OKI Semiconductor

Z684 PCI Bus Controller

0.5μm Technology Mega Macrocell

DESCRIPTION

The PCI Bus Controller Mega Macrocell is a featured element in OKI's 0.5μm Sea of Gates (SOG) and Customer Structured Array (CSA) families. Designers can significantly reduce design and simulation effort by using OKI's Z684 on PCI bus interface projects. The Z684 is fully compliant with the PCI Bus Revision 2.0 specification.

OKI's Z684 provides a PCI interface, data FIFO and control, and a configuration block in a highly integrated module for system design interfaces that implement the PCI bus protocol. The Z684 connects between an external PCI bus and a peripheral device's internal 486-like bus. Both buses are 32 bits wide and operate at clock frequencies of up to 33 MHz. Two 4x36-bit write FIFOs enhance system performance. The cell also supports address and data parity generation and error reporting. PCI-compliant I/O buffers are available to connect the Z684 to a PCI bus, and are added when the design is implemented in an SOG array or CSA. By using the Z684, the users may speed up the design cycle and accelerate market entry for the end product.

FEATURES

- Highly integrated interfaces between customer's application specific design module and PCI bus
- Compliant with PCI Bus Revision 2.0 specification
- Internal, i486-like 32-bit address and data bus interface, with byte-select-enable control
- Two 4x36-bit address, data and byte-enable write FIFOs
- 32-bit PCI address and data bus interface with byte-select-enable control
- 3-V operation
- Built-in internal 64-byte PCI configuration registers, which can be accessed both from PCI and module buses
- Built-in memory and I/O address decoding
- Burst write mode operation on both PCI and module interfaces
- Full support for PCI master and slave functions
- Full address parity generation, data parity generation and error checking and correction (ECC)
- Synchronous operation at up to 33 MHz for both the PCI and module interfaces

Supported ASIC Families

Family Name	Family Type
MSM10R0000	Sea of Gates
MSM13R0000	Sea of Gates
MSM98R000	Customer Structured Array

Recommended Operating Conditions (V_{SS} = 0V)

Parameter	Symbol	Rated Values			Unit
		Min.	Typ.	Max.	
Power supply voltage	V_{DD}	2.7	3.3	3.6	V
Operating temperature	T_j	-40	+25	+85	°C

Mega Macrocell Characteristics

Mega Macrocell	Description	Logic Gate Count	Number of Mega Macrocell Pins
Z684	PCI Bus Controller	21,600	269

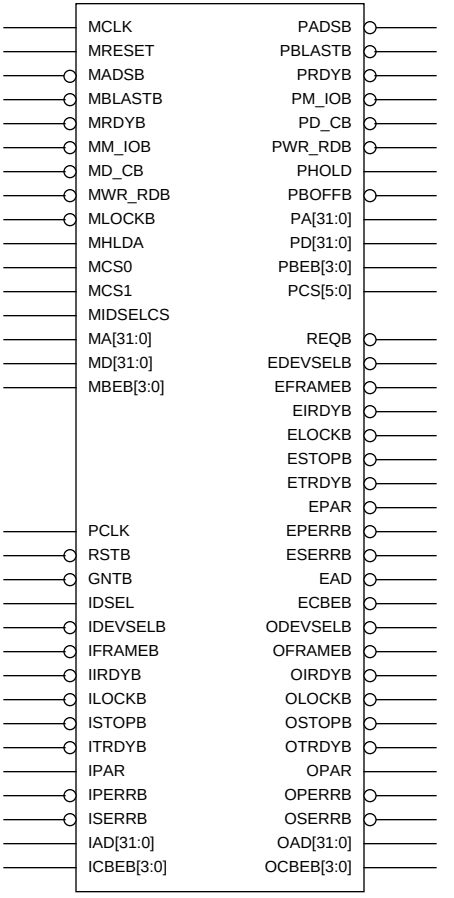

Figure 203. Logic Symbol

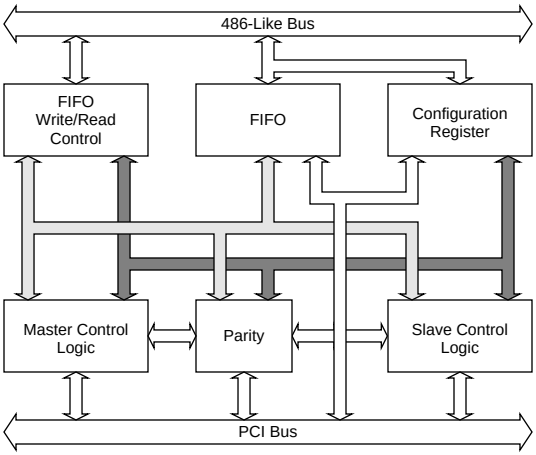

Figure 204. PCI Mega Macrocell Block Diagram

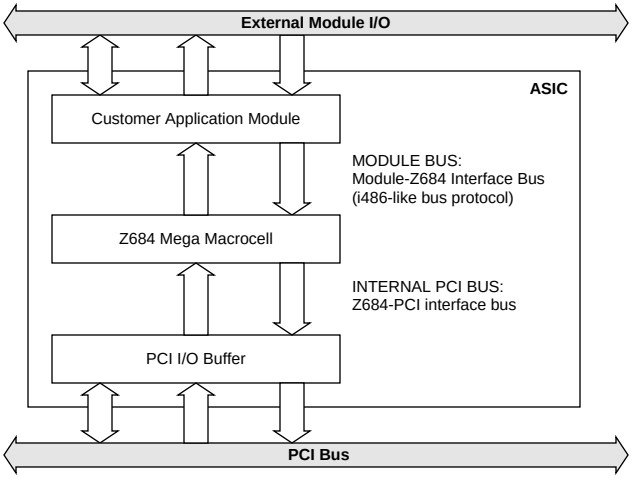

Figure 205. PCI Mega Macrocell Application

SIGNAL DESCRIPTIONS

The Z684 signals can be classified into two main groups:

- Module Interface Signals
- PCI Interface Signals

All Z684 signals are unidirectional and unbuffered.

Module Interface Signals

Module Interface Signals interface the Z684 with the customer's application module. Module Interface Signals are either Z684 inputs or Z684 outputs.

- Input signals (driven from the module to the Z684) start with an "M" prefix
- Output signals (driven from the Z684 to the module) start with a "P" prefix
- Low assertion-signals end with "B" suffix

The Module Interface Signals are derived from 486-like external bus signals and have the same function as equivalent i486 signals.

Module Interface Signals

Signal	Type	Assertion	Description
MCLK	Input	–	**Module Clock.** The MCLK signal synchronizes all bus cycles and control signals for module-Z684 transactions. The current version of the Z684 requires that both MCLK and PCLK are connected to the same clock driver.
MRESET	Input	HIGH	**Module Reset.** When asserted HIGH, MRESET resets all components related with Z684-PCI interface control logic.
MADSB	Input	LOW	**Module Address Strobe**. When asserted LOW, MADSB indicates that the address on the MA[31:0] address bus is valid. This signal also indicates the beginning of a bus cycle.
MBLASTB	Input	LOW	**Module Burst Last**. When asserted LOW, MBLASTB indicates that the transaction cycle has entered the final data phase.
MRDYB	Input	LOW	**Module Data Ready**. This signal indicates that the module has received valid data from the Z684 data bus, PD[31:0].
MM_IOB	Input	–	**Module Memory or I/O Select**. When asserted HIGH, memory access is selected. When asserted LOW, I/O access is selected.
MD_CB	Input	–	**Module Data or Code Select**. When asserted HIGH, the bus performs a data access cycle. When asserted LOW, the bus performs a code access cycle.
MWR_RDB	Input	–	**Module Write or Read.** This signal defines a write or read bus cycle. When asserted HIGH, the bus performs a write cycle. When asserted LOW, the bus performs a read cycle.
MLOCKB	Input	LOW	**Module Lock Cycle Request**. When asserted LOW, the module performs a locked transaction. This signal is not supported in the current version of the Z684 and must be tied inactive (HIGH).
MHLDA	Input	HIGH	**Module Bus Hold Acknowledge**. When asserted HIGH, this signal indicates that the customer application module is available to receive a transaction from the Z684. This signal is equivalent to the i486 bus signal HLDA.
MCS0	Input	HIGH	**Module Chip Select 0**. This signal indicates an access cycle to the internal configuration registers. When asserted HIGH, the Z684's internal configuration registers are selected.
MCS1	Input	HIGH	**Module Chip Select 1**. This signal indicates a PCI bus transaction. When asserted HIGH, this signal selects external PCI memory or I/O, or PCI configuration register space.

Module Interface Signals (Continued)

Signal	Type	Assertion	Description
MIDSELCS	Input	HIGH	**PCI Configuration Cycle Select**. This signal indicates that the module intends to access configuration registers of a PCI device.
MA[31:0]	Input	–	**Module Address Input Bus**. The MA[31:0] signals are the 32-bit input address bus from the module to the Z684. The address is valid when MADSB is asserted LOW.
MD[31:0]	Input	–	**Module Data Input Bus**. The MD[31:0] signals are the 32-bit input data bus from the module to the Z684.
MBEB[3:0]	Input	–	**Module Byte Enable**. The MBEB[3:0] signals select the byte or bytes on the MD[31:0] data bus. The byte is valid when corresponding MBEB[3:0] bit is asserted LOW.
PADSB	Output	LOW	**Module Address Strobe Output**. When asserted LOW, the PADSB output signals that the address on the PA[31:0] address bus is valid. The PADSB output also signals the beginning of a bus transaction.
PBLASTB	Output	LOW	**Module Burst Last Output**. When asserted LOW, this signal indicates that the transaction has entered the last data phase.
PRDYB	Output	LOW	**Module Data Ready Output**. This signal indicates that the Z684 has received valid data from the module data bus, MD[31:0].
PM_IOB	Output	–	**Module Memory or I/O Select Output**. When asserted HIGH, memory access is selected. When asserted LOW, I/O access is selected.
PD_CB	Output	–	**Module Data or Code Select Output**. When the Z684 asserts PD_CB HIGH, the bus performs a data access cycle. When the Z684 asserts PD_CB LOW, the bus performs a code access cycle.
PWR_RDB	Output	–	**Module Write or Read Output**. This signal defines whether a write or read bus cycle takes place. When the Z684 asserts PWR_RDB HIGH, the bus performs a WRITE cycle. When the Z684 asserts PWR_RDB LOW, the bus performs a read cycle.
PHOLD	Output	HIGH	**Module Hold Request Output**. When the Z684 intends to start a transaction targeting the module, the Z684 asserts PHOLD HIGH. The PHOLD signal is equivalent to the HOLD signal on the i486 bus.
PBOFFB	Output	LOW	**Module Back-Off Output.** When asserted LOW, this signal indicates that the Z684 is requesting the module to stop the current transaction.
PA[31:0]	Output	–	**Module Address Output**. This is the address bus output, driven from the Z684 to the module.
PD[31:0]	Output	–	**Module Data Output**. The PD[31:0] signals are the 32-bit output data bus from the module to the Z684.
PBEB[3:0]	Output	–	**Module Byte Enable Output.** These signals select the valid bytes on the PD[31:0] data bus. The byte is valid when the corresponding PBEB[3:0] is asserted LOW.
PCS[5:0]	Output	–	**PCI-to-Module** Chip Select. These outputs indicate that one of the address spaces in the base address registers has been selected.

PCI Interface Signals

The PCI interface signals are signals between the Z684 and the PCI bus. The PCI interface signals are unidirectional versions of normal PCI signals.

- Input signals (driven from the PCI bus to the Z684) start with an “I” prefix.
- Output signals (driven from the Z684 to the module) start with an “O” prefix.
- Enable signals start with an “E” prefix.
- All low assertion signals end with a “B” suffix.

PCI Interface Signals

Signal	Type	Assertion	Description
PCLK	Input	–	**PCI Clock.** The PCLK signal synchronizes all bus cycles and control signals for Z684-PCI transactions. The current version of the Z684 requires that both MCLK and PCLK are connected to the same clock driver.
RSTB	Input	LOW	**PCI Reset.** When asserted to LOW, RSTB resets all the Z684-PCI interface control logic.
REQB	Output	LOW	**PCI Bus Request**. Whenever a direct transaction to the PCI bus is pending, the Z684 asserts REQB LOW, signaling a PCI bus access request to the PCI bus arbiter.
GNTB	Input	LOW	**PCI Bus Grant**. The PCI bus arbiter asserts GNTB LOW to signal to the Z684 that the PCI bus is available for access.
IDSEL	Input	HIGH	**IDSEL PCI Configuration Access**. The IDSEL input indicates that a configuration register read or write cycle is pending.
IDEVSELB	Input	LOW	**Device Select.** The PCI target asserts IDEVSELB low to signal to the Z684 that it has been selected for access. The IDEVSELB pin is the input signal for the PCI DEVSEL# pin.
IFRAMEB	Input	LOW	**Frame Input.** When driven low by an initiator, the IFRAMEB signal indicates the beginning and duration of a transaction cycle. The initiator deasserts IFRAMEB in the clock cycle before the last data phase of the transaction cycle. The IFRAMEB pin is the input signal for the PCI FRAME# pin.
IIRDYB	Input	LOW	**Initiator Data Ready.** When the initiator drives the IIRDYB signal low, this signal indicates the initiator's ability to complete the current data phase. This signal is the input signal for the PCI IRDY# pin.
ILOCKB	Input	LOW	**ILOCKB Input.** This signal indicates that an atomic operation may require multiple transactions to complete. The ILOCKB signal is the input pin for the PCI LOCK# signal.The ILOCKB signal is used only when system memory is supported. This pin is not supported in the current Z684 version and must be tied inactive (HIGH).
ISTOPB	Input	LOW	**Stop or Retry Input**. The ISTOPB signal indicates that the current target is requesting the initiator to stop the current transaction. The ISTOPB signal is the input pin for the PCI STOP# signal.
ITRDYB	Input	LOW	**ITRDYB Input**. This signal indicates that the target device is able to complete the current data phase of the transaction. The ITRDYB signal is the input signal for the PCI TRDY# pin.
IPAR	Input	–	**Parity**. This signal provides even parity across the PCI address/data and byte enable buses. The IPAR signal is the input pin for the PCI PAR signal.
IPERRB	Input	LOW	**Parity Error**. This is the parity error reporting signal. The IPERRB signal reports data parity errors during all PCI transactions except special cycles. The IPERRB signal is the input pin for the PCI PERR# signal.
ISERRB	Input	LOW	**System Error**. This is the system error signal. This input reports address and special cycle data parity errors. The ISERRB signal is the input pin for the PCI SERR# signal.
IAD[31:0]	Input	–	**AD Address and Data Bus**. These are the PCI multiplexed address and data bus inputs.

PCI Interface Signals (Continued)

<table>
<tr><th>Signal</th><th>Type</th><th>Assertion</th><th>Description</th></tr>
<tr><td>ICBEB[3:0]</td><td>Input</td><td>–</td><td>Command and Byte Enable Bus. These are the PCI multiplexed command and byte enable bus inputs with the PCI command definitions shown below.
<table>
<tr><th>ICBEB[3:0]</th><th>Command Type</th><th>ICBEB[3:0]</th><th>Command Type</th></tr>
<tr><td>0000</td><td>Interrupt Acknowledge</td><td>1000</td><td>Reserved</td></tr>
<tr><td>0001</td><td>Special Cycle</td><td>1001</td><td>Reserved</td></tr>
<tr><td>0010</td><td>I/O Read</td><td>1010</td><td>Configuration Read</td></tr>
<tr><td>0011</td><td>I/O Write</td><td>1011</td><td>Configuration Write</td></tr>
<tr><td>0100</td><td>Reserved</td><td>1100</td><td>Memory Read Multiple</td></tr>
<tr><td>0101</td><td>Reserved</td><td>1101</td><td>Dual Access Cycle</td></tr>
<tr><td>0110</td><td>Memory Read</td><td>1110</td><td>Memory Read Line</td></tr>
<tr><td>0111</td><td>Memory Write</td><td>1111</td><td>Memory Write and Invalidate</td></tr>
</table></td></tr>
<tr><td>EDEVSELB</td><td>Output</td><td>LOW</td><td>Device Select Enable. The EDEVSELB signal is the enable output for the PCI DEVSEL# signal.</td></tr>
<tr><td>EFRAMEB</td><td>Output</td><td>LOW</td><td>Frame Enable. The EFRAMEB signal is the enable output for the PCI FRAME# pin.</td></tr>
<tr><td>EIRDYB</td><td>Output</td><td>LOW</td><td>Initiator Data Ready Enable. The EIRDYB signal is the enable output for the PCI IRDY# pin.</td></tr>
<tr><td>ELOCKB</td><td>Output.</td><td>LOW</td><td>Lock Enable. The ELOCKB signal is the enable output for the PCI LOCK# pin.</td></tr>
<tr><td>ESTOPB</td><td>Output</td><td>LOW</td><td>Transaction Stop/Retry Enable. The ESTOPB signal is the enable output for the PCI STOP# pin.</td></tr>
<tr><td>ETRDYB</td><td>Output</td><td>LOW</td><td>Target Data Ready Enable. The ETRDYB signal is the enable output for the PCI TRDY# pin.</td></tr>
<tr><td>EPAR</td><td>Output</td><td>LOW</td><td>Parity Enable. The EPAR signal is the enable output for the PCI PAR pin.</td></tr>
<tr><td>EPERRB</td><td>Output</td><td>LOW</td><td>Parity Error Enable. The EPERRB signal is the enable output for the PCI PERR# pin.</td></tr>
<tr><td>ESERRB</td><td>Output</td><td>LOW</td><td>System Error Enable. The ESERRB signal is the enable output for the PCI SERR# pin.</td></tr>
<tr><td>EAD</td><td>Output</td><td>LOW</td><td>Address and Data Bus Enable. The EAD signal is the enable output for the PCI AD[31:0] pins.</td></tr>
<tr><td>ECBEB</td><td>Output</td><td>LOW</td><td>Command and Byte Enable Bus Enable. The ECBEB is the enable output for the PCI CBE#[3:0] pins.</td></tr>
<tr><td>ODEVSELB</td><td>Output</td><td>LOW</td><td>Device Select Output. The ODEVSELB signal indicates that the Z684 has been selected as a target PCI bus device for access. The ODEVSELB signal is the output pin for the PCI DEVSEL# signal.</td></tr>
<tr><td>OFRAMEB</td><td>Output</td><td>LOW</td><td>Frame Output. The OFRAMEB signal indicates the beginning and duration of a transaction. The OFRAMEB signal is deasserted before the last data phase of the current transaction. The OFRAMEB signal is the output pin for the PCI FRAME# signal.</td></tr>
<tr><td>OIRDYB</td><td>Output</td><td>LOW</td><td>Initiator Data Ready. When the Z684 drives the OIRDYB output low, the OIRDYB signal indicates the initiator's ability to complete the current data phase. The OIRDYB signal is the output pin for the PCI IRDY# signal.</td></tr>
<tr><td>OLOCKB</td><td>Output</td><td>LOW</td><td>LOCKb Output. The OLOCKB signal indicates an atomic operation that may require multiple transactions to complete. The OLOCKB signal is the output signal for the PCI LOCK# pin and is driven by the Z684. The OLOCKB signal is only used when system memory is supported. This pin is not supported in the current Z684 version and must be tied inactive (HIGH).</td></tr>
<tr><td>OSTOPB</td><td>Output</td><td>LOW</td><td>Stop/Retry Output. The OSTOPB indicates that the Z684 is requesting the initiator to stop the current transaction. The OSTOPB signal is the output for the PCI STOP# pin.</td></tr>
<tr><td>OTRDYB</td><td>Output</td><td>LOW</td><td>TRDYb Output. This signal indicates the target device's ability to complete the current data phase of the transaction. The OTRDYB pin is the output signal for the PCI TRDY# pin.</td></tr>
<tr><td>OPAR</td><td>Output</td><td>–</td><td>Parity. The OPAR signal is an even parity output across the PCI address/data and byte enable buses. The OPAR pin is the output signal for the PCI PAR pin.</td></tr>
</table>

PCI Interface Signals (Continued)

<table>
<tr><th>Signal</th><th>Type</th><th>Assertion</th><th>Description</th></tr>
<tr><td>OPERRB</td><td>Output</td><td>LOW</td><td>Parity Error. This is the parity error reporting signal. The OPERRB signal reports data parity errors during all PCI transactions except special cycles. The OPERRB signal is the output signal for the PCI PERR# pin.</td></tr>
<tr><td>OSERRB</td><td>Output</td><td>LOW</td><td>System Error. This is the system error signal. The OSERRB signal reports address and special cycle data parity errors. The OSERRB pin is the output signal for the PCI SERR# pin.</td></tr>
<tr><td>OAD[31:0]</td><td>Output</td><td>–</td><td>AD Address and Data Bus. These are PCI multiplexed address and data bus outputs.</td></tr>
<tr><td>OCBEB[3:0]</td><td>Output</td><td>–</td><td>Command and Byte Enable Bus. These are PCI multiplexed command and byte enable bus outputs with the PCI command definitions shown below.
<table>
<tr><th>OCBEB[3:0]</th><th>Command Type</th><th>OCBEB[3:0]</th><th>Command Type</th></tr>
<tr><td>0000</td><td>Interrupt Acknowledge</td><td>1000</td><td>Reserved</td></tr>
<tr><td>0001</td><td>Special Cycle</td><td>1001</td><td>Reserved</td></tr>
<tr><td>0010</td><td>I/O Read</td><td>1010</td><td>Configuration Read</td></tr>
<tr><td>0011</td><td>I/O Write</td><td>1011</td><td>Configuration Write</td></tr>
<tr><td>0100</td><td>Reserved</td><td>1100</td><td>Memory Read Multiple</td></tr>
<tr><td>0101</td><td>Reserved</td><td>1101</td><td>Dual Access Cycle</td></tr>
<tr><td>0110</td><td>Memory Read</td><td>1110</td><td>Memory Read Line</td></tr>
<tr><td>0111</td><td>Memory Write</td><td>1111</td><td>Memory Write and Invalidate</td></tr>
</table>
</td></tr>
</table>

FUNCTIONAL DESCRIPTION

First introduced in 1992, the PCI bus offers high bandwidth and operates independently of the host CPU. As a result, the PCI bus has become widely accepted in the PC industry.

Figure 206 shows one of the most common configurations of a PCI bus system.

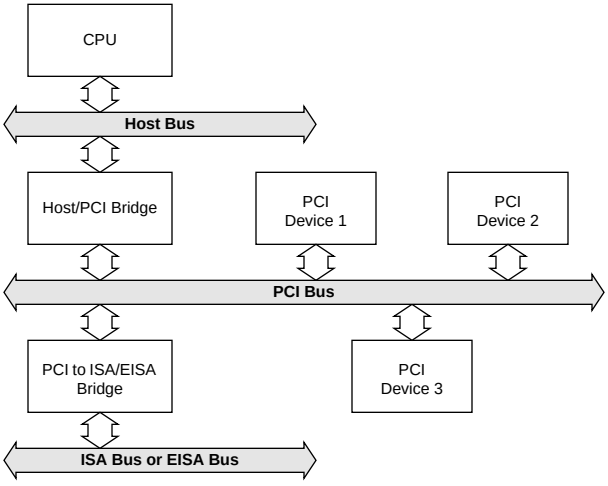

Figure 206. PCI System Configuration

In the example above, there are three bus protocols: the CPU host bus, the PCI bus, and the ISA / EISA bus protocols. The system uses a bus bridge to provide a communication between the buses. The bus bridge converts one bus protocol to another. The host / PCI bridge provides protocol linkage between a CPU host and the PCI bus. The PCI and ISA / EISA bus linkage is provided by the PCI-to-ISA / EISA bridge. With this system configuration, high-bandwidth PCI modules (either add-on cards or IC devices) can be connected to the PCI bus. The reader is recommended to consult the latest PCI specification for more detail.

The OKI Z684 provides an interface between the customer's application module and the PCI bus. The Z684 can be integrated into a customer's design and allows the integrated device to connect directly to the PCI bus. Alternatively, the Z684 can be used in a chip on a PCI daughtercard. *Figure 205* on page 3 shows one of the most common applications of the Z684. In this application, a customer's module is integrated with the Z684. A PCI I / O buffer block that converts the Z684 unidirectional signals to bidirectional PCI bus signals is also shown in *Figure 205* on page 3.

The customer application module may initiate a transaction or operation as the master for another PCI device. As an initiator of a PCI transaction, the module sends all necessary signals and data to the Z684; the Z684 then handles the rest of the transaction. As a target of a PCI transaction, the module receives data and all necessary signals from the Z684; the complex PCI control and scheduling are handled completely by the Z684. The following sections provide more detail on each component.

Customer Application Module (“Module”)

By using an i486-like module / Z684 interface bus protocol, the Z684 can be integrated into any customer-designed module. The integration is limited only by the available gates and I / O pins in the array. The

Customer Application Module may have its own external I/Os, which do not interface with the Z684 and which may have their own memory or I/O space. In this document, the term *module* by itself refers specifically to the Customer Application Module.

Module/Z684 Interface Bus ("Module Bus")

This is an i486-like bus which interfaces between the customer's module and the Z684. This protocol was chosen because more designers are familiar with the x86 protocol than any other bus protocol; in addition, the i486 bus protocol is easy to understand. On the module bus, all signals are unidirectional signals, and are either inputs or outputs of the Z684. In this manual, the term *module bus* specifically refers to the module / Z684 interface bus.

Internal PCI Bus

All Z684 interface signals are unidirectional signals. For each bidirectional PCI signal, the Z684 has three corresponding unidirectional signals: one input, one output and one enable signal.

- The input signal starts with an "I" prefix
- The output signal starts with an "O" prefix
- The enable signal starts with an "E" prefix
- Low-assertion output signals end with "B" suffix.

For example, the PCI target ready signal TRDY# has three corresponding Z684 signals: ITRDYB for input, OTRDYB for output and ETRDYB for enable. When the enable signal is asserted (the enable signal is asserted LOW for most PCI signals), the output is valid and the input signal is in a "don't care" state. When the enable signal is deasserted, the input is valid and the output is 3-stated.

To avoid confusion with formal bidirectional PCI signals, all unidirectional Z684 PCI signals are called "internal" PCI signals. These signals form an internal PCI bus interface between the Z684 and PCI I/O buffers.

PCI I/O Buffers

These are I/O buffers to convert the internal unidirectional PCI signals into formal PCI bus signals. The buffers also provide the signal drive strengths defined by PCI electrical specifications. The following table lists the available PCI I/O buffers. These buffers must be placed in the design by the designer.

PCI I/O Buffer Macrocells

Family	Cell Name	Equivalent SSO Current(mA)	
		GND	VDD
MSM13R/98R	OTPCILH	24	12
	ODPCILH	24	—
	BTPCILH	24	12
	OTPCILL	12	12
	BTPCILL	12	12
MSM10R	OTPCI	24	24
	BTPCI	24	24

Z684 PCI Mega Macrocell

The Z684 functions as a bus protocol translator. The Z684 translates i486-like module bus signals into PCI bus signals. The two four-entry FIFOs are used to improve write transactions performance. The Z684 is partitioned into the following blocks:

- Configuration registers
- FIFO control logic
- Post-write FIFOs
- Master, slave and parity blocks

The following subsection describe these blocks in turn.

Configuration Registers

There are 64 byte-wide configuration registers implemented in the Z684 to support configuration control of the PCI device. The configuration registers can either be accessed from the module bus internally or from the PCI bus externally.

FIFO Control Logic

FIFO control provides the data traffic handling and scheduling of the Z684. It arranges priority of the data flow when multiple transactions occur.

Post-Write FIFOs

All write cycles are postable. Two 4x36-bit FIFOs store the address, data, and command / byte enable information during write transactions.

- The module-to-PCI write FIFO (“MPWR FIFO”) stores data and byte enable bits during module-to-PCI memory write cycles (single or burst).
- The PCI-to-module write FIFO (“PMWR FIFO”) stores address, data and command / byte enable bits during PCI-to-module cycles (single or burst).

The FIFOs eliminate wait states in most write transaction and thus improve system performance.

Figure 207 on the following page shows the data path from the Customer Application Module to the PCI bus.

During module-to-PCI memory write cycles, the data and byte-enable bits are stored in the MPWR FIFO. Only four 36-bit data words can be stored in the FIFO. If a burst-write transaction fills up the FIFO, the Z684 issues a retry request by asserting PBOFFB LOW. The transaction then resumes when the FIFO is not full.

A request to access the PCI bus is sent whenever MPWR FIFO is not empty. The data and byte-enable bits are then retired to the PCI memory address of a specific agent.

The FIFO can be disabled by programming the FIFO enable bit, bit 0, in the Z684 Control Register. The data and byte-enable bits pass through the data and byte enable latches when the FIFO is disabled.

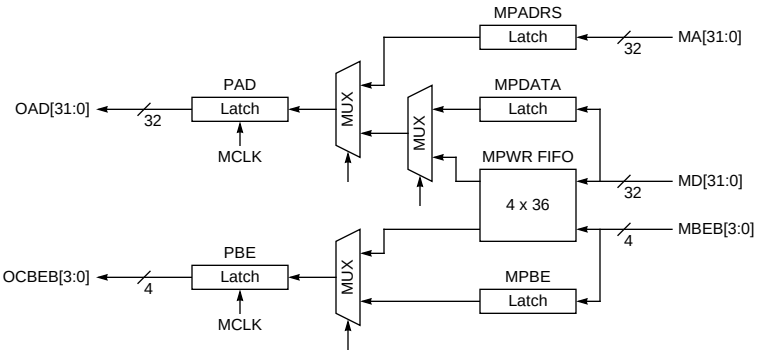

Figure 207. Data Path from the Customer Application Module to the PCI Bus

Figure 208 shows the data path of a PCI device write to module memory.

During PCI-to-module memory write cycles, the address, data and command / byte-enable bits are stored in the PMWR FIFO. As with the MPWR FIFO, only four 36-bit address, data and command / byte-enable entries can be stored in the FIFO. If a PCI burst-write transaction fills up the FIFO, the Z684 issues a retry request by asserting OSTOPB and ESTOPB LOW. The transaction then resumes when the FIFO is not full. A request to access the Customer Application Module bus is sent whenever the MPWR FIFO is not empty. The data and byte-enable bits are then retired to the Customer Application Module memory specified by the address.

As with the MPWR FIFO, the PMWR FIFO can be disabled by programming the same FIFO enable bit 0 in the Z684 Control Register. The address / data and command / byte enable bits pass through the PMAD and PMBE latches when the PMWR FIFO is disabled.

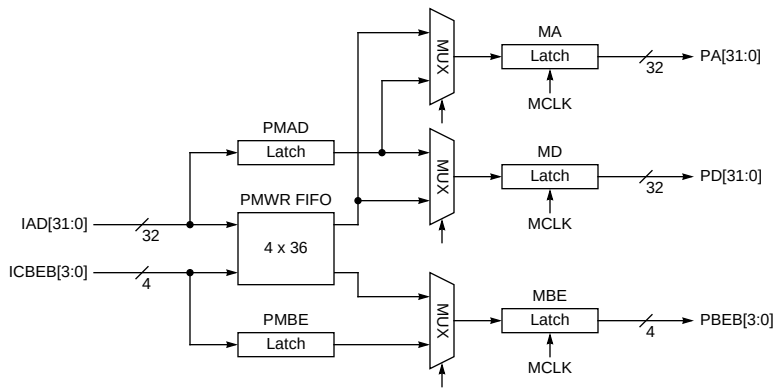

Figure 208. Data Path from the PCI Bus to the Customer Application Module

Master, Slave, and Parity Blocks

The Z684 supports both master and slave functions of the PCI protocol. When the Customer Application Module intends to access the PCI bus, the Z684 functions as a bus master. When a PCI agent targets the Customer Application Module in a transaction, the Z684 functions as a slave device. The Z684 also supports parity error checking and reporting functions. The following paragraphs provide more information on the master, slave, and parity blocks.

- *Master Block* – After detecting a grant signal from a bus arbiter to access the PCI bus, the master block initiates a transaction cycle by asserting all necessary PCI signals. The cycle continues either until the last data has been transferred, or until a retry / termination signal, generated by the target, is detected by the master block. All standard PCI master functions and bus activities are supported by the master block.
- *Slave Block* – The PCI address is monitored and decoded by the Z684 during the PCI address phase. When the PCI bus address matches the Z684's address space, the Z684 becomes the PCI transaction's target. The slave block in the Z684 responds to the bus access and claims the transaction by asserting the device select signal; the transaction then starts. The slave block signals the PCI device which initiated the transaction to stop or retry the transaction when the PMWR buffer is full. The slave block handles all PCI target functions and bus activities.
- *Parity Block* – The parity block handles parity generation, checking and reporting. The parity bit is generated during the transaction according to the PCI specification. Parity is checked one clock after it is generated. In case of address, data, or system errors, the error is reported by error reporting signals to the Z684's configuration registers and related destinations.

For More Information

For information on the Z684 register descriptions and for Z684 functional waveforms, please contact OKI ASIC Application Engineering.