LSI LOGIC

ATMizer MegaCore
Preliminary

Description

In a bold move towards the emerging ATM market, LSI Logic has introduced the ATMizer™ portfolio of ATM technology that enables developers to dramatically reduce time to market when developing ATM networking products. The ATMizer MegaCore™ can be embodied in customer specific ICs or ASSPs, or it can be unbundled to quickly develop ASICs or to differentiate networking product solutions. Figure 1 shows a typical ATMizer installation with supporting chips.

The ATMizer is a single-chip segmentation and reassembly ATM network controller. The ATMizer provides more power and flexibility than other segmentation and reassembly devices due to its ATM Processing Unit (APU), which is a 32-bit, user-programmable RISC CPU based on the MIPS R3000 architecture. The ATMizer also includes carefully chosen hardware functional blocks that can be woven together by user firmware to solve the particular problems of the user's system. The operational diversity of the ATMizer is due to user firmware downloaded to the APU. Solutions that the ATMizer offers to networking problems include cell segmentation and reassembly, cell switching, VCI/VPI translation, traffic shaping, statistics gathering, message passing, and diagnostic operation. In addition, the way that the user's system manages CS-PDU link lists, memory buffers, host-ATMizer message passing, and other structures can vary from system to system depending on the firmware.

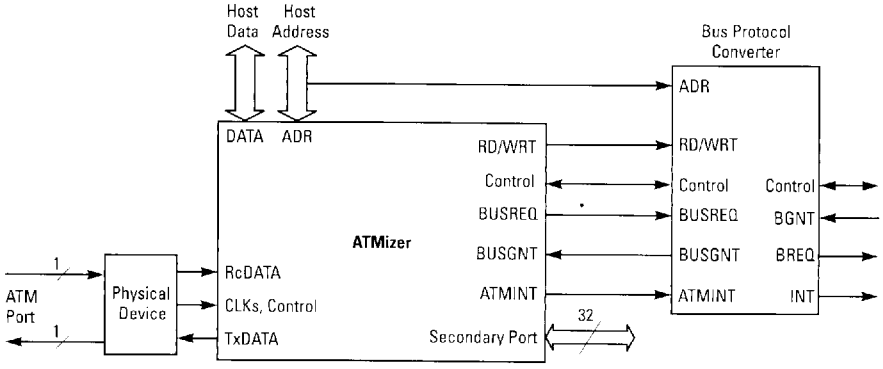


Figure 1. ATMizer with Support Chips

Features and Benefits

- Supports ATM data rates up to 155.54 megabits per second
- Supports simultaneous segmentation and reassembly of some virtual channels (VCs) and cell switching of others
- Features an ATM Processing Unit (APU), a 32-bit, MIPS RISC CPU that supports all ATM cell generation and switching processes under firmware control
- Handles contiguous and noncontiguous CS-PDUs
- Supports ATM Adaptation Layers (AALs) 1, 2, 3/4, and 5, simultaneously
- Generates and appends a 4-byte CRC32 field on AAL5 CS-PDU segmentation
- Connects directly to TAXI interface
- Connects directly to Universal Test & Operations PHY Interface for ATM (UTOPIA)
- Supports up to 65,536 VCs
- Supports user-defined cell size up to 64 bytes
- Reacts immediately to congestion notification with specified congestion control algorithms under firmware control
- Shapes traffic with peak rate pacing, maximum burst length, global pacing, and leaky bucket algorithm
- Operates from system memory in low-cost Network Interface Card (NIC) applications
- Includes a 32-bit DMA controller
- Provides a robust ATM port interface

November 1993 Order Number R15000

LSI LOGIC

ATMizer
MegaCore
Preliminary

Features and Benefits (Continued)

- Generates and checks 4-byte CRC32 field on AAL5 CS-PDU reassembly
- Eliminates buffers with 8-byte, ATM port, elastic receive buffers
- Includes a cache and write buffer that is four words deep for efficient memory bus utilization
- Supports atomic transactions
- APU controls:
  - Scatter-gather DMA algorithms
  - AAL header and trailer generation
  - ATM header generation and manipulation
  - ATMizer-host messaging
  - Error handling
  - Congestion control
  - Statistics gathering
  - Diagnostic operation
- Converts ATM cell size transformations into switch-specific format
- Supports VCI/VPI translation and cell switching
- Generates and checks CRC10 for AAL2 and 3/4 SAR PDUs
- Gathers statistics under firmware control
- Cell multiplexing/cell demultiplexing from up to 65536 VCs/VPs
- Supports Header Error Control (HEC) generation and checking
- Controls up to ten peak rate pacing counters and maximum burst length
- Marks and manipulates cell loss priority (with AAL5 high-med-low priority CS-PDU support)
- Inserts IDLE cells for automatic cell rate decoupling
- Includes 8-bit parallel transmit and receive ATM data ports
- Gathers CRC10 and CRC32 error statistics
- Forces CRC10 and CRC32 errors for diagnostic purposes
- Provides APU network management and troubleshooting
- Supports low-cost network interface cards with 4-Kbyte Virtual Channel RAM (VCR)

ATM Layers

The ATMizer addresses the portions of three ATM layers highlighted in Figure 2. In addition, it performs operation and management (OAM) functions for the supported layers.

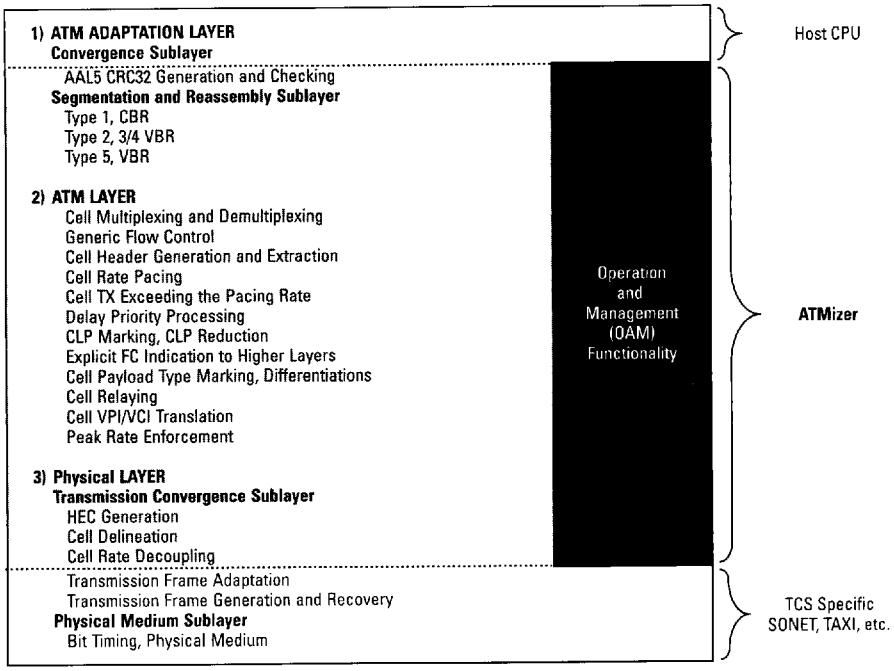


Figure 2. ATMizer-Supported ATM Layers

Block Diagram

Figure 3 is a functional block diagram showing the relation of nine hardware functions that firmware controls within the ATMizer. A brief description of each block in the diagram follows.

ATMizer Processing Unit (APU)

The 32-bit RISC APU firmware builds cells, controls messages between the ATMizer and host, and services channel sequencing. The user writes the firmware and controls almost all operational functions of the ATMizer including the following:

- SAR-PDU generation, ATM cell generation
- DMA initialization and operation
- Pacing rate unit configuration
- ATM cell interface, cell queuing, and cell processing
- Memory allocation
- ATMizer-host messaging
- Atomic transactions
- CS-PDU segmentation on priority
- Congestion control
- Other application-dependent features

The APU accesses external devices through the DMA or secondary port each having 4 Mbytes of address space. Unless the APU attempts to use data before it is written into the appropriate internal register, load scheduling prevents APU stalls. During back-to-back store operations, an APU write buffer that is four-words deep enhances throughput. The load/store effective address decodes whether the operation is a DMA, a secondary port, or an internal access.

Data Cache

The ATMizer has a write-through cache that is four-words deep and functions as a prefetch buffer. Burst or block fetch transactions load a channel parameter entry, buffer list, or other data structure. A field within the APU configuration register specifies the block size to be fetched during a cache miss. The APU programs both the configuration register and the ATMizer system control register to configure the cache block size.

The cache provides fast transfers to the APU for channel parameter entries from an external

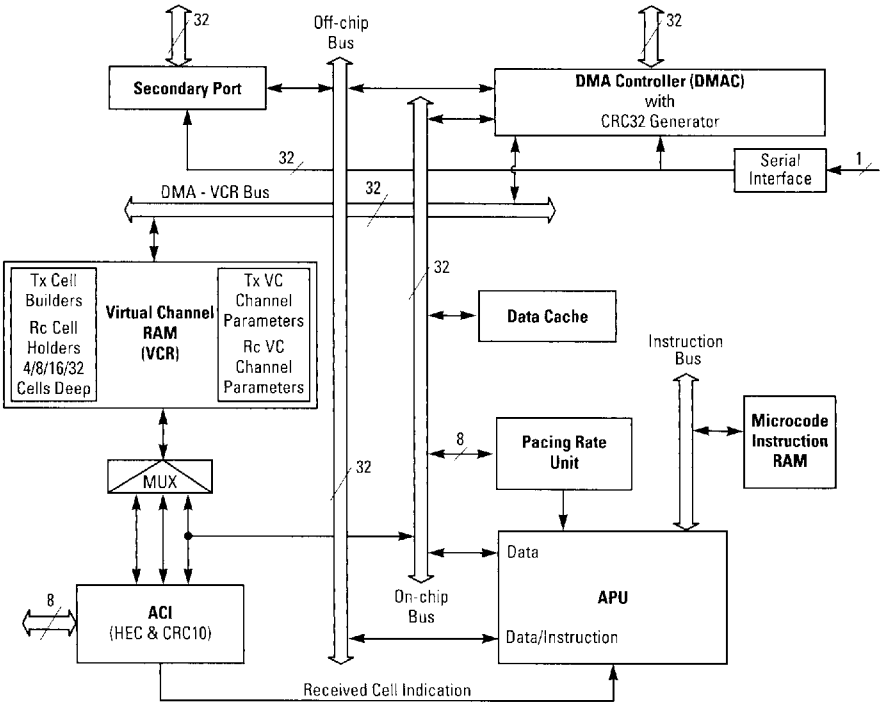


Figure 3. ATMizer Block Diagram

Block Diagram
(Continued)

buffer. When an entry is needed, the APU performs a load from external memory. If the cache block size is programmed for four words and the load instruction causes a cache miss, the ATMizer requests a burst read transaction four words in length. When the first word arrives (latency depends on the external memory), the cache controller strobes it to the APU while fetching the other three words. When the APU wants to fetch the next three words, it causes single-cycle cache hit accesses.

Microcode Instruction RAM

The 1024 x 32 instruction RAM contains 4096 bytes of user-written firmware that is downloaded during system reset. There are two download modes, serial and parallel. The serial mode is designed to work with the commonly available serial PROMs. An external signal, SRL_BOOT, distinguishes between the serial and parallel mode. Loading is serial when SRL_BOOT is zero and parallel when SRL_BOOT is one. The HBS_BOOT signal selects the secondary port when it is zero or the DMA port when it is one.

Virtual Channel RAM (VCR)

Most ATMizer operations involve data transfer to or from the 1024 x 32-bit dual-ported Virtual Channel RAM (VCR). The DMA controller, the ATM cell interface, and the APU can read and write to the VCR. All incoming cells are written into the VCR prior to processing. The APU decides whether to terminate a cell (reassemble it into a CS-PDU or a data buffer) or to switch the cell internally or externally. All outgoing cells are either constructed in the VCR (segmentation) or transferred to the VCR (external switching) prior to transmission. In addition, channel parameter entries, memory buffer lists, messages, and other parameters are stored in the VCR. Storing parameters inside the ATMizer enhances its use in a variety of cost-sensitive applications such as network interface cards without local memory.

Figure 4 shows VCR partitioning for a Network Interface Card (NIC) and for a router. In the first example, an NIC in a PC or workstation supports a limited number of open channels. All channel parameter entries, for both transmit and receive channels, are stored in the VCR, which eliminates the need for off-chip local memory. In the second example, the router supports an unlimited number of open channels, but places a restriction on the number of VCs that can have CS-PDUs under active segmentation at any one time.

Figure 5 on the next page shows the types of software structures that can be stored in the VCR.

Pacing Rate Unit (PRU)

The ATMizer implements the peak rate pacing and maximum burst length control functions. When one of the ten peak rate pacing counters (PRPCs) reaches zero, "Credit to Send" is given to all CS-PDUs associated with that PRPC. Anytime one or more PRPCs has timed out but has not yet been serviced, internal hardware asserts the APU input CpCond2 or the Interrupt1 signal depending on the time-out mode selected. Firmware running on the APU periodically checks the state of CpCond2 by executing the Branch on Coprocessor Condition 2 True instruction. If CpCond2 is true, at least one PRPC has timed out and the APU must segment the CS-PDUs attached to the PRPC or PRPCs that have reached their service intervals. The APU determines which PRPCs have timed out by reading the 10-bit Channel Group Credit Register (CGCR). Each bit set in the CGCR indicates that the corresponding PRPC has timed out since the APU last cleared its bit. The PRPC can be clocked by the system clock (CLK) or the transmission line clock connected to the PRU_CLK pin. Note that the maximum frequency of PRU_CLK cannot exceed one fourth of the system clock frequency. In some applications, the PRPC counters must count with the transmission line clock, so the TX_CLK is

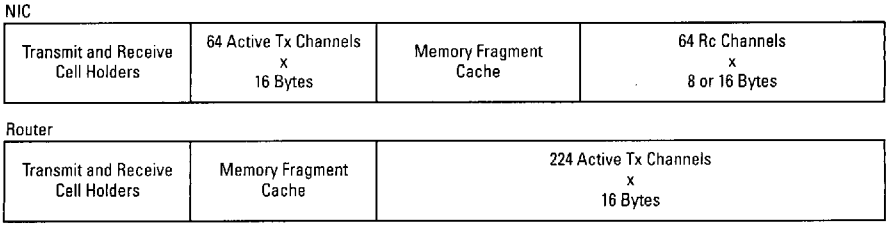


Figure 4. Sample VCR Partitioning for an NIC and a Router

Block Diagram
(Continued)

connected to PRU_CLK. If this is the case and if, for example, the system clock (CLK) is 40 MHz and TX_CLK is 19.4 MHz, the system designer must divide TX_CLK by two before connecting it to the PRU_CLK pin. Internally, the frequency is multiplied by two before PRU_CLK is fed into the appropriate counter.

Traffic Shaping – The Global Pacing Rate Register (GPRR) allows the ATMizer to limit the data rate on its transmission port within the same cell time that the congestion is recognized from an incoming cell. The amount of initial reduction as well as the algorithm by which the ATMizer returns to full speed operation can be

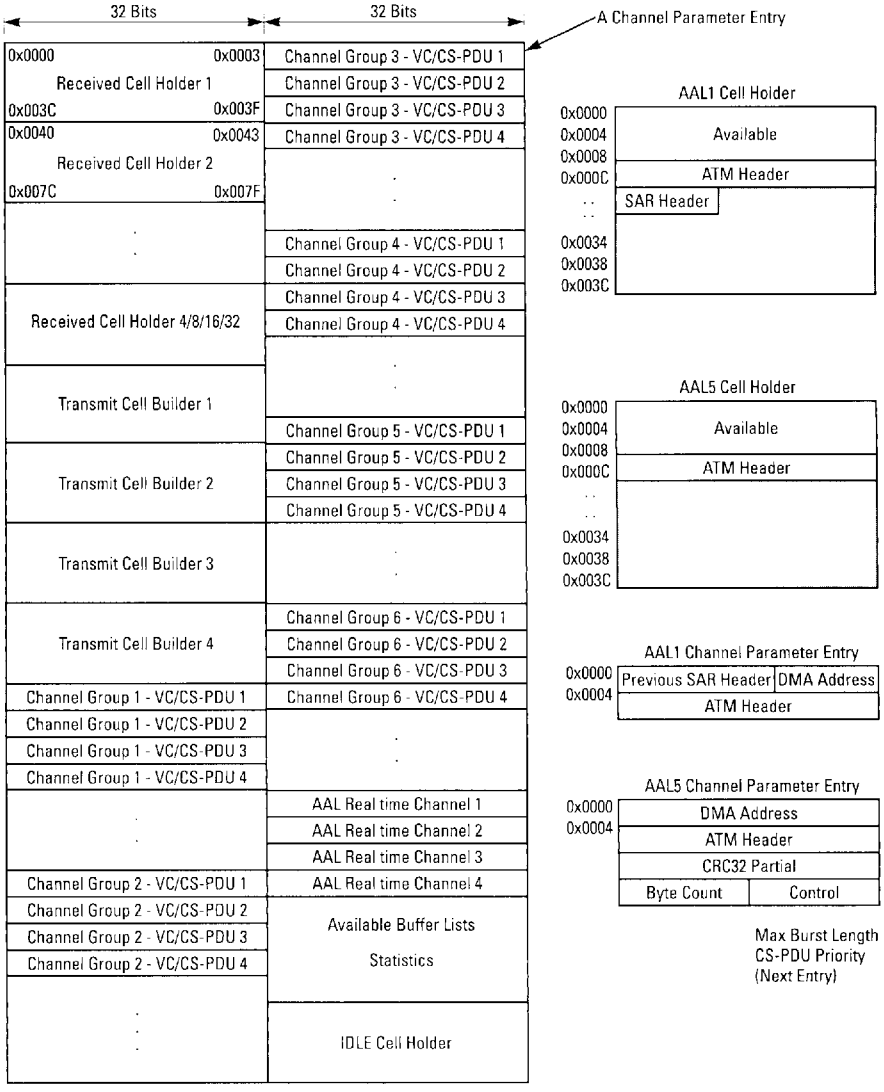


Figure 5. VCR Software Structures

Block Diagram
(Continued)

implemented intelligently in APU firmware and can be modified as more is learned about ATM network congestion. A single APU instruction modifies the GPRR, which determines the percentage of assigned cells sent out over the ATMizer's ACI transmit port. High priority channels have access to the reduced throughput while lower priority channels are blocked.

Leaky Bucket – The ATMizer has two 24-bit, general purpose counters that can be used to implement the leaky bucket algorithm. One counter tracks the rate to fill the bucket, and the second counter tracks the number of tokens in the bucket.

DMA Controller (DMAC)
The DMA controller, which can perform block transfers up to 64 bytes, includes registers, counters, and a data path that collectively control data transfer operations between the on-chip VCR and main memory. Transfer types include the following:

- Retrieve SAR user payloads from memory-based CS-PDUs during segmentation operations
- Link list pointers
- Write SAR user payloads back into memory-based CS-PDUs during reassembly operations
- Messages
- Application-specific data structures

In addition, in systems that support more simultaneously active VCs than the on-chip VCR can support directly, the DMA controller can be used to retrieve and restore memory-based channel parameters. All DMA registers and counters are initialized by the APU, and all DMA operations are initiated by the APU as part of the standard servicing of events such as "Cell Received" and "Peak Rate Pacing Counter Time-out." Because the DMAC is configured at the start of each DMA operation, it effectively provides an unlimited number of DMA channels.

The DMA controller also contains CRC32 generation circuitry that generates the CRC32 values required for AAL5 CS-PDU protection. CRC32s can be calculated individually for each CS-PDU actively undergoing either segmentation or reassembly. For CS-PDUs undergoing segmentation, the final CRC32 result is appended, under APU control, to bytes [48:44] of the SAR SDU of the last cell generated from the CS-PDU. For CS-PDUs undergoing reassembly, the CRC32 result is compared with the CRC32 received in the last cell of the CS-PDU as a checking mechanism. Because the ATMizer supports cell multiplexing and demultiplexing from up to 64K VCs, the APU must provide CRC32 partial result storage and retrieval services to allow multiple concurrently active CRC32 calculations to be performed by the single CRC32 generator.

ATM Cell Interface (ACI)
The ATM Cell Interface contains the ATM port side transmitter and receiver functions. The ACI's transmitter takes cells built in the VCR and transfers them one byte at a time to an external ATM line serializer/transmitter. The transmitter also generates and inserts the HEC and generates and appends a CRC10 field to AAL3/4 cells. The transmitter also handles cell rate decoupling. If an assigned cell that is ready for transmission does not exist in the VCR, the transmitter automatically sends an IDLE cell. The receiver accepts cells, one byte at a time, from the ATM physical interface and reconstructs them in the VCR so that the APU can either reassemble or switch the cell.

To support applications that employ extra header fields for switch-specific information, the actual size of a cell is user programmable up to 64 bytes. The size must be a multiple of four bytes. The typical ATM cell is represented in the VCR as a 52-byte entity, but the cell size in the VCR can be 56, 60, or 64 bytes. Since the HEC value is generated and inserted into the cell as it is passed out of the ATMizer, the typical ATM cell adheres to the requirement that it be a multiple of four bytes. Two fields in the system control register program the transmit and receive cell sizes.

The proposed standard interface between the PHY and ATM layers is a subset of the ACI interface. The Universal Test & Operations PHY interface for ATM (UTOPIA) is a proposed standard to interface the multiple PHY layers to the ATM layer. There are currently four PHY layers defined by the ATM forum, with a fifth one under study. By implementing the UTOPIA interface as a subset of the ACI, the user can choose from a variety of PHY layers for system requirements.

The Secondary Port
The ATMizer includes a 32-bit secondary port that can be used to accomplish a variety of data transfer and control transfer operations between the APU and the outside world. The

Block Diagram (Continued)

secondary port has a different interface to the external system compared to the direct access APU interface through the DMA port. The secondary port is also unique in that it can be accessed by the APU when the DMA engine is busy because of a dedicated DMA-VCR bus. In this fashion the secondary port gives the APU an ability to control DMA operations simultaneously with the secondary port operation. This capability may be of use in switching applications, if the APU wishes to notify another switching port that it is about to source a cell targeted to it onto the memory backplane.

Signals

Figure 6 shows the ATMizer signals, their direction, and their polarity. A brief description of each signal is given after the diagram. For convenience, signals are grouped and explained according to their functions.

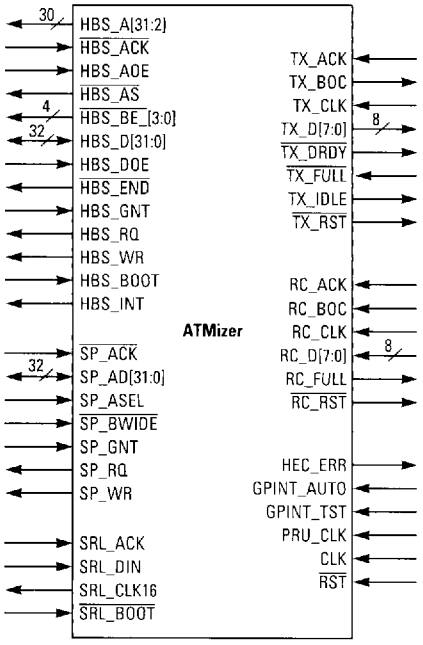


Figure 6. ATMizer Logic Symbol

DMA and Host Signals

Signals that connect the ATMizer to the host and DMA are described below.

HBS_A[31:2] **Output**
Host Bus Address Bus
During DMA or CPU operations, the host bus interface sources this address bus to provide the ATMizer with access to all system components. The ATMizer's DMA Engine increments HBS_A[23:2] in response to HBS_ACK, except on the last word of a DMA transfer. Note that signals HBS_A[31:24] are never incremented, so user firmware should never initiate DMA operations that cross 4-Mbyte boundaries.

HBS_ACK **Input**
Host Bus Read/Write Acknowledgment
External logic asserts HBS_ACK LOW when the DMA Engine or CPU initiate memory read operations to indicate that they have placed valid data on HBS_D[31:0]. During memory write operations, an external device asserts HBS_ACK to tell the ATMizer that it has retired the data for the current write operation and is ready for the next address/data pair.

HBS_AOE **Input**
Host Bus Address Output Enable
When HBS_AOE is HIGH, the address bus is enabled. When HBS_AOE is LOW, the address bus is 3-stated.

HBS_AS **Output**
Host Bus Address Strobe
At the beginning of each transaction, after HBS_GNT is asserted, the ATMizer asserts HBS_AS and drives the address bus with valid address.

HBS_BE_[3:0] **Output**
Host Bus Byte Enable
During a 32-bit word access from an external device, the host bus interface asserts the four HBS_BE_[3:0] signals LOW and drives the appropriate address onto HBS_A[31:2]. When accessing a single byte, only one of the four HBS_BE signals is asserted to indicate valid data on the corresponding byte port. The table below maps the assertion of each HBS_BE_[3:0] signal to the corresponding valid data byte on HBS_D[31:0].

HBS_BE_[3:0]	HBS_D[31:0] Byte
HBS_BE_3	[31:24]
HBS_BE_2	[23:16]
HBS_BE_1	[15:8]
HBS_BE_0	[7:0]

Signals
(Continued)

HBS_D[31:0] **Bidirectional**
DMA Memory Data Bus
During memory read operations, the host bus interface samples HBS_D[31:0] on the rising edge of CLK when HBS_ACK is asserted. During memory write operations, the host bus interface sources data onto HBS_D[31:0]. During write operations, the DMA controller responds to HBS_ACK by sourcing data for the next transfer onto HBS_D[31:0]. The HBS_D[31:0] data are not changed following acknowledgment of the last transfer of the DMA operation.

HBS_DOE **Input**
Host Bus Data Output Enable
When HBS_DOE is HIGH, the data bus is enabled. When it is LOW, the data bus is 3-stated.

HBS_END **Output**
DMA Operation Ending
The DMA controller asserts HBS_END LOW to warn the memory controller that the current transfer will end when the next HBS_ACK is returned to the ATMizer. The memory controller uses this warning to gain an early start on RAS precharge or to grant the bus to another master. HBS_END is deasserted following the rising clock edge during which the ATMizer samples the final transfer acknowledgment (HBS_ACK asserted) for the given DMA operation.

HBS_GNT **Input**
Host Bus Grant
An external arbiter grants the bus to the ATMizer by asserting HBS_GNT.

HBS_RQ **Output**
Host Bus Request
The ATMizer drives HBS_RQ HIGH when the APU has programmed the DMA or when the APU wants to execute a word, partial word, or block transfer to or from a memory mapped device (including main memory). The accessed device responds to HBS_RQ by asserting HBS_GNT to allow the ATMizer to proceed with the transfer.

HBS_WR **Output**
Host Bus Write
HBS_WR is asserted when the ATMizer performs a write transaction. HBS_WR is deasserted when the ATMizer performs a read transaction.

ATM Cell Interface Transmit Signals
Seven ATM signals are associated with data transmission control. In addition to the control signals, the ATMizer has a separate 8-bit data bus.

TX_ACK **Input**
ACI Transmitted Data Acknowledgment
The Transmission Convergence Sublayer (TCS) framing logic asserts TX_ACK HIGH when it processes data on TX_D[7:0]. The ATMizer responds to TX_ACK by placing the next byte of the existing cell or the first byte of the next cell (assigned or IDLE) onto TX_D[7:0]. TX_ACK is deasserted if external logic is unable to sample the byte on TX_D[7:0] in a given cycle.

TX_BOC **Output**
Beginning of Cell
The ACI transmitter asserts TX_BOC HIGH while the first byte of a cell is sourced on TX_D[7:0]. TX_BOC is removed after the first TX_ACK is received. TX_BOC should be qualified with TX_DRDY.

TX_CLK **Input**
ATM Cell Interface Transmitter Clock
TX_CLK is the byte clock of the external transmitter, and it drives the elastic byte buffer in the transmitter portion of the ATMizer's ATM cell interface. All data transfers from the ATMizer over TX_D[7:0] are synchronized to this clock, as are TX_DRDY, TX_BOC, and TX_IDLE. Logic inside the ATMizer synchronizes the ATMizer's system clock and the ACI transmitter's elastic data buffer circuitry, which is sequenced off of TX_CLK. The system designer need not worry about metastability at the transmitter output.

TX_D[7:0] **Output**
ATM Cell Interface Transmission Data
The ATMizer sources byte-aligned cell data onto TX_D[7:0], which supports either the Transmission Convergence Sublayer (TCS) framing logic or, for 8B/10B encoding, the TAXI chipset.

TX_DRDY **Output**
ATM Cell Interface Transmit Data Ready
The TX_DRDY signal is asserted LOW and is significant after the transmitter has been reset by system reset (RST), or on transmitter reset that is done in firmware by writing the IDLE cell holder address. After these resets, data are

Signals
(Continued)

ready a number of cycles later, and the ATMizer asserts $\overline{\text{TX\_DRDY}}$ and TX_BOC to indicate that external logic can sample TX_D[7:0] and issue a TX_ACK. Once asserted, $\overline{\text{TX\_DRDY}}$ remains asserted until the next system or transmitter reset or until $\overline{\text{TX\_FULL}}$ is asserted.

$\overline{\text{TX\_FULL}}$ **Input**
TCS Receive Buffer Full
This signal is used with the UTOPIA interface. When the TCS receiver buffer is full, it notifies the ACI by asserting $\overline{\text{TX\_FULL}}$. The ACI responds by deasserting $\overline{\text{TX\_DRDY}}$, which should be connected directly to the PHY layer enable signal.

TX_IDLE **Output**
IDLE Cell
The ACI transmitter asserts TX_IDLE HIGH during the entire period that an IDLE cell (53 bytes long) is sourced onto TX_D[7:0]. Transmission convergence framing logic that does not transmit IDLE cells must still assert TX_ACK until TX_IDLE is LOW.

$\overline{\text{TX\_RST}}$ **Output**
Transmitter Reset
After the ATMizer is powered on, $\overline{\text{TX\_RST}}$ is asserted LOW within two to four cycles of the ATMizer system clock to reset the physical layer. $\overline{\text{TX\_RST}}$ is removed approximately two clock cycles after the transmit initialize bit is set to one.

ATM Cell Interface Receive Signals
There are six control signals and an 8-bit data bus associated with ATMizer receive signals.

HEC_ERR **Output**
HEC Error
The ATMizer asserts HEC_ERR HIGH when the actual HEC field that is received (byte 5 of a cell) does not equal the HEC field that the ATMizer calculated from the ATM Header. HEC_ERR is only active when the receiver is configured to accept and check the HEC byte.

RC_ACK **Input**
ACI Receive Data Acknowledgment
Framing logic in the transmission convergence unit asserts RC_ACK HIGH when it has placed data on RC_D[7:0]. The ATMizer responds to RC_ACK by sampling RC_D[7:0] on the rising edge of RC_CLK. External logic should deassert RC_ACK if it is unable to supply a byte on RC_D[7:0] in a given cycle.

RC_BOC **Input**
Receiver Beginning of Cell
RC_BOC is an input to the ACI Receiver that signals the beginning of a cell.

RC_CLK **Input**
ATM Cell Interface Receiver Clock
The RC_CLK drives the elastic byte buffer inside the ATM cell interface receiver. All data transfers over RC_D[7:0] to the ATMizer, as well as the assertion of the HEC_ERR signal, are synchronized to this clock. Logic inside of the ATMizer handles synchronization between the ATMizer's system clock and the ACI's receive data buffer circuitry.

RC_D[7:0] **Input**
ATM Cell Interface Data for Reception
The TCS framing logic or the TAXI chipset (for 8B/10B encoding) uses RC_D[7:0] to send byte aligned cell data to the ATMizer. Bit 7 is the first bit received over the serial line.

RC_FULL **Output**
Received Cell Holder Buffer Full
The ATMizer asserts RC_FULL when its receive cell buffer has room for at least four more bytes. When the receiver asserts RC_FULL, at least four or more bytes can be accepted. In other words, the remaining bytes of the current cell are accepted. When the buffer is full, additional bytes make the ACI operation invalid.

$\overline{\text{RC\_RST}}$ **Output**
Receiver Reset
$\overline{\text{RC\_RST}}$ is asserted LOW after $\overline{\text{RST}}$ is active. Because several parameters have to be configured before the ACI can receive any cell, firmware controls the deassertion of $\overline{\text{RC\_RST}}$ by setting the $\overline{\text{RC\_RST}}$ bit in the system register to one.

Secondary Port Signals
The secondary port is controlled by a 32-bit, multiplexed address and data bus. In addition to the bus signals, the ATMizer supports six additional port control signals.

Signals
(Continued)

SP_ACK **Input**
Secondary Port Acknowledgment
External logic asserts SP_ACK during a secondary port read operation one cycle before valid data is available on SP_AD[31:0]. External logic can extend a read access indefinitely by withholding SP_ACK, which forces the bus interface unit to stall until SP_ACK is asserted. External logic also must assert SP_ACK in response to a secondary port write operation. External logic should assert SP_ACK one cycle before it retires the write data sourced on SP_AD[31:0].

SP_AD[31:0] **Bidirectional**
Secondary Port Address
SP_AD[31:0] is a multiplexed address and data bus on the secondary port. During the address phase, SP_ASEL is HIGH, and SP_AD[31:0] contains both an address and the information shown in the table below.

SP_AD Bits	Field Definition
[31:28]	Byte Enables
[27]	Asserted Low
[26]	Access Type (0 = Data, 1 = Instruction)
[25]	Block Fetch
[24]	Atomic
[23:22]	Asserted Low
[21:2]	Address
[1:0]	Asserted Low

When SP_ASEL is LOW, SP_AD[31:0] contains data. When SP_ASEL and SP_WR are LOW, the SP_AD[31:0] lines go to 3-state, which allows the external device to drive the bus and present information for the secondary port.

SP_ASEL **Input**
Secondary Port Address/Data Select
When SP_ASEL is HIGH, the address on SP_AD[31:0] is selected. When SP_ASEL is LOW, data are selected.

SP_BWIDE **Input**
Secondary Port Byte Wide Device
External logic asserts SP_BWIDE LOW when an external device is 8-bits wide. With SP_BWIDE asserted, the secondary port executes four cycles with sequential byte addresses beginning with the effective address. If the effective address is not modulo 4, the secondary port wraps around to get all of the bytes into the word in which the effective address falls.

SP_GNT **Input**
Secondary Port Bus Grant
SP_GNT grants the bus to the secondary port. When SP_GNT is LOW, SP_AD[31:0] is enabled. When SP_GNT is HIGH, SP_AD[31:0] is 3-stated.

SP_RQ **Output**
Secondary Port Request
The ATMizer asserts SP_RQ HIGH when it performs a transfer using the secondary port.

SP_WR **Output**
Secondary Port Write
SP_WR qualifies SP_RQ. If SP_WR is HIGH while SP_RQ is HIGH, the ATMizer is requesting a secondary port write operation. If SP_WR is LOW while SP_RQ is HIGH, the ATMizer is requesting a secondary port read operation. The secondary port will automatically disable the data output drivers on read operations.

Messaging Signals
Three ATMizer signals provide message control functions, as follows.

GPINT_AUTO **Input**
General Purpose APU Interrupt
GPINT_AUTO is asserted HIGH and is connected to APU Interrupt3. APU software can disable or enable interrupts as necessary.

GPINT_TST **Input**
ATMizer Interrupt
GPINT_TST connects to the CpCond0 signal on the APU. When the Branch on CpCond0 instruction is TRUE, the ATMizer senses this signal, so firmware can determine whether the host has communication for the APU. Messages might include "configure transmit channel," "activate CS-PDU segmentation," or "change pacing rates." The use of this input is entirely user programmable and the messages are user defined. When the level-sensitive GPINT_TST signal is asserted HIGH, the ATMizer can read the secondary port to determine why the host asserted GPINT_TST.

Signals
(Continued)

HBS_INT **Output**
Host Interrupt
This signal is user defined, but is likely to be used as part of the messaging system. The ATMizer may assert HBS_INT HIGH to indicate error conditions, congestion problems, CS-PDUs reassembled, and other conditions.

General Signals
The following are general signals used for housekeeping and similar chores on a typical telecommunications board.

CLK **Input**
System Clock Input
The CLK input runs the ATMizer APU, DMA controller, secondary port, and most ATM cell interfaces. CLK does not effect the transfer of byte data to or from the ATMizer over the ATM cell interface. These transactions are controlled by TX_CLK and RC_CLK.

HBS_BOOT **Input**
Host Bus Port Select
HBS_BOOT selects the DMA or secondary port during the downloading process. When HBS_BOOT is zero, the secondary port is selected. When HBS_BOOT is one, the DMA port is selected.

PRU_CLK **Input**
Pacing Rate Unit Clock
The down counters associated with the ten PRPCs count down by one every clock tick. The clock inputs to the PRPCs are selected by the clock select field in the PRPC configuration register. The system clock or the clock connected to the PRU_CLK pin provides the frequency. Note that the clock connected to the PRU_CLK pin must run at half or less of the system clock's frequency.

$\overline{\text{RST}}$ **Input**
System Reset
The ATMizer master reset allows external logic to download firmware. $\overline{\text{RST}}$ also resets the ACI transmitters and receivers.

$\overline{\text{SRL\_BOOT}}$ **Input**
Serial Boot Select
$\overline{\text{SRL\_BOOT}}$ is used to select the code download mode. When $\overline{\text{SRL\_BOOT}}$ is zero, the serial download mode is selected. When $\overline{\text{SRL\_BOOT}}$ is one, the ATMizer expects to boot from either the DMA port or the secondary port.

SRL_ACK **Input**
Serial Acknowledge
SRL_ACK controls the bit rate that is applied to the ATMizer during serial downloads. When SRL_ACK is asserted, the ATMizer latches the bit on SRL_DIN on the rising edge of SRL_CLK16. When downloading from a serial PROM, tie SRL_ACK HIGH.

SRL_DIN **Input**
Serial Data Input
When SRL_DIN is asserted in serial downloading mode, one bit of data is presented to the ATMizer into the slow PROM devices used in serial download mode.

SRL_CLK16 **Output**
Serial Clock
The SRL_CLK16 signal is derived from the system clock divided by 16. It is used to clock bits into the slow PROM devices used in serial download mode.

Specifications

This section includes the electrical specifications for the LSI Logic ATMizer. During AC testing, HIGH inputs are driven at 3.0 V, and LOW inputs are driven at 0 V. For transitions between HIGH, LOW, and invalid states, timing measurements are made at 1.5 V, as shown in Figure 7.

For 3-state outputs, timing measurements are made from the point at which the output turns ON or OFF. An output is ON when its voltage is greater than 3.5 V or less than 1.5 V. An output is OFF when its voltage is less than V_{DD} – 1.5 V or greater than 1.5 V, as shown in Figure 8.

Timing diagrams for the salient AC timing characteristics of the ATMizer are shown in Figures 9 through 20.

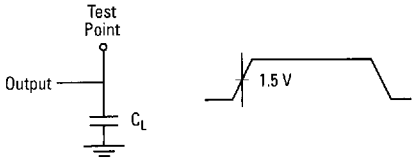


Figure 7. AC Test Load and Waveform for Standard Outputs

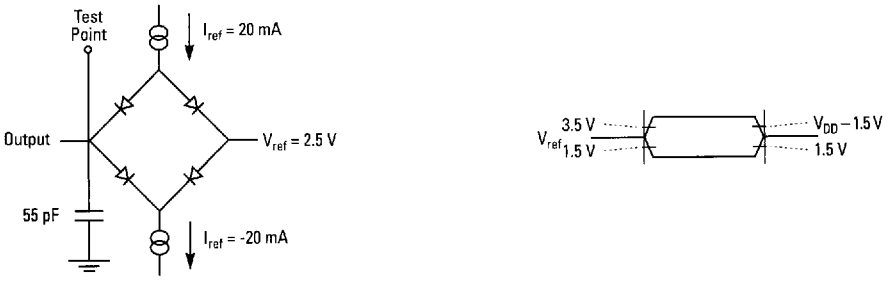
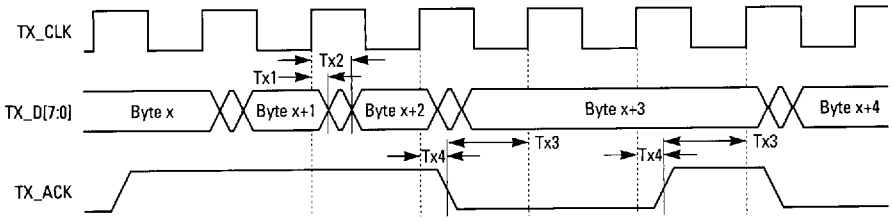


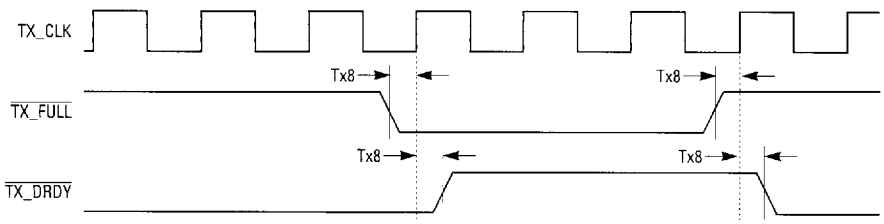
Figure 8. AC Test Load and Waveform for 3-State Outputs



Note:
1. TxClkFreq ≤ 25 MHz.

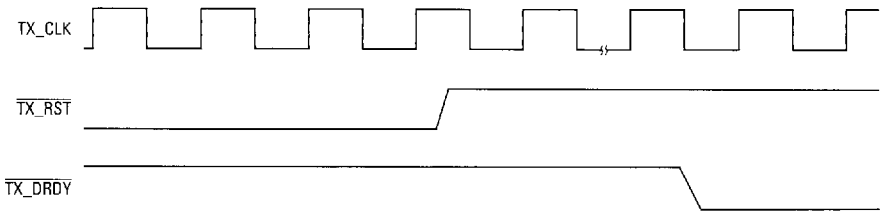
Figure 9. TX_D[7:0] and TX_ACK

Specifications
(Continued)



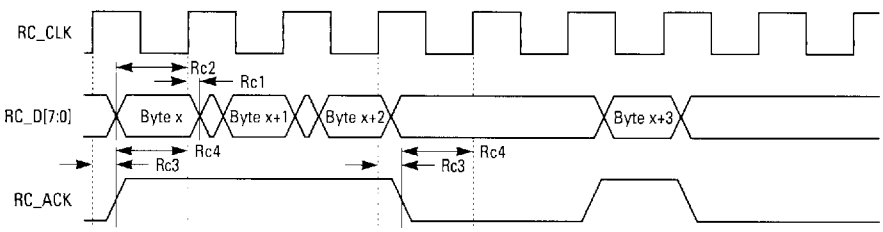
Note:
1. TxClkFreq ≤ 25 MHz.

Figure 10. TX_FULL and TX_DRDY



Note:
1. TxClkFreq ≤ 25 MHz.

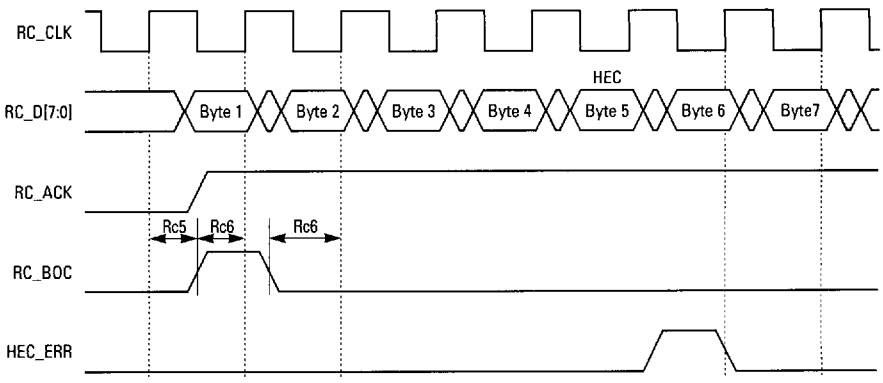
Figure 11. TX_RST



Note:
1. RcClkFreq ≤ 25 MHz.

Figure 12. RC_D[7:0] and RC_ACK

Specifications
(Continued)



Note:
1. RcClkFreq ≤ 25 MHz.

Figure 13. Receiver Synchronization (RC_BOC) and HEC_ERR

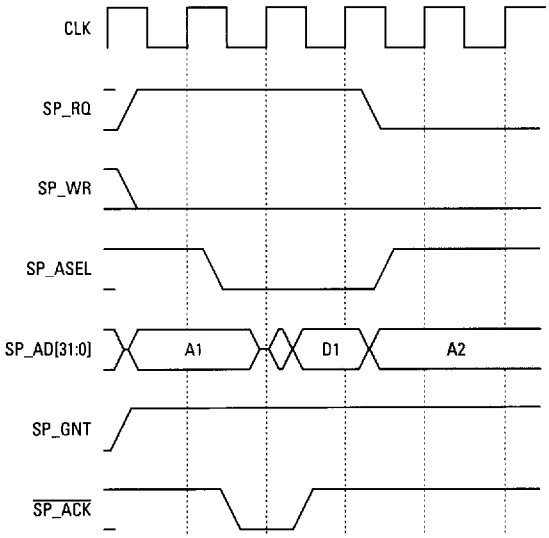


Figure 14. Secondary Port One-Word Read

Specifications
(Continued)

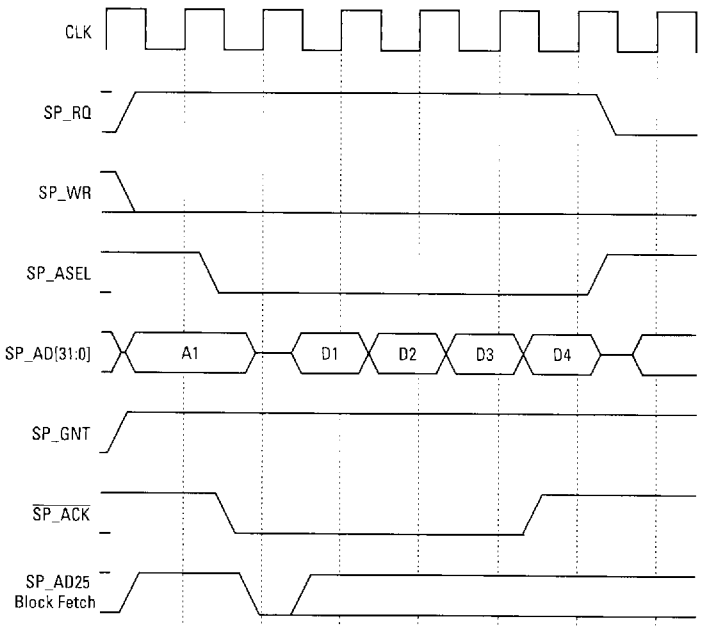


Figure 15. Secondary Port Four-Word Read

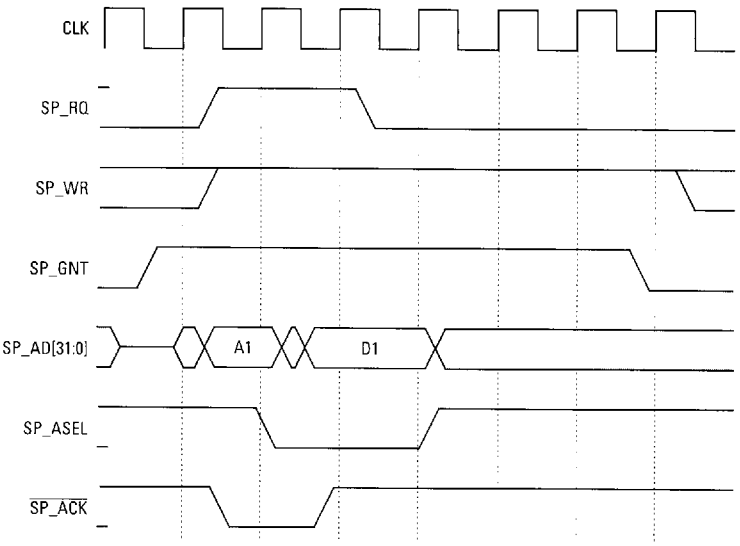


Figure 16. Secondary Port Write

Specifications
(Continued)

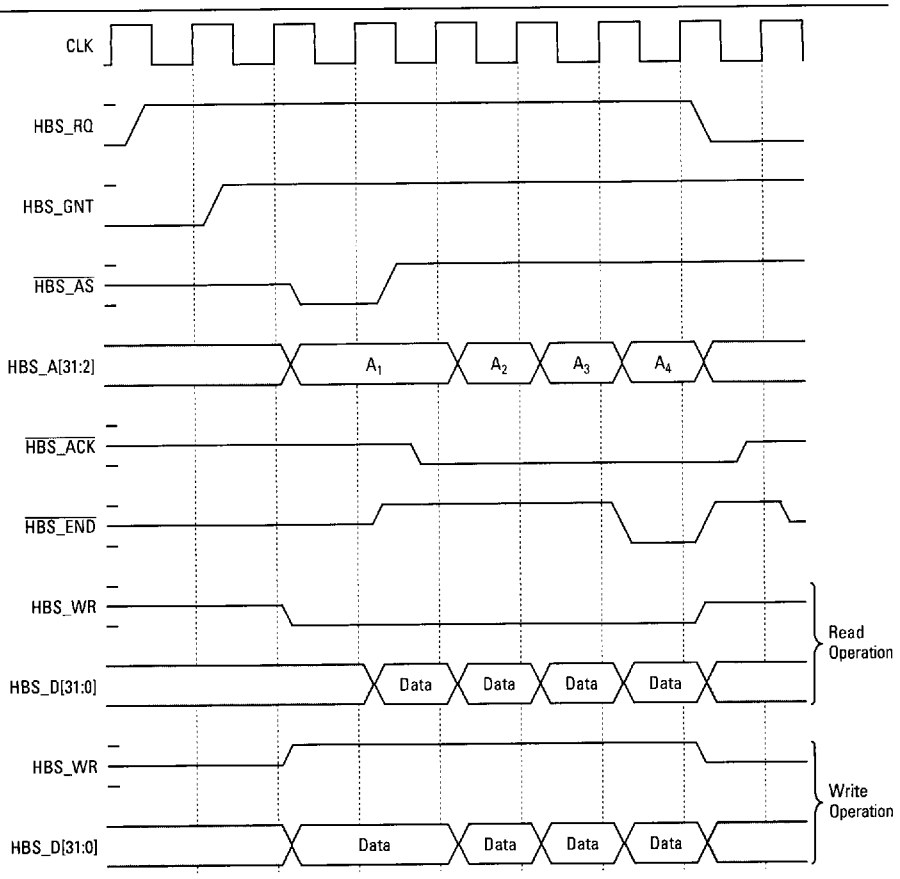


Figure 17. DMA Transaction

Specifications
(Continued)

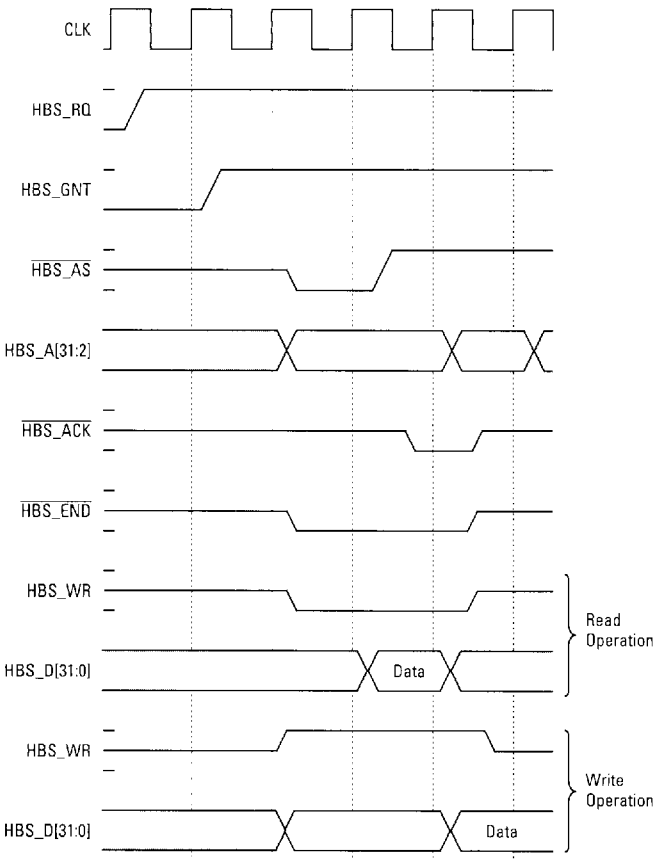


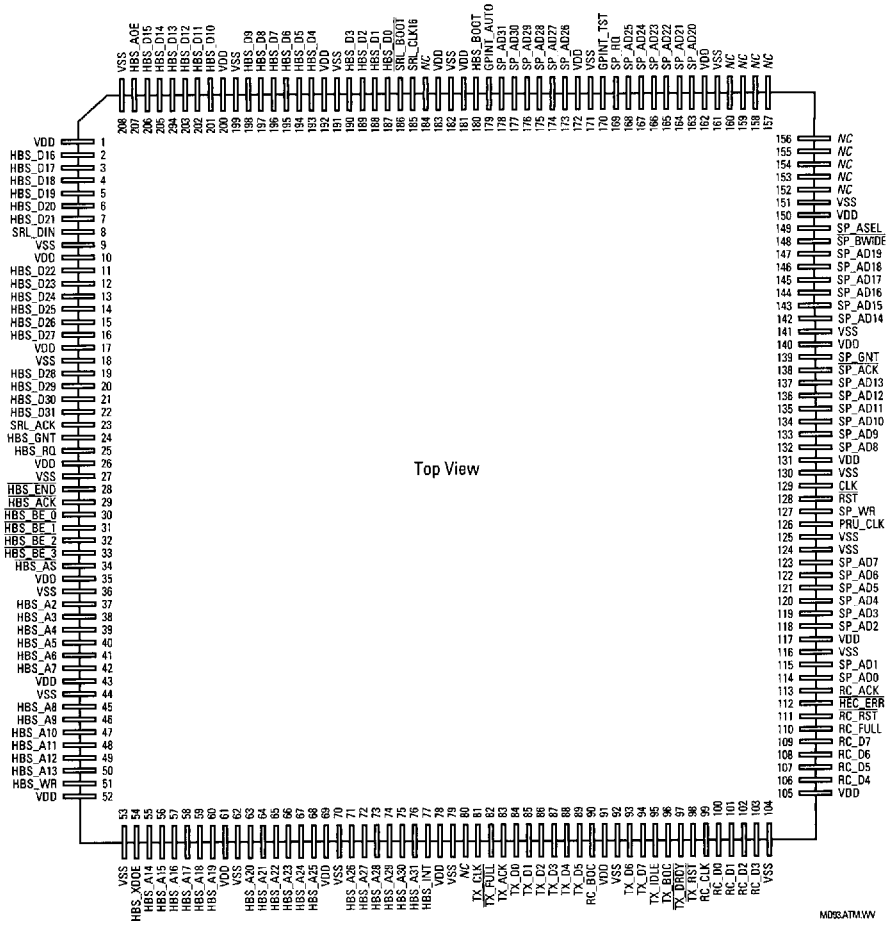
Figure 18. Direct Load/Store Through DMA Port

LSI LOGIC

ATMizer
MegaCore
Preliminary

Pinout and Package Information

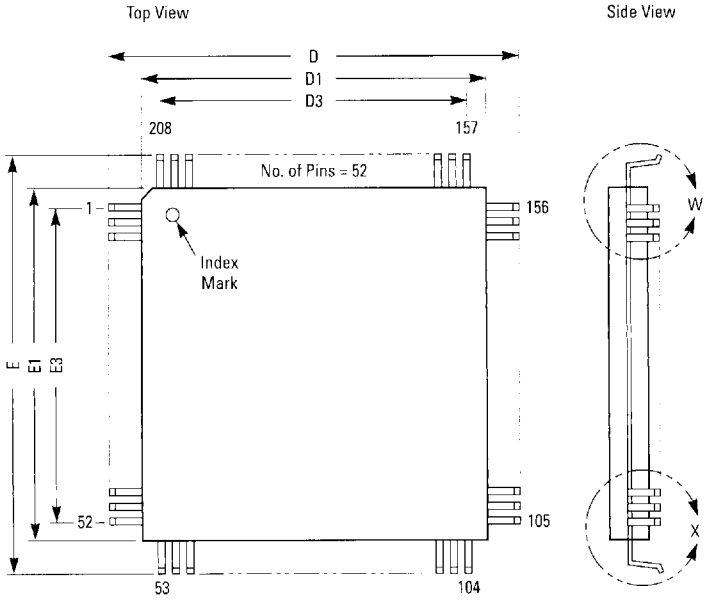
The LSI Logic ATMizer ASSP is available in a 208-pin, cavity down, metal quad flat package (MQUAD). Figure 19 illustrates the pinout of the 208-pin package, and Figure 20 is the mechanical drawing.



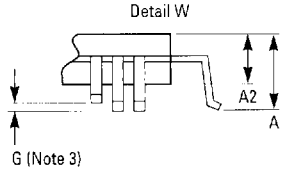
Note:
1. *NC* pins are not connected.

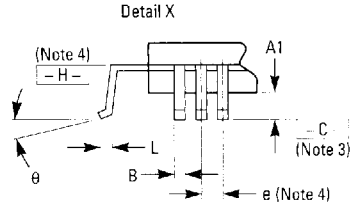
Figure 19. 208-Pin MQUAD Pinout – Cavity Down

Pinout and Package Information (Continued)



Dimension		mm
A	Max	3.86
A1	Min	0.25
	Max	0.51
A2	Min	3.17
	Max	3.43
B	Typ	0.23
D	Min	30.40
	Max	30.80
D1	Min	27.56
	Max	27.72
D3	BSC	25.50
E	Min	30.40
	Max	30.80
E1	Min	27.56
	Max	27.72
E3	BSC	25.50
θ	Min	0°
	Max	7°
e	BSC	0.50
G	Max	0.10
L	Min	0.40
	Max	0.60





Note:
1. Total number of pins is 208.
2. Drawing is not to scale.
3. Coplanarity of all leads shall be within 0.10 mm (difference between the highest and lowest lead with seating plane – C – as reference).
4. Lead pitch determined at – H –.
5. Leadframe to package offset tolerance is ±0.10 mm Max.
6. For board layout and manufacturing, you may obtain engineering drawings from your LSI Logic Products marketing representative by requesting the outline drawing for package code WV.

MD92.WVa

Figure 20. 208-Pin MQUAD Mechanical Drawing