SPWS020 – AUGUST 1995

- **Glueless Interface Between the Peripheral Component Interconnect (PCI) Bus and the TI380C2x Family of Processors**
- **Compliant With PCI Specification, Revision 2.0†**
- **Allows Use of Existing TI2000 Drivers**
- **Includes TI2000 Interface Configuration Register**
- **Supports Bus Master Operations for High Performance**
- **Provides 32-Bit Address-Data Path**
- **Implements Address/Data Parity Checking**
- **Includes Internal Error Checking for Illegal Bus Operations**
- **Supports DMA Bursts With 64-Byte FIFO**
- **Supports EPROM Interface for Remote Program Load (RPL) Operation**
- **Supports I^2C Interface for Optional Serial EEPROM for Configuration Information**
- **Allows BIA to be Implemented in Configuration EEPROM**
- **Includes NAND Tree Structure to Allow for In-Circuit Connectivity Testing**
- **144-Pin JEDEC PCM Plastic Quad Flat Package (PCM Suffix)**
- **Operating Temperature Range 0°C to 70°C**

description

The TI380PCI provides a glueless interface between a TI380C2x commprocessor and the PCI bus. The TI380PCI transfers information/data between the PCI bus and the TI380C2x System Interface (SIF) using any of these three methods:

- Direct memory access (DMA)
- Direct input/output (DIO)
- Pseudo-direct memory access (PDMA)

DMA (or PDMA) transfers all data between host memory (by way of the PCI Local Bus) and TI380C2x local memory. DIO accesses are typically used to load software into TI380C2x local memory and for initializing the TI380C2x.

The TI380PCI conforms to the PCI standards found in "PCI Local Bus Specification," Revision 2.0.†

The TI380PCI is available in a 144-pin EIAJ plastic quad flat package (PCM suffix) and is rated from 0°C to 70°C.

† The "PCI Local Bus Specification", Revision 2.0, and the TI380C2x series of data sheets (literature numbers SPWS012, SPWS013, and SPWS014) should be used as references to this document.

ADVANCE INFORMATION

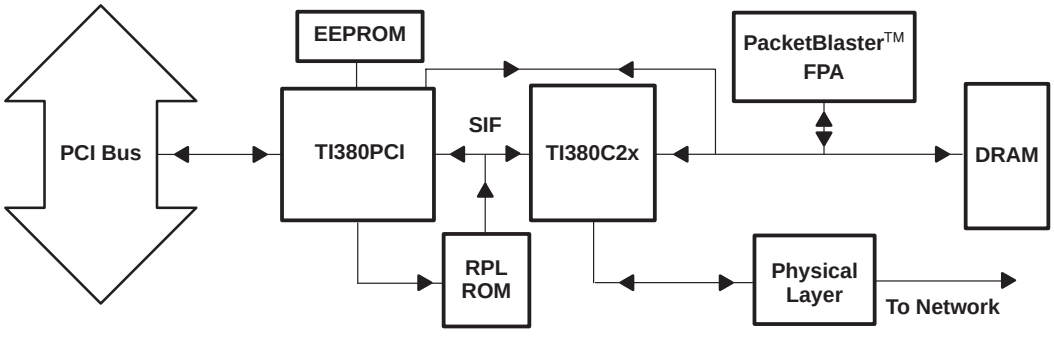


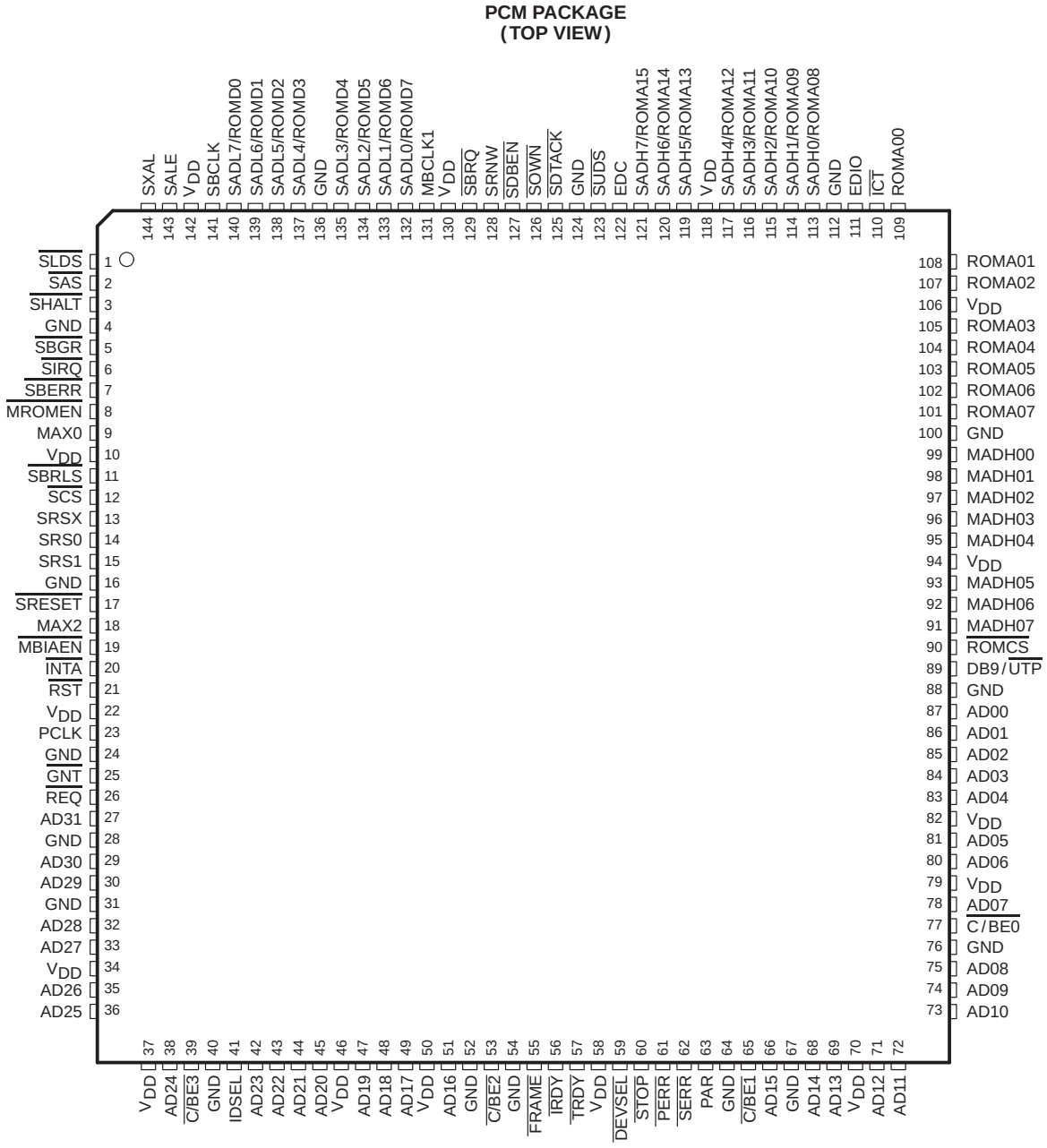
Figure 1. TI380PCI Applications Diagram

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PacketBlaster is a trademark of Texas Instruments Incorporated.

pin assignments

The pin assignments for the TI380PCI (144-pin plastic quad flat package) are shown in Figure 2.



NOTE: Pin 1 is positioned at the upper left corner.

Figure 2. TI380PCI Pinout

ADVANCE INFORMATION

block diagram

TI380PCI can connect to up to four devices in a system: the PCI bus, the SIF bus, an optional serial EEPROM, and an optional boot ROM. The major blocks of the TI380PCI include the PCI Interface (PCIIF), SIF, and ROM Interface (ROMIF) as shown in Figure 3.

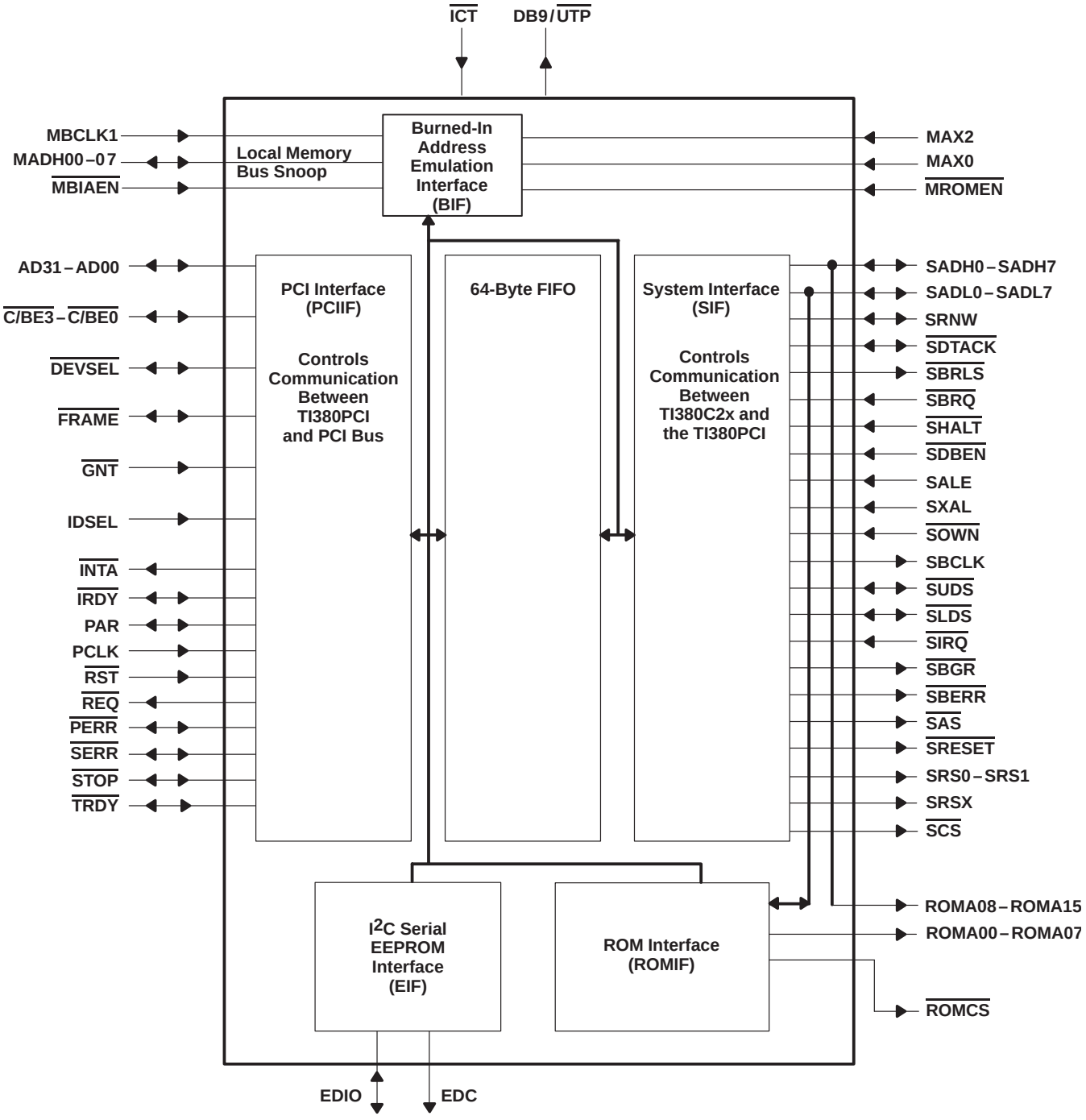


Figure 3. TI380PCI Block Diagram

ADVANCE INFORMATION

Pin Functions

PIN NAME	PIN NO.	I/O†	DESCRIPTION
GND	4 16 24 28 31 40 52 54 64 67 76 88 100 112 124 136	I	Ground. These pins must be attached to the common system ground plane.
DB9/$\overline{\text{UTP}}$	89	O	Connector. The value on DB9/$\overline{\text{UTP}}$ indicates the type of connector in use. Upon reset, DB9/$\overline{\text{UTP}}$'s value is 0. 1 = D-Shell (AUI or DB9) 0 = UTP/10BaseT
$\overline{\text{ICT}}$	110	I	$\overline{\text{ICT}}$ supports in-circuit tests. $\overline{\text{ICT}}$ should be pulled high for normal operation of the TI380PCI. When pulled along with $\overline{\text{RST}}$ to a steady low state, all bi-directional signals in the TI380PCI are configured as inputs, and all output pins of the TI380PCI are in 3-state mode.
ROMA07 ROMA06 ROMA05 ROMA04 ROMA03 ROMA02 ROMA01 ROMA00	101 102 103 104 105 107 108 109	I/O	ROM address. ROMA07–ROMA00 form the least significant eight bits of the address for the RPL ROM. The most significant bits (MSBs) of the ROM address are multiplexed onto the SADHx lines. When $\overline{\text{RST}}$ is driven high, the value on ROMA07–ROMA00 is latched into the board configuration register in the TI380PCI configuration space. The value on ROMA07–ROMA00 can be provided by pullup and pulldown resistors that do not affect operation after reset. This feature allows designers to support jumpers or board stuffing options that can be sensed by software that reads the board configuration register. If pullup and pulldown registers are not used, the contents of the board configuration register are undefined after reset.
$\overline{\text{ROMCS}}$	90	O	$\overline{\text{ROMCS}}$ enables the outputs of the ROM when the ROM has been accessed. ROMCS enable allows the data lines from the ROM to be driven.
V_{DD}	10 22 34 37 46 50 58 70 79 82 94 106 118 130 142	I	5-V supply. These pins must be attached to the common system power supply plane.

† I = in, O = out

Pin Functions – System Interface†

PIN NAME	PIN NO.	I/O‡	DESCRIPTION
SADH0/ROMA08 SADH1/ROMA09 SADH2/ROMA10 SADH3/ROMA11 SADH4/ROMA12 SADH5/ROMA13 SADH6/ROMA14 SADH7/ROMA15	113 114 115 116 117 119 120 121	I/O	System address/data bus—high byte. These lines make up the most significant byte of each TI380C2x address word (32-bit address bus) and data word (16-bit data bus). The most significant bit (MSB) is SADH0, and the least significant bit (LSB) is SADH7. Address multiplexing bits 31–24 and bits 15–8§ Data multiplexing bits 15–8§ During accesses to the ROM address space from the PCI bus, these lines provide the eight most significant address bits to the ROM.
SADL0/ROMD7 SADL1/ROMD6 SADL2/ROMD5 SADL3/ROMD4 SADL4/ROMD3 SADL5/ROMD2 SADL6/ROMD1 SADL7/ROMD0	132 133 134 135 137 138 139 140	I/O	System address/data bus—low byte. These lines make up the least significant byte of each address word (32-bit address bus) and data word (16-bit data bus). The MSB is SADL0, and the LSB is SADL7. These address lines also make up the ROM address. Address multiplexing bits 23–16 and bits 7–0§ Data multiplexing bits 7–0§ During accesses to the ROM address space from the PCI bus, these lines transfer data from the ROM to the TI380PCI.
SALE	143	I	System address latch enable. SALE is the enable pulse used to externally latch the 16 LSBs of the address from the SADH0–SADH7 and SADL0–SADL7 buses at the start of the DMA cycle. Systems that implement address parity also can externally latch the parity bits (SPH and SPL) for the latched address.
SBCLK	141	O	SIF bus clock. The TI380C2x requires SBCLK to synchronize its bus timings for all DMA transfers (see Note 1).
$\overline{\text{SBRLS}}$	11	O	SIF bus release. $\overline{\text{SBRLS}}$ indicates to the TI380C2x that a higher-priority device requires the SIF bus. The value on $\overline{\text{SBRLS}}$ is ignored by the TI380C2x when DMA is not performed. H = The TI380C2x can hold onto the system bus. L = The TI380C2x should release the system bus upon completion of current DMA cycle. If the DMA transfer is not yet complete, the SIF rearbitrates for the SIF bus.
$\overline{\text{SCS}}$	12	O	System chip select. $\overline{\text{SCS}}$ activates the system interface of the TI380C2x for a DIO read or write. H = Not selected L = Selected
$\overline{\text{SDBEN}}$	127	I	System data bus enable. $\overline{\text{SDBEN}}$ causes the TI380PCI to allow its external data buffers to begin driving data. $\overline{\text{SDBEN}}$ is accepted during both DIO and DMA. H = Keep external data buffers in high-impedance state L = Cause external data buffers to begin driving data
$\overline{\text{SOWN}}$	126	I	SIF bus owned. $\overline{\text{SOWN}}$ signals to the PIF to indicate to external devices that the TI380C2x has control of the SIF bus. $\overline{\text{SOWN}}$ drives the enable signal of the bus transceiver chips, which drive the address and bus control signals. L = TI380PCI does not have control of the SIF bus. H = TI380PCI has control of the SIF bus.

† The TI380PCI SIF pin names correspond to a subset of the system interface pins on a TI380C2x. Consult the TI380C2x data sheet for more information on individual pins. Like-named pins on the TI380PCI and TI380C2x system interfaces are intended to be connected to each other.
‡ I = in, O = out
§ Typical bit ordering for Intel and Motorola processor buses
NOTE 1: The TI380PCI allows driver software to set SBCLK output to a steady high state. This signal is driven to a steady high state during power-down operations.

ADVANCE INFORMATION

Pin Functions – System Interface† (Continued)

PIN NAME	PIN NO.	I/O‡	DESCRIPTION
$\overline{\text{SRESET}}$	17	O	System reset. $\overline{\text{SRESET}}$ is sent to initialize the TI380C2x. It is set low whenever $\overline{\text{RST}}$ goes low or by a config write to MISCCTRL. H = No system reset L = System reset Rising edge = Latch bus width for DMA operation
SRSX SRS0 SRS1	13 14 15	O	System register select. These outputs are sent to select the word or byte to be transferred during a system DIO access. The most significant bit is SRSX and the least significant bit is SRS1. MSB LSB Registers selected = SRSX SRS0 SRS1
SXAL	144	I	System extended address latch. SXAL provides the enable pulse that externally latches the most significant 16 bits of the 32-bit system address during DMA. SXAL is activated by the TI380C2x prior to the first cycle of each block DMA transfer, and thereafter as necessary (whenever an increment of the DMA address counter causes a carryout of the lower 16 bits).
$\overline{\text{SUDS}}$	123	I/O	Upper data strobe. $\overline{\text{SUDS}}$ serves as the active-low upper data strobe. $\overline{\text{SUDS}}$ is an output during DIO and an input during DMA.§ H = Not valid data on SADH0–SADH7 lines L = Valid data on SADH0–SADH7 lines
$\overline{\text{SDTACK}}$	125	I/O	System data transfer acknowledge. The purpose of $\overline{\text{SDTACK}}$ is to indicate to the bus master that a data transfer is complete. $\overline{\text{SDTACK}}$ is internally synchronized to SBCLK by the TI380C2x. During DMA cycles, it is asserted before the falling edge of SBCLK in state T2 by the TI380PCI to prevent a wait state. $\overline{\text{SDTACK}}$ is an input when the TI380C2x is selected for DIO, and an output otherwise.§ H = System bus NOT ready L = Data transfer is complete; system bus is ready.
$\overline{\text{SBERR}}$	7	O	Bus error. $\overline{\text{SBERR}}$ corresponds to the bus error signal of the 68000 microprocessor. $\overline{\text{SBERR}}$ is driven low during a DMA cycle to indicate to the TI380C2x that the cycle must be terminated. See Section 3.4.5.3 of the *TMS380 Second-Generation Token Ring User's Guide* (SPWU005) for more information.
$\overline{\text{SLDS}}$	1	I/O	Lower data strobe. $\overline{\text{SLDS}}$ is an output during DIO and an input during DMA. $\overline{\text{SLDS}}$ serves as the active-low lower data strobe.§ H = Not valid data on SADL0-SADL7 lines L = Valid data on SADL0-SADL7 lines
$\overline{\text{SHALT}}$	3	O	System halt/bus error retry. If $\overline{\text{SHALT}}$ is asserted along with bus error ($\overline{\text{SBERR}}$), the adapter retries the last DMA cycle. This is the rerun operation as defined in the 68000 specification. See Section 3.4.5.3 of the *TMS380 Second-Generation Token Ring User's Guide* (SPWU005) for more information.
$\overline{\text{SBGR}}$	5	O	System bus grant. $\overline{\text{SBGR}}$ serves as an active-low bus grant, as defined in the standard 68000 interface. H = System bus not granted L = System bus granted
$\overline{\text{SBRQ}}$	129	I	System bus request. $\overline{\text{SBRQ}}$ is used to request control of the system bus in preparation for a DMA transfer. $\overline{\text{SBRQ}}$ is internally synchronized to SBCLK. H = System bus not requested L = System bus requested

† The TI380PCI SIF pin names correspond to a subset of the system interface pins on a TI380C2x. Consult the TI380C2x data sheet for more information on individual pins. Like-named pins on the TI380PCI and TI380C2x system interfaces are intended to be connected to each other.
‡ I = in, O = out
§ The signal connecting this pin to the TI380C2x also should be connected to a 4.7-kΩ pullup resistor.

Pin Functions – System Interface† (Continued)

PIN NAME	PIN NO.	I/O‡	DESCRIPTION
$\overline{\text{SIRQ}}$	6	I	System interrupt request. TI380C2x drives $\overline{\text{SIRQ}}$ to signal an interrupt request to the host processor. H = No interrupt request L = Interrupt request by TI380C2x
$\overline{\text{SAS}}$	2	I/O	System memory address strobe. $\overline{\text{SAS}}$ is an active-low address strobe that is an output during DIO and an input during DMA.§ H = Address not valid L = Address is valid and a transfer operation is in progress
SRNW	128	I/O	System read not write. SRNW serves as a control signal to indicate a read or write cycle. H = Read Cycle L = Write Cycle

† The TI380PCI SIF pin names correspond to a subset of the system interface pins on a TI380C2x. Consult the TI380C2x data sheet for more information on individual pins. Like-named pins on the TI380PCI and TI380C2x system interfaces are intended to be connected to each other.
‡ I = in, O = out
§ The signal connecting this pin to the TI380C2x also should be connected to a 4.7-kΩ pullup resistor.

ADVANCE INFORMATION

Pin Functions – Burned-In Address Emulation Interface†

PIN NAME	PIN NO.	I/O‡	DESCRIPTION
MBLCK1	131	I	Local memory bus clock 1.
MADH00 MADH01 MADH02 MADH03 MADH04 MADH05 MADH06 MADH07	99 98 97 96 95 93 92 91	I/O	Local memory address, data, and status bus — high byte. For the first quarter of the local memory cycle, MADH00 – MADH07 monitor address bits AX4 and A0 to A6; for the second quarter they monitor status bits, and for the third and fourth quarters they carry data bits 0 to 7. The most significant bit is MADH00, and the least significant bit is MADH07.
$\overline{\text{MBIAEN}}$	19	I	Burned-in address enable. $\overline{\text{MBIAEN}}$ enables the output of data on the MADHxx lines during BIA accesses.
MAX2	18		Local memory extended address bit. For the first quarter of a local memory cycle MAX2 monitors AX2. For quarters two through four, MAX2 monitors A14.
MAX0	9	I	Local memory extended address bit. For the first quarter of a local memory cycle MAX0 monitors AX0. For quarters two through four, MAX0 monitors A12.
$\overline{\text{MROMEN}}$	8	I	$\overline{\text{MROMEN}}$ monitors the local memory bus ROM enable signal.

† The TI380PCI BIF pin names correspond to a subset of the local memory bus interface pins on a TI380C2x. Like-named pins on the two devices are intended to be connected to each other. Consult the TI380C2x data sheets for more information on individual pins.
‡ I = in, O = out

ADVANCE INFORMATION

Pin Functions – PCI Interface

PIN NAME	PIN NO.	I/O†	DESCRIPTION
AD31 AD30 AD29 AD28 AD27 AD26 AD25 AD24 AD23 AD22 AD21 AD20 AD19 AD18 AD17 AD16 AD15 AD14 AD13 AD12 AD11 AD10 AD09 AD08 AD07 AD06 AD05 AD04 AD03 AD02 AD01 AD00	27 29 30 32 33 35 36 38 42 43 44 45 47 48 49 51 66 68 69 71 72 73 74 75 78 80 81 83 84 85 86 87	I/O	PCI address and data. A bus transaction to or from one of these pins consists of an address phase followed by one or more data phases. The address phase is the clock cycle in which FRAME is asserted. During the address phase AD31 – AD00 contain a physical address (32 bits). For I/O, this is a byte address; for configuration and memory it is a DWORD address. During data phases, AD07– AD00 contain the LSB, and AD31 – AD24 contain the MSB. Write data is stable and valid when IRDY is asserted, and read data is stable and valid when TRDY is asserted. Data is transferred during those clocks where both IRDY and TRDY are asserted.
C/BE3 C/BE2 C/BE1 C/BE0	39 53 65 77	I/O	Bus command and byte enables. During the address phase of a transaction, C/BE3 – C/BE0 define the bus command. During the data phase, C/BE3 – C/BE0 are used as byte enables. The byte enables are valid for the entire data phase and they determine which byte lanes carry meaningful data. C/BE0 applies to the LSB and C/BE3 applies to the MSB.
DEVSEL	59	I/O	Device select. When DEVSEL is actively driven, it indicates the driving device has decoded its address as the target of the current access. As an input, DEVSEL indicates whether any device on the bus has been selected. If no PCI agent has asserted DEVSEL, then the TI380PCI shall remove itself as the PCI bus master.
FRAME	55	I/O	Cycle frame. FRAME is driven by the current master to indicate the beginning and duration of an access. It is asserted to indicate a bus transaction is beginning. While FRAME is asserted, data transfers continue. When FRAME is deasserted, the transaction is in the final data phase.
GNT	25	I	Grant. GNT indicates to the TI380PCI that access to the PCI bus has been granted. This is a point-to-point signal.
IDSEL	41	I	Initialization device select. IDSEL is used as a chip select during configuration read and write transactions.
INTA	20	O	Interrupt A. INTA is used to request an interrupt. The assertion and deassertion of INTA is asynchronous to PCLK.

† I = in, O = out

ADVANCE INFORMATION

Pin Functions – PCI Interface (Continued)

PIN NAME	PIN NO.	I/O†	DESCRIPTION
$\overline{\text{IRDY}}$	56	I/O	Initiator ready. $\overline{\text{IRDY}}$ indicates the bus master's ability to complete the current data phase of the transaction. If a data phase is completed on any clock, both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are sampled and asserted. During a write, $\overline{\text{IRDY}}$ indicates that valid data is present on AD31 – AD00. During a read, it indicates the master is prepared to accept data. Wait cycles are inserted until both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted together.
PCLK	23	I	Clock. PCLK provides timing for all transactions on the PCI bus. All other PCI signals, except $\overline{\text{RST}}$ and $\overline{\text{INTA}}$, are sampled on the rising edge of PCLK. PCLK is used to generate the SBCLK signal that goes to the TI380C2x.
$\overline{\text{PERR}}$	61	I/O	Parity error. $\overline{\text{PERR}}$ is for the reporting of data parity errors during all transactions. It is driven active by the agent receiving data two clocks following the data when a data parity error is detected. The minimum duration of $\overline{\text{PERR}}$ is one clock for each data phase that a data parity error is detected. There are no special conditions when a $\overline{\text{PERR}}$ can be lost or when reporting of an error can be delayed. An agent cannot report a $\overline{\text{PERR}}$ until it has claimed the access by asserting $\overline{\text{DEVSEL}}$ and completing a data phase.
$\overline{\text{REQ}}$	26	O	Request. REQ indicates to the arbiter that TI380PCI desires use of the bus. $\overline{\text{REQ}}$ is a point-to-point signal.
$\overline{\text{RST}}$	21	I	Reset. $\overline{\text{RST}}$ is used to hard reset the LAN subsystem, including TI380C2x and the TI380PCI. To prevent AD, $\overline{\text{C/BE}}$, and PAR signals from floating during reset, the central device can drive these lines during reset (bus parking) but only to a logic low level; they can not be driven high.
$\overline{\text{SERR}}$	62	I/O	System error. $\overline{\text{SERR}}$, when enabled, reports address parity errors or any other system error where the result is catastrophic. The assertion of $\overline{\text{SERR}}$ is synchronous to the clock and meets the setup and hold times of all PCI signals.
$\overline{\text{STOP}}$	60	I/O	Stop. $\overline{\text{STOP}}$ indicates the current slave is requesting the master to stop the current transaction.
$\overline{\text{TRDY}}$	57	I/O	Target ready. $\overline{\text{TRDY}}$ indicates the target agent's (the TI380PCI) ability to complete the current data phase of the transaction. If a data phase is completed on any clock, both $\overline{\text{TRDY}}$ and $\overline{\text{IRDY}}$ are sample asserted. During a read, $\overline{\text{TRDY}}$ indicates that valid data is present on AD31 – AD00. During a write, it indicates that the target is prepared to accept data. Wait cycles are inserted until both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted together.

† I = in, O = out

Pin Functions – Serial EEPROM Interface

PIN NAME	PIN NO.	I/O†	DESCRIPTION
EDIO	111	I/O	EEPROM data input/output is a bidirectional signal used to transfer data into or out of the EEPROM memory. Note I^2C memory devices typically require a pullup resistor to V_{CC} on this line. This is an open drain output. If the optional EEPROM is not present, EDIO should be tied high.
EDC	122	O	EEPROM clock signal used to synchronize all data in and data out of the memory. I^2C memory devices typically require a pullup resistor to V_{CC} on this line. If the optional EEPROM is not present, EDC should be tied low.

† I = in, O = out

architecture

The major blocks of the TI380PCI, illustrated in Figure 3, include the PCI interface (PCIIF), TI380C2x interface (SIF), ROM interface (ROMIF), serial EEPROM interface (EIF), and burned-in address emulation interface (BIF). The functionality of each block is described in the following sections.

PCI interface (PCIIF)

The PCIIF block contains all logic needed to interact with the PCI bus and allows the TI380PCI to assume the role of bus master or slave. This block controls all communication between the TI380PCI and the PCI bus. All signals entering the PCIIF and generated within it are synchronous to the PCLK signal.

The TI380PCI also checks and generates parity for data presented on the PCI bus at the PCIIF.

The PCIIF generates and responds to the following PCI cycles:

TI380PCI CAN GENERATE:	TI380PCI RESPONDS TO:
Memory read line	I/O reads and writes
Memory write and invalidate	Configuration reads and writes
Memory write	All types of memory reads and writes

The TI380PCI asserts slave-initiated termination, if the initiator exceeds one data phase transfer.

address decode

The TI380PCI uses only the most significant 27 address bits presented in AD31–AD00 to decode an access directed at the TI380PCI for a TI380C2x DIO access, and the most significant 16 bits for a ROM access. The TI380PCI's base address selects a block of eight continuous 32-bit memory or I/O locations for DIO access or 64K bytes of address space for ROM access. Access to each register within the selected memory, I/O, or configuration space is uniquely decoded. The decode logic returns $\overline{\text{DEVSEL}}$ as a medium device.

The PCIIF logic generates control signals that propagate either to configuration registers, the SIF, ROMIF, or EIF within the TI380PCI. The address, command, and byte enables are valid and sustained throughout each PCI cycle. Configuration registers can be accessed through configuration cycles when the TI380PCI's address is decoded.

address/data parity checking and generation

The PCIIF is responsible for checking and generating parity for addresses during bus master operation and for data during master and slave operations. If the PCIIF encounters $\overline{\text{SERR}}$ two PCLKs after the address phase when it is mastering the PCI bus, it terminates its transaction immediately and regains control of the TI380C2x SIF. If the PCIIF detects any address (even if not selected) or data parity (only when selected) while operating as a PCI slave, or if it detects data parity while operating as a PCI bus master, it pulses $\overline{\text{PERR}}$ and sets bit 15 in the status register. Bus transactions causing the parity error are otherwise ignored by the LAN subsystem. Any bus mastering transaction that encounters a parity error is aborted by the TI380PCI. This detection also prevents future bus master cycles from occuring until a reset occurs.

The TI380PCI performs a bus master cycle when bus mastering is enabled and the TI380C2x requests the SIF bus.

The supported PCI master cycles are:

- Memory write
- Memory read line
- Memory write and invalidate

address/data parity checking and generation (continued)

During PCI master cycles, PCIIF is able to handle the following:

- Master-abort termination: due to no $\overline{\text{DEVSEL}}$ response
- Slave-initiated termination: disconnect/retry and slave-abort
- Parity error handling

TI380C2x interface (SIF)

The SIF handles all communications between the TI380C2x and the TI380PCI.

The TI2000 software standard† requires the TI380PCI SIF to include a configuration register at I/O address 0xE–0xF. This is the same location as the TI380C2x DIO DMALEN register, therefore, access to the DIO DMALEN register from the TI380PCI is masked, and the TI380PCI configuration register information is presented instead.

When there is no DIO access pending, the TI380PCI SIF acknowledges all $\overline{\text{SBRQ}}$ from the TI380C2x and allows the TI380C2x to control the SIF bus. If the PCIIF initiates a DIO access while the TI380C2x is controlling the SIF bus and is performing a DMA access, the SIF signals the TI380C2x to relinquish control of the SIF bus and let the DIO proceed. On completion of the DIO cycle, the SIF logic again grants the SIF bus to the TI380C2x, if the interrupted DMA sequence is not complete.

The interface between the TI380PCI and the TI380C2x operates in Motorola mode. The interface is designed so it can operate with drivers that use 8- or 16-bit I/O instructions on the host. Accesses to the SIFDAT, SIFDAT/INC, and SIFADR registers are ideally made with 16-bit I/O instructions from the host. If a driver has to perform byte-accesses to these registers, then the odd byte must be accessed before the even byte.

Note that on Intel platforms the INSTR word and the OUTSTR word 16-bit I/O instructions use a little-endian byte ordering that does not work without additional byte-swapping when used with the TI380PCI Motorola-mode big-endian interface. These instructions are typically used only in 16-bit pseudo DMA drivers.

For more information on byte operations between the host and the TI380C2x, see the description of DIO operations on pages 4-16 and 4-28 in the *TMS380 Second Generation User's Guide*.

interrupt request

interrupting the PCI host

The SIF takes an $\overline{\text{SIRQ}}$ from the TI380C2x, sends it to the PCIIF, and the PCIIF then translates it into an $\overline{\text{INTA}}$ on the PCI bus. The generation of $\overline{\text{INTA}}$ on the PCI bus is held off until the data in the TI380PCI FIFO has been emptied. This interrupt is done to assure data coherency.

clearing the host interrupt request

Driver software running on the host can clear the $\overline{\text{SIRQ}}$ interrupt by writing a 0 to bit 8 of the SIFCMD register in the TI380C2x. Writing a 1 to this location has no effect (see *TI380C2x User's Guide*).

ROM interface (ROMIF)

The ROMIF supports remote program load for PCI LAN adapter cards. This interface supports a 64K-byte ROM address range aligned on a 64K-byte boundary. Since the PCI bus can access the ROM in any byte order, the ROMIF reads all 32 bits for each PCI ROM access. This latency is acceptable because the PCI bus does not execute code out of the ROM. The ROM is normally accessed during system configuration with performance is not critical. The ROMIF uses either 14 or 16 bits of the address depending on the setting of bit 17 in the MISCCTRL register. In 14-bit mode, the ROMIF appends two bits to the most significant end of the address from its internal page register. The TI2000† software standard requires that writes to the ROM location increment to the next 16K pages and that the ROM pages be cycled through only once until the next reset. Note that $\overline{\text{ROMCS}}$ does not go low when the host attempts to write to the EPROM.

† TI2000 software standard specification is available from your local TI Sales Office.

ROM interface (ROMIF) (continued)

When $\overline{RST}$ is driven high, the value on ROMA[07:00] is latched into the board configuration register in the TI380PCI configuration space. The value on ROMA[07:00] can be provided by pullup and pulldown resistors that do not affect operation after reset. This feature allows designers to support jumpers or board stuffing options sensed by software that reads the board configuration register. If pullup and pulldown registers are not used, the contents of the board configuration register are undefined after reset.

When the host computer initiates a 32-bit read to the ROMIF, the TI380PCI fetches four bytes from the ROM and presents the resulting 32-bit DWORD to the PCI bus. The fetches from the ROM start with the least significant byte, byte 0, followed by bytes 1, 2, and 3. The data is presented to the PCI data bus as shown in Figure 4.

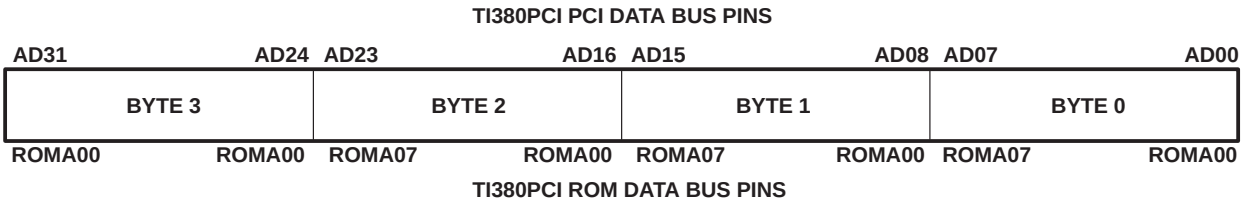


Figure 4. PCI Bus Data

The ROMIF has been designed for operation with EPROM devices with 100 ns access times.

serial EEPROM interface (EIF)

The TI380PCI includes an interface for an optional I^2C serial EEPROM. The EEPROM contains the following:

- Bytes $0 \times 0 – 0 \times 7$ contain eight bytes of PCI configuration information that automatically are loaded into the appropriate PCI configuration register by the 380PCI at power on.
- Byte 0×8 is a checksum calculated on bytes 0–7. This checksum byte automatically is read at power on by the TI380PCI. If the checksum read from byte 8 does not agree with the checksum that the 380PCI calculated from bytes 0–7, then the 380PCI sets the ECRCERR and NEP bits in the EEPROM read/write register in 380PCI configuration space.
- Bytes $0 \times 09 – 0 \times 10$ contain the six bytes of the BIA and two bytes of BIA checksum. The 380PCI reads these bytes and stores them in an internal register, the contents of which are presented to TI380C2x local memory bus when the TI380PCI detects an access to the BIA. In this way the TI380PCI emulates the presence of a BIA ROM on the TI380C2x local memory bus. Note that the TI380PCI does not verify the BIA checksum.
- Byte 0×11 is reserved for use by TI2000 drivers. This byte is not read automatically by the TI380PCI, but it is read by TI2000 drivers that load bit 0 from this byte into the TI380PCI MISCCTRL register bit 8 and load bit 1 from this byte into TI380PCI MISCCTRL register bit 9. This byte allows the TI2000 driver to read the network speed and topology from nonvolatile memory before attempting to access the network.
- Bytes $0 \times 12 – 0 \times 20$ are reserved for future use by the TI380PCI.
- Bytes $0 \times 21 – 0 \times FF$ are available for user-defined variable storage.

serial EEPROM interface (EIF) (continued)

The user can program several copies of the BIA into the EEPROM user-defined space as a protection against inadvertent modification of the copy of the BIA at addresses 0x08–0x0F.

Table 1. EIF Configuration Registors

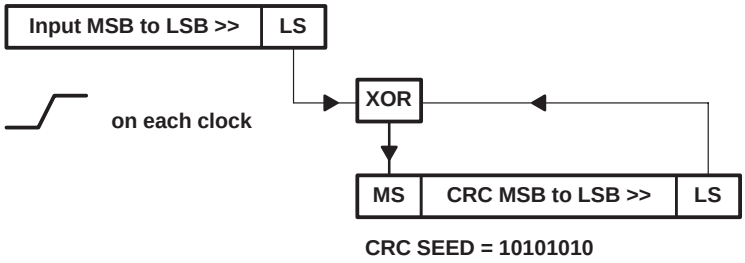
PARAMETER DESCRIPTION	EEPROM BYTE ADDRESS	DEFAULT VALUE
Vendor ID least significant byte	0x00	0x4C
Vendor ID most significant byte	0x01	0x10
Device ID least significant byte	0x02	0x08
Device ID most significant byte	0x03	0x05
Revision ID	0x04	0x10
Sub-class	0x05	0x80
Min_Gnt	0x06	0x01
Max_Lat	0x07	0x00
Checksum	0x08	0x7A
Burned-in address	0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F 0x10	BIA MSbyte BIA byte BIA byte BIA byte BIA byte BIA LSbyte BIA MS checksum BIA LS checksum
TI2000 reserved byte	0x11	0x01
Reserved for future use	0x12 ... 0x20	Reserved
User defined	0x21 ... 0xFF	User defined

Systems that do not use an EEPROM should tie the EDC pin low and the EDIO pin high. This action causes default values to be loaded into the configuration registers as shown in Table 1. EDC and EDIO are sampled at the deassertion of $\overline{\text{RST}}$. If EDC is low and EDIO is high at the deassertion of reset, the TI380PCI assumes that no EEPROM is present in the system.

The default values for the parameters stored in the EEPROM can always be accessed by software because copies of these values are stored in a series of hardwired registers in the TI380PCI configuration space. These registers are read only and are unaffected by the process of loading data from the optional EEPROM.

The algorithm that should be used to generate the cyclic redundancy check (CRC) code placed in the serial EEPROM byte 08 is included here. When the TI380PCI reads the EEPROM after power up, it checks the CRC in byte 08 to ensure that it was generated with this algorithm. If the CRC from byte 08 of the EEPROM does not match the CRC calculated from EEPROM bytes 0–7, then the TI380PCI sets the NEP and ECRCERR bits in the EEPROM read/write register. When the NEP and ECRCERR bits are set owing to a CRC mismatch on the data read from the EEPROM, the TI380PCI still continues to load the data read EEPROM into the PCI configuration registers. Note that in the event that the EEPROM is unprogrammed, the TI380PCI configuration registers may be left in an unknown state. In such a case, the EEPROM can be initialized by using PCI BIOS calls to read the hardwired configuration register addresses associated with each PCI slot. The slot containing the TI380PCI responds with the contents of the hardwired registers. Once the slot containing the TI380PCI has been identified, the EEPROM can be programmed using the EEPROM register in the PCI configuration space. After the next power cycle the TI380PCI loads the contents of the initialized EEPROM.

serial EEPROM interface (EIF) (continued)



It is assumed that data in the serial EEPROM is shifted LSB to MSB, consistent with other serial communication streams.

This is the routine that calculates the Serial EEPROM CRC over the first eight bytes of the EEPROM. The return value is written to the ninth byte of the EEPROM. The "MSB" argument passed is a copy of the bytes read from the EEPROM and stored in an array. Note that the BIA has its own checksum, which is not verified by the TI380PCI.

```
unsigned short xor-crc (unsigned short   *msb, int bytes)
{
        BYTE crc=0xaa; // seed
        int i;
        for (i=0; i<bytes; i++)
                crc^=msb[i];
        return (crc);
}
```

The BIA downloaded from the EEPROM ultimately appears in the adapter RAM at chapter 0, address 0. The high or odd byte of each 16-bit word contains two hex digits of the 12-digit BIA, the most significant digit first. The last two high bytes of the first eight 16-bit words contain the 16-bit checksum of the BIA. The checksum is performed as shown below.

example:

Here is an adapter RAM memory dump for a card with BIA = 0001fafe1093.

Address		Data
000000	:	0000 0100 fa00 fe00 1000 9300 0b00 9200

Address	Data	Checksum
01	00	0001
03	01	fafe
05	fa	+ 1093
07	fe	_____
09	10	(1) 0b92
0b	93	
0c	0b	
0d	92	

ADVANCE INFORMATION

serial EEPROM interface (EIF) (continued)

PARAMETER DESCRIPTION	HARDWIRED REGISTER NAME (CONTAINS READ-ONLY COPY OF DEFAULT VALUE)	REGISTER ADDRESS IN TI380PCI CONFIGURATION SPACE
Vendor ID	Hardwired vendor ID	4C
Device ID	Hardwired device ID	4C
Revision ID	Hardwired revision ID	50
Subclass	Hardwired subclass	50
Min_Gnt	Hardwired Min_Gnt	50
Max_Lat	Hardwired Max_Lat	50

The EEPROM also can be read and written under software control by PCI configuration register 0x48. A typical I^2C EEPROM that can be used with the TI380PCI is the 24C02. EEPROMs that provide external address pins for identification of several devices on the I^2C bus should have these address pins pulled down to logic 0.

burned-in address interface (BIF)

The BIF logic is designed to connect directly to the local memory bus of the TI380C2x. The BIF serves two purposes:

- Snoops the status cycles on the local memory bus to detect the initialization and completion of DMA transfers from the TI380C2x.
- Detects accesses to the BIA ROM from the TI380C2x. During a BIA access, the TI380PCI drives data onto MADH00−07, emulating a BIA PROM. This eliminates the need for a separate BIA PROM.

in-circuit test NAND tree operation

The terminal function table in this data sheet explains how to use the $\overline{ICT}$ pins on the TI380PCI to place the output pins of the TI380PCI in a 3-state mode to allow for in-circuit testing of a PCB.

The TI380PCI also contains a NAND tree that can be used during the in-circuit testing process to verify the integrity of the solder connections to the TI380PCI. The NAND tree is activiated by latching a control value on the SADL0–7 pins on the rising edge of the signal applied to the $\overline{ICT}$ pin.

To activate the NAND tree:

1. Drive and hold $\overline{ICT}$ low.

2. Drive and hold the MAX0 input low and drive the MAX2 input high. Drive and hold the SADL0–7 pins as follows:

SADL7	SADL6	SADL5	SADL4	SADL3	SADL2	SADL1	SADL0
0	0	0	0	0	0	1	0

3. While holding the values in step 2, pull the $\overline{ICT}$ input high to latch the values from step 2.

The connections to the TI380PCI can now be tested with the NAND tree. The NAND tree can be deactivated by repeating steps 1 through 3 with the data applied to the SADL0–7 pins in step 2 set as follows:

SADL7	SADL6	SADL5	SADL4	SADL3	SADL2	SADL1	SADL0
0	0	0	0	0	0	0	0

ADVANCE INFORMATION

TI380PCI device operation

TI380PCI behavior as a PCI bus target

The PCIIF in the TI380PCI acts as an I/O slave to the PCI bus after power up and when the LAN subsystem is not the PCI bus master. It monitors and decodes PCI commands for memory, I/O, and configuration accesses. Enabling of memory and I/O accesses are performed by setting and cleaning bits in the COMMAND register.

TI380PCI behavior as a PCI bus master

The TI380PCI requests to become a master on the PCI bus after the TI380C2x initiates a DMA transfer to the TI380PCI. The TI380C2x begins a DMA series by asserting $\overline{\text{SBRQ}}$ and requesting its own system interface bus. The TI380PCI performs a DIO access to the TI380C2x and reads the DMA length before granting the SIF bus to the TI380C2x. (Note that the read cycle occurs only on the first transaction of the DMA access). If the TI380C2x has requested DMA write access and no DIO access from the PCI bus is pending, the TI380PCI acknowledges the request and latches the DMA address and completes the write cycle from the TI380C2x to start filling the FIFO. Subsequent writes continue to fill the FIFO. When the FIFO contains 60 bytes of data, the PCIIF logic arbitrates for the PCI bus and transfer the data in a burst sequence.

If the access is a read access, the TI380PCI acknowledges the request and latches the DMA address. It then initiates a read access on the PCI bus. If the DMA length exceeds the cache line size, as defined by the contents of the TI380PCI cache line size configuration register, then the read access on the PCI bus is a burst read. As data is available from the PCI bus, it is provided to the TI380C2x. Since the PCI bus has a higher bandwidth than the TI380C2x SIF bus, the 64-byte FIFO in the TI380PCI fills as the burst read continues. If the FIFO fills before the DMA length is reached, the PCI master relinquishes the bus until the FIFO is almost empty.

If a DIO conflict occurs while the DMA is in progress, the TI380PCI forces the TI380C2x off the SIF bus to allow the DIO to complete. The DMA operation resumes at the point it was interrupted and continues until completion.

removal of TI380PCI as bus master

The TI380PCI is removed as the PCI bus master under two conditions:

- target-inititiated retry and/or termination
- master-initiated termination as no PCI agent asserted $\overline{\text{DEVSEL}}$.

TI380PCI behavior on the SIF bus

SIF behavior as a SIF master

The TI380PCI SIF logic masters the TI380C2x system interface to translate signals from the PCIIF into TI380C2x control signals and relays TI380C2x return signals to the PCIIF.

SIF behavior as a slave

The SIF acts as a slave when the TI380C2x is engaged in DMA operations. During DMA transfers, other PCI masters can access the TI380C2x's DIO registers. The TI380PCI retries all accesses from the PCI bus while posting a preempt to the TI380PCI SIF. When the TI380PCI SIF receives the preempt, it signals the TI380C2x to relinquish the SIF bus at the earliest opportunity and does not grant the TI380C2x the SIF bus until the preempt is removed.

FIFO control

The 64-byte FIFO serves to reduce the bandwidth demand on the PCI bus from the TI380PCI. It is used only when the TI380C2x is functioning as a PCI bus master during DMA accesses.

FIFO operation (write cycles)

The TI380PCI BIF snoops the status cycles on the TI380C2x local bus. When the TI380PCI detects a status code that indicates the start of a DMA write from the TI380C2x to the host computer, the TI380PCI immediately initiates a DIO read of the DMALEN register in the TI380C2x. When the DMALEN register has been read, the TI380PCI grants the TI380C2x control of the SIF bus and the DMA write can proceed.

The DMALEN register is not read again until the next DMA transaction starts. During DMA write cycles from the TI380C2x, the TI380PCI stores one cache line in the FIFO before requesting the PCI bus. It drives data on the PCI bus as long as it has sufficient data in the FIFO or until it loses the PCI bus. The FIFO control requests the PCI bus under the following conditions:

- A full cache line is available for writing and the DMA address is aligned on a cache boundary. The PCI bus cycle is a write-and-invalidate cycle.
- The TI380C2x relinquishes control of the SIF bus owing to completion of a DMA write from the TI380C2x with DMA data in the TI380PCI FIFO. If the quantity of data in the TI380PCI FIFO is greater than or equal to the cache line size and the DMA address is aligned on a cache line boundary, the PCI cycles are write-and-invalidates. If the quantity of data in the TI380PCI FIFO is less than the cache line size or the DMA address is not aligned on a cache line boundary, the PCI cycles are memory writes.
- A full 32-bit word is available for the PCI bus and the cache line size is zero.

If the host attempts a DIO access during a DMA transfer between the TI380C2x and the host, the TI380PCI signals the TI380C2x to pause the DMA transfer. When the TI380PCI regains control of the SIF bus, the DIO access from the host is allowed to complete, and then the TI380PCI allows the DMA transfer to restart and run to completion. Note that the TI380PCI signals the host to retry the DIO access until the TI380PCI gains control of the SIF bus and can allow the DIO access to complete. In the case of a DMA write from the TI380C2x to the host (which is almost complete), the TI380PCI can give up control of the SIF bus prior to transferring the last data from the FIFO to the host. If the host initiates a DIO access at this time, the DIO access is allowed to complete immediately, i.e., before the FIFO empties.

FIFO operation (read cycles)

The TI380PCI BIF snoops the status cycles on the TI380C2x local bus. When the TI380PCI detects a status code that indicates the start of a DMA read by the TI380PCI from the host computer, the TI380PCI immediately initiates a DIO read of the DMALEN register in the TI380C2x. When the DMALEN register has been read, the TI380PCI grants the TI380C2x control of the SIF bus, and the DMA read can proceed. The DMALEN register is not read again until the next DMA transaction starts. The TI380PCI then requests the PCI bus and begins filling the FIFO. As data is available, it is provided to the TI380C2x. The TI380PCI counts the bytes received from the PCI bus and decrements its DMA length counter. If the DMA count is not zero, but the TI380C2x releases the SIF bus, the TI380PCI continues to read data from the PCI bus. The TI380PCI logic terminates the PCI ownership when the DMA count goes to zero.

The TI380PCI always performs memory-read line commands on the PCI bus regardless of DMA count or address alignment.

BIA interface

On power up, BIA data is loaded from the EEPROM and held in registers inside the TI380PCI. All the pins on the local bus of the TI380C2x necessary to sense an access from the TI380C2x to an attached BIA ROM are connected to the TI380PCI. When the TI380PCI senses an access to the BIA ROM, it drives the BIA data onto the MADHxx bus, simulating the presence of a BIA ROM on the TI380C2x local bus.

TI380PCI registers

The TI380PCI supports a number of registers to facilitate communications between the host computer and the token-ring LAN subsystem. It also performs address translation to map TI380C2x DIO registers into host computer memory or I/O space. Table 2 describes the configuration space registers implemented within the TI380PCI. The registers shown as shaded are autoloaded from an external serial EEPROM at power up.

Table 2. Configuration Space Header

DWORD ADDRESS	BYTE 3	BYTE 2	BYTE 1	BYTE 0	READ/WRITE
0x00	Device ID		Vendor ID		R
0x04	Status		Command		R/W
0x08	Base-Class	Sub-Class		Revision ID	R
0x0C		Header Type	Latency Timer	Cache Line Size	R/W
0x10	Base Address I/O				R/W
0x14	Base Address Memory				R/W
0x18	Reserved (returns 0)				R
0x1C	Reserved (returns 0)				R
0x20	Reserved (returns 0)				R
0x24	Reserved (returns 0)				R
0x28	Reserved (returns 0)				R
0x2C	Reserved (returns 0)				R
0x30	Expansion ROM Base Address				R/W
0x34	Reserved (returns 0)				R
0x38	Reserved (returns 0)				R
0x3C	Max_Lat	Min_Gnt	Interrupt Pin	Interrupt Line	R/W
0x40	Miscellaneous Control (MISCCTRL) Register				R/W
0x44	Board Config Register				R
0x48	Reserved (returns 0)		EEPROM		R/W
0x4C	Hardwired DEVICE ID		Hardwired VENDOR ID		R
0x50	H/W Subclass	H/W Max Lat	H/W Min Gnt	H/W Revision ID	R
0x54	380PCI Interface Control Register				R
0x58–0xFF	Reserved (returns 0 when read)				

ADVANCE INFORMATION

device identification

The TI380PCI contains the following device identification information in its configuration space.

REGISTER	HEX VALUE
Vendor ID	104Ch†
Device ID	0508h†
Revision ID	01
Header type	00
Class code	028000

† These values are defaults and can be overwritten by the contents of the optional EEPROM.

COMMAND register—configuration space DWORD address (0x04)

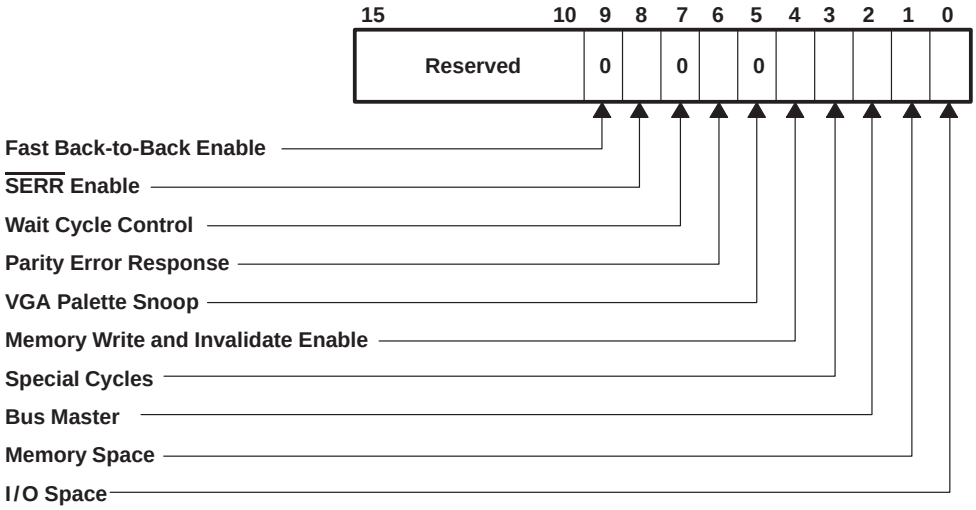


Figure 5. Command Register Layout

The COMMAND register provides coarse control over the TI380PCI's ability to generate and to respond to PCI cycles. When 0 is written to this register, the device is logically disconnected from the PCI bus for all accesses except configuration accesses.

The COMMAND register is set to 0 at power up and after hardware reset.

Bit 00: Controls TI380PCI's response to I/O space accesses. A value of 0 disables TI380PCI's response. A value of 1 allows TI380PCI to respond to I/O space accesses.

Bit 01: Controls TI380PCI's response to memory space accesses. A value of 0 disables TI380PCI's response. A value of 1 allows TI380PCI to respond to memory space accesses.

Bit 02: Controls TI380PCI's ability to act as a master on the PCI bus. A value of 0 disables TI380PCI from generating PCI accesses. A value of 1 allows the TI380PCI to behave as a bus master. Note that if the automatic software reset bit in the 380PCI interface control register (0x54) is set to a one then the action of clearing the bus master bit of the 380PCI command register from one to zero also resets the TI380PCI.

Bit 03: CONTROL[3] allows the LAN subsystem to enter low-power mode following a shutdown broadcast. All other special cycle messages are ignored. Device drivers must reinitialize and download micro code to the LAN subsystem when the subsystem exits low-power mode.

COMMAND register—configuration space DWORD address (0x04) (continued)

Bit 04: CONTROL[4] controls memory write-and-invalidate cycles. A value of 1 enables the TI380PCI to generate memory write-and-invalidate cycles.

Bit 05: CONTROL[5] applies to VGA subsystems and is hardwired to 0.

Bit 06: This bit controls the TI380PCI's response to parity errors. When this bit is set, the TI380PCI takes its normal action when a parity error is detected. When this bit is reset, the TI380PCI ignores any parity errors that it detects and continues normal operation. This bit is set to 0 after reset.

Bit 07: CONTROL[7] is hardwired to a 0 since the TI380PCI does not support address/data stepping.

Bit 08: This bit is an enable bit for the $\overline{\text{SERR}}$ driver. A value of 0 disables the $\overline{\text{SERR}}$ driver and a value of 1 enables it. This bit's state after $\overline{\text{RST}}$ is 0.

Bit 09: CONTROL[9] is hardwired to a 0 since the TI380PCI does not support fast back-to-back transaction to different devices.

Bits 10–15: Reserved. These reserved bits read as 0.

STATUS register—configuration space DWORD address (0x04)

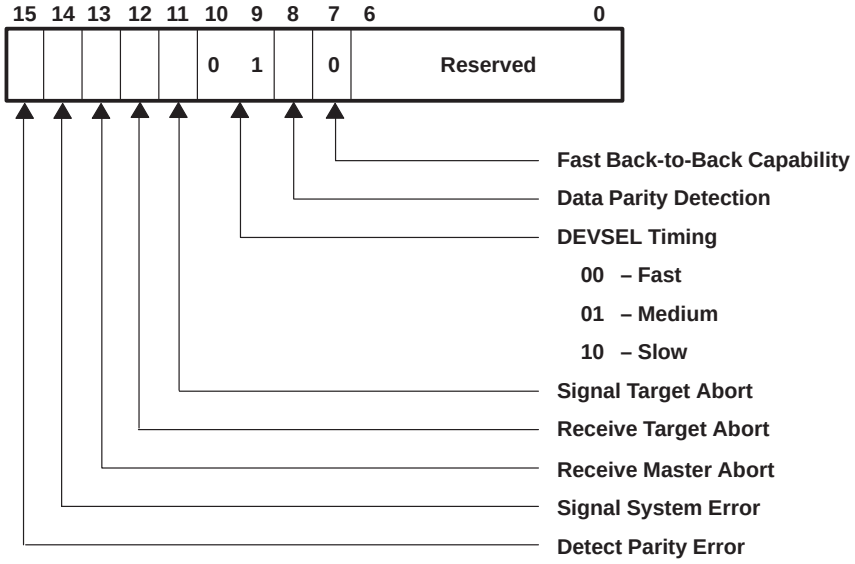


Figure 6. Status Register Layout

The STATUS Register records status information for PCI bus-related events. Reads to this register behave normally. Writes are slightly different in that bits can be reset, but not set. A bit is reset whenever the register is written, and the data in the corresponding bit location is a 1.

Bits 0–6: Reserved

Bit 7: STATUS[7] is hardwired to a 0 as the TI380PCI does not support fast back-to-back cycles as a slave.

Bit 8: This bit is set when three conditions are met: 1) the PCI device asserted $\overline{\text{PERR}}$ itself or observed $\overline{\text{PERR}}$ asserted; 2) the device setting the bit acted as the bus master for the operation in which the error occurred; 3) the parity error response bit (command register) is set.

Bits 9–10: $\overline{\text{DEVSEL}}$ timing for the TI380PCI is set to 01 (medium).

Bit 11: This bit is set whenever TI380PCI terminates a transaction with slave-abort.

STATUS register—configuration space DWORD address (0x04) (continued)

Bit 12: This bit is set whenever TI380PCI's transaction is terminated with slave-abort.

Bit 13: This bit is set whenever TI380PCI's transaction is terminated with master-abort.

Bit 14: This bit is set whenever TI380PCI asserts $\overline{\text{SERR}}$.

Bit 15: This bit is set by TI380PCI whenever it detects a parity error, even if parity-error handling is disabled (as controlled by bit 6 in the command register).

board configuration register (Board Config)—configuration space DWORD address (0x44)

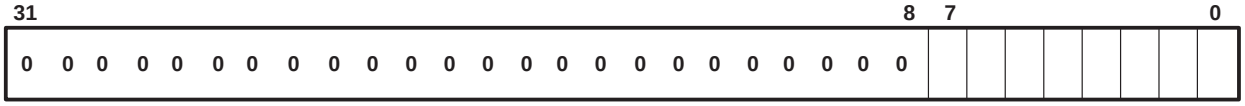


Figure 7. Board Configuration Register

Bits 08–31: These bits must read as 0.

Bits 00–07: When $\overline{\text{RST}}$ is driven high, the value on ROMA[07:00] is latched into the board configuration register in the TI380PCI configuration space. The value on ROMA[07:00] can be provided by pullup and pulldown resistors that do not affect operation after reset. This feature allows designers to support jumpers or board-stuffing options that can be sensed by software that reads the board configuration register. If pullup and pulldown registers are not used, the contents of the board configuration register are undefined after reset.

miscellaneous functions

cache line size (CLS)—configuration space DWORD address (0x0c)

The CLS register is loaded with the host system data cache line size. On reset, it is set to 0. The value in this register controls the TI380PCI FIFO fill/flush algorithm. The value in this register should be a power of two that is greater than or equal to 4, or a sum of those powers of two.

latency timer (LT)—configuration space DWORD address (0x0C)

The TI380PCI supports burst data transfer on the PCI bus; therefore, a latency timer register is needed as defined in the PCI specification.

built-in self-test (BIST)—configuration space DWORD address (0x0C)

The TI380PCI does not support built-in self-test and returns 0 when the BIST register is read.

base address register (BASE0)—configuration space DWORD address (0x10)

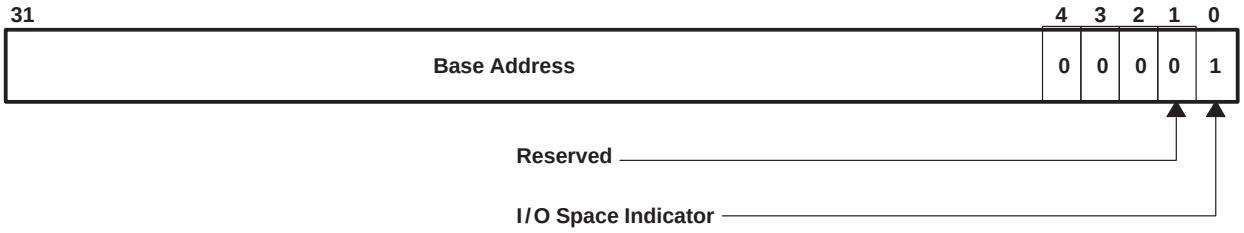


Figure 8. Base Address Register for I/O

The TI380PCI supports only one I/O address range decoded down to eight continuous 32-bit locations. After reset, the value of the address map is set to 0x1. Bits 0–4, are read only as 0x1. The LAN subsystem is expected to reside in I/O space under normal operations for conformance with the TI2000 software specification.

base address register (BASE1)—configuration space DWORD address (0x14)

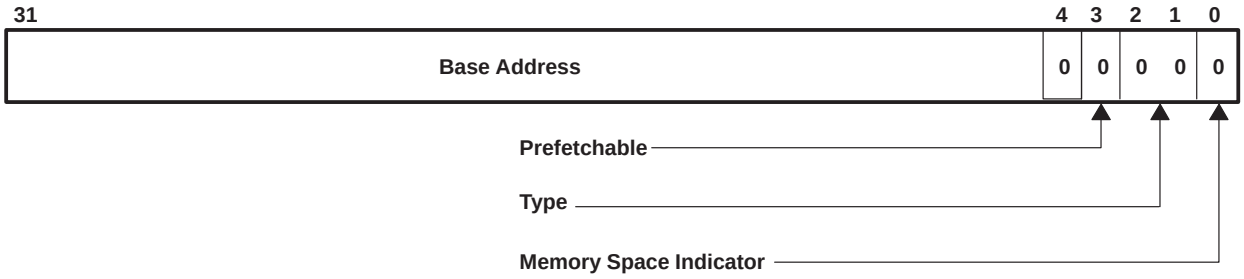


Figure 9. Base Address Register for Memory

The TI380PCI supports only one memory address range decoded down to eight continuous 32-bit locations. After reset, the value of the address map is set to 0x0. Bits 0–4 are read only as 0x0. The LAN subsystem is expected to reside in I/O space under normal operations for conformance with the TI2000 software, indicating that this register is not utilized.

interrupt line—configuration space DWORD address (0x3C)

TI380PCI uses a PCI bus interrupt so this register is implemented as an 8-bit read/write register. The value after reset is 0x00. The least significant four bits are passed to the MISCCTRL register and the TI2000 configuration register.

interrupt pin—configuration space DWORD address (0x3C)

This is hardwired to 0x01 indicating $\overline{\text{INTA}}$ is used as the interrupt pin.

maximum latency (Max_Lat)—configuration space DWORD address (0x3C)

The TI380C2x has a larger than 1-Kbyte buffer for storing network data; thus, at the maximum network data rate of 16 Mbps, the TI380C2x can be serviced at intervals greater than 64 μs. This register is loaded with a value from the EEPROM. If no EEPROM is present, the register is loaded with 0x00.

minimum grant (Min_Gnt)—configuration space DWORD address (0x3C)

This register is loaded with a value from the EEPROM. If no EEPROM is present, the register is loaded with 0x01.

expansion ROM base address register (BASEROM)—configuration space DWORD address (0x30)

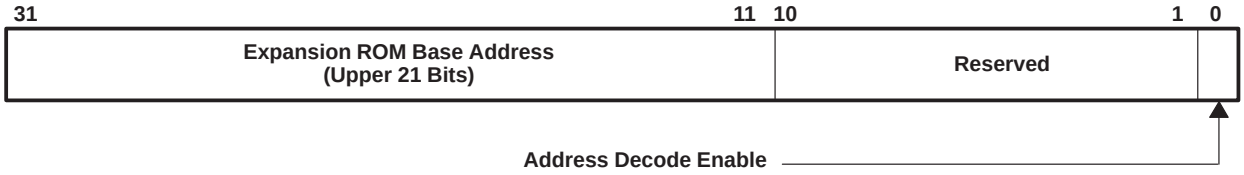


Figure 10. Expansion ROM Base Address Register Layout

The TI380PCI supports an RPL expansion ROM on a PCI bus add-in card. This register is used to configure the location of the expansion ROM. The BASEROM supports address alignment on a 64Kbyte boundary. Bits 16–31 and bit 0 of BASEROM are read/write, the rest are read only as 0x0. TI380PCI supports 64Kbyte ROM. Bit 0 is intended to be set by the host.

subsystem-specific registers

The TI380PCI contains the following registers in addition to the PCI-prescribed registers.

miscellaneous control register (MISCCTRL)—configuration space DWORD address (0x40)

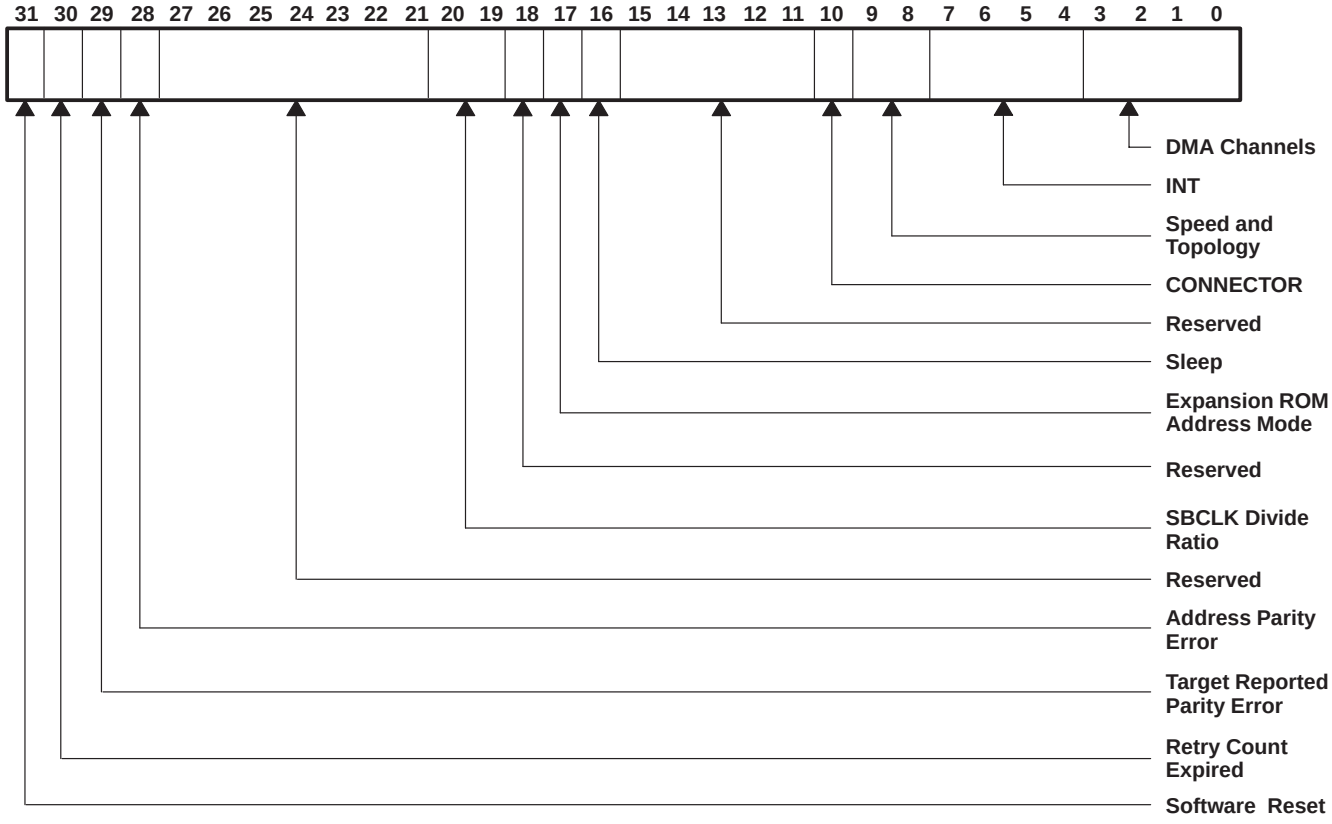


Figure 11. Miscellaneous Control Register

The MISCCTRL is a collection of LAN subsystem control functions. The value of the bit fields in this register after reset are as indicated.

Bits 0–3: DMA channels: DMA channel has no meaning in a PCI context. These four bits also appear in the TI2000 configuration register where they select between bus master DMA operation and pseudo-DMA operation as defined by the TI2000 software specification. When this field has the value zero, a TI2000 driver should operate in pseudo-DMA mode. When the field is nonzero, a TI2000 driver should operate in bus master DMA mode.

Bits 4–7: The four INT bits indicate the interrupt level. At reset, these bits are set to 0x0. They are read only and echo the least significant bits of the Interrupt Line Register.

miscellaneous control register (MISCCTRL)—configuration space DWORD address (0x40) (continued)

Bits 8–9: Speed and topology. These bits are echoed in bits 8–9 of TI2000 configuration registers. At reset, these bits are set to 0x1.

SPEED BIT 9	TOPOLOGY BIT 8	DESCRIPTION
0	0	Full-duplex Ethernet
1	0	Ethernet
0	1	Token Ring 16 Mbps
1	1	Token Ring 4 Mbps

Bit 10: The CONNECTOR bit indicates the connector in use: TI2000 drivers do not currently use this bit; however, it must always read as 0 if it is not implemented in adapter logic. The value of this bit is presented on pin DB9/$\overline{\text{UTP}}$ of the TI380PCI. Upon reset, this bit is 0.

1 = D-Shell (AUI or DB9)
0 = UTP/10BaseT

Bits 11–15: Reserved and is read as 0x9.

Bit 16: Sleep. Setting this bit to a 1 causes the SBCLK and output from the TI380PCI to be driven to a steady high state. SBCLK is used as SBCLK for the TI380C2x, holding it high causes the TI380C2x to suspend operations and go into power-saving mode. When this bit changes from a 1 to 0, the SBCLK returns to normal operation. This bit is set to 0 on reset.

Bit 17: Expansion ROM address mode, 14/$\overline{16}$. This bit indicates to the ROMIF that it should either use full 16-bit addressing or support the TI2000 paging protocol. When using TI2000 paging, the ROMIF uses the least significant 14 bits of the 16-bit ROM address and the most significant two bits are provided by an internal TI380PCI page register. This bit is set to 0x0 after reset, indicating full 16-bit mode.

Bit 18: This bit is reserved and is set to zero during reset by the TI380PCI. This bit is read as zero and should always remain set to zero.

Bits 19–20: SBCLK divide ratio. These bits specify the relationship between the PCI clock and the TI380C2x's SBCLK. After reset, these bits are set to 0x1.

BIT 20	BIT 19	SBCLK FREQUENCY
0	0	PCLK/1
0	1	PCLK/2
1	0	PCLK/3
1	1	PCLK/4

Bits 21–27: Reserved

Bit 28: Address Parity Error is set to 1 when the TI380PCI detects an address-parity error when acting as a PCI slave. This bit is set to 0 after reset.

Bit 29: Target Reported Parity Error is set to 1 when the TI380PCI receives a data parity error (that is, receives $\overline{\text{PERR}}$ during a master write and/or detected a parity error during master read). This bit is set to 0 after reset.

Bit 30: Retry Count Expired is set to 1 when the TI380PCI has exceeded the maximum retry count with a master transaction. This bit is set to 0 after reset.

miscellaneous control register (MISCCTRL)—configuration space DWORD address (0x40) (continued)

Bit 31: Software Reset is a programmable reset. Setting this bit to a 1 causes the $\overline{\text{SRESET}}$ output to pulse low for a minimum of 14 μs and results in a hard reset to the LAN subsystem. (This function is provided primarily for hardware and driver-software debug purposes.) This bit is set to 0 after reset. When set to 1, this bit resets itself to 0 after four PCI clock cycles.

EEPROM read/write register (0 × 48)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	ECR-CERR	NEP	ECLK	EEN	EDATA	rsvd	rsvd	rsvd	rsvd

Bit 8	ECRCERR	EEPROM CRC error detected. This bit is normally 0 but is set to a 1 if an error is detected in the CRC read from the EEPROM at power up. If this bit is set, the TI380PCI also sets bit 7, the NEP bit.
Bit 7	NEP	EEPROM not present. This bit indicates that the EEPROM interface was disabled either by a pulldown resistor on the EDIO pin or by the detection of an error in the checksum during the EEPROM download process.
Bit 6	ECLK	EEPROM SIO Clock: This bit controls the state of the EDC pin. 1 = pin high 0 = pin low
Bit 5	EEN	EEPROM enable: This bit controls the direction of the EDIO pin. When this bit is set to a 1, the EDIO pin is driven with a value in the EDATA bit.
Bit 4	EDATA	EEPROM data: This bit is used to read or write the EDIO data pin on the EEPROM. When bit 5 is a 1, the EEPROM data pin is driven with the value of this bit. If bit 5 is a 0, this bit is driven by the level on the EEPROM EDIO data pin.

380PCI interface control register—configuration space DWORD address (0x54)

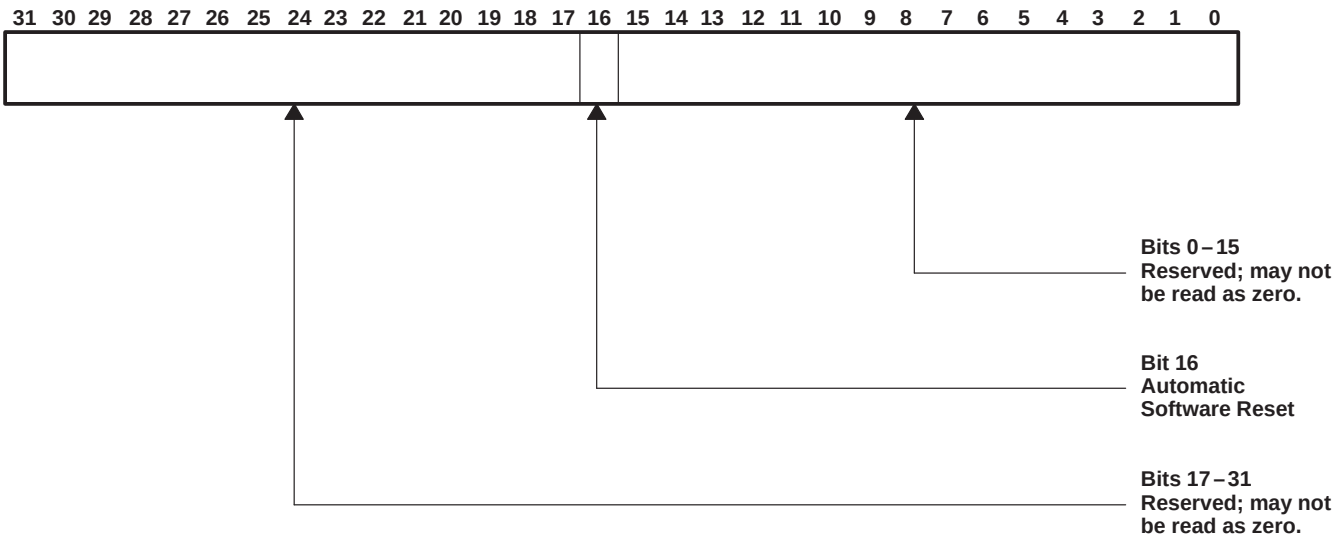


Figure 12. Interface Control Register

Bits 0 – 15: Reserved may not read as zero.

Bit 16: The automatic software reset controls the impact of writing a zero to the bus master bit (bit 2) in the TI380PCI command register 0x04. When this bit is set to zero, the TI380PCI resets itself if the host processor causes a transition from one to zero of the bus master bit. When bit 16 is set to one, the TI380PCI does not reset itself when the host processor causes a transition from one to zero of the master bus.

380PCI interface control register—configuration space DWORD address (0x54) (continued)

Typically, in personal computer (PC) systems the TI380PCI automatic software reset bit is set to a default value of zero. In a PC system, when a CTRL-ALT-DEL is issued from the keyboard, the PCI bus does not receive a reset. A TI380PCI bus master in the middle of a DMA transaction when the CTRL-ALT-DEL is issued can be left in an unknown state. In this condition, the bus master could perform unexpected memory accesses to the host after the TI380PCI device is enabled by the PCI BIOS during the boot process.

To eliminate this problem, the automatic software reset bit should be left set to the default value of zero. The action of clearing the bus master bit in the TI380PCI control register (which is a direct consequence of issuing the CTRL-ALT-DEL) also automatically resets the TI380PCI and clears any pending DMA transactions. The reset induced by clearing the TI380PCI bus master bit is the same as the software reset that takes place when bit 31 of the TI380PCI MISCCNTRL register is set to a one.

Bits 17–31: Reserved may not read as zero.

TI380C2x interface configuration register (TI2000)

The TI380PCI contains the following register which is defined in I/O space.

This is a READ ONLY register with the following functions:

CONFIG bit 15: The most significant bit of the configuration register, when set, indicates the existence of the configuration register. This bit is hardwired to 1, to indicate the presence of a configuration register.

CONFIG=1 There is a configuration register.
CONFIG=0 There is no configuration register.

RESERVED bits 13–14: These bits must read as 0.

DEDMACTL bit 12: The Don't Enable ISA DMA Controller (DEDMACTL) bit indicates whether the TI2000 driver must enable the ISA DMA controller. The ISA DMA controller is not used to control operations on the PCI bus. In order to allow the TI2000 software specification to be expanded to comprehend the PCI bus, this bit is hardwired to a 1 to indicate that a TI2000 driver controlling a PCI-based network interface card should not enable an ISA DMA controller. (The setting of this bit does not affect DMA on the PCI bus.)

DMA MODE	DEDMACTL BIT VALUE
DMA	1 = Driver must NOT enable ISA DMA controller
DMA	0 = Driver must enable ISA DMA controller
PDMA	x = Driver must NOT enable ISA DMA controller

INST8 bit 11: This bit indicates whether the driver can use 16-bit instructions to access the adapter registers. This bit is hardwired to 0 because PCI-based machines are able to perform 16-bit instructions.

1 = The TI2000 driver must use 8-bit instructions to access the LAN adapter registers.
0 = The TI2000 driver can use 16-bit instructions to access the LAN adapter registers.

TI380C2x interface configuration register (TI2000) (continued)

CONNECTOR bit 10: The CONNECTOR bit indicates the connector in use: TI2000 drivers do not currently use this bit; however, it must always read as 0 if it is not implemented in adapter logic. The value of this bit is presented on pin DB9/$\overline{\text{UTP}}$ of the TI380PCI. Upon reset, this bit is 0. This bit echoes bit 10 of the MISCCNTL register.

1 = D-Shell (AUI or DB9)
0 = UTP/10BaseT

SPEED bit 9: Indicates whether the driver must initialize 4- or 16-Mbps token-ring operation. See TOPOLOGY bit 8.

TOPOLOGY bit 8: Indicates whether the driver must initialize token-ring or Ethernet™ operation.

TI2000 drivers interpret the speed and topology bits together to determine whether to initialize the adapter under 4- or 16-Mbps token-ring, or 10-Mbps Ethernet™ operation:

SPEED BIT 9	TOPOLOGY BIT 8	DESCRIPTION
0	0	Reserved
1	0	Ethernet
0	1	Token Ring, 16 Mbps
1	1	Token Ring, 4 Mbps

These two bits echo bits 8 and 9 in the MISCCNTL Register.

INT bits 4–7: The four INT bits indicate the interrupt level. At reset these bits are set to 0x0. These bits echo the LSBs of the Interrupt Line Register.

7	6	5	4	INTERRUPT LEVEL
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
.	.	.	.	...
1	1	1	1	15

DMA bits 0–3: The four DMA bits indicate the DMA channel. At reset, these bits are 0x0. They are used by TI2000-compatible software, and the channel number has no meaning in terms of PCI bus DMA. However, TI2000 drivers read this field, and if it is set to 0, they operate in pseudo-DMA mode. Any other value allows TI2000 drivers to use PCI DMA operation.

3	2	1	0	DMA CHANNEL
0	0	0	0	DMA bus master
0	0	0	1	DMA bus master
0	0	1	0	DMA bus master
.	.	.	.	DMA bus master
1	1	1	1	Pseudo DMA operation

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage range, V_{DD} (see Note 2) . . . . . 7 V
Input voltage range (see Note 2) . . . . . – 0.3 V to 20 V
Output voltage range . . . . . – 2 V to 7 V
Power dissipation . . . . . 1 W
Operating free-air temperature range, T_A . . . . . 0°C to 70°C
Storage temperature range, T_{stg} . . . . . – 65°C to 150°C

† Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 2: Voltage values are with respect to V_{SS}.

recommended operating conditions

		MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage	4.75	5.0	5.25	V
V_{IH}	High-level DC input voltage (TTL)‡	2.0		V_{CC}	V
V_{IL}	Low-level DC input voltage (TTL)‡	0		0.8	V
t_t	Input transitions (t_r and t_f, 10% to 90%)	0		25	ns
V_O	Output voltage§	0		V_{CC}	V
T_A	Operating free-air temperature range	0		70	°C

‡ Applies for external input buffers without hysteresis
§ Applies for external output buffers

electrical characteristics over recommended range of supply voltage and operating free-air temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS (see Note 3)	MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage, TTL-level signal (see Note 4)	V_{DD} = MIN, I_{OH} = MAX	2.4			V
V_{OL}	Low-level output voltage, TTL-level signal	V_{DD} = MIN, I_{OL} = MAX			0.6	V
I_O	High-impedance output current	V_{DD} = MAX, V_O = 2.4 V V_{DD} = MAX, V_O = 0.4 V			20 –20	µA
I_I	Input current, any input or input/output pin	$V_I = V_{SS}$ to V_{DD}			± 20	µA
I_{DD}	Supply current	V_{DD} = MAX			TBD	mA
C_i	Input capacitance, any input	f = 1 MHz, other pins at 0 V			15	pF
C_o	Output capacitance, any output or input /output	f = 1 MHz, other pins at 0 V			15	pF

NOTES: 3. For conditions shown as MIN or MAX, use the appropriate value specified under the recommended operating conditions.
4. The following signals require an external pullup resistor: SRAS/$\overline{\text{SAS}}$, $\overline{\text{SRDY}}$/$\overline{\text{SDTACK}}$, $\overline{\text{SRD}}$/$\overline{\text{SUDS}}$, $\overline{\text{SWR}}$/$\overline{\text{SLDS}}$.

ADVANCE INFORMATION

PARAMETER MEASUREMENT INFORMATION

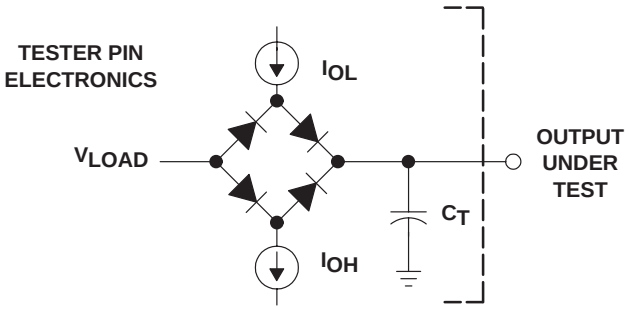
Outputs are driven to a minimum high-logic level of 2.4 volts and to a maximum low-logic level of 0.6 volts. These levels are compatible with TTL devices.

Output transition times are specified as follows: For a high-to-low transition on either an input or output signal, the level at which the signal is said to be no longer high is 2 volts, and the level at which the signal is said to be low is 0.8 volts. For a low-to-high transition, the level at which the signal is said to be no longer low is 0.8 volts, and the level at which the signal is said to be high is 2 volts, as shown below.

The rise and fall times are not specified but are assumed to be those of standard TTL devices, which are typically 1.5 ns.

test measurement

The test load circuit shown in Figure 13 represents the programmable load of the tester pin electronics used to verify timing parameters of TI380PCI output signals.



Where: I_{OL} = 2.0 mA DC level verification (all outputs)
I_{OH} = 400 μA (all outputs)
V_{LOAD} = 1.5 V typical DC level verification
0.7 V typical timing verification
C_T = 65 pF typical load circuit capacitance and includes probe and jig capacitance.

Figure 13. Test Load Circuit

ADVANCE INFORMATION

PCI INTERFACE TIMING

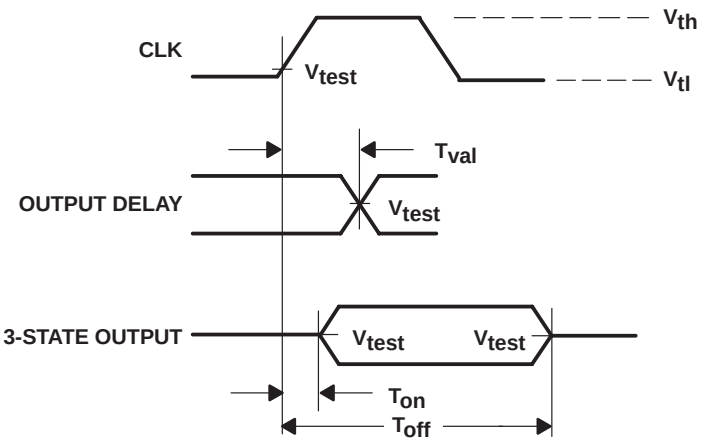


Figure 14. Output Timing Measurement Conditions

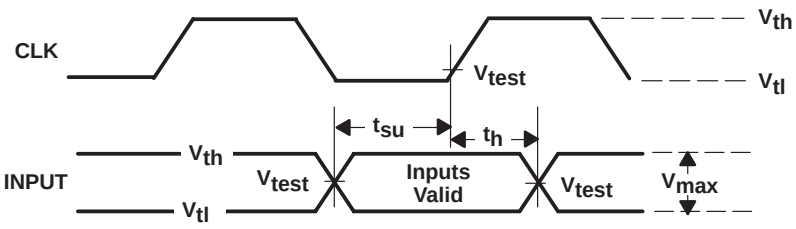


Figure 15. Input Timing Measurement Conditions

measure and test condition parameters

SYMBOL	5-V SIGNALING	UNITS
V_{th}	2.4	V
V_{tl}	0.4	V
V_{test}	1.5	V
V_{max}	2.0	V
Input signal edge rate	1 V / ns	

ADVANCE INFORMATION

timing parameters (5-V signaling environment)

PARAMETER		MIN	MAX	UNIT
t_{val}	CLK to signal valid delay-bussed signals (see Notes 5, 6, 7)	2	11	ns
$t_{val(ptp)}$	CLK to signal valid delay-point-to-point (see Notes 5, 6, 7)	2	12	ns
t_{on}	Float to active delay (see Note 5)	2		ns
t_{off}	Active to float delay (see Note 5)		28	ns
t_{su}	Input setup time to CLK-bussed signals (see Notes 7, 8)	7		ns
$t_{su(ptp)}$	Input setup time to CLK-point-to-point (see Notes 7, 8)	10, 12		ns
t_h	Input hold time from CLK (see Note 8)	0		ns
t_{rst}	Reset active time after power stable (see Note 9)	1		ms
$t_{rst-clk}$	Reset active time after CLK stable (see Note 8)	100		µs
$t_{rst-off}$	Reset active to output float delay (see Notes 8, 10)		40	ns

NOTES: 5. See timing measurement conditions in the Output Timing Measurement Conditions diagram of the PCI Specification 2.0.
6. Minimum times are measured with 0 pF equivalent load; maximum times are measured with 50 pF equivalent load. Actual test capacitance can vary, but results should be correlated to these specifications.
7. $\overline{REQ}$ and $\overline{GNT}$ are point-to-point signals, and have different output valid delay and input setup times than do bussed signals. $\overline{GNT}$ has a setup of 10; $\overline{REQ}$ has a setup of 12. All other signals are bussed.
8. See timing measurement conditions in the Input Timing Measurement Conditions diagram of the PCI Specification 2.0.
9. $\overline{RST}$ is asserted and deasserted asynchronously with respect to CLK. Refer to PCI Specifications 2.0 for more information.
10. All output drivers must be floated when $\overline{RST}$ is active.

ADVANCE INFORMATION

Serial EEPROM Interface

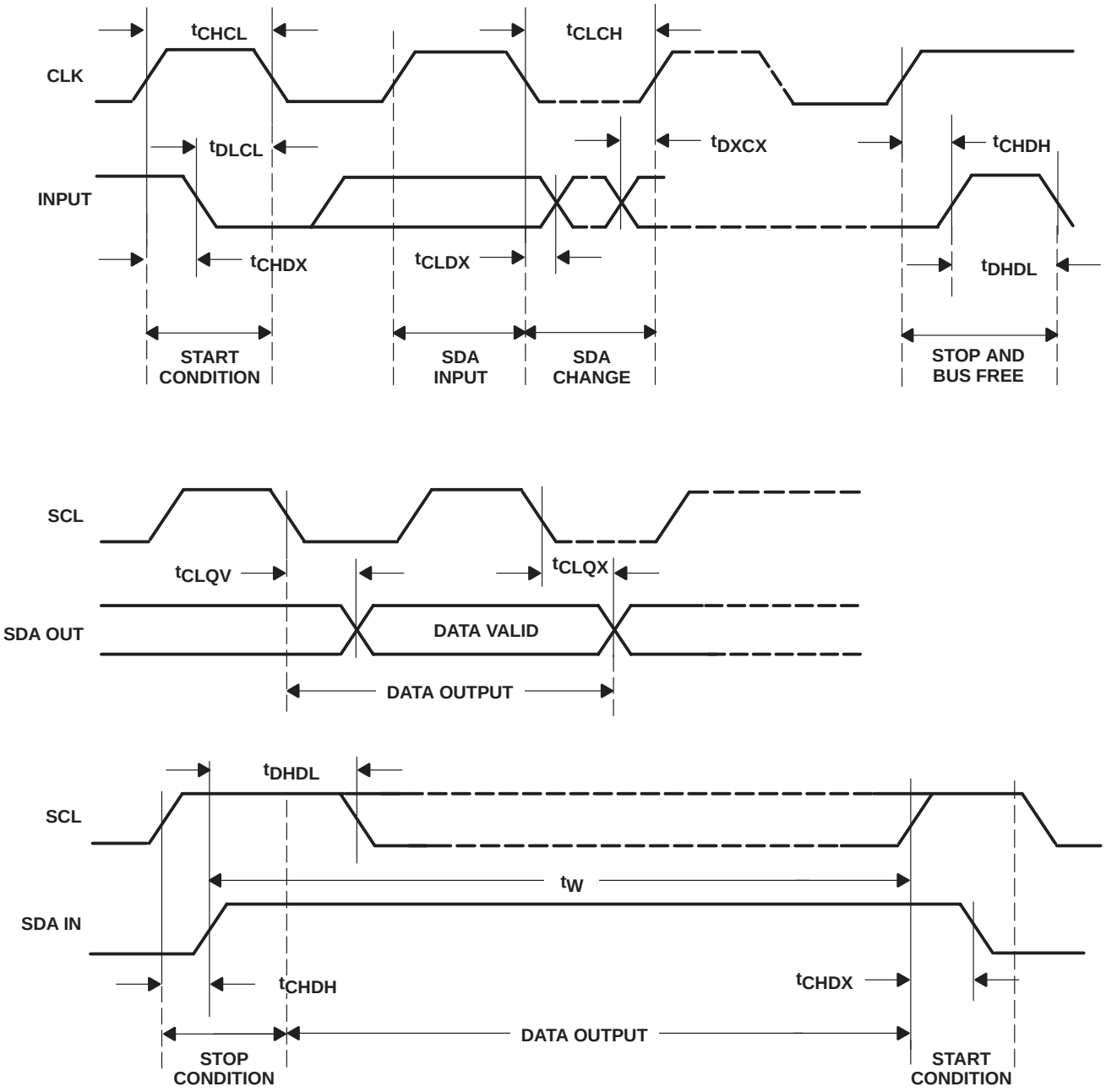


Figure 16. AC Waveforms

ADVANCE INFORMATION

AC characteristics

PARAMETER		ALT	MIN	MAX	UNIT
t_{CH1CH2}	Rise time, clock	t_R		1	µs
t_{CL1CL2}	Fall time, clock	t_F		300	ns
t_{DH1DH2}	Rise time, input	t_R		1	µs
t_{DL1DL1}	Fall time, input	t_F		300	ns
t_{CHDX}	Setup time, clock high to input transition (see Note 11)	$t_{SU:STA}$	4.7		µs
t_{CHCL}	Pulse duration, clock high	t_{-HIGH}	4		µs
t_{DLCL}	Input low to clock low (START)	$t_{HD:STA}$	4		µs
t_{CLDX}	Clock low to input transition	$t_{HD:DAT}$	0		µs
t_{CLCH}	Pulse duration, clock low	t_{LOW}	4.7		µs
t_{DXCX}	Input transition to clock transition	$t_{SU:DAT}$	250		ns
t_{CHDH}	Clock high to input high (STOP)	$t_{SU:STO}$	4.7		µs
t_{DHDL}	Input high to input low (bus free)	t_{BUF}	4.7		µs
t_{CLQV}	Clock low to output valid	t_{AA}	0.3	3.5	µs
t_{CLQX}	Clock high to output transition	t_{DH}	300		ns
f_C	Clock frequency	f_{SCL}		100	KHz
t_{LPF}	Input low pass first order filter time constant (SCL and SDA inputs)	T_I		100	ns
t_W	Write time (see Note 12)	t_{WR}		10	ms

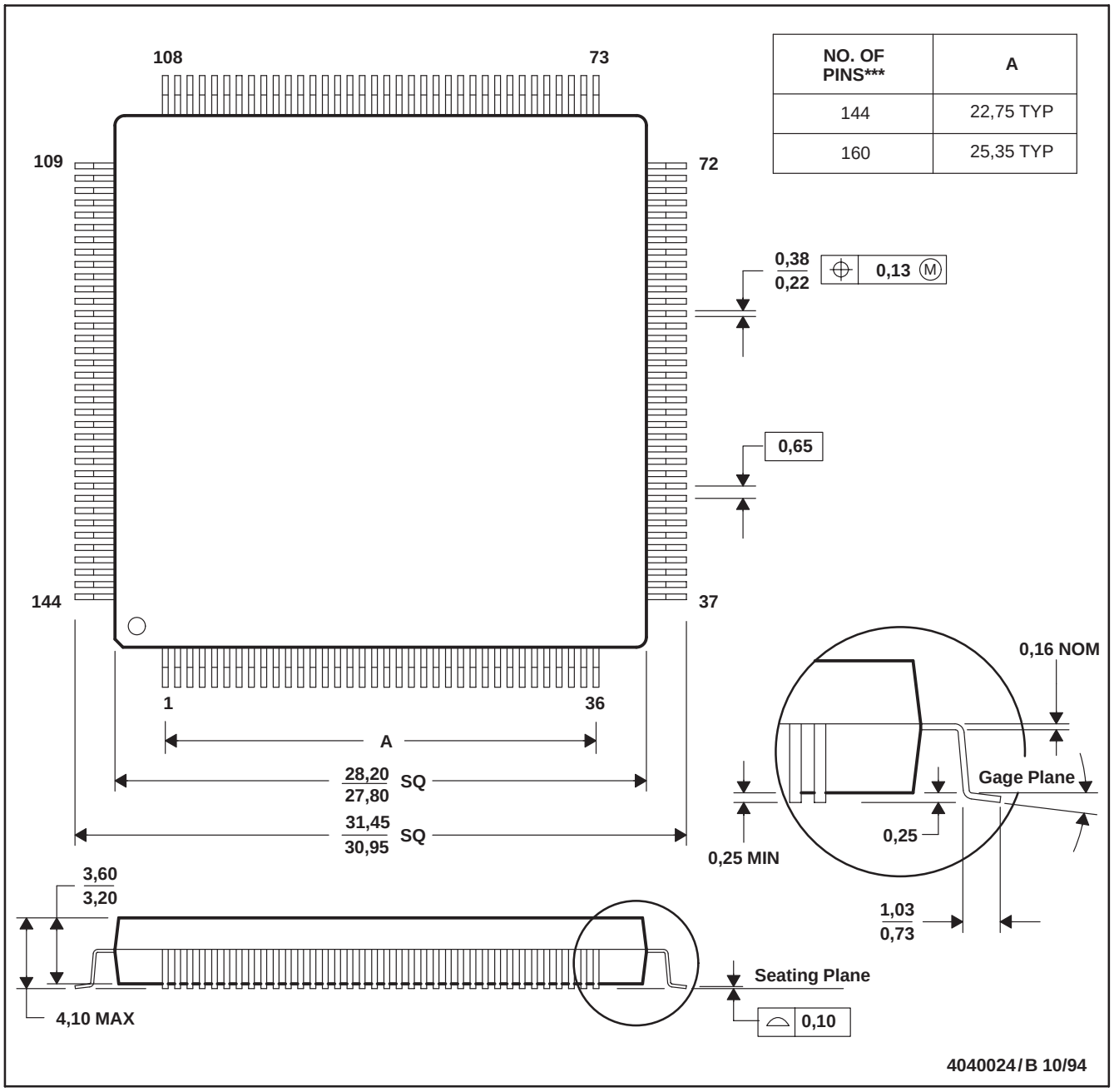
NOTES: 11. For a reSTART condition, or following a write cycle
12. In the multibyte write mode only, if accessed bytes are on two consecutive rows (upper 5 MSB must not change), the maximum programming time is doubled to 20 ms.

ADVANCE INFORMATION

MECHANICAL DATA

PCM (S-PQFP-G*)** **PLASTIC QUAD FLATPACK**

144 PIN SHOWN



NO. OF PINS***	A
144	22,75 TYP
160	25,35 TYP

NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Falls within JEDEC MS-022
D. The 144 PCM is identical to the 160 PCM except that four leads per corner are removed.

ADVANCE INFORMATION

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgement, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its semiconductor products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

CERTAIN APPLICATIONS USING SEMICONDUCTOR PRODUCTS MAY INVOLVE POTENTIAL RISKS OF DEATH, PERSONAL INJURY, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE ("CRITICAL APPLICATIONS"). TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF TI PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE FULLY AT THE CUSTOMER'S RISK.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.