description
7	**Multifunction Device.** 0 indicates a single function device.
6:0	**Layout Code.** The value of 000 0001b indicates that the configuration register space is that of a PCI-to-PCI bridge. The layout is defined in *PCI-to-PCI Bridge Architecture Specification.*

3.10. P_BUSNUM—Primary PCI Bus Number Register

Address Offset: 18h
Default Value: 00h
Attribute: Read/Write

This register contains MPCI2's primary PCI bus number.

Bit	Description
7:0	**Primary Bus Number.** MPCI2's primary PCI bus number.

3.11. S_BUSNUM—Secondary PCI Bus Number Register

Address Offset: 19h
Default Value: 00h
Attribute: Read/Write

This register contains MPCI2's secondary PCI bus number.

Bit	Description
7:0	**Secondary Bus Number.** MPCI2's secondary PCI bus number.

3.12. SUB_BUSNUM—Subordinate PCI Bus Number Register

Address Offset: 1Ah
Default Value: 00h
Attribute: Read/Write

This register contains the number of the highest numbered PCI bus that is behind the MPCI2 (see PCI-to-PCI Bridge Operations section for additional information).

Bit	Description
7:0	**Subordinate Bus Number.** Number of the last PCI bus that is behind the MPCI2.

3.13. S_PCISTS—Secondary PCI Status Register

Address Offset: 1E–1Fh
Default Value: 0400h
Attribute: Read Only, Read/Write Clear

The S_PCISTS register provides secondary PCI bus event information. Reads to this register behave normally. Status bits in this register are set by events on the MPCI2 secondary PCI bus. Software sets the R/WC bits in this register to 0 by writing a 1 to them.

Bit	Description
15	**Reserved.**
14	**Received System Error—R/WC.** This bit is set when the MPCI2 detects S_SERR# on the secondary PCI bus.
13	**Received Master Abort—R/WC.** This bit is set when a MPCI2 initiated cycle on the secondary PCI bus terminates with a master abort.
12	**Received Target Abort—R/WC.** This bit is set when a MPCI2 initiated cycle on the secondary PCI bus terminates with a target abort.
11	**Signaled Target Abort—R/WC.** This bit is set when the MPCI2, as a target on the secondary PCI bus, terminates a transaction with a target abort.
10:9	**DEVSEL# Timing—RO.** This 2-bit field is read only and always returns a 10b. Since MPCI2 performs positive decode on configuration cycles and subtractive decode on all others, these bits will not be used by the system.
8	**Data Parity Detected.** Hardwired to 0. MPCI2 does not check parity.
7:0	**Reserved.**

3.14. PMEMBASE—Prefetchable Memory Base Register

Address Offset: 24–25h
Default Value: 0010h
Attribute: Read/Write, Read Only

This register specifies the upper 12 bits of the base address for the prefetchable memory address range. AD[19:0] is assumed to be all 0's. For a description of MPCI2's Read prefetch operations see the PCI-to-PCI Bridge Operations section.

Bit	Description
15:4	**AD[31:20]—R/W.** Bits [15:4] correspond to AD[31:20], respectively. This field locates the prefetchable memory base address on 1 MB boundaries.
3:0	**64-bit addressing—RO.** Hardwired to 0s. The MPCI2 supports only 32-bit addressing.

3.15. PMEMLMT— Prefetchable Memory Limit Register

Address Offset: 26–27h
Default Value: 0000h
Attribute: Read/Write, Read Only

This register specifies the upper memory address for the prefetchable memory address range. AD[19:0] is assumed to be all 1's. For a description of MPCI2's Read prefetch operations, see the PCI-to-PCI Bridge Operations section.

Bit	Description
15:4	**AD[31:20] —R/W.** Bits [15:4] correspond to AD[31:20], respectively. This field locates the prefetchable memory upper address on 1 MB boundaries.
3:0	**64-bit addressing—RO.** Hardwired to 0s. The MPCI2 supports only 32-bit addressing.

3.16. BRDG_C—Bridge Control Register

Register Location: 3F–3Eh
Default Value: 0000h
Attribute: Read/Write

This register controls the forwarding of S_SERR# assertions to P_SERR# on the primary PCI bus.

Bit	Description
15:2	**Reserved.**
1	**S_SERR#-to-P_SERR# Enable.** 1=Enable forwarding of S_SERR# to P_SERR# (P_SERR# Enable bit in the P_PCICMD register must be set to 1). 0=Disable.
0	**Reserved.**

3.17. CONCNTL—Connector Control Register

Register Location: 40h
Default Value: 00h
Attribute: Read/Write

This register contains the two power enable signals for two docking station isolation buffers: one for the parallel port and floppy disk drive, and the other for the external keyboard and mouse ports.

Bit	Description
7:4	**Reserved.**
3	**QEN2 Output Signal.** 1=QEN2 output is high. 0=QEN2 output is low.
2	**QEN1 Output Signal.** 1=QEN1 output is high. 0=QEN1 output is low.
1:0	**Reserved.**

3.18. CONEVNT—Connector Event Register

Register Location: 41h
Default Value: Read: 11xx 1111b
Attribute: Read/Write, Read Only

This register provides a combination of status bits reflecting how several configuration pins are set, and an interrupt status register for the DCKINTR# interrupt. For a detailed description of the docking process, see the Docking Control Logic section.

NOTE

Reads from this register return the inverted values of the register bits.

Bit	Description
7	**Reserved.** Read as 1.
6	**Power-on Suspend (PWRONSUS)—R/W.** This bit controls the MPCI2 powered-on suspend state and should be set to 0 during normal operation (see Power Management section).
5	**MISAEN Pin Status—RO.** This bit reflects the inverted state of the MISAEN pin.
4	**DSTYP Pin Status—RO.** This bit reflects the inverted state of the DSTYP pin.
3	**UDKPERMIT—R/W.** 1=DCKINTR# pin is asserted. MPCI2's docking state machine starts an undock sequence when this bit is cleared (set to 0). Software must wait a minimum of 32 µs (one dock clock period) between setting and clearing this bit or the event may not be recognized. When this bit is set to 0, MPCI2's docking state machine pulses the OPENACK signal for a duration of 256 µs to signal external hardware to "eject" the notebook.
2	**Reserved.** Read as 1.
1	**DOCKED—R/W.** MPCI2 sets this bit when its internal docking state machine completes its docking sequence. The MPCI2 then asserts its DCKINTR# pin. DCKINTR# can be negated by clearing the OPENREQ, DOCKED, and UDKPERMIT bits in this register.
0	**OPENREQ Pin.** MPCI2 sets this bit to 1 when the OPENREQ input pin is pulsed (asserted high). Software can assert the DCKINTR# pin by writing a 1 to this bit. DCKINTR# can be negated by clearing the OPENREQ, DOCKED, and UDKPERMIT bits in this register.

3.19. SIF_BURSTEN—Serial Interface and Burst Enable Register

Address Offset: 44–45h
Default Value: 0000 0000 100x 0000b
Attribute: Read/Write

This register controls MPCI2 memory read bursting, prefetching, and memory write posting (see the PCI-to-PCI Bridge Operations section for more details). SIF_BURSTEN also controls the E^2PROM interface of the MPCI2.

Bit	Description
15:13	**Reserved.**
12	**Discard Write Posted Data Enable (EXP2SERR_EN).** 1=P_SERR# is asserted when MPCI2 discards posted write data and P_PCICMD [bit 8]=1.
11	**Reserved.**
10	**SDATA Signal Direction (SDATA_DIR).** 0=Output, 1=Input.
9	**Write Posting Enable (WR_POST_EN).** 1=Enable. 0=All write bursts are disconnected by MPCI2 at the first data phase.
8	**Read Bursting Enable (RD_BURST_EN).** 1=Enables non-prefetchable memory burst cycles (in both directions) to be translated into non-prefetchable memory read burst cycles. 0=Any non-prefetchable memory read burst access that crosses the MPCI2 bridge will be disconnected at the first data phase.
7	**Upstream Blind Prefetch Enable (US_BPF_EN).** 1=Enable. 0=Disable. Enables blind prefetching by MPCI2 for any upstream memory read burst cycle.
6	**Reserved.**
5	**SDATA Signal State (SDATA_SS).** When SDATA_DIR (bit 10)=0, this bit programs the state of the SDATA pin (1=SDATA asserted. 0=SDATA negated). When SDATA_DIR=1, this bit reflects the state of the SDATA pin (1=SDATA asserted. 0=SDATA negated).
4	**SDIN Signal State (SDIN_SS).** 1=SDIN asserted (high). 0=SDIN negated (low).
3	**S_PERR#-to-P_SERR# Map Disable.** 0=Pulse P_SERR# when S_PERR# is asserted (if P_SERR enable bit is set). 1=Do not pulse P_SERR# when S_PERR# is asserted.
2	**Cascade Determination Delay/Frame Determination Delay (CDD/FDD).** 1=10 P_CLKs. 0=20 P_CLKs. This bit is used during PC/PCI transactions to determine the location of the bus master and to set the FRAME# delay during PC/PCI DMA retries.
1	**Serial E^2PROM Chip Select (SCS).** 1=SCS Output Pin asserted (high). 0=Negated.
0	**Serial E^2PROM Clock (SCK).** 1=SCK Output Pin asserted (high). 0=Negated.

3.20. MISC_STS—MPCI2 Miscellaneous Status Register

Address Offset: 46h
Default Value: 00h
Attribute: Read/Write Clear

This register contains MPCI2 miscellaneous status bits that are not recorded in the PCI standard status registers.

Bit	Description
7:1	**Reserved.**
0	**STEXP_ERR.** 1=P_SERR# was asserted by MPCI2 after invalidation of posted write data. Software resets this bit by writing a 1 to it.

3.21. SPCYC_MSG—Special Cycle Message Encode Register

Register Location: 4C–4Dh
Default Value: FFFEh
Attribute: Read/Write

This register contains the message to be broadcast in the Encoded Message portion of a Special Cycle forwarded by MPCI2.

Bit	Description
15:0	**Message Encode.** Encoded message portion of Special Cycle forwarded by MPCI2. See the *PCI Local Bus Specification* and *PCI to PCI Bridge Specification* for more details.

4.0. FUNCTIONAL DESCRIPTION

4.1. PCI-to-PCI Bridge Operations

MPCI2 operates as specified in *PCI Local Bus Specification*, revision 2.1 and *PCI to PCI Bridge Architecture Specification*, revision 1.0, with the options and exceptions described in this section.

MPCI2 uses a subtractive decoding mechanism rather than memory and I/O base and limit registers to determine where to forward cycles. All cycles initiated on the primary bus are repeated on the secondary bus, except when MPCI2 is flushing its write buffers downstream or the secondary bus is busy finishing up a previous cycle. All cycles initiated on the secondary bus are repeated on the primary bus, except for interrupt acknowledge cycles. Memory write posting and interrupt acknowledge cycles are discussed later in this section. If an MISA device (PCI-to-ISA Bridge) is present in the system, memory or I/O cycles not claimed by a primary or secondary bus target are assumed to be subtractively claimed by the PCI to ISA bridge.

MPCI2 does not support Fast Back-to-Back cycles of any type (even to the same agent) with one exception: two Fast Back-to-Back cycles on the primary bus in which the first cycle is two clocks long.

4.1.1. CONFIGURATION CYCLES

MPCI2 repeats or converts all PCI configuration cycles in both directions (either primary-to-secondary or secondary-to-primary transfers). All type 0 configuration cycles are converted across MPCI2 as Configuration type 1 cycles with AD[23:2] all high. This includes primary interface type 0 configuration cycles that are targeted at MPCI2. Configuration type 0 burst cycles to MPCI2 are disconnected after the first data phase by MPCI2.

Configuration type 1 cycles that are to be ignored per the PCI to PCI Bridge Architecture Specification, revision 1.0, are forwarded by MPCI2 to the non-initiating bus as Configuration type 1 cycles with AD[23:2] all high.

4.1.2. SPECIAL CYCLES

Special cycles received by MPCI2 on the initiating bus are converted to special cycles on the non-initiating bus. This cycle is used to maintain synchronization between the Primary and Secondary busses. The message portion of the converted special cycle is programmable using the Special Cycle Message Encode Register (4Ch–4Dh).

4.1.3. INTERRUPT ACKNOWLEDGE CYCLES

MPCI2 forwards Interrupt Acknowledge cycles, unchanged, in the downstream direction. Interrupt Acknowledge cycles on the secondary bus will be ignored by MPCI2.

4.1.4. SUBTRACTIVE DECODE

MPCI2 and MISA implement a proprietary signal (NOGO signal) so that the target of a secondary bus master can be communicated to MISA. When S_FRAME# is asserted by a device other than MISA, source and target bus information is communicated to MISA. MPCI2 uses the NOGO signal to inform MISA about the originating bus. NOGO is sampled by MISA on the first PCICLK after S_FRAME# is asserted. NOGO is high if the master is on the Primary PCI bus. NOGO is low if the master is on the secondary PCI bus. If MISA is the originator of the cycle, MPCI2 ignores NOGO.

For secondary bus masters, MPCI2 acts as the subtractive decode agent and asserts S_DEVSEL#, if the cycle is not claimed by another device. For primary bus masters, MISA acts as the system's subtractive decode agent.

If the bus master is on the primary bus and MISA claims the cycle by asserting DEVSEL#, it is assumed that the target is on the ISA bus. If the bus master is on the secondary bus and MPCI2 claims the cycle, the target may reside on the primary PCI bus or ISA bus. MPCI2 monitors the P_DEVSEL# signal to check for a primary bus target. If the cycle is not claimed on the primary bus before the subtractive decode phase, MPCI2 assumes that the target is on the ISA bus. MPCI2 indicates this to MISA on the NOGO signal three clocks after the secondary bus decode phase. A high indicates that the target is on the primary bus. A low indicates that the target is on the ISA bus.

4.1.5. MEMORY AND I/O CYCLES

The Primary Bus Enable field (bits[1:0]) in the P_PCICMD register control access to I/O and memory space. These accesses must be enabled (bits[1:0]=11) for MPCI2 to repeat cycles across the bridge.

4.1.5.1. Memory and I/O Transaction Control

For a transaction on the primary interface, if a memory or I/O transaction is not claimed before the subtractive decode phase, MPCI2 asserts P_DEVSEL#. If the transaction is not claimed on the secondary interface, MPCI2 terminates the transaction.

The secondary interface may contain an MISA device which is a subtractive decode agent. MPCI2 and MISA implement a proprietary NOGO protocol interface which determines the owner of a secondary bus cycle that is unclaimed by a positive decode device. If a transaction initiated on the secondary bus is not positively claimed by a PCI device, MPCI2 asserts S_DEVSEL#. If the transaction is not claimed on the primary interface, MPCI2 passes the cycle to MISA using the NOGO protocol or terminates the transaction if MISA is not present.

MPCI2 identifies an initiating bus master accessing an initiating bus target by the DEVSEL# being positively asserted on the initiating bus. During these peer-to-peer transactions, MPCI2 normally repeats the cycle on the non-initiating bus including the terminating mechanism.

For I/O writes, configuration writes, special cycles, locked memory write cycles, or cycles initiated by any master on the x_PCPCIGNT# signal:

- If the cycle attempted is a single cycle access, data is written to the non-initiating bus before the cycle is completed on the initiating bus.
- If the cycle attempted is a burst access, MPCI2 converts the burst to single cycle transfers by performing disconnect, retry, or abort at the first data phase (depending on the cycle termination on the non-initiating bus). For each of these single cycles, data is written to the non-initiating bus before the cycle is completed on the initiating bus.

4.1.5.2. Write Posting

MPCI2 supports write bursting of up to 1-Kbyte DWords. Data is posted on memory write or memory write invalidate cycles per the *PCI to PCI Bridge Architecture Specification* with the following exceptions and clarifications:

1. The initiating bus master is disconnected if the write buffer is full or if the data phase address hits a 4-KB aligned boundary.
2. Data is not posted during locked cycles.
3. Data is not posted during PC/PCI DMA cycles.
4. WR_POST_EN bit in the SIF_BURSTEN register (44h) must be set to enable write posting.
5. Data must be in linear burst order (AD[1:0]=00) to be posted.

When MPCI2 has valid posted data in either direction

1. MPCI2 does not pass PCI or PC/PCI grants to the secondary bus.
2. S_PCPCIREQ# is passed to the primary bus.
3. MPCI2 asserts P_CLKRUN# if any device attempts to stop P_CLK.
4. If a PCPCI DMA cycle is retried, MPCI2 keeps P_PCPCIREQ# active. If P_PCPCIGNT# is still pending after flushing is complete, it is passed to S_PCPCIGNT#. The first cycle through the bridge after flushing is completed is delayed by FDD clocks (see SIF_BURSTEN register, bit 2) after S_PCPCIGNT# is asserted.

When MPCI2 has valid posted data pointing in the downstream direction

1. MPCI2 ensures that the secondary PCI bus is idle and then initiates its flush.
2. Primary PCI bus cycles claimed by primary bus targets are ignored by MPCI2.
3. Primary PCI bus cycles not claimed on the primary bus are subtractively claimed and retried. They are not reflected on the secondary PCI bus.

When MPCI2 has valid posted data pointing in the upstream direction

1. Downstream memory, I/O, and configuration write cycles (not PC/PCI DMA) pass to the secondary PCI bus unposted. For burst cycles, the cycle is disconnected at the first data phase and converted to a single cycle on the secondary PCI bus.
2. All other downstream cycles are retried.
3. Posted data is flushed upstream when MPCI2 receives a P_GNT# or intercepts P_PCPCIGNT# intended for an ISA bus master.
   a. See the Arbitration section for a description of MPCI2's use of P_PCPCIGNT# during flushing of write posts buffers.
   b. When using P_GNT#, MPCI2 uses the Master Latency Timer register per the PCI Local Bus Specification, revision 2.1. The Master Latency Timer is not used when MPCI2 uses P_PCPCIGNT# since such flushing cycles are single data phase cycles.

If the docking station goes into the powered-off suspend when MPCI2 has posted data to be flushed, the write post buffer is invalidated (data is lost). However, since MPCI2 invalidates its write post buffer 64 µs after SUSSTAT# is sampled asserted (low), if software handles the suspend process correctly, there should be no active masters on the PCI bus and MPCI2 will have flushed all posted data long before the invalidation occurs.

Posted write data is invalidated 1 ms after it is posted. The timer for discarding write data is a 5 bit counter clocked by the 32-KHz DOCKCLK. When the MPCI2 posted write data discard timer expires, P_SERR# is asserted (and bit 0 of the MISC_STS register is set), if enabled via bit 12 in the SIF_BURSTEN register and bit 8 in the P_PCICMD register.

4.1.5.3. Memory Read Bursting and Prefetching

MPCI2 is capable of bursting and prefetching memory read data that crosses the bridge. Memory Read Line (MRL) and Memory Read Multiple (MRM) cycles are always burstable and prefetchable in both directions (except when MPCI2 is in PC/PCI DMA Retry mode). The handling of Burst Memory Read cycles that cross the MPCI2 bridge is shown in Table 2. Prefetching is controlled via the SIF_BURSTEN register (44h). Prefetchable Memory address range is described in the *PCI to PCI Bridge Architecture Specification*.

Table 2. Bursting and Prefetching During Burst Memory Read Cycles Crossing the Bridge

Master Bus	US_BPF_EN	RD_BRST_EN	PA Range	MPCI2 Cycle
Secondary	0	0	x	SRD
Secondary	0	1	x	BRD
Secondary	1	x	x	PBRD
Primary	x	0	out of range	SRD
Primary	x	1	out of range	BRD
Primary	x	x	in range	PBRD

NOTES:

1. SRD=Single Memory Read cycle on the target bus.
2. BRD=Burst Memory Read cycle on the target bus.
3. PBRD=Prefetchable Burst Memory Read cycle on the target bus. Read data is prefetched.

MRL, MRM, and Burst Memory Read cycles during prefetching within the Prefetchable Memory Address Range are disconnected on the target bus, if the address reaches a 4-KB address boundary. The target bus is disconnected at the next 32-byte boundary, if these cycles are outside the Prefetchable Memory Address Range for downstream cycles or, for upstream cycles, if bit 7 (upstream blind prefetch enable) in the SIF_BURSTEN register is 0.

If a target abort is issued and there is valid prefetched data in MPCI2's buffers, MPCI2 continues feeding data to the master until it asks for the target aborted address. MPCI2 then target aborts the master. If the master terminates prior to reaching the target aborted address, the target abort is not passed to the initiating bus but the received target abort status bit in the target bus's status register is set.

4.1.6. PARITY SUPPORT

Address and data are not checked for parity errors. MPCI2 generates PAR for each cycle transferred across the bridge in either direction.

4.1.7. ERROR SUPPORT

MPCI2 generates parity for all accesses that cross the bridge; however, MPCI2 does not check parity. If MPCI2 receives a transaction on one bus with incorrect parity, it will generate a second transaction on the other side with new parity corresponding to the received address, data, and byte enables on the initiating bus. Parity errors on the secondary bus indicated by S_PERR# are forwarded to the primary bus P_SERR# signal, if enabled by the SIF_BURSTEN register.

As described in the *PCI to PCI Bridge Architecture Specification*, MPCI2 pulses P_SERR# upon detecting S_SERR# active, if enabled by the P_PCICMD and BRDG_C registers.

4.1.8. ARBITRATION

MPCI2 supports secondary bus PCI masters and the PC/PCI DMA expansion protocol. MPCI2 provides an arbiter that prioritizes system resources for up to four secondary PCI bus masters. PC/PCI requests are passed to the primary bus for arbitration by an upstream PC/PCI arbiter.

4.1.8.1. Secondary PCI Bus Master Arbitration

P_REQ# Generation

MPCI2 asserts P_REQ# when a secondary PCI bus master asserts one of the S_REQ[3:0]# inputs or when there is valid write data to flush in the upstream direction. P_REQ# remains asserted until all S_REQ[3:0]# inputs are inactive (negated) and the upstream write buffers are flushed.

The Secondary PCI Bus and Primary PCI Bus Enable bits in the P_PCICMD register must be set to enable the arbiter. Otherwise, MPCI2 ignores secondary bus master requests.

S_GNT[3:0]# Generation

The highest priority secondary PCI bus requester is granted the bus after P_GNT# is received from the system's primary bus arbiter, both busses are idle, and MPCI2's internal buffers are empty.

The asserted S_GNT[3:0]# remains active until S_FRAME# is asserted or 12 clocks have elapsed since S_GNT# assertion without S_FRAME# assertion. MPCI2 only supports secondary PCI bus masters that assert S_FRAME# within 12 PCI clocks after S_GNT[3:0]# is asserted.

Arbitration Algorithm

A rotating priority queue determines the highest priority requester. If the highest priority PCI input at arbitration time does not have an active request, the next highest priority active request will be granted the bus. The priority algorithm always drops the granted PCI device to the lowest priority after each grant event.

4.1.8.2. PC/PCI REQ#/GNT#

MPCI2 passes PC/PCI requests to the notebook's PC/PCI compatible arbiter. S_PCPCIREQ# is passed to P_PCPCIREQ#. When MPCI2 receives P_PCPCIGNT#, it is passed to S_PCPCIGNT#, unless MPCI2 is flushing its write post buffers upstream. See below for a description of PC/PCI grant during flushing of posted write data. With the exception of the flushing of posted write data, there is a two clock delay when PC/PCI request/grant signals are passed through MPCI2.

MPCI2 supports devices that meet the PC/PCI DMA protocol. See the *PC/PCI DMA Protocol specification* for more details. In addition, devices must be legacy ISA DMA compatible which includes the requirement that once a request is issued, that request may not be removed prior to the channel being granted.

NOTE

1. MPCI2 does not support PCI masters on the PCPCI REQ#/GNT# pair.
2. The Secondary PCI Bus and Primary PCI Bus Enable bits in the P_PCICMD register must be set to enable the PC/PCI functions of MPCI2.

PC/PCI and flushing posted write data

While flushing posted write data upstream, MPCI2 attempts to gain access to the primary PCI bus using P_REQ#. If P_PCPCIGNT# is issued to an ISA bus master prior to MPCI2 receiving a P_GNT#, the PC/PCI grant is intercepted for use in flushing the posted data. If the P_PCPCIGNT# is issued to an ISA bus DMA device, MPCI2 retries the cycle. In any case, during flushing of posted write data upstream, P_PCPCIGNT# is not passed to the secondary bus.

A P_PCPCIGNT# for an ISA master is identified when P_FRAME# is not asserted within CDD clocks from primary bus idle state after P_PCPCIGNT# asserted. The CDD bit in the SIF_BURSTEN register determines the CDD value.

Once the posted write data has been flushed, if a PC/PCI cycle has been retried, the system's PC/PCI arbiter will still be in retry mode and the PC/PCI cycle will be run again. In the case of a PC/PCI retry, P_FRAME# is asserted 1 clock after the P_PCPCIGNT# start bit (see the *Mobile PC/PCI DMA Arbitration and Protocols specification*). Since the grant was not passed to the secondary bus, if the retried cycle was for a DMA device, the device will not be prepared for FRAME# to be asserted until the entire PCPCIGNT serial channel message is completed. Therefore, after a PC/PCI retry has occurred, MPCI2 passes P_PCPCIGNT# to S_PCPCIGNT# with standard 2 clock delay but will delay passing P_FRAME# to S_FRAME# for FDD clocks. The FDD bit in the SIF_BURSTEN register determines the FDD value. This delay allows the downstream DMA device and PC/PCI compatible ISA bridge to decode the grant message and set up the transaction before S_FRAME# begins the cycle.

4.1.8.3. Distributed DMA

D-DMA slave devices that are not PCI-to-ISA bridges may be connected to the MPCI2 secondary bus S_REQ[3:0]#/S_GNT[3:0] pin pairs. These devices must behave as PCI Specification Compliant 2.0 or 2.1 PCI Master devices and must not cause the PCI bus to become deadlocked. Specifically, if a D-DMA device is retried, it must release its grant and accept any write cycles targeted to its address. (Failure to do so may create a deadlock situation, if a Primary bus master must flush its posted data to the retried device after the device retries that master.)

PCI-to-ISA bridges are only supported using the PC/PCI protocol via the S_PCPCIREQ#/S_PCPCIGNT# signals. The recommended PCI-to-ISA bridge docking solution is the Mobile PC/PCI Specification compliant 380AB (MISA).

4.1.9. CLOCK RUN PROTOCOL

MPCI2 supports the PCI specification's CLKRUN protocol. MPCI2's secondary bus clock always follows that of the primary bus clock. If the secondary bus is idle when P_CLKRUN# is negated, the MPCI2 generates a secondary bus stop clock request by negating the S_CLKRUN# signal. MPCI2 then asserts P_CLKRUN# giving secondary bus devices the time to assert S_CLKRUN#. If, when P_CLKRUN# is asserted again, S_CLKRUN# has not been asserted, MPCI2 allows the clock stop request to proceed.

4.1.10. BUS LOCK

4.1.10.1. Bus Lock for Standard PCI Masters

MPCI2 does not support secondary PCI bus initiated locked cycles; thus, the S_LOCK# signal does not reflect to P_LOCK# on the primary PCI bus. If a master initiates a locked cycle from the primary PCI bus, the P_LOCK# passes through to S_LOCK# on the secondary bus except in the following scenario:

- When MPCI2 subtractively claims a locked cycle on the primary PCI bus, it passes the cycle to secondary along with the S_LOCK# signal. If the MPCI2 terminates the cycle with a master abort on the secondary bus, it negates the secondary S_LOCK#. MPCI2 terminates the locked cycle on primary with 'disconnect', but the initiating master may continue asserting P_LOCK#. Although the P_LOCK# and S_LOCK# signals now appear to be out of synch, it will not cause a problem, because MPCI2's arbiter does not grant the bus to a secondary PCI master as the notebook system arbiter does not assert P_GNT# until P_LOCK# is removed (i.e., until the current primary master completes its locked operation).

4.1.10.2. Bus Lock for PC/PCI Masters

Masters using PC/PCI protocol cannot lock the bus and should not assert the LOCK# signal.

4.1.11. RESET BEHAVIOR

The secondary reset signal, S_RST#, is a logical OR of the primary interface P_RST# signal and the docking Control State Machine's RESET (see Docking Control Logic section for additional information). The S_RST# signal is asynchronously asserted and synchronously negated by MPCI2.

4.2. Docking Control Logic

MPCI2 provides the hardware support necessary to perform safe docking and undocking of a notebook computer. Together with BIOS and operating system support, MPCI2 monitors system status, controls electrical isolation, issues docking event interrupts, and performs docking mechanism control. The docking and undocking operations may be performed while the notebook is in operation, suspended, or powered down.

MPCI2's docking state machine is designed to wake-up the notebook, if suspended, and allow the operating system to reconfigure the system. Once the notebook has reconfigured itself, it can return to its prior state or execute a preprogrammed user sequence (such as uploading and downloading e-mail and voice mail, synchronizing files, or sending out faxes).

4.2.1. DOCKING CONTROL INTERFACE SIGNALS

Figure 2 shows MPCI2's docking control interface in a typical docking station application.

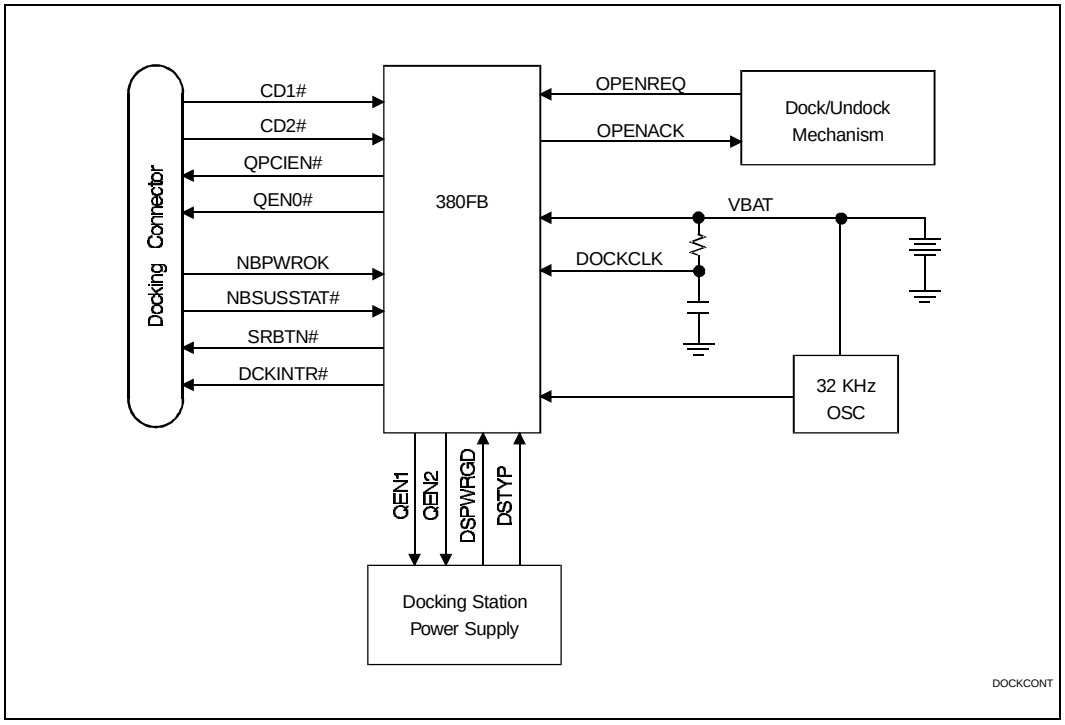


Figure 2. Docking Control Interface

MPCI2 controls the docking operations using a set of buffer controls, commands to the notebook, and an undock command signal for user notification or mechanical capture. The QPCIEN# and QEN[2:0] output pins can be used to control buffers for primary PCI bus isolation. These buffers are typically located in the notebook. Alternatively, QEN[2:1] can be used by power management logic to control power plane switches in the power supply.

The OPENACK output produces a 256 µs pulse when the notebook is permitted to be removed from the docking station. This signal may be used to control an indicator to notify the user of the undocking status. It may alternatively be used to release a mechanical interlock that captures the notebook until released by the docking controls.

SRBTN# and DCKINTR# are command signals to the notebook. SRBTN# is intended to be combined with the notebook's suspend/resume button to allow MPCI2 to wake up the notebook to perform docking or undocking procedures. DCKINTR# is an interrupt signal to be used by MPCI2 to notify the operating system of a docking or undocking event.

MPCI2 monitors system status and configuration using docking station status and notebook status signals. The docking station status signals consist of CD1#, CD2#, DSTYP, OPENREQ, and DSPWRGD. CD1# and CD2# indicate to MPCI2 that the docking connectors of the notebook and docking station are firmly seated. DSTYP

indicates the power source for the docking station; external power or the notebook. OPENREQ signals a request to undock. DSPWRGD reports the status of the docking station power supply to MPCI2.

There are two notebook status signals; NBPWROK and NBSUSSTAT#. NBPWROK from the notebook power supply signals the status of the notebook power. NBSUSSTAT# indicates the power management state of the notebook. NBPWROK high indicates that the notebook is powered on. NBSUSSTAT# high indicates that the notebook is in operation. NBSUSSTAT# low indicates that the notebook is suspended. If NBPWROK transitions to a low while NBSUSSTAT# is high, MPCI2 assumes that the notebook is powered off and not suspended.

The docking state machine in MPCI2 is clocked by the DOCKCLK input. This clock needs to be running at all times, even when the docking station is not powered. This allows MPCI2 to begin the docking sequence and the docking station power to be turned ON, if a notebook is inserted into a powered-down docking station.

The docking state machine in MPCI2 is powered by the VBAT pin permitting battery backup when system power is turned off. A reset pin (DOCKRST#) is provided to initialize MPCI2's docking state machine, if necessary.

4.2.2. DOCKING OPERATION

4.2.2.1. Initialization

When first powered on, MPCI2 monitors both CD1# and CD2#. The secondary PCI bus is placed in a reset condition at this time by the assertion of S_RST#. After both CD1# and CD2# are sampled low, MPCI2 waits 8 ms for the signals to become stable. Once CD1# and CD2# are stable the NBPWROK and NBSUSSTAT# signals are checked to determine the notebook's state.

If NBPWROK and NBSUSSTAT# are high, indicating that the notebook is powered on and not suspended, the process of establishing the PCI bus connections begins. If the notebook is powered off or in a suspend mode, MPCI2 attempts a resume by asserting the NBSRBTN# signal for 256 µs. MPCI2 then waits for the notebook power supply to assert NBPWROK indicating that the notebook is resuming from a suspend (responding to the NBSRBTN# pulse) or initializing upon power-up.

4.2.2.2. Connecting the PCI Bus

When MPCI2 determines that the notebook is powered on (NBPWROK is high), QEN0 is asserted (low) to enable the notebook to connect P_REQ#, P_GNT#, P_RST#, and PCICLK. After MPCI2 has determined that the notebook is in operation (see initialization above), MPCI2 asserts P_REQ# to the notebook. When P_GNT# is asserted, MPCI2 is assured that the primary PCI bus is idle and that the remainder of the primary PCI bus signals can be connected. QPCIEN# is asserted (low) enabling the notebook to connect the remainder of the primary PCI bus. P_REQ# is then negated to release the primary bus.

With the connection to the primary PCI bus established, MPCI2 releases the secondary PCI bus by negating S_RST#. S_RST# is asserted for 100 ms if the MISAEN input pin on MPCI2 is pulled high, and 1.2 ms if it is low. MPCI2 also sets the Docked bit in the CONEVNT register and asserts DCKINTR# to notify the notebook's operating system that the process of reconfiguring the system may begin.

4.2.3. UNDOCKING OPERATION

4.2.3.1. Undocking events

The undocking process can be initiated by a 256 µs high pulse on MPCI2's OPENREQ input or by setting the UNDCKPERMIT bit in the CONEVNT register. Software setting of the UNDKPERMIT bit causes MPCI2 to generate a docking interrupt through the assertion of the DCKINTR# signal. The assertion of the OPENREQ input causes the OPENREQ bit in the CON-EVNT register to be set. If this occurs while the notebook is

powered and not suspended, MPCI2 generates a docking interrupt through the assertion of the DCKINTR# signal and waits for the operating system to allow the PCI bus to be disconnected (see following section).

If the OPENREQ input is asserted when the notebook is suspended, NBSRBTN# is pulsed for 256 µs to resume the notebook. When the notebook resumes, indicated by NBSUSSTAT# being negated, the system continues normal operation. In this case, the OPENREQ input is being used for the resume function.

If the OPENREQ input is asserted while the notebook is powered off and not suspended, a 256 µs pulse is generated on MPCI2's OPENACK pin. This pulse can be used by the system to unlock or eject the notebook from the docking station. The OPENACK pin is also pulsed if OPENREQ is asserted during the docking sequence and the PCI bus has not yet been connected.

If CD1# and/or CD2# are negated prior to the normal updocking events described above, both QPCIEN# and QEN0 will be synchronously negated. The DCKINTR# signal will not be asserted to notify the system BIOS or operating system.

4.2.3.2. PCI Bus Disconnect and Notebook Release

The docking interrupt routine along with the operating system is responsible for preparing the system for undocking and then resetting the UNDCKPERMIT bit. Once this bit is cleared, QPCIEN# is synchronously negated to isolate the primary PCI bus and a 256 µs pulse is generated on MPCI2's OPENACK pin. This pulse can be used by the system to unlock or eject the notebook from the docking station.

When CD1# and/or CD2# are negated, QEN0 is negated (low). MPCI2 then awaits the assertion of both CD1# and CD2# to begin the docking process.

If the notebook goes into powered off mode, QPCIEN# is negated and MPCI2 waits for the OPENREQ input to be asserted or for NBPWROK to be asserted. If the OPENREQ input is asserted, the OPENACK pin is asserted (see Undocking Events section). If NBPWROK is asserted while the notebook is powered off, MPCI2 initiates a docking sequence (see Connecting the PCI Bus section).

The PWRONSUS bit in the CONENVT register is used to control the operation of QPCIEN# when NBSUSSTAT# is asserted. When CONENVT[PWRONSUS]=1, QPCIEN# remains asserted when NBSUSSTAT# is asserted. In this case, MPCI2 assumes a powered on suspend. If CONENVT[PWRONSUS]=0, QPCIEN# is negated when NBSUSSTAT# is asserted. In this case, MPCI2 assumes a powered off suspend.

4.3. Power Management Description

4.3.1. SUSPEND MODES

MPCI2 supports power-on and power-off suspend modes of the notebook. Power-on suspend mode is indicated when CONENVT[PWRONSUS]=1 and NBSUSSTAT# is asserted. Power-off suspend mode is indicated when CONENVT[PWRONSUS]=0 and NBSUSSTAT# is asserted. Power-off suspend is also entered from power-on suspend if NBPWROK is negated.

4.3.1.1. Power-on Suspend

During power-on suspend, the primary PCI bus between MPCI2 and the notebook remains connected. When NBSUSSTAT# is negated, the system resumes normal operation. If NBPWROK is negated before the notebook is resumed, MPCI2 assumes power-off suspend mode.

4.3.1.2. Power-off Suspend

During power-off suspend, the primary PCI bus between MPCI2 and the notebook is disconnected. When NBSUSSTAT# is negated and NBPWROK is asserted, the system reconnects the PCI bus (see Docking Operation section). Assertion of P_RST# upon a notebook resume from power-off suspend causes assertion of S_RST# and the initialization of MPCI2 registers.

4.3.1.3. Hardware Control signals

The QEN1 and QEN2 output signals are general purpose output bits controlled by the CONCNTL register. These signals can be used to control the power planes, isolation buffers, or low power modes of devices such as disk controllers, bus slots, pass-through peripheral connections, and other peripherals.

5.0. PINOUT AND PACKAGE INFORMATION

5.1. MPCI2 PINOUT INFORMATION

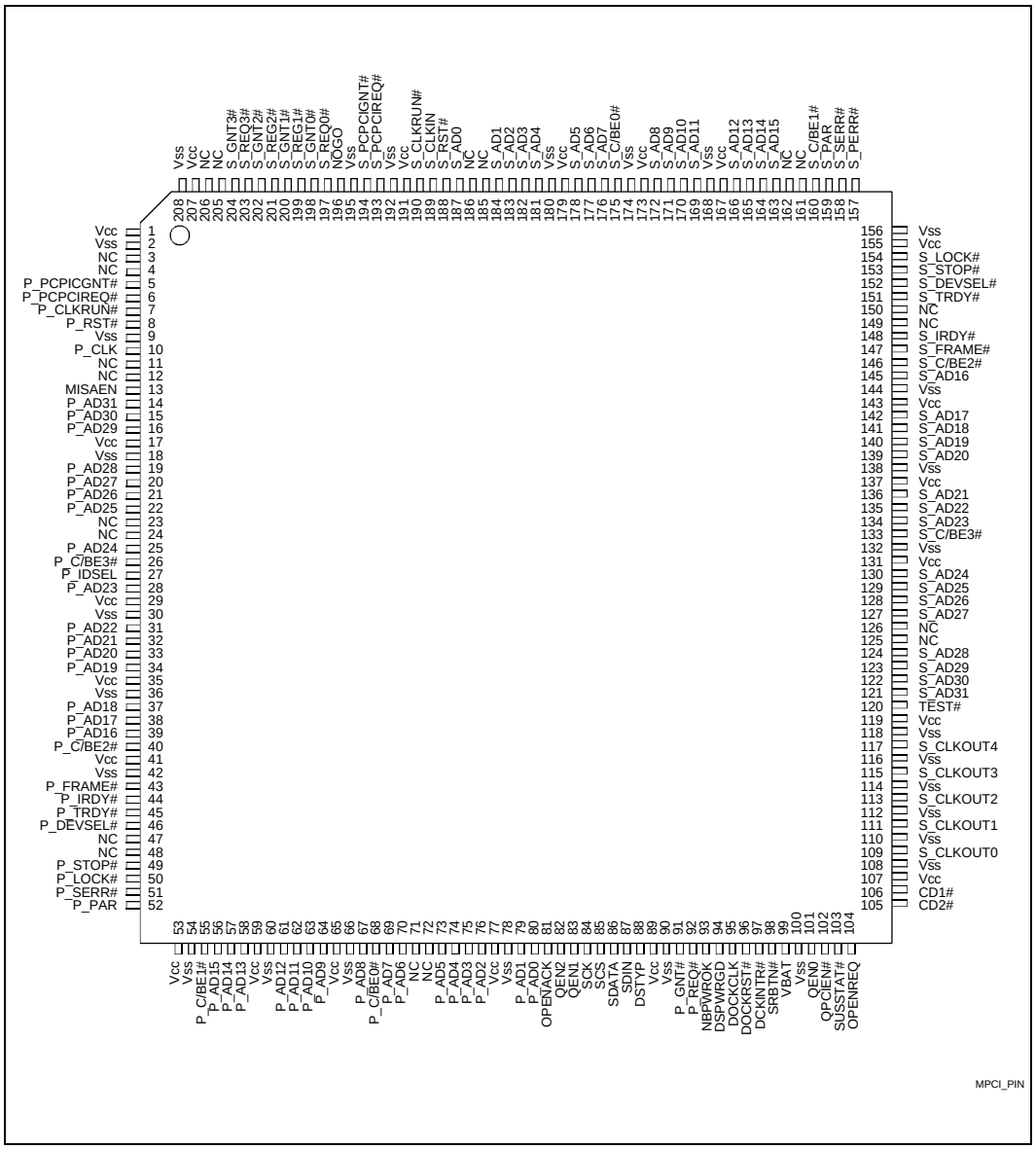


Figure 3. 82380FB Pinout

Table 3. MPCI2 Alphabetical Pin Assignment

Name	Pin	Type
CD1#	106	I
CD2#	105	I
DCKINTR#	97	od
DOCKCLK	95	I
DOCKRST#	96	I
DSPWRGD	94	I
DSTYP	88	I
MISAEN	13	I
NBPWROK	93	I
NC	3	—
NC	4	—
NC	11	—
NC	12	—
NC	23	—
NC	24	—
NC	47	—
NC	48	—
NC	71	—
NC	72	—
NC	125	—
NC	126	—
NC	149	—
NC	150	—
NC	161	—
NC	162	—
NC	185	—
NC	186	—
NC	205	—
NC	206	—

Table 3. MPCI2 Alphabetical Pin Assignment

Name	Pin	Type
NOGO	196	O
OPENACK	81	O
OPENREQ	104	I
P_AD0	80	I/O
P_AD1	79	I/O
P_AD2	76	I/O
P_AD3	75	I/O
P_AD4	74	I/O
P_AD5	73	I/O
P_AD6	70	I/O
P_AD7	69	I/O
P_AD8	67	I/O
P_AD9	64	I/O
P_AD10	63	I/O
P_AD11	62	I/O
P_AD12	61	I/O
P_AD13	58	I/O
P_AD14	57	I/O
P_AD15	56	I/O
P_AD16	39	I/O
P_AD17	38	I/O
P_AD18	37	I/O
P_AD19	34	I/O
P_AD20	33	I/O
P_AD21	32	I/O
P_AD22	31	I/O
P_AD23	28	I/O
P_AD24	25	I/O
P_AD25	22	I/O

Table 3. MPCI2 Alphabetical Pin Assignment

Name	Pin	Type
P_AD26	21	I/O
P_AD27	20	I/O
P_AD28	19	I/O
P_AD29	16	I/O
P_AD30	15	I/O
P_AD31	14	I/O
P_C/BE0#	68	I/O
P_C/BE1#	55	I/O
P_C/BE2#	40	I/O
P_C/BE3#	26	I/O
P_CLK	10	I
P_CLKRUN#	7	I/O,od
P_DEVSEL#	46	s/t/s
P_FRAME#	43	s/t/s
P_GNT#	91	I
P_IDSEL	27	I
P_IRDY#	44	s/t/s
P_LOCK#	50	s/t/s
P_PAR	52	I/O
P_PCPCIGNT#	5	I/O
P_PCPCIREQ#	6	I/O
P_REQ#	92	I/O
P_RST#	8	I
P_SERR#	51	od
P_STOP#	49	s/t/s
P_TRDY#	45	s/t/s
QEN0	101	O
QEN1	83	O
QEN2	82	O

Table 3. MPCI2 Alphabetical Pin Assignment

Name	Pin	Type
QPCIEN#	102	O
S_AD0	187	I/O
S_AD1	184	I/O
S_AD2	183	I/O
S_AD3	182	I/O
S_AD4	181	I/O
S_AD5	178	I/O
S_AD6	177	I/O
S_AD7	176	I/O
S_AD8	172	I/O
S_AD9	171	I/O
S_AD10	170	I/O
S_AD11	169	I/O
S_AD12	166	I/O
S_AD13	165	I/O
S_AD14	164	I/O
S_AD15	163	I/O
S_AD16	145	I/O
S_AD17	142	I/O
S_AD18	141	I/O
S_AD19	140	I/O
S_AD20	139	I/O
S_AD21	136	I/O
S_AD22	135	I/O
S_AD23	134	I/O
S_AD24	130	I/O
S_AD25	129	I/O
S_AD26	128	I/O
S_AD27	127	I/O

Table 3. MPCI2 Alphabetical Pin Assignment

Name	Pin	Type
S_AD28	124	I/O
S_AD29	123	I/O
S_AD30	122	I/O
S_AD31	121	I/O
S_C/BE0#	175	I/O
S_C/BE1#	160	I/O
S_C/BE2#	146	I/O
S_C/BE3#	133	I/O
S_CLKIN	189	I
S_CLKOUT4	117	O
S_CLKOUT3	115	O
S_CLKOUT2	113	O
S_CLKOUT1	111	O
S_CLKOUT0	109	O
S_CLKRUN#	190	I,s/t/s
S_DEVSEL#	152	s/t/s
S_FRAME#	147	s/t/s
S_GNT0#	198	O
S_GNT1#	200	O
S_GNT2#	202	O
S_GNT3#	204	O
S_IRDY#	148	s/t/s
S_LOCK#	154	s/t/s
S_PAR	159	I/O
S_PCPCIGNT#	194	I/O
S_PCPCIREQ#	193	I/O
S_PERR#	157	I
S_REQ0#	197	I
S_REQ1#	199	I

Table 3. MPCI2 Alphabetical Pin Assignment

Name	Pin	Type
S_REQ2#	201	I
S_REQ3#	203	I
S_RST#	188	O
S_SERR#	158	I
S_STOP#	153	s/t/s
S_TRDY#	151	s/t/s
SCK	84	O
SCS	85	O
SDATA	86	I/O
SDIN	87	I
SRBTN#	98	od
SUSSTAT#	103	I
TEST#	120	I
VBAT	99	—
Vcc	1	—
Vcc	17	—
Vcc	29	—
Vcc	35	—
Vcc	41	—
Vcc	53	—
Vcc	59	—
Vcc	65	—
Vcc	77	—
Vcc	89	—
Vcc	107	—
Vcc	119	—
Vcc	131	—
Vcc	137	—
Vcc	143	—

Table 3. MPCI2 Alphabetical Pin Assignment

Name	Pin	Type
VCC	155	—
VCC	167	—
VCC	173	—
VCC	179	—
VCC	191	—
VCC	207	—
VSS	2	—
VSS	9	—
VSS	18	—
VSS	30	—
VSS	36	—
VSS	42	—

Table 3. MPCI2 Alphabetical Pin Assignment

Name	Pin	Type
VSS	54	—
VSS	60	—
VSS	66	—
VSS	78	—
VSS	90	—
VSS	100	—
VSS	108	—
VSS	110	—
VSS	112	—
VSS	114	—
VSS	116	—
VSS	118	—

Table 3. MPCI2 Alphabetical Pin Assignment

Name	Pin	Type
VSS	132	—
VSS	138	—
VSS	144	—
VSS	156	—
VSS	168	—
VSS	174	—
VSS	180	—
VSS	192	—
VSS	195	—
VSS	208	—

5.2. MPCI2 Package Information

The MPCI2 comes in an industry standard 208-lead SQFP package.

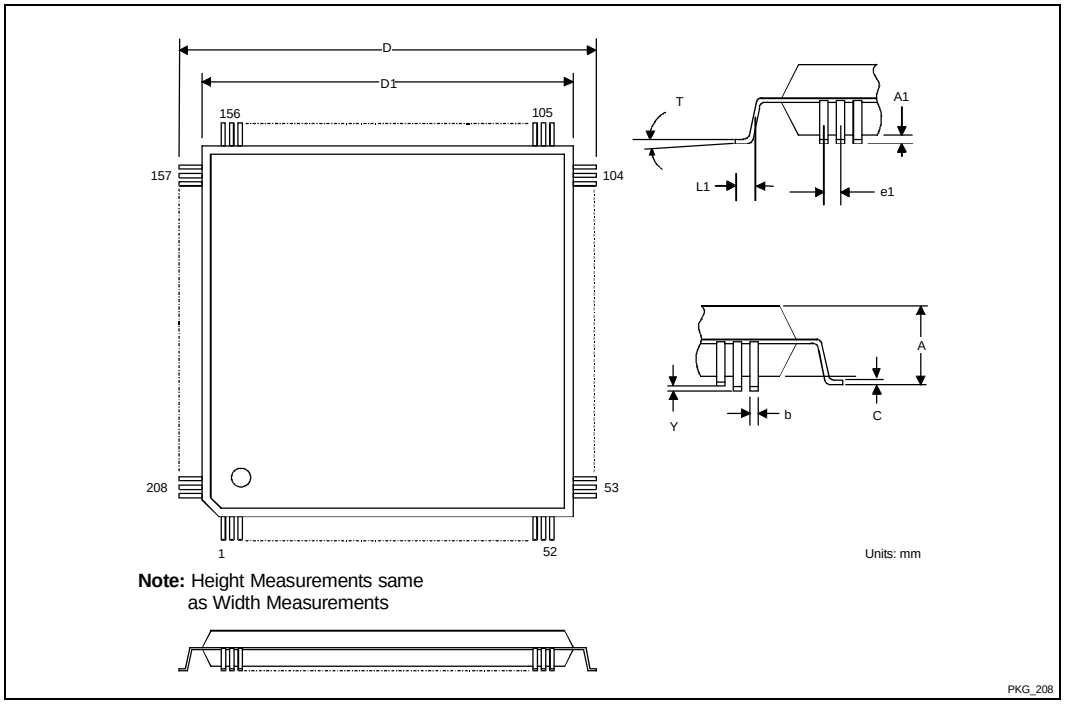


Figure 4. MPCI2 208-Pin SQFP Package

Table 4. MPCI2 208 Pin SQFP Package Dimensions

Symbol	Dimension in Millimeters		
	Minimum	Nominal	Maximum
A		3.5	3.75
A1	0.05	0.15	0.25
A2		3.37	
b	0.13	0.20	0.28
C	0.10	0.15	0.25
D	30.3	30.6	30.9
D1	27.9	28.0	28.1
e1	0.4	0.5	0.6
L1	0.5	0.6	0.7
Y			0.08
T	0		10

6.0. TESTABILITY

The test modes are decoded from the TEST#, DOCKRST#, and OPENREQ signals as shown in Table 5. Modes. The following table shows which signals are used and how they are decoded to enter.

Table 5. Test Modes

Test Mode	TEST#	DOCKRST#	OPENREQ
Normal operation	1	x	x
Tri-state Test	0	1	0
NAND Tree Test	0	0	0

Tri-State Test Mode

This test mode tri-states all output buffers of MPCI2. Besides that TEST#, DOCKRST# and OPENREQ have to be set to the correct value to get into this mode, DSPWG and DOCKCLK have to be maintained high. Also DOCKRST# should to be low for a minimum of 1 µs before it is set high to ensure complete chip reset.

NAND Tree Test Mode

Figure 5 illustrates the MPCI2 NAND Tree structure. Every MPCI2 pin except TEST#, DOCKRST#, OPENREQ, QEN0 and QPCIEN# is configured as an input in NAND Tree mode and included in the NAND chain. TEST#, DOCKRST# and OPENREQ are used as inputs for test mode entrance decoding while QEN0 and QPCIEN# are used as the outputs for the first half and the end of the NAND Tree. Therefore, these five pins are not included in the NAND Tree. The DSPWG pin is part of the NAND Tree but it is NOT toggled during the NAND Tree test due to the NAND Tree output pins dependency upon DSPWG signal. All through the test, DSPWG is driven to a 1. The QEN0 output is provided so that the NAND tree test can be divided into two sections.

The first input of the NAND chain is P_PCPCIGNT#, and the NAND chain is routed counter-clockwise around the chip (e.g., P_PCPCIGNT#, P_PCPCIREQ#,...). SRBTN# is the mid-point of the NAND Trees and S_GNT4# is the end of the NAND Tree. Output buffers in the NAND Tree are tri-stated.

To perform a NAND Tree test, all pins included in the NAND tree should be first driven to 1. Then the walking '0' could be used to perform the test. (i.e., the first pin in NAND chain is driven input a 0 for one period, then the next pin is driven a 0, and so on.) The result is seen on the NAND Tree output pins QEN0 and QPCIEN#. The Half-NAND-Out pin (QEN0) drives out a 0 for one period and the Full-NAND-Out pin (QPCIEN#) drives out a 1 for the same period. When the 'walking 0' reaches the DSPWG pin, this pin is skipped from driving a 0 and the test goes to the next pin in chain (DOCKCLK). During this period, QEN0 drives a 1 for 2 periods, and QPCIEN# drives out a 0 for 2 periods. Note that this behavior should not be misinterpreted to be an open in the NAND chain at the DSPWG pin.

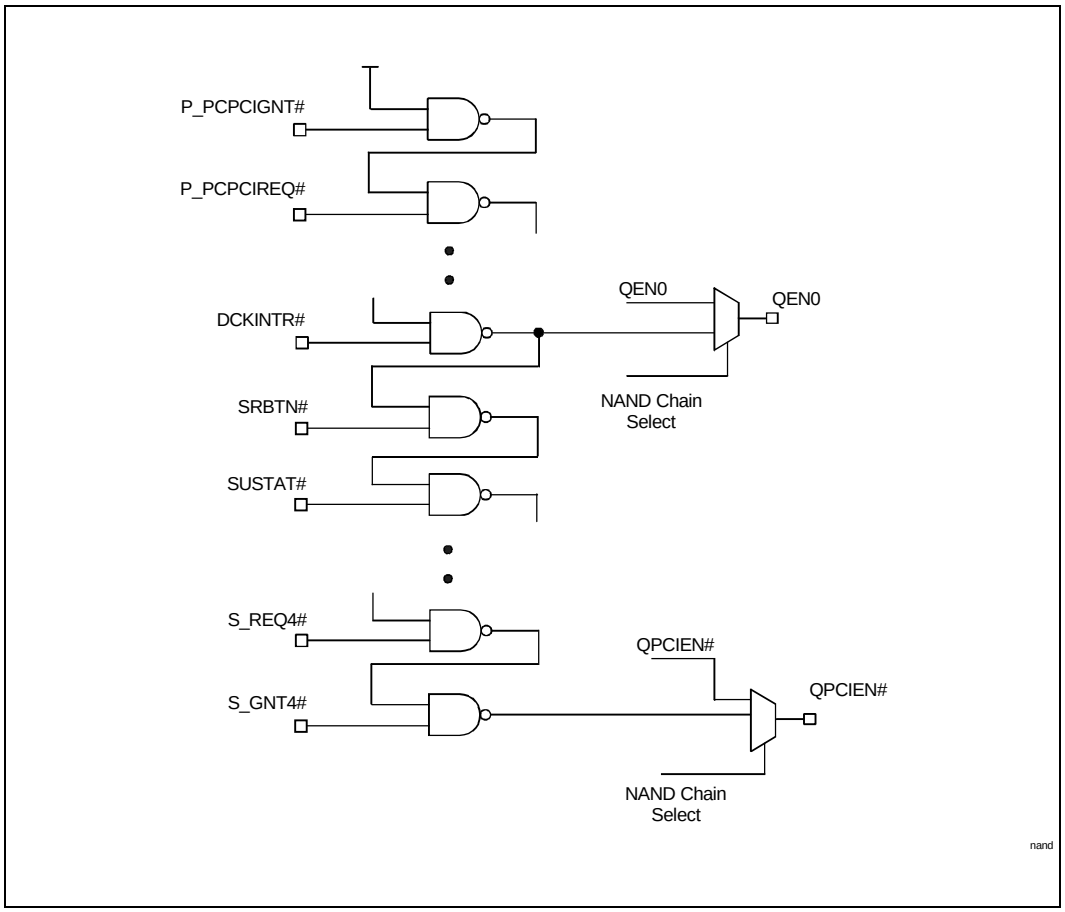


Figure 5. NAND Tree Diagram

Table 6. NAND Tree Order

Pin #	Signal Names	Note
1	Vcc	
2	Vss	
3	NC	
4	NC	
5	P_PCPCIGNT#	Input to NAND Chain
6	P_PCPCIREQ#	
7	P_CLKRUN#	
8	P_RST#	Inverted input signal
9	Vss	
10	P_CLK	Inverted input signal
11	NC	
12	NC	
13	MISAEN	
14	P_AD31	
15	P_AD30	
16	P_AD29	
17	Vcc	
18	Vss	
19	P_AD28	
20	P_AD27	
21	P_AD26	
22	P_AD25	
23	NC	
24	NC	
25	P_AD24	
26	P_C/BE3#	
27	P_IDSEL	
28	P_AD23	
29	Vcc	

Table 6. NAND Tree Order

Pin #	Signal Names	Note
30	Vss	
31	P_AD22	
32	P_AD21	
33	P_AD20	
34	P_AD19	
35	Vcc	
36	Vss	
37	P_AD18	
38	P_AD17	
39	P_AD16	
40	P_C/BE2#	
41	Vcc	
42	Vss	
43	P_FRAME#	
44	P_IRDY#	
45	P_TRDY#	
46	P_DEVSEL#	
47	NC	
48	NC	
49	P_STOP#	
50	P_LOCK#	
51	P_SERR#	
52	P_PAR	
53	Vcc	
54	Vss	
55	P_C/BE1#	
56	P_AD15	
57	P_AD14	
58	P_AD13	

Table 6. NAND Tree Order

Pin #	Signal Names	Note
59	Vcc	
60	Vss	
61	P_AD12	
62	P_AD11	
63	P_AD10	
64	P_AD9	
65	Vcc	
66	Vss	
67	P_AD8	
68	P_C/BE0#	
69	P_AD7	
70	P_AD6	
71	NC	
72	NC	
73	P_AD5	
74	P_AD4	
75	P_AD3	
76	P_AD2	
77	Vcc	
78	Vss	
79	P_AD1	
80	P_AD0	
81	OPENACK	
82	QEN2	
83	QEN1	
84	SCK	
85	SCS	
86	SDATA	
87	SDIN	
88	DSTYP	

Table 6. NAND Tree Order

Pin #	Signal Names	Note
89	Vcc	
90	Vss	
91	P_GNT#	
92	P_REQ#	
93	NBPWROK	Inverted input signal
94	DSPWG	Inverted input signal
95	DOCKCLK	
96	DOCKRST#	Inverted input signal. Not in NAND Chain
97	DCKINTR#	
98	SRBTN#	Middle of NAND Chain
99	VBAT	
100	Vss	
101	QEN0	Midpoint output of NAND Chain. Not in NAND Chain
102	QPCIEN#	Final Output of NAND Chain. Not in NAND Chain
103	SUssTAT#	Inverted input signal
104	OPENREQ	Inverted input signal. Not in NAND Chain
105	CD2#	Inverted input signal
106	CD1#	Inverted input signal
107	Vcc	
108	Vss	
109	S_CLKOUT0	
110	Vss	
111	S_CLKOUT1	
112	Vss	
113	S_CLKOUT2	
114	Vss	

Table 6. NAND Tree Order

Pin #	Signal Names	Note
115	S_CLKOUT3	
116	Vss	
117	S_CLKOUT4	
118	Vss	
119	Vcc	
120	TEST#	Not in NAND Chain
121	S_AD31	
122	S_AD30	
123	S_AD29	
124	S_AD28	
125	NC	
126	NC	
127	S_AD27	
128	S_AD26	
129	S_AD25	
130	S_AD24	
131	Vcc	
132	Vss	
133	S_C/BE3#	
134	S_AD23	
135	S_AD22	
136	S_AD21	
137	Vcc	
138	Vss	
139	S_AD20	
140	S_AD19	
141	S_AD18	
142	S_AD17	
143	Vcc	
144	Vss	

Table 6. NAND Tree Order

Pin #	Signal Names	Note
145	S_AD16	
146	S_C/BE2#	
147	S_FRAME#	
148	S_IRDY#	
149	NC	
150	NC	
151	S_TRDY#	
152	S_DEVSEL#	
153	S_STOP#	
154	S_LOCK#	
155	Vcc	
156	Vss	
157	S_PERR#	
158	S_SERR#	
159	S_PAR	
160	S_C/BE1#	
161	NC	
162	NC	
163	S_AD15	
164	S_AD14	
165	S_AD13	
166	S_AD12	
167	Vcc	
168	Vss	
169	S_AD11	
170	S_AD10	
171	S_AD9	
172	S_AD8	
173	Vcc	
174	Vss	

Table 6. NAND Tree Order

Pin #	Signal Names	Note
175	S_C/BE0#	
176	S_AD7	
177	S_AD6	
178	S_AD5	
179	Vcc	
180	Vss	
181	S_AD4	
182	S_AD3	
183	S_AD2	
184	S_AD1	
185	NC	
186	NC	
187	S_AD0	
188	S_RST#	
189	S_CLKIN	Inverted input signal
190	S_CLKRUN#	
191	Vcc	
192	Vss	

Table 6. NAND Tree Order

Pin #	Signal Names	Note
193	S_PCPCIREQ#	
194	S_PCPCIGNT#	
195	Vss	
196	NOGO	
197	S_REQ0#	
198	S_GNT0#	
199	S_REQ1#	
200	S_GNT1#	
201	S_REQ2#	
202	S_GNT2#	
203	S_REQ3#	
204	S_GNT3#	End point of NAND Chain
205	NC	
206	NC	
207	Vcc	
208	Vss	