1.0 INTRODUCTION

The AHA3101 Data Compression chip provides the basis for a complete data compression system. The chip uses an external static RAM chip, to store compression information, and also to buffer data.The chip is fabricated in a CMOS process, and is packaged in a 100 pin plastic quad flat pack (PQFP).

The chip architecture provides a low cost, low part count data compression system. It supports several configurations, and provides the system designer with the flexibility to choose between several cost performance tradeoffs.

The data compression algorithm is an enhanced version of the Lempel-Ziv algorithm, known as the DCLZ algorithm. It has been proven in silicon by HP, for a large variety of data sizes and types. The algorithm adapts to the data it is compressing, to ensure high compression ratios. Also, the compression dictionary can be shared between multiple records, allowing high compression of small records.

The primary access to the static RAM data buffer is through two fully programmable, independent DMA ports. The ports support a 5 MByte/sec data transfer rate, for high performance data buffering. The DMA registers allow complete, and easy software control of DMA operations and data buffer management. For added flexibility, the processor can also independently access any byte in the buffer.

The chip has several features which allow easy interfacing to host interface chips, such as SCSI bus interface controllers. These include twenty four bit wide data byte counters, and the capability to support multiple record transfers automatically.

1.1 FEATURES

PERFORMANCE:

- Open Standard compression technology
- Standards include: QIC, DDS/DAT and ECMA
- DCLZ adaptive lossless compression algorithm
- 2 to 1 average compression ratio
- 2.5 MBytes/sec compression rate (max) with a 20 MHz single-phase clock
- 5 MBytes/sec DMA transfer rate
- Compression algorithm uses an embedded dictionary for low overhead

FLEXIBILITY:

- Dynamic compression ratio monitoring and control
- In-line and look-aside modes

SYSTEM INTERFACE:

- 2 independent DMA ports
- 24-bit data counters
- Automatic multi-record transfers without microprocessor intervention
- Programmable interrupt output

OTHERS:

- Low power CMOS technology
- Small 100 pin Plastic Quad Flat Pak (PQFP)

1.2 APPLICATIONS

- DDS/Data DAT helical scan tape drives
- QIC tape drives (Quarter-Inch Cartridge Drive Standards, Inc.)
- High performance disk drives
- High speed data communications systems
- SCSI host-bus adapters
- Embedded controllers
- Any other data transfer/storage system that requires increased capacity or bandwidth

2.0 DATA COMPRESSION/DECOMPRESSION ALGORITHM

The algorithm used in the chip is based on the widely known Lempel-Ziv algorithm. It has been enhanced at HP, and is called the DCLZ algorithm. It is more flexible and offers better performance than the Lempel-Ziv scheme.

2.1 ALGORITHM BASICS

The underlying principle behind data compression is to remove redundancy from the data. The DCLZ scheme performs this by recognizing and encoding patterns of input characters. Each time that a unique string of input characters occurs, it is entered into a "dictionary" and assigned a numeric value. Once a dictionary entry exists, subsequent occurrences of that entry within the data stream can be replaced by its numeric value or "code word".

Each dictionary entry consists of two items: (1) a unique string of data bytes which the algorithm has found within the data; (2) a code word that represents this combination of bytes. The dictionary can contain up to 4096 logical entries and is organized in the following manner: the first eight entries are reserved code words which are used to flag specific conditions (e.g. a dictionary reset or an End of Record (EOR) condition); the next 256 entries contain the byte values 0 through 255; and the remaining locations can contain entries that are pointers to other locations that eventually terminate by pointing at one of the byte values 0 through 255. In this way, the possible byte combinations can grow from 2 bytes to 128 bytes without requiring an excessively large external RAM to store the actual byte strings.

This dictionary is built and stored logically in a bank of external RAM that may be either 8K, 10K, or 16K by 22-bits. Each logical RAM address contains the byte value or pointer in the lower 8 bits, the code word representing the entry in the next 12 bits, and three condition flags in the upper three bits. This information is physically stored in eight bit wide static RAM chips. There are three dictionary sizes supported: 8K entries (24K bytes of physical external RAM), 10K entries (30K bytes of physical external RAM), and 16K entries (48K bytes of physical external RAM). The average throughput rate of the chip is increased with a larger dictionary size.

The dictionary code words range in length from 9 bits to 12 bits and correspond to dictionary entries that range from 0 to 4095. Thus, the first 512 entries possess 9-bit code words, the next 512 entries possess 10-bit code words, the next 1024 entries possess 11-bit code words, and the final 2048 entries possess 12-bit code words. The actual RAM address for each dictionary entry is determined by a hashing function performed on the entry value. More than 4K of RAM is needed for occasions when a dictionary collision occurs. When this happens, the two colliding values must be re-hashed to two new locations and the original location is flagged as a collision site.

Another important property of the algorithm is the coupling between compression and decompression. They are tied together through two operations: (1) the compression and decompression processes; (2) the packing and unpacking of code words into a byte stream. The very nature of the compression algorithm requires that the compression process and the decompression process be synchronized. Stated differently, decompression cannot begin at an arbitrary point in the compressed data. It begins at the point where the dictionary is known to be empty or "reset". This allows for one of the fundamental advantages of the DCLZ algorithm, which is the dictionary is embedded in the code words thus saving time and space of having to send it along with the data separately. Similarly, the packing and unpacking process must be synchronized. This implies that compressed data must be presented to the algorithm in the proper order.

Data is divided into records. An End of Record is used to denote the division between records. The compression and decompression processes include internal buffering of data. Therefore, it is necessary that the chip be on a record boundary to ensure that the compression and decompression process is complete for a given record. The chip has several programmable features to handle record boundary conditions.

The underlying method used by all data compression schemes is to remove redundancy from data. Truly random data contains no redundancy. As a result, data compression methods cannot be effectively performed on random data. As is the case for all data compression algorithms, the DCLZ algorithm will actually expand random data. This means that it will output more data than the original data stream contained. Fortunately, in almost all real world applications truly random data does not exist. But as data compression becomes more widely accepted and used, much more compressed data will exist. Compressed data is essentially random data, since the act of compressing it has removed all redundancy. Thus, it is not possible to compress previously compressed data and obtain even higher compression results. The second compression will exhibit expansion.

2.2 COMPRESSION RATIO RESULTS

The compression ratio is defined as the size of the uncompressed input data, divided by the size of the compressed output data. Table 1 gives a summary of the compression ratios with various benchmark data.

Table 1: Summary of Data Compression Benchmark Results

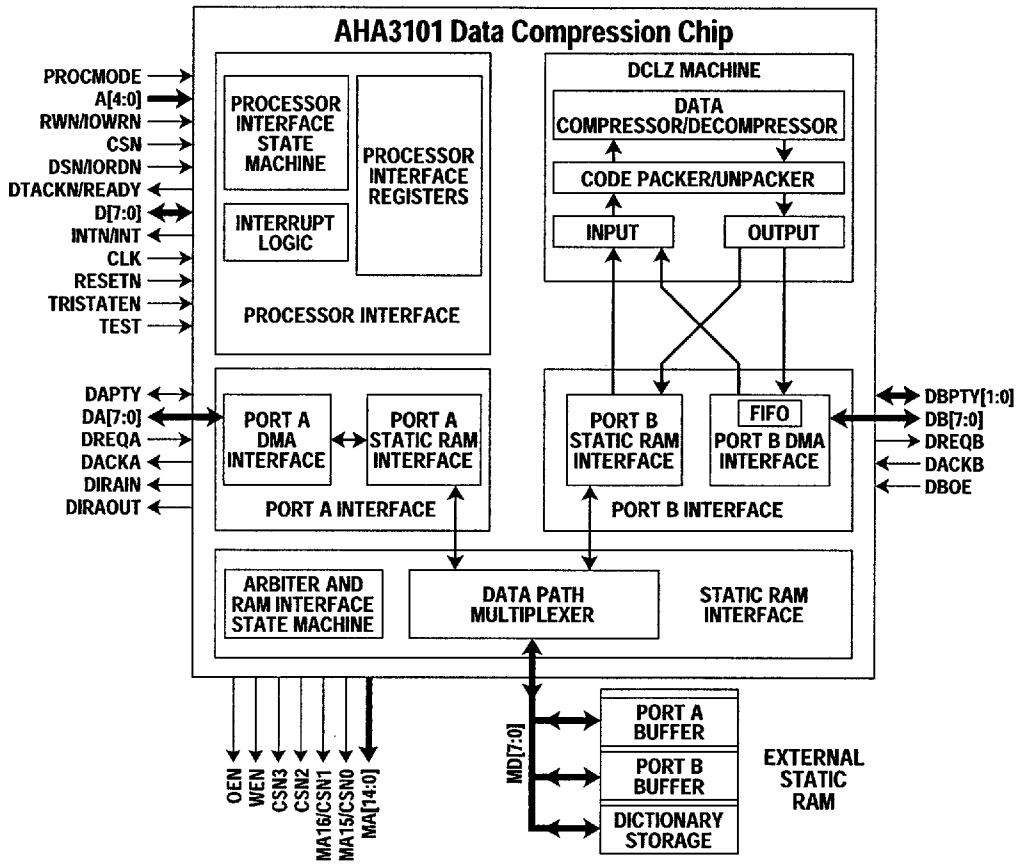
DATA DESCRIPTION	*VOLUME (MBytes)*	*COMPRESSION RATIO*
MPE/MPE-XL on HP3000s		
Series 68 (HPDESK)	528	3.93
Series 68 (DATA BASE)	2924	4.31
Series 68 (MISC)	1559	4.30
Series 70 (MANUFACTURING)	2924	4.31
Series 930 (CODE)	311	3.44
HP-UX on HP9000s		
Series 800 (COMMERCIAL HP-UX)	226	2.06
Series 500 (CODE)	363	2.38
Series 500 (DATA BASE)	336	4.07
Series 500 (VLSI)	785	2.52
Series 300 (ARCHIVE)	329	2.30
DEC		
VAX (CODE)	423	2.31
PASCAL WORKSTATION		
Series 200 (MISC)	467	2.47
AMDAHL		
(CORPORATE DATA)	5000	3.79

3.0 FUNCTIONAL BLOCK DESCRIPTION

The AHA3101 Data Compression chip contains the compression and decompression circuitry, which forms the core of a data compression system. This includes the circuitry to implement the DCLZ algorithm, and is called the DCLZ machine.

The AHA3101 has four main interfaces. The processor interface allows a microprocessor to access information inside the chip. There are two DMA ports, called Port A and Port B, which are used to transfer uncompressed and compressed data through the chip. External static RAM is used to store DCLZ algorithm information, and also to buffer data. The chip has a static RAM interface, which allows high performance access to the external static RAM.

Figure 1: Functional Block Diagram



3.1 PROCESSOR INTERFACE

There are two modes for the processor interface, which are controlled by the PROCMODE input pin (see Table 2). When PROCMODE is a high voltage, the processor interface is controlled by a data strobe (DSN), a read/write signal (RWN), with an open drain data transfer acknowledge output (DTACKN), and an open drain, low active interrupt (INTN). Connect PROCMODE to a low voltage to select a processor interface controlled by an I/O read strobe (IORDN), an I/O write strobe (IOWRN), with a high active ready output (READY), and a high active interrupt (INT).

The processor interface uses three control input signals to perform a hardware handshake with the processor. The CSN, DSN/IORDN, and RWN/IOWRN inputs provide the chip with the timing and control information to do read and write cycles with the processor. The five address inputs, A[4:0], decode which register is accessed. All processor data transfers take place on the eight bit data bus, D[7:0].

The processor interface is asynchronous to the chip. The processor interface control inputs are synchronized to the chip's clock, and then go to the processor interface state machine. This has a handshake output, called the DTACKN/READY pin. This is used to signal the processor that the chip has completed the data transfer.

The chip has an interrupt output, called the INTN/INT pin. The *Interrupt Status* register, *Interrupt Disable* register, and the *Interrupt Clear* register provide programmable interrupt handling.

The processor can read and write any byte in the external static RAM. The *Processor Buffer Address* register can be programmed with the target static RAM address. A processor read of the *Processor Buffer Data* register will read the byte in static RAM that the *Processor Buffer Address* register points to. A processor write to the *Processor Buffer Data* register will write the byte in static RAM that the *Processor Buffer Address* register points to. The *Processor Buffer Address* register increments the address after each access to the *Processor Buffer Data* register, allowing contiguous blocks of data to be transferred quickly.

3.2 DCLZ MACHINE

3.2.1 HARDWARE BLOCK DESCRIPTION

There are four main blocks in the DCLZ machine - the INPUT block, the CODE PACKER/UNPACKER block, the DATA COMPRESSOR/DECOMPRESSOR block, and the OUTPUT block.

In compression mode, the INPUT block receives uncompressed data bytes, and sends them to the CODE PACKER/UNPACKER block. The number of these bytes is counted by the Input Byte Counter. The uncompressed data bytes then flow transparently into the DATA COMPRESSOR/DECOMPRESSOR. This block implements the DCLZ algorithm. It compresses the data bytes into compressed data codewords. The CODE PACKER/UNPACKER block converts the compressed data codewords into compressed data bytes. The Output Byte Counter counts the number of compressed data bytes that flow out of the CODE PACKER/UNPACKER block and into the OUTPUT block. The OUTPUT block handshakes the compressed data bytes out of the DCLZ block.

During decompression, the INPUT block passes the compressed data bytes to the CODE PACKER/UNPACKER. The CODE PACKER/UNPACKER converts the compressed data bytes into compressed data codewords. The DATA COMPRESSOR/DECOMPRESSOR converts the compressed data codewords into uncompressed data bytes. The uncompressed data bytes then pass through the CODE PACKER/UNPACKER, and out through the OUTPUT block.

Table 2: Processor Pin Modes

PIN NAME	*PROCMODE = 1*	*PROCMODE = 0*
DSN/IORDN	DSN - low active data strobe	IORDN - low active read strobe
RWN/IOWRN	RWN - read/write signal	IOWRN - low active write strobe
DTACKN/READY	DTACKN - low active data transfer acknowledge	READY - high active ready signal
INTN/INT	INTN - low active interrupt	INT - high active interrupt

3.2.2 PROGRAMMER'S VIEW

3.2.2.1 RECORDS AND RECORD BOUNDARIES

The DCLZ machine uses the End of Record flag to separate the data stream it operates on into distinct logical entities called records. When reading data from the SRAM buffer in compression mode, the Record Length Counter determines when the end of a record occurs, and passes a flag to the DCLZ machine. If enabled, the DCLZ machine will insert an EOR codeword, delineating a record boundary in the compressed data stream. This is required in order to preserve the record boundaries, and is enabled by setting the WRITE EOR CODEWORD ON EOR bit in *Control 0* register. This is the default mode of operation.

When the DCLZ machine writes the EOR codeword, the internal pipeline of the DCLZ machine, including all Port B internal buffering, is emptied. This maintains a clean record boundary inside the AHA3101 chip.

When decompressing a data stream, the EOR codewords are recognized by the DCLZ machine, and the internal pipeline of the DCLZ machine and the Port B Static RAM Interface gets emptied. Note that the Port B DMA FIFO will continue to read bytes of the following record. Reading the FIFO Byte Count enables the user to determine how much data is in the FIFO.

3.2.2.2 PAUSING THE DCLZ MACHINE

The DCLZ machine and Port B interface can pause at the end of records by setting the PAUSE ON EOR bit in *Control 0* register. The processor can force a pause by setting the PAUSE REQUEST bit in *Control 1* register. Once the chip is paused, the processor can interrogate the status of AHA3101. This is summarized below. To allow the DCLZ machine to continue, the processor must clear the PAUSE REQUEST bit in *Control 1* register.

In compression mode, the interaction of the PAUSE ON EOR and WRITE EOR CODEWORD ON EOR control bits in *Control 0* register controls the functionality of the End of Record handling in the AHA3101 (see Table 3).

3.2.2.3 RESETTING THE DICTIONARY

The compression dictionary can be programmed to reset on end of record by setting the RESET ON EOR bit in *Control 0* register. The processor can force a reset by setting the RESET REQUEST bit in *Control 1* register. The compression optimization circuit can also reset the dictionary. A dictionary reset causes the Reset codeword to be written after the current compression match sequence has been completed. The dictionary is then re-initialized. Note that the compression match sequence may match up to 128 bytes, delaying the dictionary reset until the match completes.

When in decompression, resets are performed automatically whenever a reset codeword is detected. The reset codeword is always the first codeword of a compressed data transfer. The processor cannot force a reset to the decompressor.

3.2.2.4 BYTE COUNTS

The data compression ratio can be determined by using the Input Byte Count and the Output Byte Count. Note that when comparing input bytes and output bytes, these counts should be looked at when the chip is paused and the pipeline has been flushed. The pipeline in the DCLZ machine is variable in length and could lead to an erroneous comparison during a compression sequence.

Each byte read into the DCLZ Input block increments the Input Byte Count. Each byte written out of the Code Packer/Unpacker increments the Output Byte Count, so it is not accurate for compression ratio calculations until the DCLZ machine has written the EOR codeword and paused. These counters operate in compression and decompression modes.

Table 3: Compression Mode End of Record Handling

PAUSE ON EOR	*WRITE EOR CODEWORD ON EOR*	*AHA3101 OPERATION*
0	0	No special end of record handling occurs. No interrupts are generated. No EOR codeword is output. The Port B Static RAM interface, the DCLZ machine, and the Port B DMA Interface continue on compressing input data.
0	1	The EOR codeword is added to the record. The record is transferred out of chip. If enabled and the *Record Count* register is zero, the COMP END OF TRANSFER INTERRUPT is generated after the last byte of the record is transferred out of the chip. The chip then pauses. If the *Record Count* register is not zero, the Port B Static RAM interface, the DCLZ machine, and the Port B DMA Interface continue on compressing input data.
1	0	The DCLZ machine pauses on the last byte of the record. The Static RAM Interface is flushed. The DCLZ machine's internal pipeline contains data. No interrupt is generated. The PAUSED status bit gets set.
1	1	The EOR codeword is added to the record. The record is transferred out of chip. If enabled and the *Record Count* register is zero, the COMP END OF TRANSFER INTERRUPT is generated after the last byte of the record is transferred out. No data is in the Port B Static RAM interface, the DCLZ machine, or the Port B DMA interface. The Port B Static RAM interface, DCLZ machine, and the Port B DMA interfaces then pause. The PAUSED status bit gets set.

3.2.2.5 COMPRESSION RATIO OPTIMIZATION

A key feature of the AHA3101's compression algorithm is the ability to perform compression ratio optimization. The optimization circuit monitors the compression performance and resets the dictionary automatically if the compression ratio drops below a preset threshold. If the dictionary is less than half full the optimization circuits will check for expanding data every 1024 input bytes and reset the dictionary if expansion occurs. After the dictionary is greater than half full the optimizer will check the compression ratio every n bytes, where n is determined by the value of the PERIOD bits in *Control 2* register. The threshold is set by the value of the THRESHOLD bits in *Control 2* register.

Optimization is enabled by setting OPT ENABLE bit in *Control 0* register.

3.3 DMA INTERFACES

There are two bidirectional DMA ports, called Port A and Port B. The two DMA ports each have a set of control and status registers. Port A is completely independent from Port B.

The DMA ports normally transfer data between the port's pins, and the external static RAM. Each port is split into two sections - a DMA interface, and a static RAM interface. The DMA interface contains the state machine that controls the port's hardware DMA handshake at the pins of the chip. The static RAM interface contains the control circuitry that interfaces with the STATIC RAM INTERFACE block. It also contains the registers that point to the target address in the external static RAM memory.

Each port defines a data buffer area in the external static RAM, with the use of two pointers. For Port A, the *Port A Buffer Address* register is the pointer that defines the beginning of the Port A data buffer area. The *Port A Buffer Stop Address* register is programmed with the address that is one byte above the last byte in the buffer. The *Port A Buffer Address* register is incremented after each byte is transferred through the Port A STATIC RAM INTERFACE block. This value is then compared to the Port A Buffer Stop Address. If these are equal, then the buffer transfer has completed.

These registers address the entire static RAM address space. Therefore, the data buffer location in static RAM memory and the buffer size are variable. Port B has an identical *Port B Buffer Address* register, and a *Port B Buffer Stop Address* register.

In an in-line application, Port A is typically used to transfer data between the host bus interface circuit, and the external static RAM data buffer. Port B is typically used to transfer data between the static RAM buffer, the DCLZ machine, and the system data buffer. Port B is the master of the Port B data bus, DB[7:0]. Port A is the slave of the Port A data bus, DA[7:0]. This allows the AHA3101

Data Compression chip to be easily inserted into an existing in-line architecture, which does not have data compression.

3.3.1 PORT A INTERFACE

The Port A interface is a high performance data link between the Port A data bus, and the external static RAM. It supports a 5 Mbytes/sec sustained data rate with the Port A data buffer area in the external static RAM. Note that Port A typically transfers uncompressed data.

Port A DMA is controlled by the *Port A DMA Control* register. The DIR bit determines the direction of DMA transfer - in from the DA[7:0] bus out to the external static RAM, or in from the external static RAM and out to the DA[7:0] bus. The GO bit is used to begin and abort DMA transfers. Also, the DREQA POLARITY bit controls the polarity of the DREQA pin, and the DIRA POLARITY bit controls the polarity of the DIRAIN and DIRAOUT pins. The DAPTY pin can be programmed to generate and check parity on the DA[7:0] bus.

Port A DMA transfers are complete after Port A transfers a byte and the *Port A Buffer Address* register increments and is then equal to the *Port A Buffer Stop Address* register. The processor can be programmed to interrupt when this event occurs.

3.3.2 PORT B INTERFACE

The Port B static RAM interface transfers data between the Port B data buffer in the external static RAM, and the DCLZ machine. This data is typically uncompressed. The Port B DMA interface transfers data between the DCLZ machine, and the Port B data bus. This is typically compressed data. Since Port B interfaces with the DCLZ machine, the direction of data transfer through Port B is controlled by the DCLZ machine's COMP DECOMP bit. During compression, when COMP DECOMP is a one, data flows from the external static RAM, through the Port B static RAM interface, and into the DCLZ machine. There, the data is compressed, and sent into the eight byte Port B FIFO. Data then flows from the FIFO, through the Port B DMA interface, and out the Port B data bus. The direction of data flow is reversed in decompression mode.

The Port B static RAM block contains the *Record Length* register, and the *Record Count* register. These are used during compression, to allow the chip to automatically compress multiple, fixed sized records.

Since Port B transfers data for the DCLZ machine, the RESET DCLZ bit in *Control* register 0 is used to reset the Port B Static RAM Interface and the Port B DMA Interface. Also, the PAUSE REQUEST bit in *Control 1* register is used to pause the Port B Static RAM Interface, the DCLZ machine, and the Port B DMA Interface. This allows Port B registers to be accessed in the middle of compression and decompression operations.

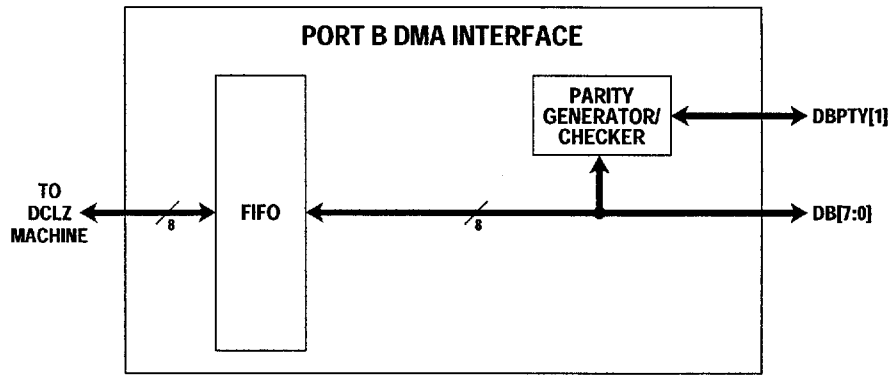
The Port B DMA Interface state machine transfers data between the Port B FIFO and the Port B DMA Interface pins. The state machine attempts to transfer data when the FIFO is ready. When the PAUSE REQUEST bit is programmed to go to its active state, the Port B DMA Interface may be requesting a DMA transfer. The Port B DMA Interface will pause, and the FIFO BYTE COUNT will be stable at its previous value. The DMA cycle is allowed to complete when the Port B DMA Interface is paused. If it does, the FIFO BYTE COUNT will still remain in its previous state. Once the PAUSE REQUEST is removed, the Port B DMA Interface will recognize that the DMA cycle has been completed, and will then update the FIFO BYTE COUNT.

The Port B data transfers between the FIFO and the Port B data bus can be eight or sixteen bit transfers. This is controlled by the DMA SIZE bit in Port B DMA *Control 1* register. The two modes are shown in Figure 2.

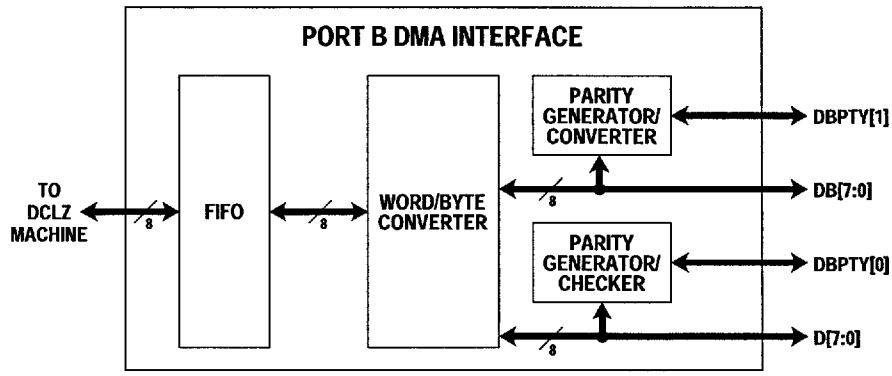
In eight bit mode, data is transferred on the DB[7:0] bus, and parity on the DBPTY[1] pin. During transfers out of Port B, the output drivers are enabled only when the DBOE input pin is active. During transfers into Port B, data is internally latched on the rising edge of DACKB.

In sixteen bit mode, data is transferred on DB[7:0] and on the processor data bus, D[7:0]. This mode features a shared bus for processor read and write cycles, and for Port B DMA cycles. Parity for DB[7:0] is on the DBPTY[1] pin. Parity for D[7:0] is on the DBPTY[0] pin. The DREQB output goes active to begin a DMA cycle. The DMA acknowledge input, DACKB, then goes active to denote a DMA cycle is in process.

Figure 2: Port B DMA Transfer Sizes



PORT B DMA 8 BIT MODE



PORT B DMA 16 BIT MODE

In sixteen bit mode, the actual data transfer is accomplished using the processor interface control inputs, in the same manner as for processor read and write cycles. A DMA cycle occurs when the DACKB pin is a low voltage (active). The chip select pin, CSN, must be high voltage (inactive) during 16 bit Port B DMA cycles. The function of the PROCMODE pin selects which signals control the timing of the DMA data transfer. When PROCMODE is low voltage, the RWN/IOWRN pin is a low active I/O write strobe (IOWRN), and DSN/IORDN is a low active I/O read strobe (IORDN). During transfers into Port B, data is latched inside the chip on the rising edge of RWN/IOWRN. During data transfers into Port B, the output drivers on D[7:0], DB[7:0], and DBPTY[1:0] are enabled when DSN/IORDN is at low voltage. When PROCMODE is high voltage, the RWN/IOWRN pin is read (high voltage) - write (low voltage) signal, and DSN/IORDN is a low active data strobe (DSN). During transfers into Port B, data is latched on the rising edge of DSN/IORDN when RWN/IOWRN is low voltage. During data transfers out of Port B, the output drivers on D[7:0], DB[7:0], and DBPTY[1:0] are enabled when DSN is low voltage and RWN is high voltage.

The WORD bit in *Port B DMA Control 1* register controls which data bus transfers the first byte of data. During the first DMA cycle when WORD is a one, the first byte is transferred on D[7:0], and the second byte is transferred on DB[7:0]. On the next DMA cycle, the third byte is transferred on D[7:0], and the fourth byte on DB[7:0].

During the first DMA cycle when WORD is a zero, the first byte is transferred on DB[7:0], and the data on D[7:0] is ignored. On the next DMA cycle, the second byte is transferred on D[7:0], and the third byte on DB[7:0].

This section describes End of Record handling in compression mode and compression passthru mode, when data is transferred out of Port B in 16 bit mode. When the last byte of the record occurs on a word boundary, a DMA cycle always occurs. The second to the last byte of the record is transferred out of Port B on D[7:0] and the last byte of the record is transferred on DB[7:0].

When the last byte of the record occurs on an odd byte boundary (i.e., when the last byte of the record is the first byte in a sequence to output a word) there are several special cases which determine the chip's behavior. If the *Record Count*

register is zero (i.e., the last byte of the record is the last byte of a compression transfer of multiple, fixed length records), the byte is always transferred out on D[7:0] and the data on DB[7:0] is undefined. When the *Record Count* register is non-zero and the last byte of a record occurs on an odd byte boundary, the HOLD BYTE EOR bit in *Port B DMA Control 1* register controls the operation of the chip. When HOLD BYTE EOR is zero, a Port B DMA cycle occurs with the last byte of the record transferred on D[7:0] and undefined data on DB[7:0]. When HOLD BYTE EOR is one, the last byte of the record is held in the Port B FIFO. The BYTE EOR HELD bit in the *Port B DMA Status* register is set. The chip then begins processing the next record. After the first byte of the next record enters the Port B FIFO, a DMA cycle occurs. The last byte of the previous record is then transferred on D[7:0] and the first byte of the next record is transferred on DB[7:0]. The BYTE EOR HELD status bit is then cleared. This mode allows multiple records to be compressed, with a contiguous stream of valid data flowing out of Port B.

During DMA transfers of compressed data out of Port B, the number of valid data bytes on the last transfer can be calculated from two items - the state of the WORD bit and the Output Byte Counter value. The WORD bit tells if the first DMA cycle contained one or two valid bytes. The Output Byte Counter tells the number of bytes transferred. Thus, the number of valid bytes in the last DMA cycle can be calculated.

During DMA transfers in to Port B, both bytes of the last DMA cycle are always written into the FIFO. The DCLZ machine should be programmed to pause on End of Record. When End of Record is reached, the DCLZ machine will pause. The FIFO will contain only the last extra byte, which does not contain valid data. At this point, the CLEAR FIFO bit should be set, to initialize the FIFO byte counter, which purges the byte from the FIFO.

When parity is enabled, it is always checked and generated on both the D[7:0] bus and the DB[7:0] bus. This occurs even when only one valid byte of data is being transferred at the beginning or the end of a transfer.

The processor can be programmed to interrupt when the *Port B Buffer Address* register matches the *Port B Buffer Stop Address* register. During compression, this signals that the Port B static RAM interface is out of data to compress. On decompression, it signals that the Port B static RAM interface has filled the Port B data buffer in the external static RAM.

3.4 STATIC RAM INTERFACE

There are several configuration options for the static RAM external to the AHA3101 chip. The hardware for the static RAM, and the software usage of the static RAM can be tailored to meet specific system requirements.

3.4.1 HARDWARE INTERFACE

The hardware supports up to 128K bytes of external static RAM. Two types of static RAM are supported, as selected by the RAMTYPE bit in the *Configuration Control* register. One to four 32K by 8 static RAM chips can interface directly to the AHA3101, without any additional components. Alternatively, one 128K by 8 static RAM chip can replace the four 32K by 8 RAM chips.

The AHA3101 supports both sizes of memory chips on one PC board layout. Twenty eight pins on the 128K static RAM chip are the same as for the twenty eight pin 32K static RAM chip. The thirty two pin 128K static RAM chip can be placed over either location for the 32K static RAM chips connected to CSN3 or CSN2. Since CSN3 and CSN2 are low voltage when 128K mode is selected, the 128K static RAM low active chip select is active, as required.

The average compression throughput is effected by the bus bandwidth to the static RAM chips. The RAM WAIT STATE bit in the *Configuration Control* register determines whether zero or one wait state is used in read and write cycles to the static RAM chips. This allows faster, more expensive static RAM chips to be used in higher performance applications.

The following diagrams show how to interface the two sizes of static RAM to the chip.

Figure 3: 128K x 8 Static RAM Interface Diagram

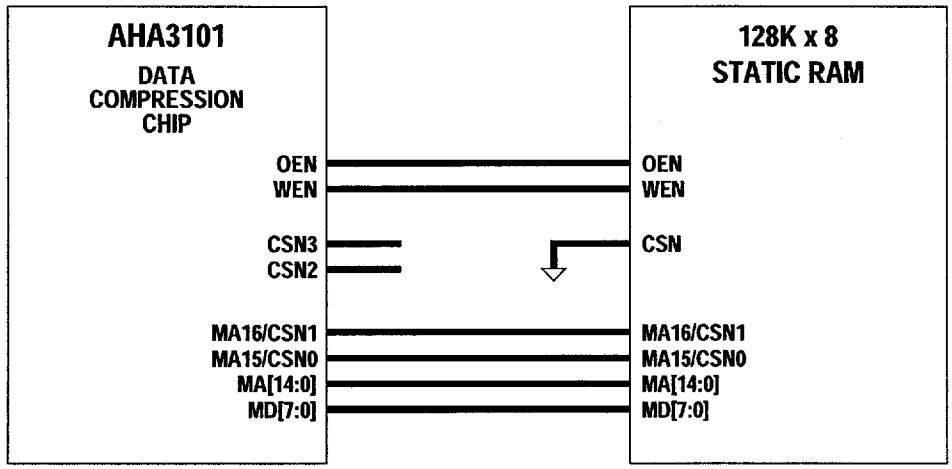
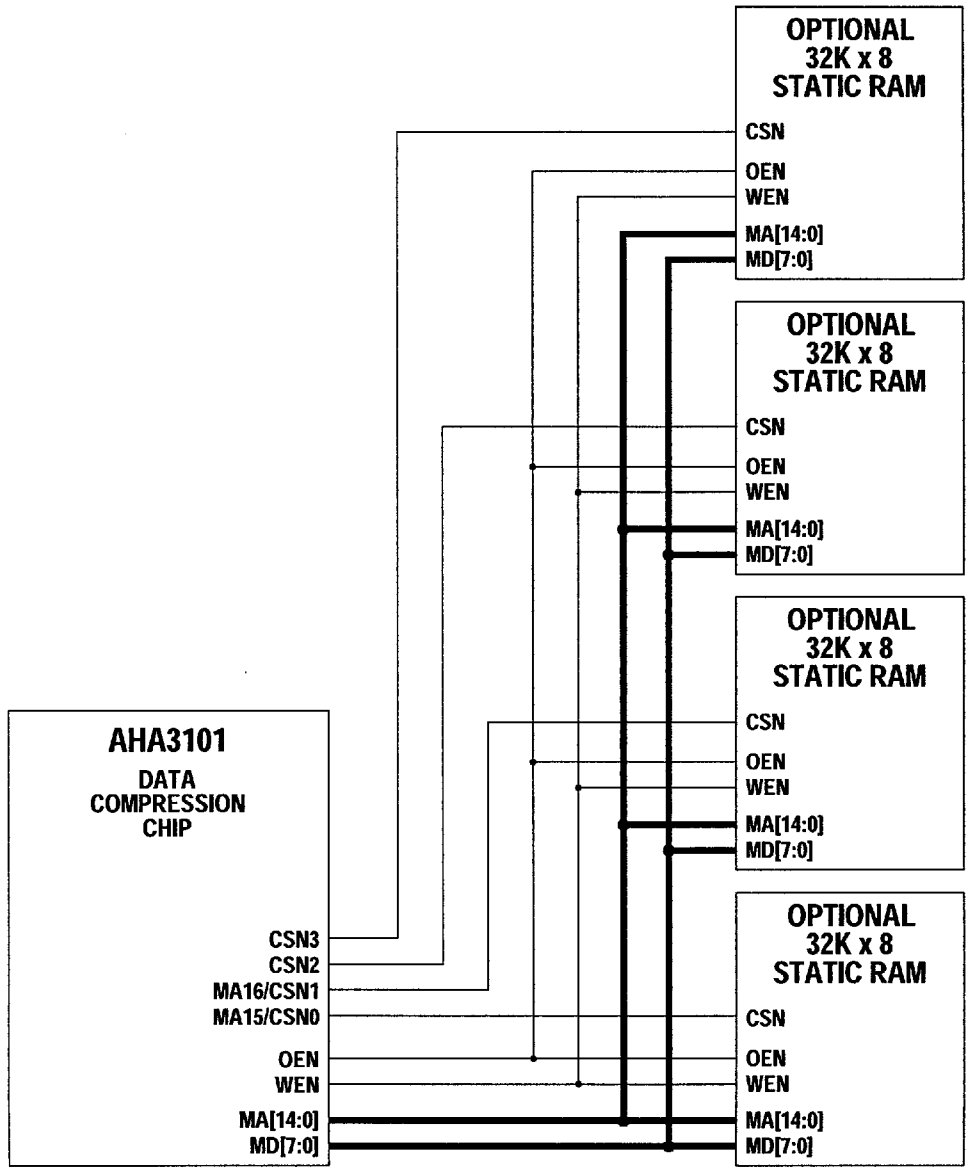


Figure 4: 32K x 8 Static RAM Interface Diagram



3.4.2 MEMORY AREA PARTITIONS

The static RAM is divided into two main areas. The dictionary storage area is used exclusively by the compression and decompression algorithms. The remaining area is available for data buffer usage.

Each compression - decompression dictionary entry location uses three bytes of static RAM. The DICT SIZE bits in *Control 3* register can be programmed to select the size of the dictionary, as shown in Table 4.

One external 32K by 8 static RAM chip can support 8K dictionary entry locations (24K bytes) with 8K bytes of available data buffer storage, or 10K dictionary entries (30K bytes) with 2K bytes of available data buffer storage. Two external 32K by 8 static RAM chips can support 16K dictionary entry locations (48K bytes), with 16K bytes available for data buffer storage. The additional memory gained by adding more 32K by 8 RAM chips is dedicated to additional data buffer storage.

3.4.3 DATA BUFFER OPERATION

There are three access ports into the data buffer storage area in the static RAM. There are two bidirectional DMA ports, called Port A and Port B. There is also an access port for the processor, to read and write any byte in the buffer.

The two DMA ports each have a set of registers to control and monitor their DMA operations. Data transfer operations use an address pointer into the static RAM. For Port A, this is called the *Port A Buffer Address* register. A buffer area begins at the initial programmed address. The buffer address pointers auto-increment, after each byte has been transferred. Thus, data buffers exist in memory address space above the initial address. Each port has a stop address pointer. For Port A, this is called the Port A Buffer Stop Address. It is programmed with the address that is one byte above the last byte in the buffer. This value is then compared to the incremented Port A Buffer Address. If these are equal, then the buffer transfer has completed.

The buffer size is variable, and is limited only by the amount of static RAM used. Also, there are several interrupt conditions that can be used for data buffer management and control. All the above mentioned registers are further described in Section 6.0 *Programmer's Overview*.

The Port A and Port B data buffer areas must never overlap in the static RAM address space. The two sets of buffers operate as a set of swing buffers. One port must first completely fill a buffer area. Once this operation is complete, the other port can be programmed to access the data in the original buffer area.

The access port for the processor has a static RAM address pointer stored in the *Processor Buffer Address* register. The processor can read and write bytes into the static RAM by reading or writing the *Processor Buffer Data* register. After each access to the *Processor Buffer Data* register, the *Processor Buffer Address* register contents are incremented, to point to the next byte in the static RAM address space.

Table 4: Memory Area Partitions

DICT SIZE 1	DICT SIZE 0	DICTIONARY ENTRIES	RAM BYTES	DICTIONARY ADDRESS
0	0	8K	24K	05FFF-00000 hex
0	1	16K	48K	0BFFF-00000 hex
1	X	10K	30K	077FF-00000 hex

3.4.4 PROGRAMMER'S CAVEATS

The flexibility of the software interface to the static RAM data buffer places some responsibility on the programmer to ensure data integrity.

The static RAM interface hardware does not have information on the amount of static RAM that exists in each application. It is the programmer's responsibility not to access non-existent memory addresses. For example, if only 32K bytes static RAM are used, the programmer should not access addresses from 08000 hex to 1FFFF hex.

Since Port A DMA operation is independent from Port B, there is no tracking of pointers between the two ports. The programmer must ensure that the data buffers defined for the two ports do not overlap when both DMA operations are active.

The *Port A Buffer Address* register, the *Port B Buffer Address* register, and the *Processor Buffer Address* register all wrap around from 1FFFF hex to 00000 hex, when they auto-increment. The programmer should be careful to keep the pointers from wrapping around, and then accessing the dictionary storage area in the bottom of the memory address space.

The *Port A* and *Port B Buffer Address* registers increment after each byte is transferred. The incremented value is compared to the *Buffer Stop Address* register. Remember to program the Buffer Stop Address one byte above the last byte in a buffer storage area.

The data buffer pointers can address the full 128K byte static RAM address space. This includes the area reserved for compression and decompression dictionary storage. The Port A, Port B, and *Processor Buffer Address* register pointers should never address the area reserved for dictionary entry storage. Alterations to the dictionary storage area will cause failures in the compression and decompression algorithms, and corrupt the data being processed.

4.0 TYPICAL MODES OF OPERATION

The following section is a set of typical operating modes in which AHA3101 Data Compression chip may be used. There are constraints regarding the times that modes may be switched. Most of the modes should only be switched to another mode when the chip is paused and flushed of data at an End of Record boundary. Switching from compression to decompression can only take place if RESET DCLZ bit is set (active), when the Port B Static RAM Interface, the DCLZ machine, and the Port B DMA Interface are reset.

4.1 COMPRESSION DATA FLOW

The processor first sets up a Port A data buffer area in the external static RAM, by writing to the *Port A Buffer Address* register, and the *Port A Buffer Stop Address* register. It then writes to the *Port A DMA Control* register, to begin the DMA transfer. Data comes into the chip through the Port A DMA interface, passes through the Port A static RAM interface, and is stored in a Port A buffer area in the external static RAM. Once the Port A buffer is full, Port A DMA activity halts, and the processor is interrupted by the Port A Pointer Interrupt.

To begin compression of multiple records, the processor should first write a one to the RESET DCLZ bit in *Control 0* register. This resets the Port B Static RAM interface, the DCLZ machine, and the Port B DMA interface. The processor then programs the *Record Length* register, and the *Record Count* register, to determine the total transfer size of uncompressed data.

The processor then sets the Port B Buffer Address and Port B Buffer Stop Address to point to the original Port A buffer memory area. This allows the buffer to swing under control of Port B. The Port A Buffer Address and Port A Buffer Stop Address can now be set up to point to an unused memory area, to begin another transfer of more data.

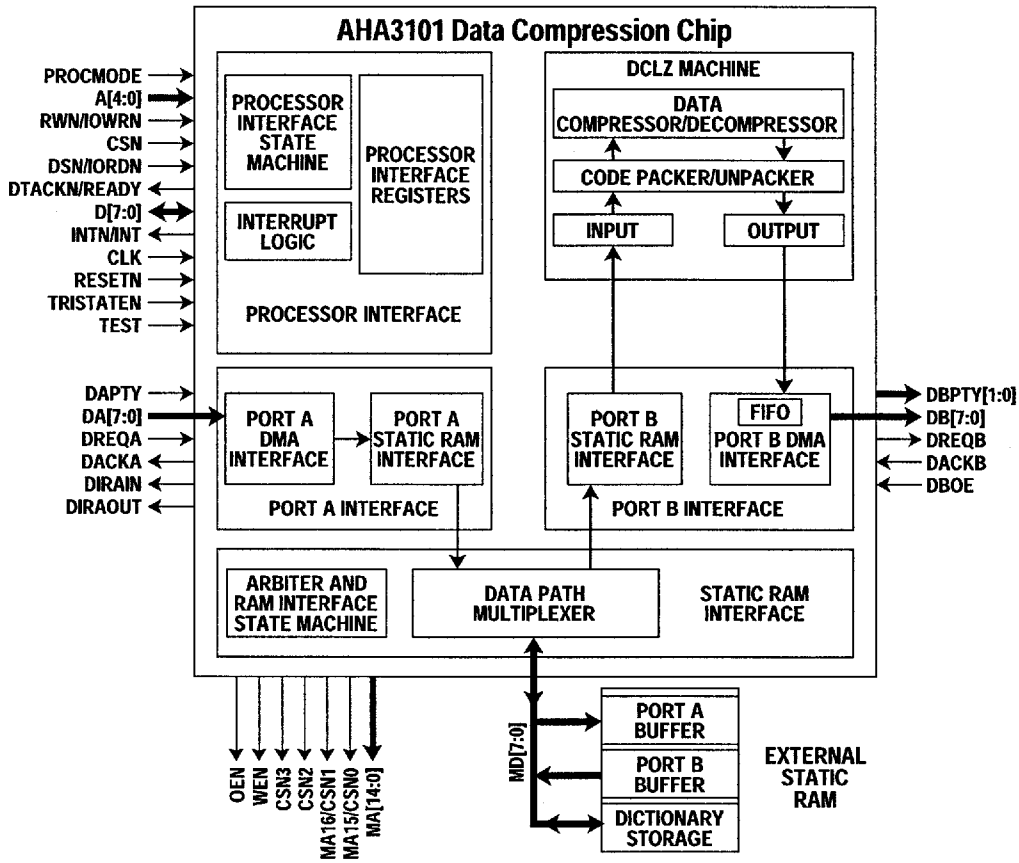
The processor should then initialize *Control 1* register as follows: DCLZ ENABLE = 1; PASSTHRU ENABLE = 0; DCLZ OUTPUT ENABLE = 1. *Control 0* register should then be initialized as follows: RESET DCLZ = 0; COMP DECOMP = 1; WRITE EOR CODE WORD ON EOR = 1; PAUSE ON EOR = 0. The WRITE EOR CODE WORD ON EOR bit must be set to one, to get compression end of transfer interrupts.

Compression begins when a zero is written to the RESET DCLZ bit in *Control 0* register. The Port B static RAM interface fetches data from the static RAM, and sends it to the DCLZ machine. The data is compressed, and sent out of the chip through the Port B DMA interface.

The *Record Length* register determines the number of uncompressed bytes in a record. It sends an End of Record (EOR) signal into the DCLZ machine. The DCLZ machine inserts EOR codewords in the appropriate place in the compressed data stream. The *Record Count* register determines the number of records in a compression transfer. The compression process completes after the number of records programmed in the *Record Count* register have been compressed. After this occurs, the Compression End of Transfer (COMP END OF TRANSFER) interrupt is generated. The Port B Static RAM Interface, the DCLZ machine, and the Port B DMA Interface then pause.

The Port B SRAM data buffer size is independent of the number of bytes in a record, and the number of records to be compressed. When the data in the Port B SRAM data buffer has been compressed, the Port B Static RAM interface, the DCLZ machine, and the Port B DMA interface pause. The PAUSED bit in *Status 0* register gets set to a one. The processor is interrupted with the Port B Pointer Interrupt. The processor then programs the *Port B Buffer Address* register, and the *Port B Buffer Stop Address* register to set up a new Port B SRAM data buffer, which has been previously filled with data by Port A. The processor then writes a zero to the PAUSE REQUEST bit in *Control 1* register to allow the compression process to continue. The procedure of swinging buffers from Port A to Port B is repeated as many times as is necessary, until the compression end of transfer occurs.

Figure 5: Compression Data Flow



4.2 DECOMPRESSION DATA FLOW

To begin decompression operation, the processor should first write a one to the RESET DCLZ bit in *Control 0* register. This resets the Port B Static RAM Interface, the DCLZ machine, and the Port B DMA Interface.

The processor then sets up a Port B data buffer area in the external static RAM, by writing to the *Port B Buffer Address* register, and the *Port B Buffer Stop Address* register.

The processor should then initialize *Control 1* register as follows: DCLZ ENABLE = 1; PASSTHRU ENABLE = 0; DCLZ OUTPUT ENABLE = 1. *Control 0* register should then be initialized as follows: RESET DCLZ = 0; COMP DECOMP = 0; PAUSE ON EOR = 1.

Decompression begins when a zero is written to the RESET DCLZ bit in *Control 0* register. Data comes into the chip through the Port B DMA Interface, and is decompressed by the DCLZ machine. The decompressed data then passes through the Port B Static RAM Interface, and is stored in a Port B buffer area in the external static RAM.

Once the Port B SRAM buffer is full, the Port B Static RAM Interface, the DCLZ machine, and the Port B DMA Interface pause, and the PAUSED bit in *Status 0* register is set to one. The processor is interrupted with the Port B Pointer Interrupt.

The processor then sets the Port A Buffer Address and Port A Buffer Stop Address to point to

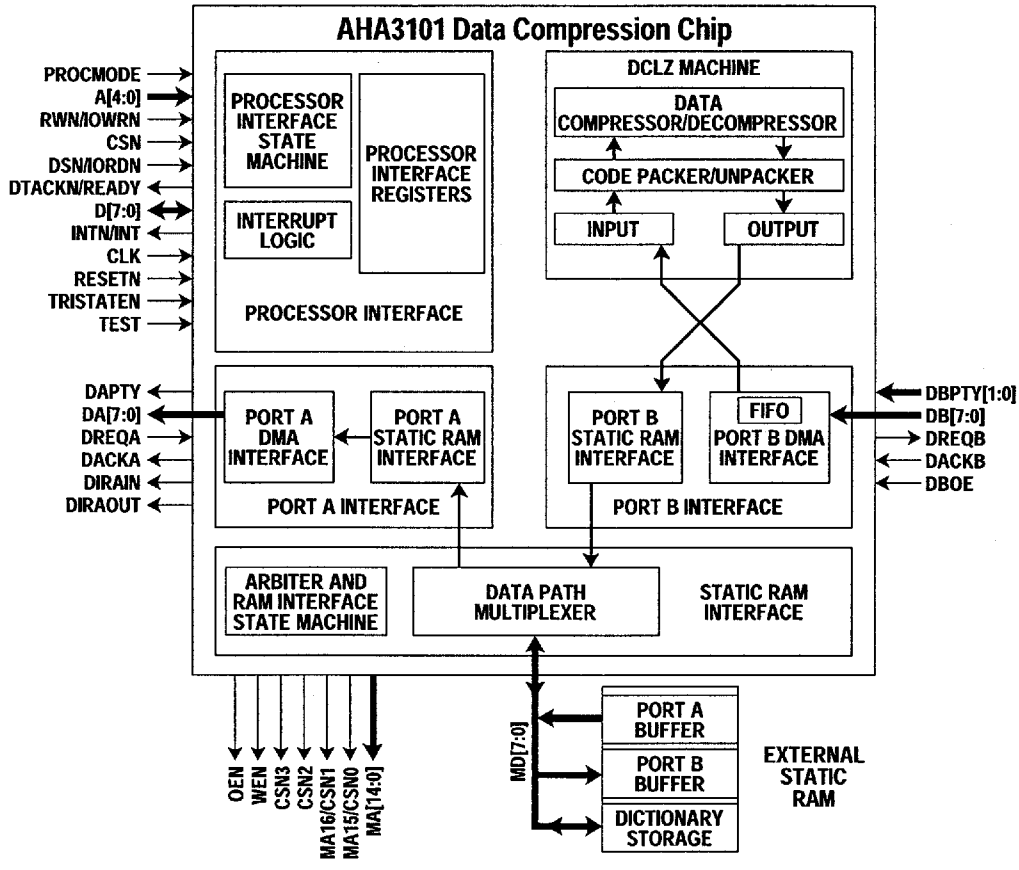
the original Port B buffer memory area. This allows the buffer to swing under control of Port A. The processor then sets the GO bit in the *Port A DMA Control* register to a one. This allows the Port A static RAM Interface to fetch data from the static RAM, send it through the Port A DMA Interface, and out of the chip on the Port A data bus, DA[7:0].

The Port B Buffer Address and Port B Buffer Stop Address can now be set up to point to an unused memory area, to begin another transfer of more data. The processor writes a zero to the PAUSE REQUEST bit in *Control 1* register, to allow the Port B Static RAM Interface, the DCLZ machine, and the Port B DMA Interface to continue.

When the DCLZ machine detects an End of Record (EOR) codeword in the compressed data stream, the Decompression End of Record (DECOMP EOR) interrupt is generated after the last byte of the record has been stored in the Port B SRAM data buffer. Since the PAUSE ON EOR bit in *Control 0* register was previously set, the DCLZ machine and the Port B Static RAM Interface will then pause, and wait for the processor. The Port B DMA Interface will continue to fill the Port B FIFO.

The procedure of swinging buffers from Port B to Port A is repeated as many times as is necessary, until the processor determines that the proper number of records have been decompressed.

Figure 6: Decompression Data Flow



4.3 PASS THROUGH MODE

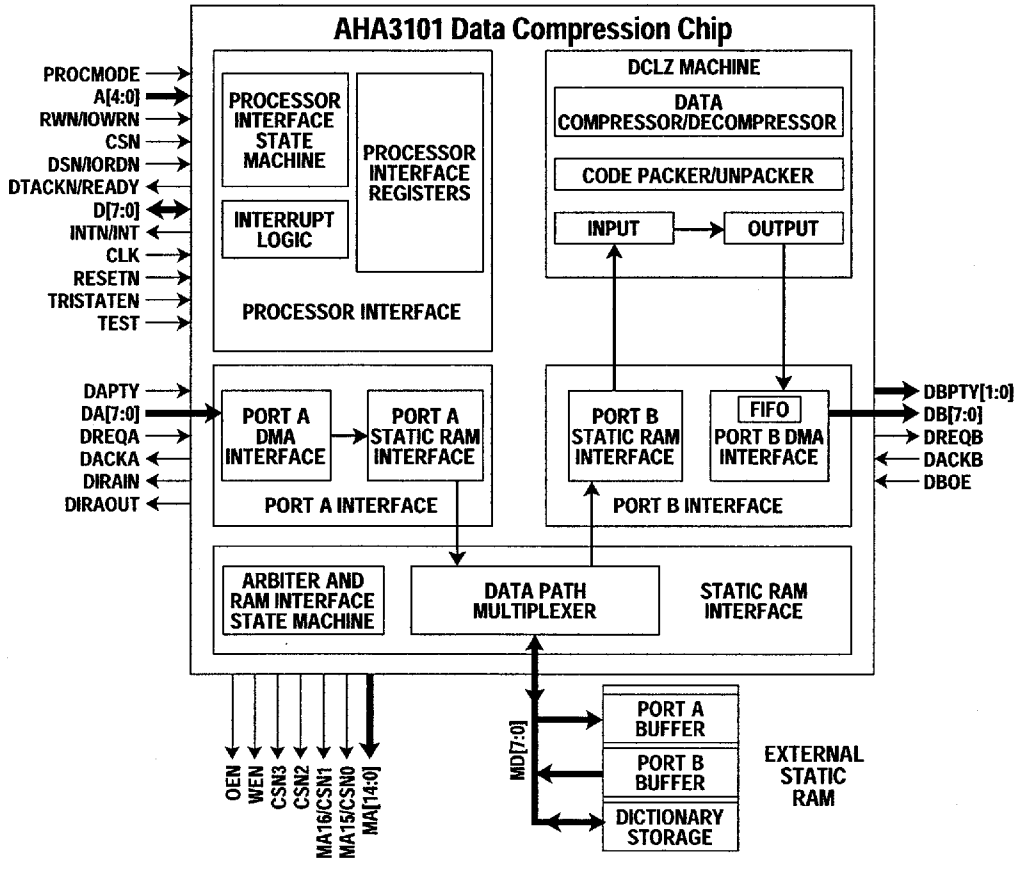
Pass through mode operates for compression and decompression (see Figure 7). In this case, data flows through the Port A interface and Port B interface as in normal compression mode. However, the DCLZ machine passes data from the INPUT block to the OUTPUT block. The CODE PACKER/UNPACKER and DATA COMPRESSOR/DECOMPRESSOR blocks are idle. This is useful when the chip is used in an in-line application, to allow data to pass through the data compression system unaltered.

The register programming sequence for pass through and compression modes is the same as for normal compression mode. The static RAM swing buffer operation for Port A and Port B, the *Record Length* register, the *Record Count* register, and the Compression End of Transfer interrupt operate as for normal compression mode.

For compression and pass through mode, *Control 1* register should be programmed to the following state: DCLZ ENABLE = 0; PASSTHRU ENABLE = 1; DCLZ OUTPUT ENABLE = 1. *Control 0* register should be programmed to the following state: COMP DECOMP = 1; WRITE EOR CODE WORD ON EOR = 1. The WRITE EOR CODE WORD ON EOR bit must be set to one, to get compression end of transfer interrupts.

Pass through mode also operates in the decompression direction. Because the data is not decompressed, end of record codewords in the data stream are not recognized. Therefore, Decompression End of Record Interrupts are not supported. When the static RAM buffer is enabled, the Port B Pointer Interrupt can be used to control data flow. The *Port B Buffer Address* register and *Port B Buffer Stop Address* register will limit the data transfer, and then interrupt the processor.

Figure 7: Pass Through Mode



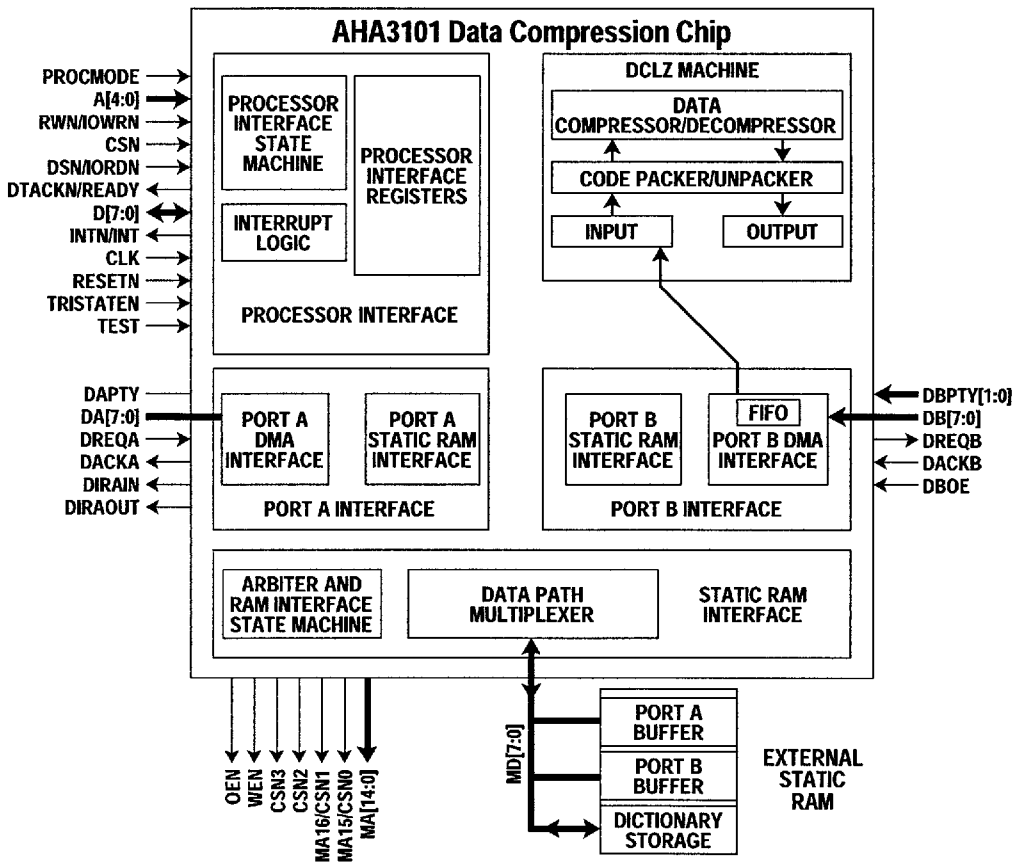
4.4 OUTPUT DISABLED MODE

Output disabled mode operates for decompression mode only. This is useful in decompression mode, when multiple records share a common dictionary. To recover data that occurs down stream from a dictionary reset boundary, decompression is begun (with the output disabled) at the dictionary reset boundary so that the dictionary is built correctly. However, data is not transferred out of the chip. When the processor determines that the desired record is reached, the output disabled mode is turned off and the data is then output.

The figure "Output Disabled Mode" shows the data flow for decompression. Data flows the same as in normal decompression mode, except that data does not pass from the OUTPUT block of the DCLZ machine, through the Port B static RAM interface. The Port A interface is idle. For decompression and output disabled modes, *Control 1* register should be programmed to the following state: DCLZ ENABLE = 1; PASSTHRU ENABLE = 0; DCLZ OUTPUT ENABLE = 0. *Control 1* register should be programmed as follows: COMP DECOMP = 0; PAUSE ON EOR = 1.

The DCLZ machine, and the Port B Static RAM Interface will pause on every End of Record boundary in the compressed data stream, and the processor will be interrupted with the Decompression End of Record (DECOMP EOR) interrupt. When the chip is paused at the End of Record boundary for the record proceeding the desired record, the processor should write a one to the DCLZ OUTPUT ENABLE bit, and a zero to the PAUSE REQUEST bit in *Control 1* register. This allows the record to be decompressed and the data to be stored in the Port B SRAM data buffer.

Figure 8: Output Disabled Mode

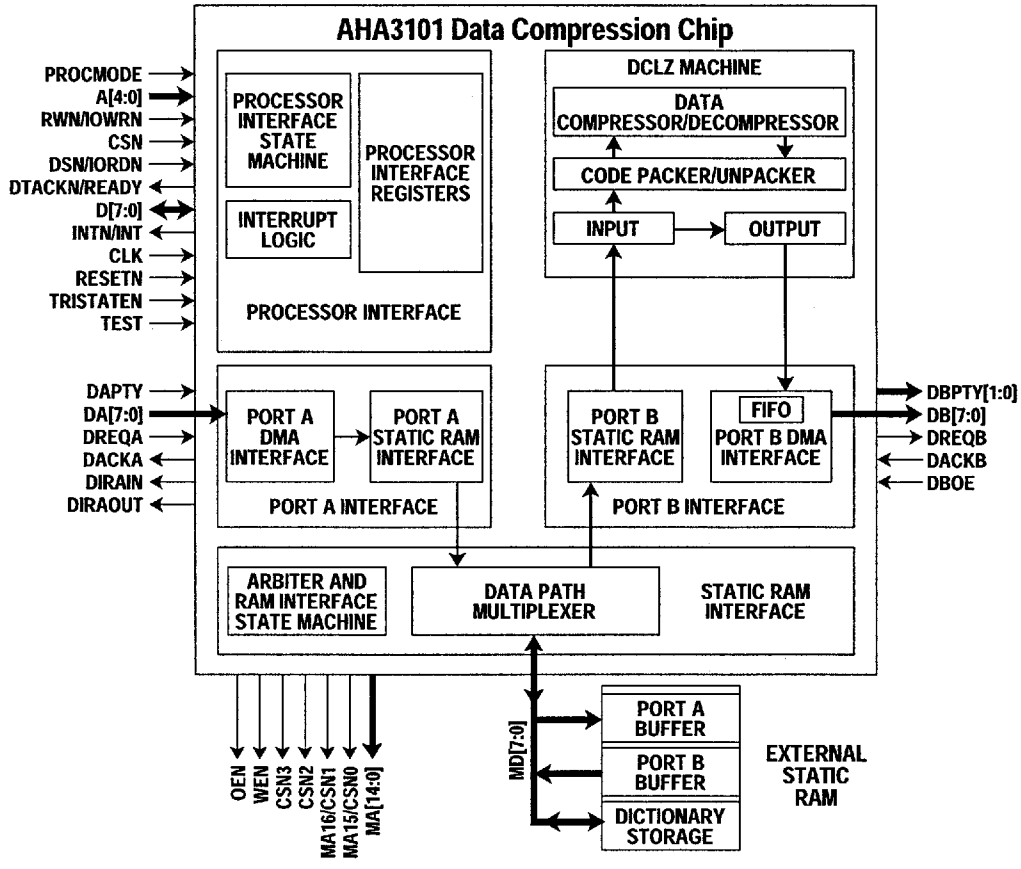


4.5 OBSERVATION MODE

Observation mode is supported in compression mode only. Figure 9 shows the flow of data through the chip when it is set up for compression. In observation mode, the chip will pass raw data from input to output (as in pass through mode), but will try to compress the data. This mode is useful for implementing simple expansion protection schemes. By first passing a record, or a number of records of data through the chip while in observation mode, the compression ratio that can be realized can be calculated. After the records have passed through the chip, the Input Byte Counter and Output Byte Counter will contain valid counts and can be compared to determine the compression ratio for that particular block of data. If the compression ratio is less than one (i.e., the Output Byte Count is greater than the Input Byte Count), the user may decide to turn off compression for that particular block of data.

For observation and compression modes, *Control 1* register should be programmed to the following state: DCLZ ENABLE = 1; PASSTHRU ENABLE = 1; DCLZ OUTPUT ENABLE = 1. *Control 0* register should be programmed to the following: COMP DECOMP = 1; COUNTER ENABLE = 1.

Figure 9: Observation Mode



5.0 TYPICAL SYSTEM APPLICATIONS

One typical application is to add data compression between a host interface bus and a tape drive in a computer peripheral. The AHA3101 Data Compression chip supports this in two methods, each with a different architecture. These architectures are known as in-line and look-aside.

5.1 IN-LINE APPLICATION

In the in-line application, the data compression sub-system is in series with the data flow between the host interface, and the system data buffer. (Refer to Figure 10.) For compression, data flows from the host interface bus, through the bus interface controller, and is buffered in the external static RAM by the AHA3101 data compression chip. The data is then compressed by the DCLZ machine, flows into the system buffer, and then on to the tape drive interface. The data flow is reversed during decompression.

In order to maintain the host interface bus data transfer performance, the data compression sub-system must be able to handle the data rate that the host interface supports. The AHA3101 supports a 5 Mbyte/sec data transfer rate, between the Port A interface, and the Port A data buffer storage area in the external static RAM. The ENABLE RAM BUFFER bit in the *Configuration Control* register should be set, to allow data storage in the external static RAM.

The system processor data bus, the Port A data bus, and the Port B data bus are separate. This partitioning allows full bus bandwidth utilization with no arbitration overhead.

5.2 LOOK-ASIDE APPLICATION

In the look-aside application, the system buffer is in series with the data flow. For compression, data flows from the host interface bus, through the bus interface controller, and into the system buffer. Data then flows from the system buffer and into the AHA3101 data compression chip, where it is compressed and sent back to the system buffer. The data then flows from the system buffer, out through the tape drive interface. The data flow is reversed during decompression.

Since the data is buffered in the system buffer as it enters and leaves the AHA3101, there is no need to also buffer the data in the external static RAM. Data buffering in the external static RAM is disabled, by writing a zero to the ENABLE RAM BUFFER bit in the *Configuration Control* register.

Figure 10: In-Line Application

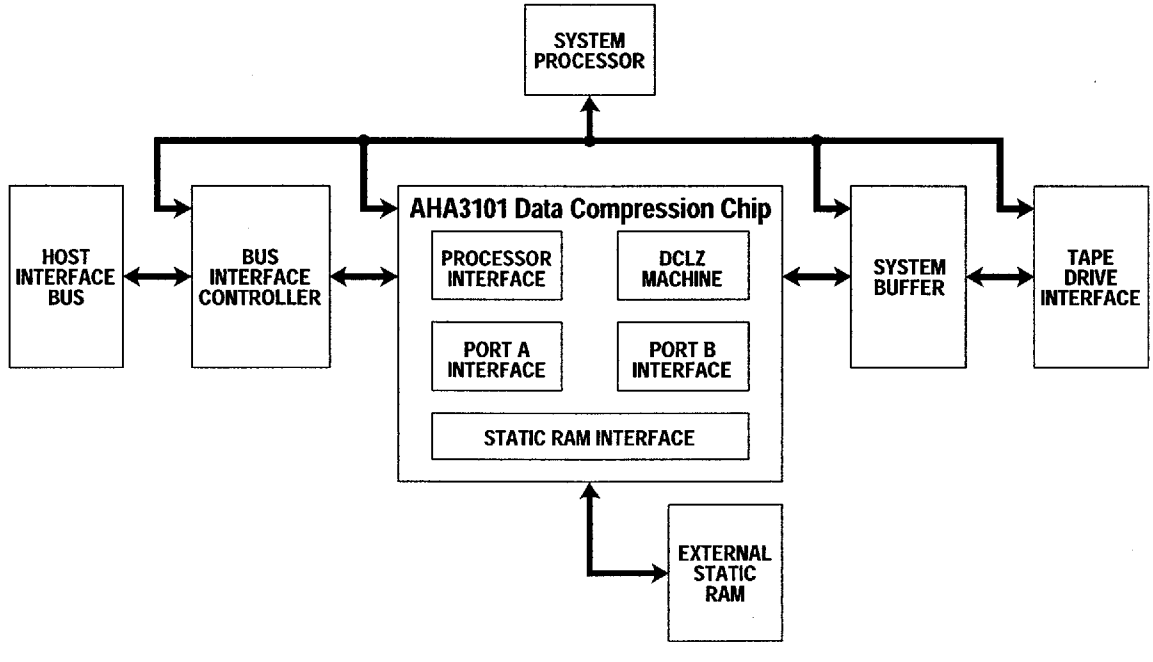
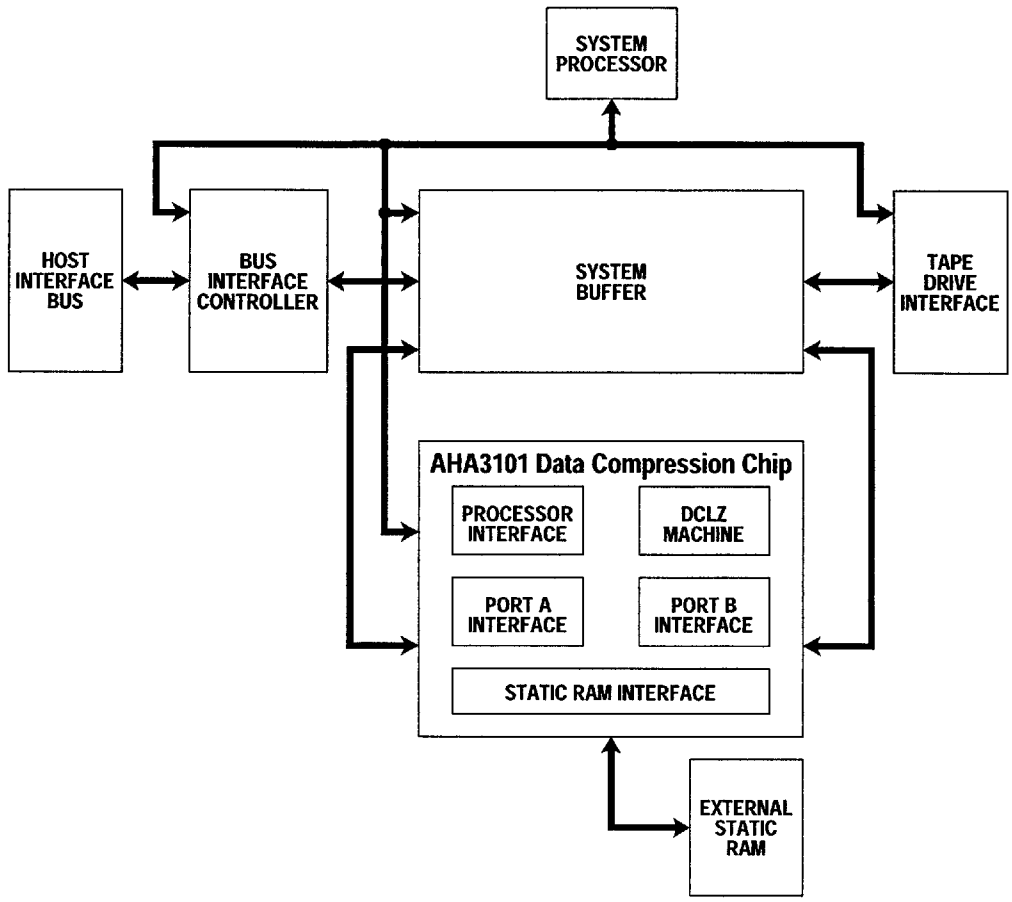


Figure 11: Look Aside Application



6.0 PROGRAMMER'S OVERVIEW

6.1 REGISTER ADDRESS MAP

The following table shows the register map for the chip. All addresses are in hexadecimal.

Table 5: Register Address Map

ADDRESS	*READ*	*WRITE*
00	Status 0	Control 0
01	Status 1	Control 1
02	Input Byte Count [23:16]	Control 2
03	Input Byte Count [15:8]	Control 3
04	Input Byte Count [7:0]	Configuration Control
05	Output Byte Count [23:16]	Port A Buffer Stop Address [7:0]
06	Output Byte Count [15:8]	Port A Buffer Stop Address [15:8]
07	Output Byte Count [7:0]	Port A Buffer Stop Address [16]
08	Port A Buffer Address [7:0]	Port A Buffer Address [7:0]
09	Port A Buffer Address [15:8]	Port A Buffer Address [15:8]
0A	Port A Buffer Address [16]	Port A Buffer Address [16]
0B	Port A DMA Status	Port A DMA Control
0C	*Reserved*	Port B Buffer Stop Address [7:0]
0D	*Reserved*	Port B Buffer Stop Address [15:8]
0E	*Reserved*	Port B Buffer Stop Address [16]
0F	Port B Buffer Address [7:0]	Port B Buffer Address [7:0]
10	Port B Buffer Address [15:8]	Port B Buffer Address [15:8]
11	Port B Buffer Address [16]	Port B Buffer Address [16]
12	Port B DMA Status	Port B DMA Control 0
13	*Reserved*	Port B DMA Control 1
14	Processor Buffer Address [7:0]	Processor Buffer Address [7:0]
15	Processor Buffer Address [15:8]	Processor Buffer Address [15:8]
16	Processor Buffer Address [16]	Processor Buffer Address [16]
17	Processor Buffer Data [7:0]	Processor Buffer Data [7:0]
18	Record Length [7:0]	Record Length [7:0]
19	Record Length [15:8]	Record Length [15:8]
1A	Record Length [23:16]	Record Length [23:16]
1B	Record Count [7:0]	Record Count [7:0]
1C	Record Count [15:8]	Record Count [15:8]
1D	Record Count [23:16]	Record Count [23:16]
1E	Interrupt Status	Interrupt Clear
1F	*Reserved*	Interrupt Disable

6.2 REGISTER DEFINITION

6.2.1 CONTROL 0: ADDRESS 00 HEX - WRITE ONLY

bit7	*bit6*	*bit5*	*bit4*	*bit3*	*bit2*	*bit1*	*bit0*
RESET DCLZ	COMP DECOMP	OPT ENABLE	COUNTER ENABLE	X	WRITE EOR CODE WORD ON EOR	RESET ON EOR	PAUSE ON EOR

(X denotes unused bit position)

RESET DCLZ - Writing a one initializes all the logic involved in the compression and decompression operations - the DCLZ machine, the Port B Static RAM Interface, and the Port B DMA Interface. It clears the FIFO Byte Count. The RESET DCLZ bit does not affect any registers that can be written through the processor interface, except to clear the PAUSE REQUEST bit and set the RESET REQUEST bit in *Control 1* register. This bit does not affect the Port A Interface. A zero must be written to this bit to allow the DCLZ machine, the Port B Static RAM Interface, and the Port B DMA Interface to leave their reset states. This bit is a one when the chip is reset from the RESETN pin.

COMP DECOMP - High level selects compression. Low level selects decompression. This bit should only be changed when the RESET DCLZ bit is a one (active). This bit is a one when the chip is reset from the RESETN pin.

OPT ENABLE - A high level enables the optimization circuitry to optimize the compression ratio, based on the current data being compressed. This bit enables the function of the THRESH[5:0] bits and the PERIOD[1:0] bits in *Control 2* register. This bit is a one when the chip is reset from the RESETN pin.

COUNTER ENABLE -A high level allows the Input Byte Counter and the Output Byte Counter to count incoming bytes to the DCLZ machine and outgoing bytes from the CODE PACKER/ UNPACKER of the DCLZ machine. This bit is a one when the chip is reset from the RESETN pin.

WRITE EOR CODE WORD ON EOR -Active high. This bit operates in compression mode only. If set, the End of Record code word is output in the compressed data stream, when the *Record Length* register signals End of Record. The internal buffering in the Port B Static RAM Interface, the DCLZ machine and the Port B DMA Interface will be flushed. The FLUSHED status bit will then be set until the next byte comes in. If the *Record Count* register is zero, signaling end of compression transfer, the COMP END OF TRANSFER Interrupt is then generated. If the WRITE EOR CODE WORD ON EOR bit is zero, there will not be End of Record code words in the compressed output data, which correspond to the End of Record boundaries generated by the *Record Length* register in the uncompressed input data. Also, COMP END OF TRANSFER Interrupts will not be generated, and the chip will not RESET ON EOR or PAUSE ON EOR. This bit is a one when the chip is reset from the RESETN pin.

RESET ON EOR - Active high. This bit operates in compression mode only. If set, the dictionary will be reset when the *Record Length* register signals End of Record. A reset code word will be output after the code words for the previous record have been output. After the DCLZ machine reset sequence is completed, the DCLZ RESET COMPLETE status bit will be set until the next byte comes in. This bit is a zero when the chip is reset from the RESETN pin.

PAUSE ON EOR - Active high. In compression mode, the Port B Static RAM Interface, the DCLZ machine, and the Port B DMA Interface will be paused when the *Record Length* register detects an End of Record Condition, and after the last compressed byte of the record has been sent out of the chip. The PAUSED status bit will then be set until a low level is written to PAUSE REQUEST (bit 0 in *Control 1* register). In decompression mode the DCLZ machine and the Port B Static RAM Interface will be paused when an incoming End of Record code word is detected by the DCLZ machine in the incoming compressed data stream, and after the last byte of the record has been transferred out of the chip. If enabled, the DECOMP EOR Interrupt will then be generated. Note that during decompression the Port B DMA Interface will continue to fill the FIFO, regardless of the state of the PAUSE ON EOR bit. To continue, a low must be written to the PAUSE REQUEST bit. This bit is a one when the chip is reset from the RESETN pin.

6.2.2 STATUS 0: ADDRESS 00 HEX - READ ONLY

bit7	*bit6*	*bit5*	*bit4*	*bit3*	*bit2*	*bit1*	*bit0*
0	UNKNOWN CODE LENGTH	HALF	FULL	FROZEN	FLUSHED	DCLZ RESET COMPLETE	PAUSED

UNKNOWN CODE LENGTH -During decompression an attempt to bump the code word length beyond 12 bits was encountered. The DCLZ ERROR Interrupt will be set when this occurs. The reset state of this bit is zero.

HALF - The dictionary is over half full if this bit is set to a one. The reset state of this bit is zero.

FULL - Active high, indicating the dictionary is full. The reset state of this bit is zero.

FROZEN - Active high, indicating no more entries are possible in the dictionary. The reset state of this bit is zero.

FLUSHED - If high, the internal buffering in the Port B Static RAM Interface, the DCLZ machine, and the Port B DMA Interface have been flushed. The reset state of this bit is zero.

DCLZ RESET COMPLETE -Active high, indicating that the DCLZ machine reset sequence has been completed. The reset state of this bit is zero.

PAUSED - If high, the Port B Static RAM Interface, the DCLZ machine, and the Port B DMA Interface are paused. When the PAUSED bit is set, the *Record Length* register, the *Record Count* register, the *Port B Buffer Address* register, Input Byte Count, Output Byte Count, FIFO Byte Count are stable, and can be read through the Processor Interface. Also, the *Record Length* register, the *Record Count* register, the *Port B Buffer Address* register and the *Port B Buffer Stop Address* register can be written when PAUSED is set. The PAUSED bit can be set by the processor with the PAUSE REQUEST bit, or the PAUSE ON EOR bit. The PAUSED bit gets set on the following interrupt conditions: Port B pointer match; DCLZ error; Port B parity error; and Port A parity error, when the static RAM buffer is disabled. A low must be written to the PAUSE REQUEST bit of *Control 1* register to continue. The reset state of this bit is zero.

6.2.3 CONTROL 1: ADDRESS 01 HEX - WRITE ONLY

bit7	*bit6*	*bit5*	*bit4*	*bit3*	*bit2*	*bit1*	*bit0*
RESET INPUT COUNTER	RESET OUTPUT COUNTER	DCLZ ENABLE	PASS-THRU ENABLE	DCLZ OUTPUT ENABLE	WRITE EOR CODE WORD REQUEST	RESET REQUEST	PAUSE REQUEST

RESET INPUT COUNTER -A high level resets the Input Byte Counter. The bit automatically returns low following the reset. This bit is a one when the chip is reset from the RESETN pin.

RESET OUTPUT COUNTER -A high level resets the Output Byte Counter. The bit automatically returns low following the reset. This bit is a one when the chip is reset from the RESETN pin.

DCLZ ENABLE - A high level enables the CODE PACKER/UNPACKER and DATA COMPRESSOR/ DECOMPRESSOR blocks in the DCLZ machine. This bit should only be changed when the RESET DCLZ bit is active. This bit is a one when the chip is reset from the RESETN pin.

PASSTHRU ENABLE -A high level causes data to be passed through the DCLZ machine unaltered. A low level causes the output to come from the DCLZ CODE PACKER/UNPACKER block. This bit should only be changed when the RESET DCLZ bit is active. This bit is a zero when the chip is reset from the RESETN pin.

DCLZ OUTPUT ENABLE -A high level enables data out from the DCLZ machine. A low level disables data from coming out of the DCLZ machine. The output disable feature is only supported in decompression mode. This bit is a one when the chip is reset from the RESETN pin.

WRITE EOR CODE WORD REQUEST -This bit operates in compression mode only. After the current SRAM cycle for the Port B Static RAM Interface is completed, the DCLZ machine will issue an End of Record code word. All data in the Port B Static RAM Interface, the DCLZ machine, and the Port B DMA Interface will be transferred out of the chip. If the *Record Count* register is zero, the COMP END OF TRANSFER Interrupt is generated after the last byte of the record has been transferred out of the chip. The FLUSHED bit in *Status 0* register will then be set until the next byte is input. This request will clear itself following the flush. This bit is a one when the chip is reset from the RESETN pin.

RESET REQUEST -This bit operates in compression mode only. After the current compression match sequence has been completed, the DCLZ machine will reset the dictionary. The reset code word is output in the compressed data stream. After the DCLZ machine reset sequence is completed, the DCLZ RESET COMPLETE bit will be set until the next byte is input. This request will clear itself following the reset. This bit is a one when the chip is reset from the RESETN pin, or when the RESET DCLZ bit is active.

PAUSE REQUEST -If high, the Port B Static RAM Interface, the DCLZ machine, and the Port B DMA Interface will pause after the Port B Static RAM Interface has completed its current SRAM access. The PAUSED status bit will then be set. To continue, a low level must be written here. This bit is a one when the chip is reset from the RESETN pin, or when the RESET DCLZ bit is active.

6.2.4 STATUS 1: ADDRESS 01 HEX - READ ONLY

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	0	DECOMP BIT2	DECOMP BIT1	DECOMP BIT0	COMP BIT2	COMP BIT1	COMP BIT0

The decompression status bits are encoded as follows:

DECOMP BIT2	DECOMP BIT1	DECOMP BIT0	DESCRIPTION
0	0	0	IDLE
0	0	1	WAITING FOR INPUT
0	1	0	WAITING FOR OUTPUT
0	1	1	ACTIVE
1	0	0	PERFORMING RESET
1	0	1	BAD WORD LENGTH
1	1	0	UNKNOWN CODEWORD
1	1	1	RESERVED

IDLE - Decompression has started, or the chip is in compression mode.

WAITING FOR INPUT -The data decompressor is ready to input another codeword, but one is not available from the CODE PACKER/UNPACKER input.

WAITING FOR OUTPUT -The data decompressor has a byte to output, but the CODE PACKER/ UNPACKER output is not ready.

ACTIVE - The data decompressor is not waiting for I/O, and is actively decompressing codewords.

PERFORMING RESET -The decompression dictionary is being initialized. This is not a static state, and lasts only a few clock cycles.

BAD WORD LENGTH -A codeword was input that represents a string longer than 128 bytes. This will cause a DCLZ ERROR interrupt.

UNKNOWN CODEWORD -A codeword was input for a dictionary entry that does not exist, or is a reserved codeword. This will cause a DCLZ ERROR Interrupt.

The DECOMP BIT [2:0] bits are cleared to zero when the chip is reset by the RESETN pin, or when the RESET DCLZ bit is active.

The compression status bits are encoded as follows:

DECOMP BIT2	DECOMP BIT1	DECOMP BIT0	DESCRIPTION
0	0	0	IDLE
0	0	1	WAITING FOR INPUT
0	1	0	WAITING FOR OUTPUT
0	1	1	ACTIVE
1	0	0	PERFORMING RESET
1	0	1	SEARCHING FOR AN EMPTY RAM SPACE
1	1	0	HARDWARE ERROR
1	1	1	RESERVED

IDLE - Compression has not started, or the chip is in decompression mode.

WAITING FOR INPUT -The data compressor is waiting for a byte from the CODE PACKER/ UNPACKER input.

WAITING FOR OUTPUT -The data compressor has a codeword to output, but the CODE PACKER/ UNPACKER output is not ready.

ACTIVE - The data compressor is not waiting for I/O, or searching for an empty RAM space, and is actively compressing data.

PERFORMING RESET -The compression dictionary is being initialized. This is not a static state, and lasts only a few clock cycles.

SEARCHING FOR AN EMPTY RAM SPACE -The data compressor is searching for an empty dictionary location.

HARDWARE ERROR -The data compressor tried to generate a codeword for a string longer than 128 bytes. This causes a DCLZ ERROR Interrupt.

The COMP BIT [2:0] bits are cleared to zero when the chip is reset by the RESETN pin, or when the RESET DCLZ bit is active.

6.2.5 CONTROL 2: ADDRESS 02 HEX - WRITE ONLY

bit7	*bit6*	*bit5*	*bit4*	*bit3*	*bit2*	*bit1*	*bit0*
THRESH BIT5	THRESH BIT4	THRESH BIT3	THRESH BIT2	THRESH BIT1	THRESH BIT0	PERIOD BIT1	PERIOD BIT0

This register is used to control optimized mid record resets of the dictionary. A compression ratio is specified by the threshold bits according to the following formula:

$$\text{compression ratio} = 64 \div (64 - \text{THRESH})$$

For example, if THRESH is set at 32 the compression ratio is 2. This compression ratio is a target. After every N number of bytes as specified by the PERIOD field has been input, the actual compression ratio is checked against the target. If the actual is less than the target, the dictionary is automatically reset.

The THRESH[5:0] and PERIOD[1:0] bits are cleared to zero when the chip is reset by the RESETN pin.

PERIOD		
bit 1	*bit 0*	*SIZE*
0	0	512 bytes
0	1	1024 bytes
1	0	2048 bytes
1	1	4096 bytes

COMPRESSION RANGES	
Compression Ratio	*Threshold Value*
$1 \rightarrow 2$	$0 \rightarrow 32$
$2 \rightarrow 3$	$33 \rightarrow 42$
$3 \rightarrow 4$	$43 \rightarrow 48$
$4 \rightarrow 8$	$49 \rightarrow 56$
$8 \rightarrow 64$	$57 \rightarrow 63$

6.2.6 CONTROL 3: ADDRESS 03 HEX - WRITE ONLY

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
X	X	X	X	BAD CDWD CK DISAB	X	DICT SIZE 1	DICT SIZE 0

(X denotes unused bit position)

BAD CODEWORD CHECK DISABLE -Active high. Disables checking for bad code words and producing the associated status error. The reset state of this bit is zero.

DICT SIZE [1:0] - These bits determine the size of the compression and decompression dictionary storage area. Three bytes of static RAM are required for each dictionary entry. The encoding of the two DICT SIZE bits is shown below. The reset state of these two bits is zero.

DICT SIZE 1	DICT SIZE 0	DICTIONARY ENTRIES	RAM BYTES
0	0	8K	24K
0	1	16K	48K
1	X	10K	30K

6.2.7 CONFIGURATION CONTROL: ADDRESS 04 HEX - WRITE ONLY

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
X	X	X	X	X	RAMTYPE	ENABLE RAM BUFFER	RAM WAIT STATE

(X denotes unused bit position)

RAMTYPE - This bit determines what type of static RAM is used for the dictionary memory and RAM buffer memory. A one denotes 32K by 8 static RAM chips. A zero denotes that a 128K by 8 static RAM chip is used. This bit determines the function of the MA15/CSN0 and MA16/CSN1 pins. This bit is a one when the chip is reset.

ENABLE RAM BUFFER -Writing a one enables the RAM buffer, allowing data to be stored in the external static RAM, and accessed through Port A and Port B. Writing a zero bypasses data storage between Port A, the static RAM buffer, and Port B. When ENABLE RAM BUFFER is a zero during compression, data enters the chip from the DA[7:0] bus. It flows through the Port A DMA interface and the Port A static RAM interface. It then passes directly to the Port B static RAM interface, flows through the DCLZ machine, and out of the chip through the Port B DMA interface. The data flow is reversed during decompression. This bit is a one when the chip is reset.

RAM WAIT STATE -A one in this bit adds a clock cycle, as a wait state, to the read and write cycles to the static RAMs. A zero causes no clock cycles to be added to the read and write to the static RAMs. This bit is zero when the chip is reset.

6.2.8 INPUT BYTE COUNT: ADDRESS 02,03,04 HEX - READ ONLY

Most Significant Byte (address 02 hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
23	22	21	20	19	18	17	16

Middle Byte (address 03 hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
15	14	13	12	11	10	9	8

Least Significant Byte (address 04 hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
7	6	5	4	3	2	1	0

The twenty four bit *Input Byte Count* is a read only register. It counts the number of bytes that are transferred into the INPUT block of the DCLZ machine. This operates in compression and decompression modes.

This register should be read when the PAUSED bit in *Status 0* register is a one (active).

6.2.9 OUTPUT BYTE COUNT: ADDRESS 05,06,07 HEX - READ ONLY

Most Significant Byte (address 05 hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
23	22	21	20	19	18	17	16

Middle Byte (address 06 hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
15	14	13	12	11	10	9	8

Least Significant Byte (address 07 hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
7	6	5	4	3	2	1	0

The twenty four bit *Output Byte Count* is a read only register which operates in compression and decompression modes. It counts the number of bytes that are transferred out of the CODE PACKER/ UNPACKER block inside the DCLZ machine.

In compression mode, this register should be read when the DCLZ machine is paused at an End of Record boundary, when the internal pipeline of the DCLZ machine is flushed of data.

6.2.10 PORT A BUFFER STOP ADDRESS: ADDRESS 05,06,07 HEX - WRITE ONLY

Least Significant Byte (address 05 hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
7	6	5	4	3	2	1	0

Middle Byte (address 06 hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
15	14	13	12	11	10	9	8

Most Significant Byte (address 07 hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
X	X	X	X	X	X	X	16

(X denotes unused bit position)

The *Port A Buffer Stop Address* register contains 17 bits, so the register can point to any address location in the 128K byte memory address space. This register is programmed with the address that is one byte above the last byte in the buffer for Port A DMA transfers. After each Port A DMA cycle, the *Port A Buffer Address* register contents are incremented, and then compared to the Port A Buffer Stop Address. If these two pointers are equal, Port A DMA transfer is completed. The chip can be programmed to interrupt once this event occurs. This register is write only, and can not be read through the processor interface.

6.2.11 PORT A BUFFER ADDRESS; ADDRESS 08,09,0A HEX - READ/WRITE

Least Significant Byte (address 08 hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
7	6	5	4	3	2	1	0

Middle Byte (address 09 hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
15	14	13	12	11	10	9	8

Most Significant Byte (address 0A hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	0	0	0	0	0	0	16

The *Port A Buffer Address* register contains the pointer into the memory space that data is read from or written to during Port A DMA transfers. This is a 17 bit register, and so it can point to any byte location in the 128K byte memory space. Since any memory address can be accessed through Port A, it is the programmer's responsibility that the Port A DMA transfers do not affect the dictionary memory space, or the Port B DMA memory space.

The initial contents of this register should be written through the processor interface, when Port A DMA is inactive. On subsequent DMA cycles, the Port A Buffer Address is incremented after each static RAM access, to point to the next byte in the static RAM buffer. Thus, data buffers in the memory begin at the initial address, and grow upward in the memory address space. When incrementing, the address will wrap around from 1FFFF hex to 00000 hex.

Writes through the processor interface initialize the *Port A Buffer Address* register. Bits [7:1] of the Most Significant Byte (address 0A hex) are unused during write cycles. Reads from the *Port A Buffer Address* register give the current Port A DMA address, containing the address of the next byte to be accessed. Bits [7:1] of the Most Significant Byte (address 0A hex) are zero on read cycles. All three bytes should only be read when the DMA ACTIVE bit in the *Port A DMA Status* register is zero (inactive).

6.2.12 PORT A DMA CONTROL: ADDRESS 0B HEX - WRITE ONLY

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
DIRA POLARITY	ENABLE PARITY	ODD PARITY	ENABLE DAPTY PULLUP	ENABLE DA BUS PULLUP	DREQA POLARITY	DIR	GO

DIRA POLARITY -A one makes the DIRAIN and DIRAOUT pins high active. A zero makes the DIRAIN and DIRAOUT pins low active. The reset state of this bit is zero.

ENABLE PARITY -A one enables the DAPTY output driver during data transfers out of Port A, and enables parity error checking during data transfers into Port A. A zero in the ENABLE PARITY bit disables the DAPTY output driver during data transfers out of Port A, and disables parity error checking during data transfers into Port A. The reset state of this bit is zero.

ODD PARITY - A one selects odd parity on Port A. A zero selects even parity on Port A. The reset state of this bit is one.

ENABLE DAPTY PULLUP -A one enables the pullup resistor on the DAPTY pin. A zero tristates the pullup resistor. The reset state of this bit is zero.

ENABLE DA BUS PULLUP -A one enables the pullup resistors on the DA[7:0] bus. A zero tristates the pullup resistors. The reset state of this bit is zero.

DREQA POLARITY -A one makes a high voltage on the DREQA input pin denote that the signal is active. A zero makes a low voltage on the DREQA pin denote that the signal is active. The DREQA POLARITY bit should only be changed when the GO bit has previously been set to zero (inactive), to prevent a spurious DMA cycle from occurring. The reset state of this bit is one.

DIR - A zero denotes that the direction of DMA transfer is inward from the Port A interface out to the static RAM memory. A one denotes that the direction of DMA transfer is inward from the static RAM memory and out to the Port A interface. Note that the direction of Port A DMA transfer is independent from any other data transfer operations in the chip. The reset state of this bit is zero.

GO - A one will start Port A DMA transfer. A zero will abort a Port A DMA transfer in progress after the current DMA cycle completes normally. Once a zero is written to the GO bit, the processor should wait until the DMA ACTIVE bit in the *Port A DMA Status* register is a zero, to signal that Port A DMA is inactive. The reset state of this bit is zero.

6.2.13 PORT A DMA STATUS: ADDRESS 0B HEX - READ ONLY

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	0	0	0	0	DAPTY	DIR	DMA ACTIVE

DAPTY - This bit shows the state of the DAPTY pin. This bit is a one when the DAPTY pin is a high voltage. This bit is a zero when the DAPTY pin is a low voltage.

DIR - A zero denotes that the direction of DMA transfer is inward from the Port A interface out to the static RAM memory. A one denotes that the direction of DMA transfer is inward from the static RAM memory and out to the Port A interface. This bit reflects the state of the DIR bit in the *Port A DMA Control* register. The reset state of this bit is a zero.

DMA ACTIVE - A one denotes that Port A DMA cycle is in process. A zero denotes that Port A DMA is either between DMA cycles, or is stopped. Port A DMA is stopped (DMA ACTIVE is zero) when: a zero is written to the GO bit in the *Port A DMA Control* register and any DMA cycle in process has been completed; Port A parity is enabled, and a parity error occurs; the Port A Buffer Address increments and is equal to the Port A Buffer Stop Address when the Port A DMA is completed. The reset state of this bit is zero.

6.2.14 PORT B BUFFER STOP ADDRESS: ADDRESS 0C,0D,0E HEX - WRITE ONLY

Least Significant Byte (address 0C hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
7	6	5	4	3	2	1	0

Middle Byte (address 0D hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
15	14	13	12	11	10	9	8

Most Significant Byte (address 0E hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
X	X	X	X	X	X	X	16

(X denotes unused bit position)

The *Port B Buffer Stop Address* register contains 17 bits, so the register can point to any address location in the 128K byte memory address space. This register is programmed with the address that is one byte above the last byte in the buffer for Port B DMA transfers. After each Port B DMA cycle, the *Port B Buffer Address* register contents are incremented, and then compared to the Port B Buffer Stop Address. If these two pointers are equal, Port B DMA transfer is completed. The chip can be programmed to interrupt once this event occurs. When this occurs, the Port B Static RAM Interface will pause itself, the DCLZ machine, and the Port B DMA Interface. This allows the processor to change the *Port B Buffer Stop Address* register and the *Port B Buffer Address* register.

This register is write only, and can not be read through the processor interface. It should be written when the RESET DCLZ bit in *Control 0* register is active, or when the PAUSED bit in *Status 0* register is active.

6.2.15 PORT B BUFFER ADDRESS: ADDRESS 0F,10,11 HEX - READ/WRITE

Least Significant Byte (address 0C hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
7	6	5	4	3	2	1	0

Middle Byte (address 0D hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
15	14	13	12	11	10	9	8

Most Significant Byte (address 0E hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	0	0	0	0	0	0	16

The *Port B Buffer Address* register contains the pointer into memory space that data is read from or written to during Port B DMA transfers. This is a 17 bit register, and so it can point to any byte location in the 128K byte memory space. Since any memory address can be accessed through Port B, it is the programmer's responsibility that the Port B DMA transfers do not affect the dictionary memory space, or the Port A DMA memory space.

The Port B Buffer Address should be read from, or written to, only when the RESET DCLZ bit in *Control 0* register is active, or when the PAUSED bit in *Status 0* register is active. The processor initializes the *Port B Buffer Address* register by writing through the processor interface. On subsequent DMA cycles, the Port B Buffer Address is incremented after each static RAM access, to point to the next byte in the static RAM buffer. Thus, data buffers in the memory begin at the initial address, and grow upward in the memory address space. When incrementing, the address will wrap around from 1FFFF hex to 00000 hex.

Port B Buffer Address register always transfers data between the static RAM memory and the compression/decompression circuit. Therefore, it is typically transferring uncompressed data bytes.

Writes through the processor interface initialize the *Port B Buffer Address* register. Bits [7:1] of the Most Significant Byte (address 11 hex) are unused during write cycles. Reads from the *Port B Buffer Address* register give the current Port B DMA address, containing the address of the next byte to be accessed. Bits [7:1] of the Most Significant Byte (address 11 hex) are zeroes during read cycles.

6.2.16 PORT B DMA CONTROL 0: ADDRESS 12 HEX - WRITE ONLY

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
X	X	ENABLE PARITY	ODD PARITY	ENABLE DBPTY PULLUP	ENABLE DB BUS PULLUP	DREQB POLARITY	DBOE POLARITY

(X denotes unused bit position)

Note that since Port B supports the DCLZ machine, the Port B DMA direction is controlled by the COMP DECOMP bit in *Control 0* register. This is why Port B does not have a DIR bit (as is found in Port A).

ENABLE PARITY -A one enables the DBPTY[1:0] output drivers during data transfers out of Port B, and enables parity error checking during data transfers into Port B. A zero in the ENABLE PARITY bit disables the DBPTY[1:0] output drivers during data transfers out of Port B, and disables parity error checking during data transfers into Port B. The reset state of this bit is zero.

ODD PARITY - A one selects odd parity per byte on Port B. A zero selects even parity per byte on Port B. The reset state of this bit is one.

ENABLE DBPTY PULLUP -A one enables the pullup resistors on the DBPTY[1:0] pins. A zero tristates these pullup resistors. The reset state of this bit is zero.

ENABLE DB BUS PULLUP -A one enables the pullup resistors on the DB[7:0] bus. A zero tristates the pullup resistors. The reset state of this bit is zero.

DREQB POLARITY -A one makes a high voltage on the DREQB output pin denote that the signal is active. A zero makes a low voltage on the DREQB pin denote that the signal is active. The reset state of this bit is one.

DBOE POLARITY -A one makes a high voltage on the DBOE input pin denote that the signal is active. A zero makes a low voltage on the DBOE pin denote that the signal is active. The reset state of this bit is zero.

6.2.17 PORT B DMA STATUS: ADDRESS 12 HEX - READ ONLY

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	BYTE EOR HELD	DBPTY[1]	DBPTY[0]	FIFO BYTE COUNT[3]	FIFO BYTE COUNT[2]	FIFO BYTE COUNT[1]	FIFO BYTE COUNT[0]

BYTE EOR HELD -This bit is used when Port B is in 16 bit DMA mode, for compression and compression pass through modes, when the HOLD BYTE EOR bit in Port B DMA *Control 1* register is a one. The BYTE EOR HELD bit is set when the HOLD BYTE EOR bit is a one, the *Record Count* register is non-zero, and the last byte of a record occurs on an odd byte boundary. The BYTE EOR HELD bit remains set, and the byte is held in Port

B, until the first byte of the subsequent record arrives in the Port B FIFO. A Port B DMA cycle with these two bytes occurs, with the last byte of the first record transferred on D[7:0], and the first byte of the subsequent record transferred on DB[7:0]. Then, the BYTE EOR HELD bit is cleared. This bit is always zero when Port B is in 8 bit DMA mode. The reset state of this bit is zero.

DBPTY[1] - This bit shows the state of the DBPTY[1] pin. This bit is a one when the DBPTY[1] pin is a high voltage. This bit is a zero when the DBPTY[1] pin is a low voltage.

DBPTY[0] - This bit shows the state of the DBPTY[0] pin. This bit is a one when the DBPTY[0] pin is a high voltage. This bit is a zero when the DBPTY[0] pin is a low voltage.

FIFO BYTE COUNT [3:0] -These bits determine the number of bytes in the Port B FIFO. The valid binary values are 0000 to 1000, for zero to eight bytes in the FIFO. These bits are initialized to zero when the chip is reset from the RESETN pin, when the RESET DCLZ bit in *Control 0* register is a one (active), or when a one is written to the CLEAR FIFO bit.

6.2.18 PORT B DMA CONTROL 1: ADDRESS 13 HEX - WRITE ONLY

bit7	*bit6*	*bit5*	*bit4*	*bit3*	*bit2*	*bit1*	*bit0*
X	X	X	X	HOLD BYTE EOR	DMA SIZE	WORD	CLEAR FIFO

(X denotes unused bit position)

HOLD BYTE EOR -This bit controls the function of the chip in compression and compression pass through modes during 16 bit Port B DMA mode, when the last byte of a record occurs on an odd byte boundary. This applies only when the *Record Count* register is non-zero, and the last byte of the record is the only valid byte in the compression pipeline. In this scenario, when HOLD BYTE EOR is zero, the last byte of the record is transferred out of Port B on D[7:0]. The data on DB[7:0] is undefined. In this scenario, when HOLD BYTE EOR is a one, the last byte of the record is held in the Port B FIFO. The BYTE EOR HELD bit in the *Port B DMA Status* register is set. The chip then fetches the first byte of the next record. When this byte enters the Port B FIFO, a Port B DMA cycle with these two bytes occurs. The last byte of the first record is transferred on D[7:0], and the first byte of the subsequent record is transferred on DB[7:0]. The reset state of HOLD BYTE EOR is zero.

DMA SIZE - A zero selects eight bit Port B DMA transfer on the DB[7:0] pins. A one selects sixteen bit Port B DMA transfer on the DB[7:0] and D[7:0] pins. This bit should only be changed when the RESET DCLZ bit in *Control 0* register is a one (active), when Port B is reset. The reset state of this bit is zero.

WORD - The WORD bit determines which bytes contain valid data on the first Port B DMA cycle when Port B is in sixteen bit mode. When the WORD bit is a zero in sixteen bit mode for Port B DMA, the first DMA cycle only transfers one byte of valid data on DB[7:0]. The data on D[7:0] is ignored. When the WORD bit is a one in sixteen bit mode for Port B DMA, the first DMA cycle transfers two bytes of valid data on DB[7:0], and D[7:0]. The WORD bit only affects the first DMA cycle after the RESET DCLZ bit has been active, or after Port B and the DCLZ machine have been paused on an End of Record (EOR) boundary in compression or decompression modes. This bit has no affect during eight bit Port B DMA transfers, when the DMA SIZE bit is a zero. The reset state of this bit is one.

CLEAR FIFO - A one will create a pulse, which will clear the FIFO Byte Count. This effectively removes any data in the FIFO. A zero has no affect on the FIFO Byte Count.

6.2.19 PROCESSOR BUFFER ADDRESS: ADDRESS 14,15,16 HEX - READ/WRITE

Least Significant Byte (address 14 hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
7	6	5	4	3	2	1	0

Middle Byte (address 15 hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
15	14	13	12	11	10	9	8

Most Significant Byte (address 16 hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	0	0	0	0	0	0	16

The *Processor Buffer Address* register is the pointer for the processor to access any byte in the static RAM. It is the programmer's responsibility to ensure that processor accesses do not ever interfere with the dictionary address space, or cause spurious interference with the Port A and Port B data buffer areas.

The processor writes the target static RAM address to the three bytes of this register. The order of the writes to the three bytes is arbitrary. The *Processor Buffer Address* register contents will be incremented after each access to the *Processor Buffer Data* register, allowing a contiguous block of static RAM to be accessed quickly. The auto-incrementing function of the *Processor Buffer Address* register will wrap around from 1FFFF hex to 00000 hex.

When reading from the *Processor Buffer Address* register the current, incremented value of the *Processor Buffer Address* register is returned. This is the address of the next byte that the processor would access.

6.2.20 PROCESSOR BUFFER DATA: ADDRESS 17 HEX - READ/WRITE

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
7	6	5	4	3	2	1	0

The *Processor Buffer Data* register supports read and write operations into the static RAM. The processor has the highest priority for access into the static RAM. The DTACKN/READY handshake with the processor guarantees that the transfer has completed with valid data. Since the processor has the highest priority for access into the static RAM, the DTACKN/READY handshake will complete in the shortest amount of time possible. This minimizes the amount of time that the processor data bus is tied up with accesses into the static RAM buffer.

6.2.21 RECORD LENGTH: ADDRESS 18,19,1A HEX - READ/WRITE

Least Significant Byte (address 18 hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
7	6	5	4	3	2	1	0

Middle Byte (address 19 hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
15	14	13	12	11	10	9	8

Most Significant Byte (address 1A hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
23	22	21	20	19	18	17	16

The twenty four bit *Record Length* register is used to count the number of bytes of uncompressed data that comprise one record. It counts the number of uncompressed bytes that the Port B static RAM interface sends into the DCLZ machine. Thus, the counter operates in compression mode only. The counter is a binary down counter. When it reaches zero, an End of Record signal is sent into the DCLZ machine with the last uncompressed byte of the record.

The initial value of the record length is written into the *Record Length* register. The current value of the down counter is transferred during read cycles from this register.

The three bytes of the *Record Length* register should be read from, or written to, only when the RESET DCLZ bit in *Control 0* register is active, or when the PAUSED bit in *Status 0* register is active.

This register is used in conjunction with the *Record Count* register. When the *Record Length* register reaches zero, the *Record Count* register is decremented. If the *Record Count* register is greater than zero, the *Record Length* register down counter is reloaded, to allow another record to be compressed automatically.

6.2.22 RECORD COUNT: ADDRESS 1B,1C,1D HEX - READ/WRITE

Least Significant Byte (address 1B hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
7	6	5	4	3	2	1	0

Middle Byte (address 1C hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
15	14	13	12	11	10	9	8

Most Significant Byte (address 1D hex):

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
23	22	21	20	19	18	17	16

The twenty four bit *Record Count* register is used to count the number of records in a multi-record transfer. The number of bytes in a record is determined by the *Record Length* register. The number of records multiplied by the number of bytes in a record determines the total transfer size.

These two sets of registers allow multiple records to be compressed, without processor intervention. If only one record is to be compressed, then the *Record Count* register should be initialized to one.

The *Record Count* register operates on data compression only. It counts the number of uncompressed records that the Port B static RAM interface sends to the DCLZ machine. The counter is a binary down counter. When it reaches zero, and the Record Length counter is also zero, the fetching of uncompressed data is completed.

The initial value of the record count is written into the *Record Count* register. The current value of the down counter is transferred during read cycles from this register.

The three bytes of the *Record Count* register should be read from, or written to, only when the RESET DCLZ bit in *Control 0* register is active, or when the PAUSED bit in *Status 0* register is active.

6.2.23 INTERRUPT STATUS: ADDRESS 1E HEX - READ ONLY

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	DCLZ ERROR INT	DBPTY INT	DAPTY INT	DECOMP EOR INT	COMP END OF TRANSFER INT	PORT B POINTER INT	PORT A POINTER INT

This register is not affected by the *Interrupt Disable* register. Therefore, software polling can be used on this register to monitor internal chip events, without generating interrupts.

DCLZ ERROR INT -The DCLZ ERROR interrupt occurs when the DCLZ machine detects an error condition. A one denotes an active interrupt state. This bit is zero when the chip is reset from the RESETN pin.

DBPTY INT - The DBPTY interrupt occurs when the parity is incorrect when data is latched inside the chip on a Port B DMA cycle. This will cause the Port B Static RAM Interface, the DCLZ machine, and the Port B DMA Interface to pause. A one denotes an active interrupt state. This bit is zero when the chip is reset from the RESETN pin.

DAPTY INT - The DAPTY interrupt occurs when the parity is incorrect when data is latched on the Port A data bus, DA[7:0]. A one denotes an active interrupt state. This bit is zero when the chip is reset from the RESETN pin.

DECOMP EOR INT -The DECOMP EOR interrupt occurs during decompression, after an End of Record (EOR) code word is processed in the compressed data stream, and the last byte of the record has been sent out of the chip. This happens when decompression of a record is completed. When the static RAM buffer is enabled, this interrupt occurs when the last byte has been stored in the Port B data buffer in the static RAM. When the static RAM buffer is disabled, the DECOMP EOR INT occurs when the last byte has been transferred out of Port A. A one denotes an active interrupt state. This bit is zero when the chip is reset from the RESETN pin.

COMP END OF TRANSFER INT -The COMP END OF TRANSFER interrupt occurs during compression, after the *Record Length* register is zero, and the *Record Count* register is zero, and the last byte of the compressed record has been transferred out of the chip. This happens when a compression transfer is completed. This bit is also set when the WRITE EOR CODE WORD REQUEST bit in *Control 1* register is set, when the *Record Count* register is zero, and after the last byte of the record has been transferred out of the chip. This happens when a compression transfer is terminated through the processor interface. A one denotes an active interrupt state. This bit is zero when the chip is reset from the RESETN pin.

PORT B POINTER INT -When the Port B Buffer Address auto-increments and is equal to the Port B Buffer Stop Address, the last access occurs to the external static RAM for the currently defined Port B data buffer. When this access is completed, the PORT B POINTER INT bit is set to a one, and the Port B Static RAM Interface, the DCLZ machine, and the Port B DMA Interface will pause. This functions for compression and decompression modes. This bit is zero when the chip is reset from the RESETN pin.

PORT A POINTER INT -When the Port A Buffer Address auto-increments and is equal to the Port A Buffer Stop Address, the last access occurs to the external static RAM for the currently defined Port A data buffer. When this access is completed, the PORT A POINTER INT bit is set to a one. This functions for compression and decompression modes. This bit is zero when the chip is reset from the RESETN pin.

6.2.24 INTERRUPT CLEAR: ADDRESS 1E HEX - WRITE ONLY

bit7	*bit6*	*bit5*	*bit4*	*bit3*	*bit2*	*bit1*	*bit0*
X	CLEAR DCLZ ERROR INT	CLEAR DBPTY INT	CLEAR DAPTY INT	CLEAR DECOMP EOR INT	CLEAR COMP END TRANSFER INT	CLEAR PORT B POINTER INT	CLEAR PORT A POINTER INT

(X denotes unused bit position)

CLEAR DCLZ ERROR INT -Writing a one clears the DCLZ ERROR interrupt bit in the *Interrupt Status* register. The RESET DCLZ bit in *Control 0* register should be set to one before writing a one to the CLEAR DCLZ ERROR INT bit. This ensures that the cause of the error is cleared first, followed by the DCLZ ERROR interrupt status bit. Writing a zero has no affect on the *Interrupt Status* register.

CLEAR DBPTY INT -Writing a one clears the DBPTY interrupt bit in the *Interrupt Status* register. Writing a zero has no affect on the *Interrupt Status* register.

CLEAR DAPTY INT -Writing a one clears the DAPTY interrupt bit in the *Interrupt Status* register. Writing a zero has no affect on the *Interrupt Status* register.

CLEAR DECOMP EOR INT -Writing a one clears the DECOMP EOR interrupt bit in the *Interrupt Status* register. Writing a zero has no affect on the *Interrupt Status* register.

CLEAR COMP END TRANSFER INT -Writing a one clears the COMP END OF TRANSFER interrupt bit in the *Interrupt Status* register. Writing a zero has no affect on the *Interrupt Status* register.

CLEAR PORT B POINTER INT -Writing a one clears the PORT B POINTER interrupt bit in the *Interrupt Status* register. Writing a zero has no affect on the *Interrupt Status* register.

CLEAR PORT A POINTER INT -Writing a one clears the PORT A POINTER interrupt bit in the *Interrupt Status* register. Writing a zero has no affect on the *Interrupt Status* register.

6.2.25 INTERRUPT DISABLE: ADDRESS 1F HEX - WRITE ONLY

bit7	*bit6*	*bit5*	*bit4*	*bit3*	*bit2*	*bit1*	*bit0*
X	DIS DCLZ ERROR INT	DIS DBPTY INT	DIS DAPTY INT	DIS DECOMP EOR INT	DIS COMP END TRANSFER INT	DIS PORT B POINTER INT	DIS PORT A POINTER INT

(X denotes unused bit position)

DIS DCLZ ERROR INT -Writing a one disables the DCLZ ERROR interrupt. Writing a zero enables the DCLZ ERROR interrupt. This bit goes to one when the chip is reset from the RESETN pin.

DIS DBPTY INT - Writing a one disables the DBPTY interrupt. Writing a zero enables the DBPTY interrupt. This bit goes to one when the chip is reset from the RESETN pin.

DIS DAPTY INT - Writing a one disables the DAPTY interrupt. Writing a zero enables the DAPTY interrupt. This bit goes to one when the chip is reset from the RESETN pin.

DIS DECOMP EOR INT -Writing a one disables the DECOMP EOR interrupt. Writing a zero enables the DECOMP EOR interrupt. This bit goes to one when the chip is reset from the RESETN pin.

DIS COMP END OF TRANSFER INT -Writing a one disables the COMP END OF TRANSFER interrupt. Writing a zero enables the COMP END OF TRANSFER interrupt. This bit goes to one when the chip is reset from the RESETN pin.

DIS PORT B POINTER INT -Writing a one disables the PORT B POINTER interrupt. Writing a zero enables the PORT B POINTER interrupt. This bit goes to one when the chip is reset from the RESETN pin.

DIS PORT A POINTER INT -Writing a one disables the PORT A POINTER interrupt. Writing a zero enables the PORT A POINTER interrupt. This bit goes to one when the chip is reset from the RESETN pin.

7.0 PIN DESCRIPTION

This section describes the function of the pins of the chip. A low active signal has an "N" appended to the end of the signal name.

7.1 PROCESSOR INTERFACE

PROCESSOR INTERFACE		
NAME	**TYPE**	**DESCRIPTION**
PROCMODE	I	PROCessor MODE select pin. Connect to VDD to select a processor interface controlled by a data strobe (DSN), a read/write signal (RWN), with an open drain data transfer acknowledge output (DTACKN), and an open drain, low active interrupt (INTN). Connect to GND to select processor interface controlled by an I/O read strobe (IORDN), an I/O write strobe (IOWRN), with a high active ready output (READY), and a high active interrupt (INT).
A[4:0]	I	Address for registers accessed through the processor interface.
RWN/IOWRN	I	When the PROCMODE pin is a high voltage this signal functions as ReadWriteN. A high voltage denotes a processor read cycle. A low voltage denotes a write cycle. When the PROCMODE pin is a low voltage, this signal functions as I/O WRiteN. A low voltage denotes a processor I/O write cycle is occurring, and the rising edge denotes the end of the processor access. As IOWRN, this signal is used as a strobe signal, and must not glitch.
CSN	I	Chip SelectN. When the PROCMODE pin is a high voltage, a low voltage on this signal and on the DSN/IORDN signal denotes the start of a processor access. This signal can glitch when DSN/IORDN is a high voltage. It must not glitch once DSN/IORDN is a low voltage. When the PROCMODE pin is a low voltage, a low voltage on CSN and either DSN/IORDN or RWN/IOWRN denotes the start of a processor access. The CSN signal can glitch when both DSN/IORDN and RWN/IOWRN are at high voltage. CSN must not glitch once DSN/IORDN or RWN/IOWRN are at low voltage. CSN is active low.
DSN/IORDN	I	When the PROCMODE pin is a high voltage, this pin functions as DataStrobeN. A low voltage on this signal and on the CSN signal denotes the start of a processor access. The rising edge of DSN/IORDN denotes the end of a processor access. This signal is used as a strobe signal. It must not glitch. DSN/IORDN is active low. When the PROCMODE pin is a low voltage, this pin functions as I/O ReaDN. A low voltage denotes a processor I/O read cycle is occurring, and the rising edge denotes the end of the processor access. As IORDN, this signal is used as a strobe signal, and must not glitch.
DTACKN/READY	O	When the PROCMODE pin is a high voltage, this signal functions as a Data Transfer Acknowledge open drain output. A low voltage indicates that processor data has been latched on processor write cycles. On read cycles, a low voltage indicates that data is valid on the D[7:0] bus for the processor to latch. When the PROCMODE pin is a low voltage, this signal functions as a READY output. At the beginning of processor cycles, this output is driven to a low voltage, indicating that the chip is not ready. The pin is driven high when data is valid on the D[7:0] bus during read cycles, and after data has been internally latched during write cycles. This signal is tristated when processor cycles are inactive. The reset state of this pin is high impedance.
D[7:0]	I/O	Bidirectional processor data bus, to access all registers internal to the chip. The reset state of these pins is high impedance.

PROCESSOR INTERFACE		
NAME	**TYPE**	**DESCRIPTION**
INTN/INT	O	When the PROCMODE pin is a high voltage, this signal functions as a low active interrupt, with an open drain output. A low voltage indicates that an internal interrupt is active. The reset state of the pin in this mode is tristate. When the PROCMODE pin is a low voltage, this signal functions as a high active interrupt. A high voltage denotes that an internal interrupt is active. In this mode, the pin is never tristated. The reset state of the pin in this mode is low voltage.
CLK	I	20 MHz input clock.
RESETN	I	A low voltage on this pin will reset the chip.
TRISTATEN	I	A low voltage on this pin will tristate all I/O and output signal drivers, and will disable the pad pullup resistors on all other pins. The TRISTATEN pin has a pullup resistor on the pin. For normal operation, it should be left open circuited on the PC board.
TEST	I	TEST mode enable. A high voltage enables IC production test mode. A low voltage disables test mode. This pin should always be grounded on the PC board.

7.2 PORT A DMA INTERFACE

PORT A DMA INTERFACE		
NAME	**TYPE**	**DESCRIPTION**
DREQA	I	Port A DMA channel DMA request input signal. Polarity is programmable through the DREQA POLARITY bit in the *Port A DMA Control* register. This signal is used to start and end DMA cycles.
DACKA	O	Port A DMA channel DMA acknowledge output. This signal pulses once for every DMA transfer into or out of Port A. The reset state of this output is high voltage.
DIRAIN	O	Port A DMA DIRection IN signal. This signal pulses during each DMA transfer into Port A, and can be used to enable an external device's output drivers. The polarity of DIRAIN is controlled by the DIRA POLARITY bit in the *Port A DMA Control* register. The reset state of this signal is high voltage.
DIRAOUT	O	Port A DMA DIRection OUT signal. This signal pulses during each DMA transfer out of Port A, and can be used by an external device to latch data out of Port A. The polarity of DIRAOUT is controlled by the DIRA POLARITY bit in the *Port A DMA Control* register. The reset state of this signal is high voltage.
DA[7:0]	I/O	Port A DMA bidirectional data bus. These pins have internal 10K ohm pullup resistors, which are enabled by the ENABLE DA BUS PULLUP bit in the *Port A DMA Control* register. The reset state of these pins has the output drivers tristated, and the internal pullup resistors disabled.
DAPTY	I/O	Bidirectional parity bit for the DA[7:0] bus. The ODD PARITY bit in the *Port A DMA Control* register selects odd or even parity on the DAPTY pin. This pin can be programmed to output parity when the AHA3101 drives the DA[7:0] bus, and to check parity when the AHA3101 reads data on the DA[7:0] bus. This pin has an internal 10K ohm pullup resistor, which is enabled with the ENABLE DAPTY PULLUP bit in the *Port A DMA Control* register. If parity on the DA[7:0] bus is not used, this pin should be left open circuited, with the pullup resistor enabled. The reset state of this pin is high impedance.

7.3 PORT B DMA INTERFACE

PORT B DMA INTERFACE		
NAME	**TYPE**	**DESCRIPTION**
DREQB	O	Port B DMA channel DMA request output signal. Polarity is programmable through the DREQB POLARITY bit in *Port B DMA Control 0* register. This signal is used to start and end DMA cycles. The reset state of this pin is low voltage.
DACKB	I	Port B DMA channel DMA acknowledge input. A low voltage on DACKB denotes that a DMA cycle is in process. This causes the DREQB output to go inactive. The rising edge of DACKB signals the end of a DMA cycle.
DBOE	I	Port B DMA Output Enable signal, for Port B DMA transfers in eight bit mode. When DBOE is inactive, it disables the Port B DMA data bus DB[7:0], and the Port B parity bit DBPTY[1], from driving out data during Port B DMA transfers in compression mode. When DBOE is active, it enables DB[7:0] and DBPTY[1] to drive data out during Port B DMA transfers in compression mode. Note that DB[7:0] and DBPTY[1] never drive data out in decompression mode. The polarity of DBOE is controlled by the DBOE POLARITY bit in *Port B DMA Control 0* register. DBOE is not used in sixteen bit mode for Port B DMA. This signal is intended to resolve data bus contention on DB[7:0] and DBPTY[1]. It should be connected to the active state if this function is not required.
DB[7:0]	I/O	Port B DMA bidirectional data bus. These pins have internal 10K ohm pullup resistors, which are enabled by the ENABLE DB BUS PULLUP bit in *Port B DMA Control 0* register. The reset state of these pins has the output drivers tristated, and the internal pullup resistors disabled.
DBPTY[1]	I/O	Bidirectional parity bit for the DB[7:0] bus. The ODD PARITY bit in *Port B DMA Control 0* register selects odd or even parity on the DBPTY[1] pin. This pin can be programmed to output parity when the AHA3101 drives the DB[7:0] bus, and to check parity when the AHA3101 reads data on the DB[7:0] bus. This pin has an internal 10K ohm pullup resistor, which is enabled with the ENABLE DBPTY PULLUP bit in *Port B DMA Control 0* register. If parity on the DB[7:0] bus is not used, this pin should be left open circuited, with the pullup resistor enabled. The reset state of this pin is high impedance.
DBPTY[0]	I/O	Bidirectional parity bit for Port B DMA transfers on the D[7:0] bus. The ODD PARITY bit in *Port B DMA Control 0* register selects odd or even parity on the DBPTY[0] pin. This pin can be programmed to output parity when the AHA3101 drives Port B DMA data on the D[7:0] bus, and to check parity when the AHA3101 reads Port B DMA data on the D[7:0] bus. This pin has an internal 10K ohm pullup resistor, which is enabled with the ENABLE DBPTY PULLUP bit in *Port B DMA Control 0* register. If Port B parity on the D[7:0] bus is not used, this pin should be left open circuited, with the pullup resistor enabled. The reset state of this pin is high impedance.

7.4 STATIC RAM INTERFACE

STATIC RAM INTERFACE		
NAME	**TYPE**	**DESCRIPTION**
MA15/CSN0	O	This pin has two functions, depending on the state of the RAMTYPE bit in the *Configuration Control* register. When the RAMTYPE bit is a one, this pin becomes a low active chip select output, for the 32K bytes of static RAM memory address from 00000 hex to 07FFF hex. When the RAMTYPE bit is zero, this output becomes the memory address bit 15. The state of this pin is undefined when the chip is reset.
MA16/CSN1	O	This pin has two functions, depending on the state of the RAMTYPE bit in the *Configuration Control* register. When the RAMTYPE bit is a one, this pin becomes a low active chip select output, for the 32K bytes of static RAM memory address from 08000 hex to 0FFFF hex. When the RAMTYPE bit is zero, this output becomes the memory address bit 16. The state of this pin is undefined when the chip is reset.
CSN2	O	When the RAMTYPE bit is a one, this pin is a low active chip select for the 32K bytes of memory address from 10000 hex to 17FFF hex. When the RAMTYPE bit is a zero, this pin is low voltage. The state of this pin is undefined when the chip is reset.
CSN3	O	When the RAMTYPE bit is a one, this pin is a low active chip select for the 32K bytes of memory address from 18000 hex to 1FFFF hex. When the RAMTYPE bit is a zero, this pin is low voltage. The state of this pin is undefined when the chip is reset.
MA[14:0]	O	Static RAM Memory Address bus. Fifteen bits address 32K bytes of memory address space. The address driven out in the reset state is undefined.
MD[7:0]	I/O	Bidirectional 8 bit static RAM Memory Data bus. These pins have internal pullup resistors, with a nominal resistance of 10K ohms. The reset state of these pins has the output drivers tristated, and the internal pullup resistors sourcing current onto the pin.
WEN	O	WriteEnableN. This signal is the write strobe for the static RAMs. A low voltage denotes a write cycle is occurring. The reset state of this pin is high voltage.
OEN	O	OutputEnableN. This signal is the output enable for the static RAMs. A low voltage denotes a read cycle is occurring. The reset state of this pin is high voltage.

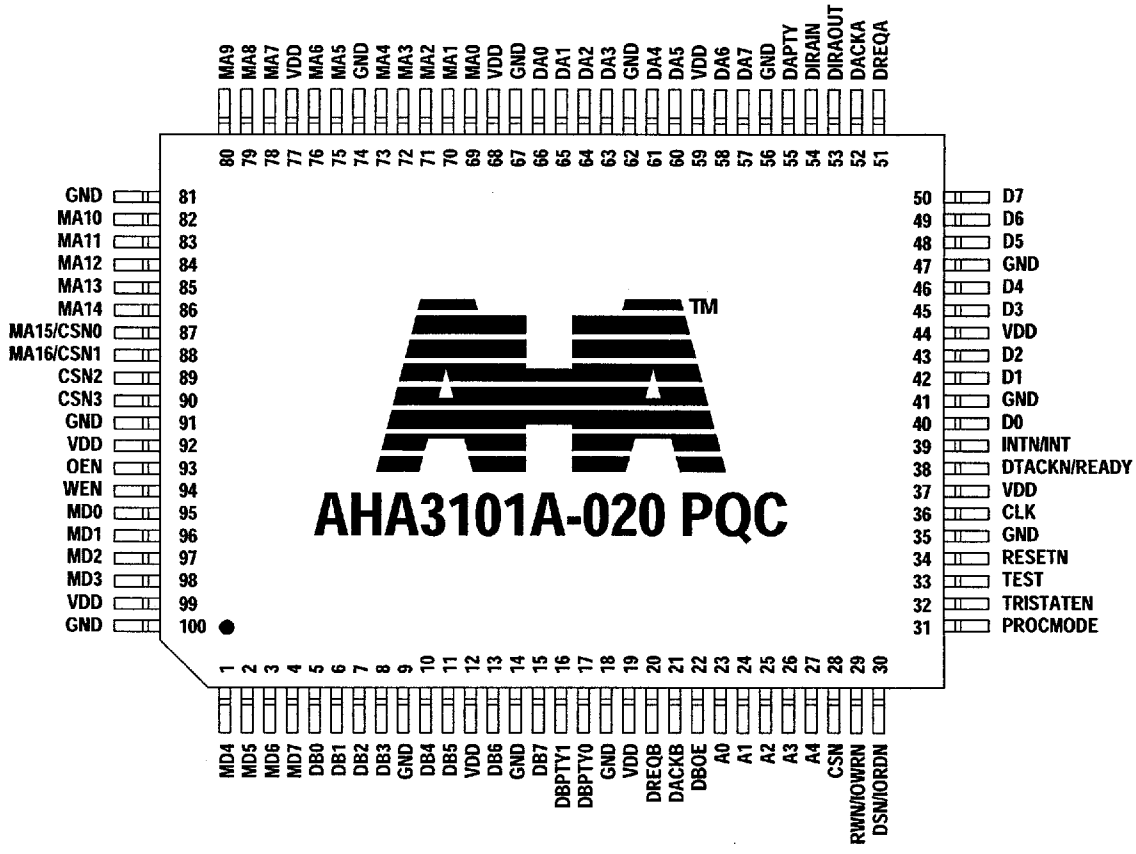
7.5 POWER SUPPLY

POWER SUPPLY		
NAME	**TYPE**	**DESCRIPTION**
VDD	Power	Positive power supply input
GND	Power	Ground input

8.0 PINOUT

The chip is packaged in an EIAJ standard 100 pin plastic quad flat pack (PQFP)

Figure 12: Pinout



9.0 ELECTRICAL SPECIFICATIONS

9.1 ABSOLUTE MAXIMUM RATINGS

SYMBOL	PARAMETER	MIN	MAX	UNITS
VDD	Power Supply Voltage		7.0	Volts
Vpin	Voltage Applied to Any Pin	-0.5	7.0	Volts

Absolute maximum voltage ratings are for voltage excursions which are transitory in nature

9.2 RECOMMENDED OPERATING CONDITIONS

SYMBOL	PARAMETER	MIN	MAX	UNITS
VDD	Power Supply Voltage	4.75	5.25	Volts
Ta	Operating Temperature	0	70	Degrees C

9.2.1 DC SPECIFICATIONS

SYMBOL	PARAMETER	CONDITIONS	MIN	MAX	UNITS
Vil	Input low voltage			0.8	Volts
Vih	Input high voltage		2.4		Volts
Vol	Output low voltage DTACKN/READY INTN/INT pins All other outputs	 Iol = 8.0 mAmps Iol = 4.0 mAmps		 0.4 0.4	 Volts Volts
Voh	Output high voltage	Ioh = -0.4 mAmps	2.4		Volts
Iil	Input low current	Vin = 0 Volts	-10		µAmps
Iih	Input high current	Vin = VDD Volts		10	µAmps
Iozl	Output tristate low current	Vout 0 Volts		10	µAmps
Iozh	Output tristate high current	Vout VDD Volts	-10		µAmps
IddA	Active Idd current		43	50	mAmps

9.2.2 AC SPECIFICATIONS

PIN NAMES	MAXIMUM CAPACITIVE LOAD
DTACKN/READY, D[7:0], INTN/INT, DB[7:0], DBPTY[1:0]	100 pF
DACKA, DIRAIN, DIRAOUT, DA[7:0], DAPTY, DREQB, WEN, OEN	75 pF
MA15/CSN0, MA16/CSN1, CSN2, CSN3, MA[14:0], MD[7:0]	60 pF

9.2.3 PIN CAPACITANCE

SYMBOL	PARAMETER	MIN	MAX	UNITS
Cin	Input capacitance		10	pF
Cout	Output capacitance		10	pF
Cio	I/O capacitance		10	p

10.0 TIMING SPECIFICATIONS

Figure 13: Clock Timing

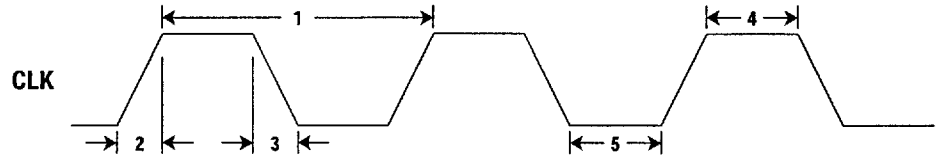


Table 6: Clock Timing Specifications

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	CLK period	50		nsec	
2	CLK rise time		5	nsec	
3	CLK fall time		5	nsec	
4	CLK high pulsewidth	20		nsec	
5	CLK low pulsewidth	20		nsec	

Figure 14: Processor Read Cycle Specifications - DSN, RWN Controlled

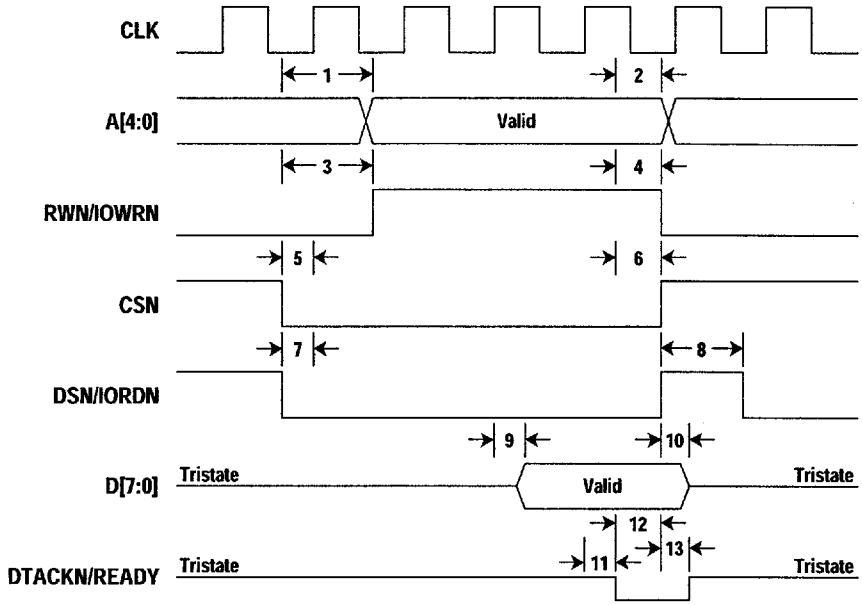


Table 7: Processor Read Cycle Specifications - DSN,RWN Controlled

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	DSN/IORDN low to A[4:0] valid		40	nsec	
2	A[4:0] hold from DTACKN/READY low	0		nsec	
3	DSN/IORDN low to RWN/IOWRN high		40	nsec	
4	RWN/IOWRN hold from DTACKN/READY low	0		nsec	
5	CSN low setup to CLK rise	10		nsec	1
6	CSN hold from DTACKN/READY low	10		nsec	
7	DSN/IORDN low setup to CLK rise	10		nsec	1
8	DSN/IORDN high pulsewidth	50		nsec	
9	CLK high to D[7:0] valid	0	30	nsec	
10	DSN/IORDN high to D[7:0] tristate	0	20	nsec	3
11	CLK rise to DTACKN/READY low	0	30	nsec	2
12	DTACKN/READY low to DSN/IORDN high	10		nsec	
13	DSN/IORDN high to DTACKN/READY tristate	0	20	nsec	3

Notes:

1) *CSN and DSN/IORDN can be asynchronous to CLK. They are synchronized internally to the rising edge of CLK. The cycle begins when both CSN and DSN/IORDN are low voltage. When DSN/IORDN is high, CSN is allowed to glitch. Once DSN/IORDN has gone low, CSN is not allowed to glitch. The cycle ends when DSN/IORDN rises to a high voltage.*
2) *The diagram shows the minimum time from DSN/IORDN low to DTACKN/READY low, for accesses to registers internal to the chip. Accesses to the Processor Buffer Data register, which require accessing the external static RAM, will take more clock cycles to complete.*
3) *This specification has been proven by worst case timing simulations. It is not fully tested in production.*

Figure 15: Processor Write Cycle Diagram - DSN, RWN Controlled

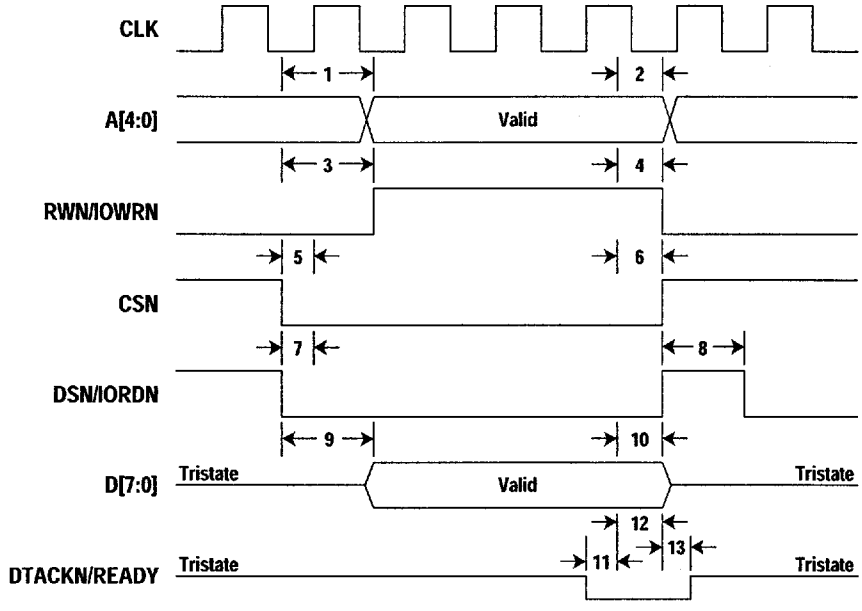


Table 8: Processor Write Cycle Specifications - DSN,RWN Controlled

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	DSN/IORDN low to A[4:0] valid		40	nsec	
2	A[4:0] hold from DTACKN/READY low	0		nsec	
3	DSN/IORDN low to RWN/IOWRN high		40	nsec	
4	RWN/IOWRN hold from DTACKN/READY low	0		nsec	
5	CSN low setup to CLK rise	10		nsec	1
6	CSN hold from DTACKN/READY low	10		nsec	
7	DSN/IORDN low setup to CLK rise	10		nsec	1
8	DSN/IORDN high pulsewidth	50		nsec	
9	DSN/IORDN low to D[7:0] valid		40	nsec	
10	D[7:0] hold from DTACKN/READY low	0		nsec	
11	CLK rise to DTACKN/READY low	0	30	nsec	2
12	DTACKN/READY low to DSN/IORDN high	10		nsec	
13	DSN/IORDN high to DTACKN/READY tristate	0	20	nsec	3

Notes:

1) *CSN and DSN/IORDN can be asynchronous to CLK. They are synchronized internally to the rising edge of CLK. The cycle begins when both CSN and DSN/IORDN are low voltage. When DSN/IORDN is high, CSN is allowed to glitch. Once DSN/IORDN has gone low, CSN is not allowed to glitch. The cycle ends when DSN/IORDN rises to a high voltage.*
2) *The diagram shows the minimum time from DSN/IORDN low to DTACKN/READY low, for accesses to registers internal to the chip. Accesses to the Processor Buffer Data register, which require accessing the external static RAM, will take more clock cycles to complete.*
3) *This specification has been proven by worst case timing simulations. It is not fully tested in production.*

Figure 16: Processor Read Cycle Diagram - IORDN Controlled

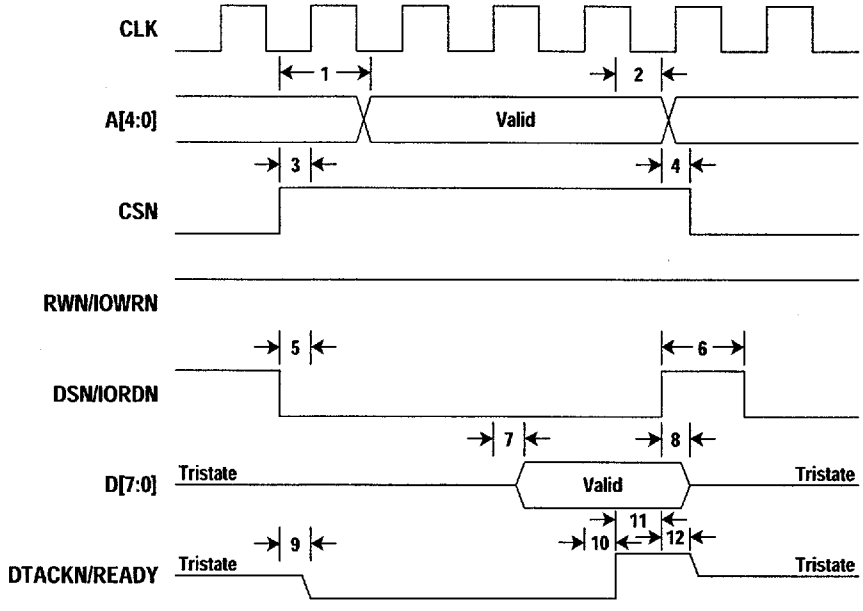


Table 9: Processor Read Cycle Specifications - IORDN Controlled

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	DSN/IORDN low to A[4:0] valid		40	nsec	
2	A[4:0] hold from DTACKN/READY high	0		nsec	
3	CSN low setup to CLK rise	10		nsec	1
4	CSN hold from DSN/IORDN high	0		nsec	
5	DSN/IORDN low setup to CLK rise	10		nsec	1
6	DSN/IORDN high pulsewidth	50		nsec	
7	CLK high to D[7:0] valid	0	30	nsec	
8	DSN/IORDN high to D[7:0] tristate	0	20	nsec	3
9	DSN/IORDN low and CSN low to DTACKN/READY low	0	30	nsec	
10	CLK rise to DTACKN/READY high	0	30	nsec	2
11	DTACKN/READY high to DSN/IORDN high	10		nsec	
12	DSN/IORDN high to DTACKN/READY tristate	0	20	nsec	3

Notes:

1)CSN and DSN/IORDN can be asynchronous to CLK. They are synchronized internally to the rising edge of CLK. The cycle begins when both CSN and DSN/IORDN are low voltage. When DSN/IORDN is high, CSN is allowed to glitch. Once DSN/IORDN has gone low, CSN is not allowed to glitch. The cycle ends when DSN/IORDN rises to a high voltage.

2)The diagram shows the minimum time from DSN/IORDN low to DTACKN/READY high, for accesses to registers internal to the chip. Accesses to the Processor Buffer Data register, which require accessing the external static RAM, will take more clock cycles to complete.

3)This specification has been proven by worst case timing simulations. It is not fully tested in production.

Figure 17: Processor Write Cycle Diagram - IOWRN Controlled

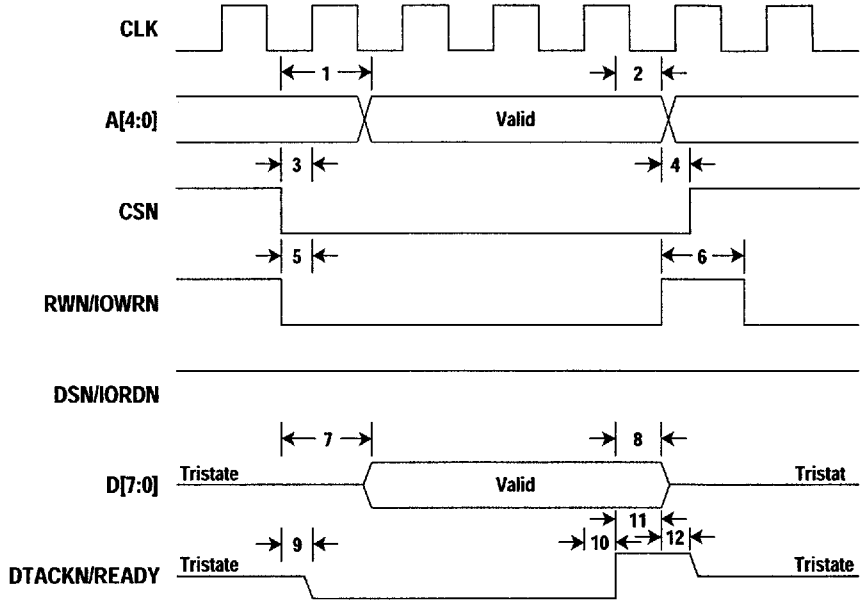


Table 10: Processor Write Cycle Specifications - IOWRN Controlled

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	RWN/IOWRN low to A[4:0] valid		40	nsec	
2	A[4:0] hold from DTACKN/READY low	0		nsec	
3	CSN low setup to CLK rise	10		nsec	1
4	CSN hold from RWN/IOWRN high	0		nsec	
5	RWN/IOWRN low setup to CLK rise	10		nsec	1
6	RWN/IOWRN high pulsewidth	50		nsec	
7	RWN/IOWRN low to D[7:0] valid		40	nsec	
8	D[7:0] hold from DTACKN/READY high	0		nsec	
9	RWN/IOWRN low, CSN low to DTACKN/READY low	0	30	nsec	
10	CLK rise to DTACKN/READY high	0	30	nsec	2
11	DTACKN/READY high to RWN/IOWRN high	10		nsec	
12	RWN/IOWRN high to DTACKN/READY tristate	0	20	nsec	3

Notes:

1) *CSN and RWN/IOWRN can be asynchronous to CLK. They are synchronized internally to the rising edge of CLK. The cycle begins when both CSN and RWN/IOWRN are low voltage. When RWN/IOWRN is high, CSN is allowed to glitch. Once RWN/IOWRN has gone low, CSN is not allowed to glitch. The cycle ends when RWN/IOWRN rises to a high voltage.*
2) *The diagram shows the minimum time from RWN/IOWRN low to DTACKN/READY high, for accesses to registers internal to the chip. Accesses to the Processor Buffer Data register, which require accessing the external static RAM, will take more clock cycles to complete.*
3) *This specification has been proven by worst case timing simulations. It is not fully tested in production.*

Figure 18: Static RAM Read Cycle Diagram

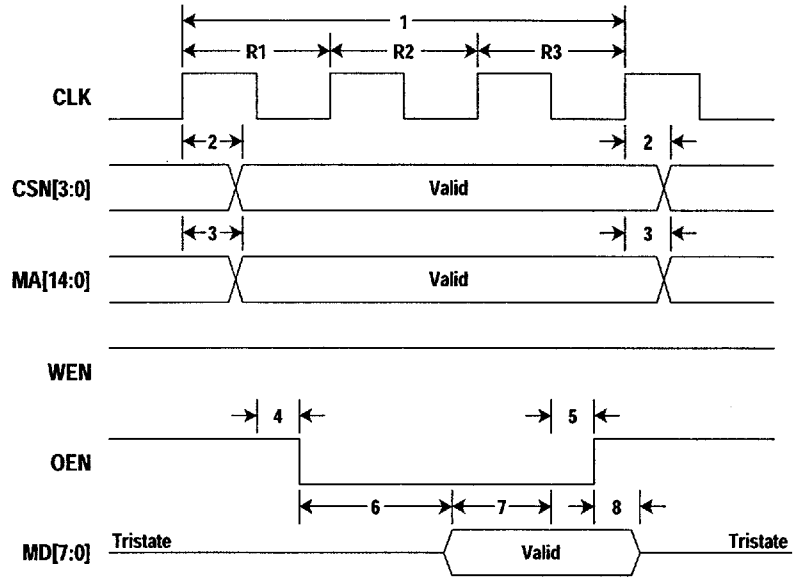


Table 11: Static RAM Read Cycle Specifications

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	Read cycle time	2	3	clocks	1
2	CLK rise to CSN[3:0] valid	0	20	nsec	2
3	CLK rise to MA[14:0] valid	0	20	nsec	
4	CLK fall to OEN low	0	15	nsec	
5	CLK fall to OEN high	0	15	nsec	
6	OEN low to MD[7:0] driven	0		nsec	
7	MD[7:0] valid setup to CLK fall	5		nsec	
8	OEN rise to MD[7:0] tristate	0	35	nsec	

Notes:

1) *The RAM WAIT STATE bit in the Configuration Control register determines the number of wait states for the static RAM interface. A zero causes the read cycle sequence to be cycle R1, followed by cycle R3, for a cycle time of two clock cycles. A one causes the read cycle sequence to be cycle R1, R2, and then R3, for a cycle time of three clock cycles.*
2) *CSN[3:0] denotes pins CSN3, CSN2, MA16/CSN1, and MA15/CSN0.*

Figure 19: Static RAM Write Cycle Diagram

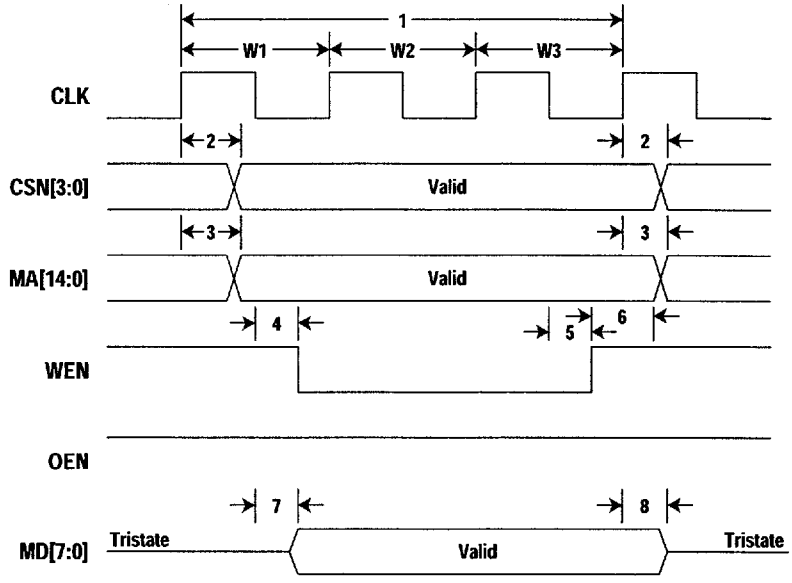


Table 12: Static RAM Write Cycle Specifications

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	Write cycle time	2	3	clocks	1
2	CLK rise to CSN[3:0] valid	0	20	nsec	2
3	CLK rise to MA[14:0] valid	0	20	nsec	
4	CLK fall to WEN low	0	15	nsec	
5	CLK fall to WEN high	0	15	nsec	
6	MA[14:0] hold form WEN rise	10		nsec	
7	CLK fall to MD[7:0] valid	0	20	nsec	
8	CLK rise to MD[7:0] tristate	0	20	nsec	3

Notes:

1) *The RAM WAIT STATE bit in the Configuration Control register determines the number of wait states for the static RAM interface. A zero causes the write cycle sequence to be cycle W1, followed by cycle W3, for a cycle time of two clock cycles. A one causes the write cycle sequence to be cycle W1, W2, and then W3, for a cycle time of three clock cycles.*
2) *CSN[3:0] denotes pins CSN3, CSN2, MA16/CSN1, and MA15/CSN0.*
3) *This specification has been proven by worst case timing simulations. It is not fully tested in production.*

Figure 20: DMA Transfer Timing for Data Out of Port A

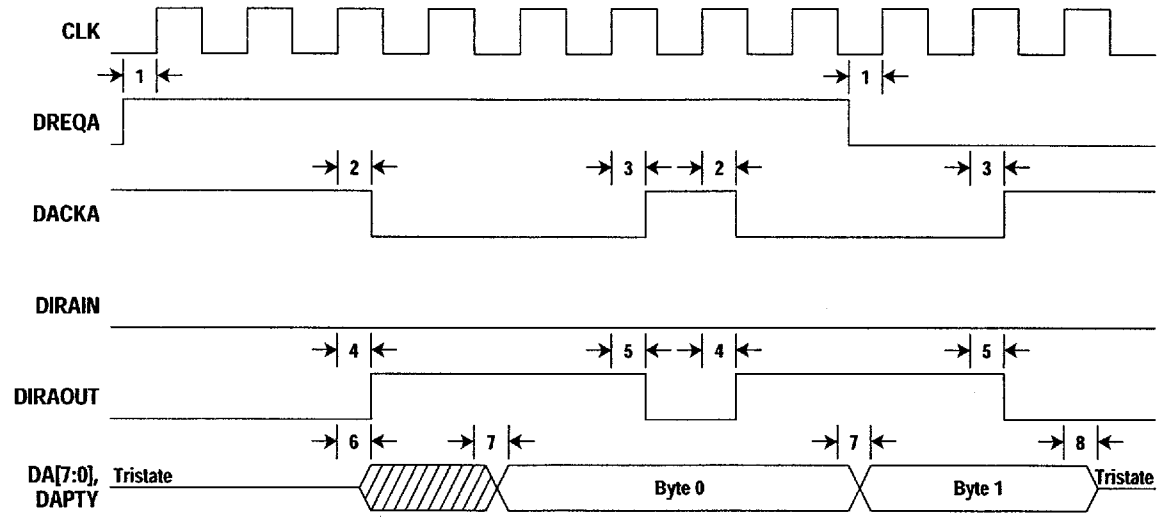


Table 13: DMA Transfer Timing for Data Out of Port A

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	DREQA valid setup to CLK rise	5		nsec	1
2	CLK rise to DACKA low	0	30	nsec	
3	CLK rise to DACKA high	0	30	nsec	
4	CLK rise to DIRAOUT high	0	30	nsec	
5	CLK rise to DIRAOUT low	0	30	nsec	
6	CLK rise to DA[7:0], DAPTY driven	0		nsec	2
7	CLK fall to DA[7:0], DAPTY valid	0	30	nsec	2
8	CLK rise to DA[7:0], DAPTY tristate	0	30	nsec	2, 3

Notes:

1) *The DREQA signal can be asynchronous to the CLK signal. It is internally synchronized to the rising edge of CLK. If the setup time is met, the maximum data transfer rate will be achieved.*
2) *If the ENABLE PARITY bit in the Port A DMA Control register is zero (inactive), the DAPTY pin will always be tristated.*
3) *This specification has been proven by worst case timing simulations. It is not fully tested in production.*

The timing diagram is for a transfer of two consecutive bytes of data.

The signals DACKA, DIRAOUT, DIRAIN, DAPTY, and DA[7:0] are chip outputs. DREQA is a chip input. The polarity of DREQA is controlled by the DREQA POLARITY bit in the *Port A DMA Control* register. In the above diagram, the DREQA POLARITY bit is a one, making DREQA active high. The polarity of DIRAOUT and DIRAIN is controlled by the DIRA POLARITY bit in the *Port A DMA Control* register. In the above diagram, the DIRA POLARITY bit is a one, making DIRAOUT and DIRAIN active high. When the static RAM buffer is enabled, arbitration delay for the static RAM bus may insert clock cycles after DACKA goes low, until Port A data out is valid.

Figure 21: DMA Transfer Timing for Data Into Port A

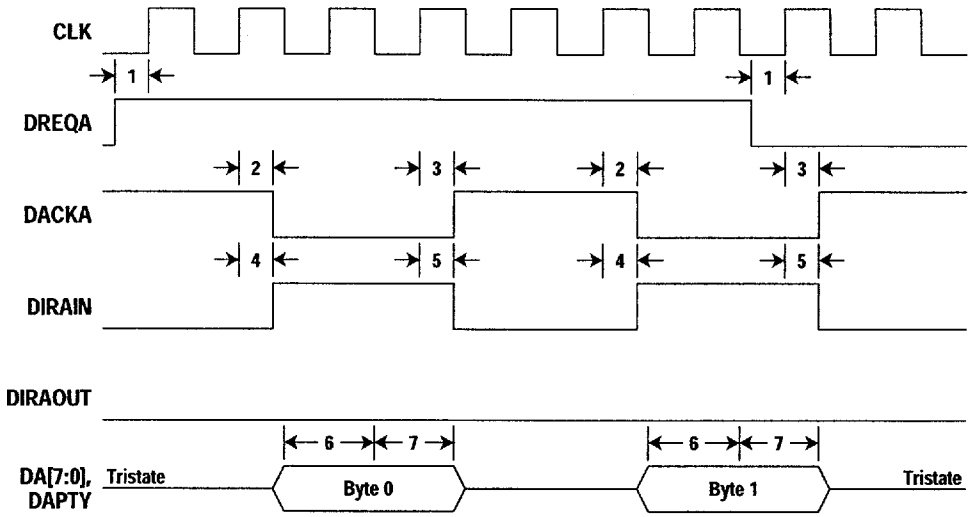


Table 14: DMA Transfer Timing for Data Into Port A

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	DREQA valid setup to CLK rise	5		nsec	1
2	CLK rise to DACKA low	0	30	nsec	
3	CLK rise to DACKA high	0	30	nsec	
4	CLK rise to DIRAIN high	0	30	nsec	
5	CLK rise to DIRAIN low	0	30	nsec	
6	DA[7:0], DAPTY setup to CLK fall	0		nsec	
7	DA[7:0], DAPTY hold from CLK fall	20		nsec	

Notes:

1) The DREQA signal can be asynchronous to the CLK signal. It is internally synchronized to the rising edge of CLK. If the setup time is met, the maximum data transfer rate will be achieved.

The Timing diagram is for a transfer of two consecutive bytes of data. The signals DACKA, DIRAOUT, and DIRAIN are chip outputs. DREQA, DAPTY and DA[7:0] are chip inputs.

The polarity of DREQA is controlled by the DREQA POLARITY bit in the *Port A DMA Control* register. In the above diagram, the DREQA POLARITY bit is a one, making DREQA active high. The polarity of DIRAOUT and DIRAIN are controlled by the DIRA POLARITY bit in the *Port A DMA Control* register. In the above diagram, the DIRA POLARITY bit is a one, making DIRAOUT and DIRAIN active high.

When the static RAM buffer is enabled, arbitration delay may insert clock cycles after Port A data in has been latched, and before DACKA rises.

Figure 22: 8-bit DMA Transfer Timing for Data Out of Port B

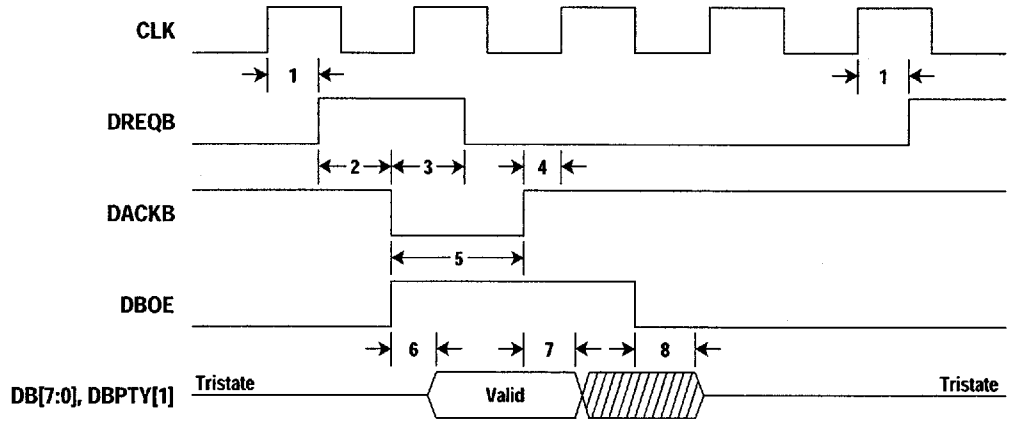


Table 15: 8-bit DMA Transfer Timing for Data Out of Port B

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	CLK rise to DREQB high	0	30	nsec	
2	DREQB high to DACKB low	0		nsec	
3	DACKB low to DREQB low	0	25	nsec	
4	DACKB high setup to CLK rise	5		nsec	1
5	DACKB low pulsewidth	40		nsec	
6	DBOE rise to DB[7:0], DBPTY[1] valid	0	30	nsec	3
7	DB[7:0], DBPTY[1] hold form DACKB rise	50		nsec	2, 3
8	DBOE fall to DB[7:0], DBPTY[1] tristate	0	30	nsec	3, 4

Notes:

1) *The DACKB signal can be asynchronous to the CLK signal. It is internally synchronized to the rising edge of CLK. If the setup time is met, the maximum data transfer rate will be achieved.*
2) *The function of DBOE is only to enable the DB[7:0] and DBPTY[1] output drivers. The DMA cycle ends with the rising edge of DACKB. If DBOE is active after DACKB rises, data out is guaranteed to be held for one clock cycle, after which it may be updated in preparation for the next DMA cycle.*
3) *If the ENABLE PARITY bit in Port B DMA Control 0 register is zero (inactive), the DBPTY[1] pin will always be tristated.*
4) *This specification has been proven by worst case timing simulations. It is not fully tested in production.*

The signals DREQB, DBPTY[1] and DB[7:0] are chip outputs. DACKB, and DBOE are chip inputs.

The polarity of DREQB is controlled by the DREQB POLARITY bit in Port B DMA Control Regiter 0. In the above diagram, the DREQB POLARITY bit is a one, making DREQB active high.

The polarity of DBOE is controlled by the DBOE POLARITY bit in Port B DMA Control Reqister 0. In the above diagram, the DBOE POLARITY bit is a one, making DBOE active high.

Figure 23: 8-bit DMA Transfer Timing for Data Into Port B

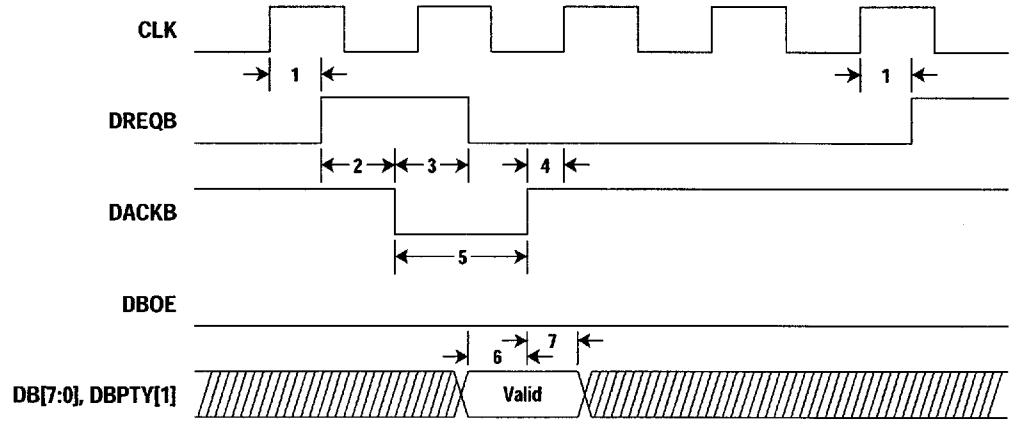


Table 16: 8-bit DMA Transfer Timing for Data Into Port B

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	CLK rise to DREQB high	0	30	nsec	
2	DREQB high to DACKB low	0		nsec	
3	DACKB low to DREQB low	0	25	nsec	
4	DACKB high setup to CLK rise	5		nsec	1
5	DACKB low pulsewidth	40		nsec	
6	DB[7:0], DBPTY[1] setup to DACKB rise	5		nsec	
7	DB[7:0], DBPTY[1] hold from DACKB rise	5		nsec	

Notes:

1) The DACKB signal can be asynchronous to the CLK signal. It is internally synchronized to the rising edge of CLK. If the setup time is met, the maximum data transfer rate will be achieved.

The signal DREQB is a chip output. DACKB, DBOE, DBPTY[1] and DB[7:0] are chip inputs.

The polarity of DREQB is controlled by the DREQB POLARITY bit in *Port B DMA Control* register 0. In Figure 23, the DREQB POLARITY bit is a one, making DREQB active high.

The polarity of DBOE is controlled by the DBOE POLARITY bit in *Port B DMA Control 0* register. In Figure 23, the DBOE POLARITY bit is a one, making DBOE active high.

Figure 24: 16-bit DMA Transfer Timing for Data Out of Port B - IORDN Controlled

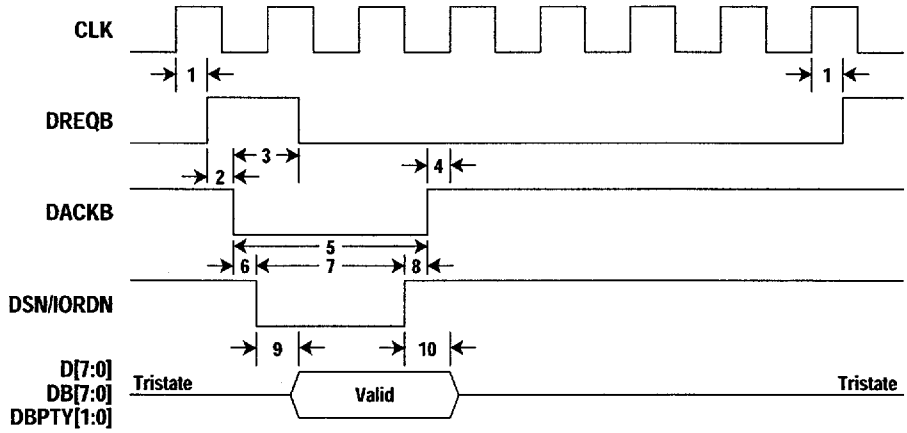


Table 17: 16-bit DMA Transfer Timing for Data Out of Port B - IORDN Controlled

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	CLK rise to DREQB high	0	30	nsec	
2	DREQB high to DACKB low	0		nsec	
3	DACKB low to DREQB low	0	25	nsec	
4	DACKB high setup to CLK rise	5		nsec	1
5	DACKB low pulsewidth	100		nsec	
6	DACKB low to DSN/IORDN low	10		nsec	
7	DSN/IORDN low pulsewidth	80		nsec	
8	DSN/IORDN high to DACKB high	10		nsec	
9	DSN/IORDN low to D[7:0], DB[7:0], DBPTY[1:0] valid	0	30	nsec	2
10	DSN/IORDN high D[7:0], DB[7:0], DBPTY[1:0] tristate	0	30	nsec	2, 3

Notes:

1) *The DACKB signal can be asynchronous to the CLK signal. It is internally synchronized to the rising edge of CLK. If the setup time is met, the maximum data transfer rate will be achieved.*
2) *If the ENABLE PARITY bit in Port B DMA Control 0 register is zero (inactive), the DBPTY[1:0] pins will always be tristated.*
3) *This specification has been proven by worst case timing simulations.*

The signals DREQB, DBPTY[1:0], DB[7:0], D[7:0] are chip outputs. DACKB, and DSN/IORDN are chip inputs. The polarity of DREQB is controlled by the DREQB POLARITY bit in *Port B DMA Control 0* register. In the above diagram, the DREQB POLARITY bit is a one, making DREQB active high. The chip select pin, CSN, must be high voltage (inactive) during Port B DMA cycles in 16 bit mode.

Figure 25: 16-bit DMA Transfer Timing for Data Into Port B - IOWRN Controlled

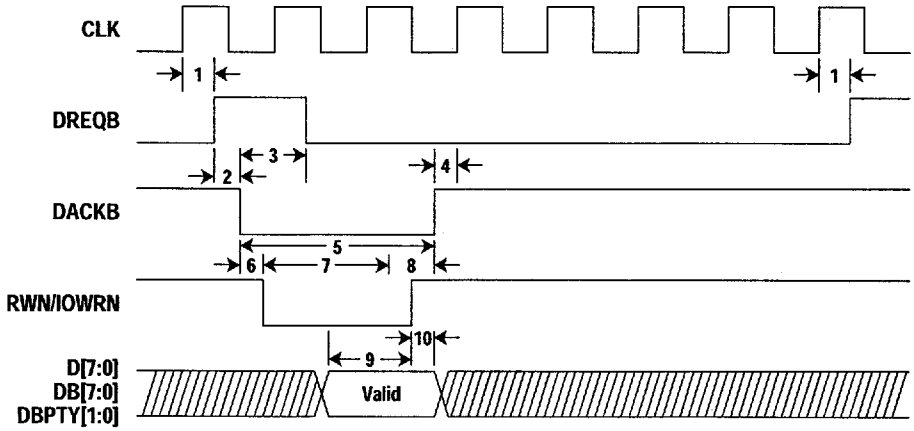


Table 18: 16-bit DMA Transfer Timing for Data Into Port B - IOWRN Controlled

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	CLK rise to DREQB high	0	30	nsec	
2	DREQB high to DACKB low	0		nsec	
3	DACKB low to DREQB low	0	25	nsec	
4	DACKB high setup to CLK rise	5		nsec	1
5	DACKB low pulsewidth	100		nsec	
6	DACKB low to RWN/IOWRN low	10		nsec	
7	RWN/IOWRN low pulsewidth	80		nsec	
8	RWN/IOWRN high to DACKB high	10		nsec	
9	D[7:0], DB[7:0], DBPTY[1:0] setup to RWN/IOWRN rise	5		nsec	
10	D[7:0], DB[7:0], DBPTY[1:0] hold RWN/IOWRN rise	5		nsec	

Notes:

1) The DACKB signal can be asynchronous to the CLK signal. It is internally synchronized to the rising edge of CLK. If the setup time is met, the maximum data transfer rate will be achieved.

The signal DREQB is a chip output. DACKB, RWN/IOWRN, D[7:0], DBPTY[1:0] and DB[7:0] are chip inputs.

The polarity of DREQB is controlled by the DREQB POLARITY bit in *Port B DMA Control 0* register. In the above diagram, the DREQB POLARITY bit is a one, making DREQB active high.

The chip select pin, CSN, must be high voltage (inactive) during Port B DMA cycles in 16 bit mode.

Figure 26: 16-bit DMA Transfer Timing for Data Out of Port B - DSN, RWN Controlled

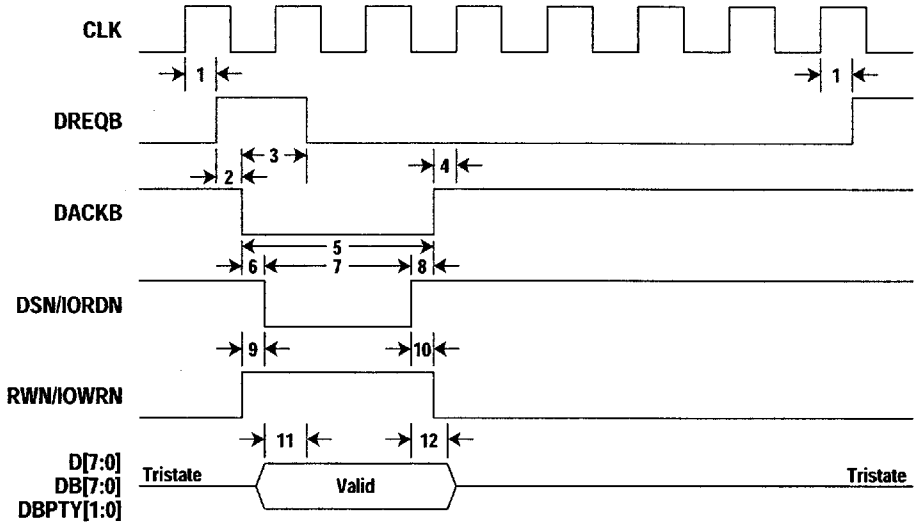


Table 19: 16-bit DMA Transfer Timing for Data Out of Port B - DSN,RWN Controlled

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	CLK rise to DREQB high	0	30	nsec	
2	DREQB high to DACKB low	0		nsec	
3	DACKB low to DREQB low	0	25	nsec	
4	DACKB high setup to CLK rise	5		nsec	1
5	DACKB low pulsewidth	100		nsec	
6	DACKB low to DSN/IORDN low	10		nsec	
7	DSN/IORDN low pulsewidth	80		nsec	
8	DSN/IORDN high to DACKB high	10		nsec	
9	RWN/IOWRN high setup to DSN/IORDN low	10		nsec	
10	RWN/IORDN high hold from DSN/IORDN high	10		nsec	
11	DSN/IOWRN low to D[7:0], DB[7:0], DBPTY[1:0] valid	0	30	nsec	2
12	DSN/IORDN high D[7:0], DB[7:0], DBPTY[1:0] tristate	0	30	nsec	2, 3

Notes:

1) *The DACKB signal can be asynchronous to the CLK signal. It is internally synchronized to the rising edge of CLK. If the setup time is met, the maximum data transfer rate will be achieved.*
2) *If the ENABLE PARITY bit in Port B DMA Control 0 register is zero (inactive), the DBPTY[1:0] pins will always be tristated.*
3) *This specification has been proven by worst case timing simulations. It is not fully tested in production.*

The signals DREQB, DBPTY[1:0], DB[7:0] are chip outputs. DACKB, DSN/IORDN, and RWN/IOWRN are chip inputs.

The polarity of DREQB is controlled by the DREQB POLARITY bit in *Port B DMA Control 0* register. In the above diagram (Figure 26), the DREQB POLARITY bit is a one, making DREQB active high.

The chip select pin, CSN, must be high voltage (inactive) during Port B DMA cycles in 16 bit mode.

Figure 27: 16-bit DMA Transfer Timing for Data Into Port B - DSN, RWN Controlled

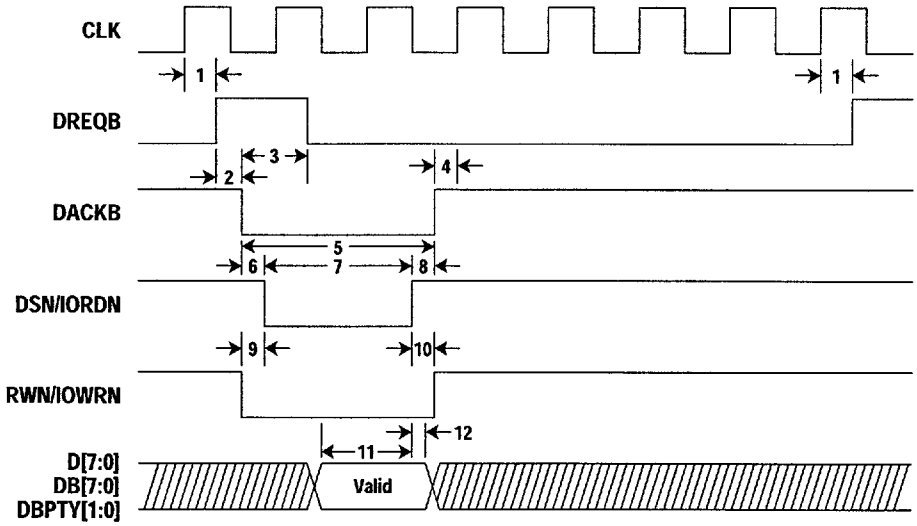


Table 20: 16-bit DMA Transfer Timing for Data Into Port B - DSN,RWN

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	CLK rise to DREQB high	0	30	nsec	
2	DREQB high to DACKB low	0		nsec	
3	DACKB low to DREQB low	0	25	nsec	
4	DACKB high setup to CLK rise	5		nsec	1
5	DACKB low pulsewidth	100		nsec	
6	DACKB low to DSN/IORDN low	10		nsec	
7	DSN/IORDN low pulsewidth	80		nsec	
8	DSN/IORDN high to DACKB high	10		nsec	
9	RWN/IOWRN low setup to DSN/IORDN low	10		nsec	
10	RWN/IOWRN low hold from DSN/IORDN high	10		nsec	
11	D[7:0], DB[7:0], DBPTY[1:0] setup to DSN/IORDN rise	5		nsec	
12	D[7:0], DB[7:0], DBPTY[1:0] hold DSN/IORDN rise	5		nsec	

Notes:

1) The DACKB signal can be asynchronous to the CLK signal. It is internally synchronized to the rising edge of CLK. If the setup time is met, the maximum data transfer rate will be achieved.

The signal DREQB is a chip output. DACKB, DSN/IORDN, RWN/IOWRN, D[7:0], DBPTY[1:0] and DB[7:0] are chip inputs. The polarity of DREQB is controlled by the DREQB POLARITY bit in *Port B DMA Control 0* register. In the above diagram, the DREQB POLARITY bit is a one, making DREQB active high. The chip select pin, CSN, must be high voltage (inactive) during Port B DMA cycles in 16 bit mode.

Figure 28: Interrupt Timing

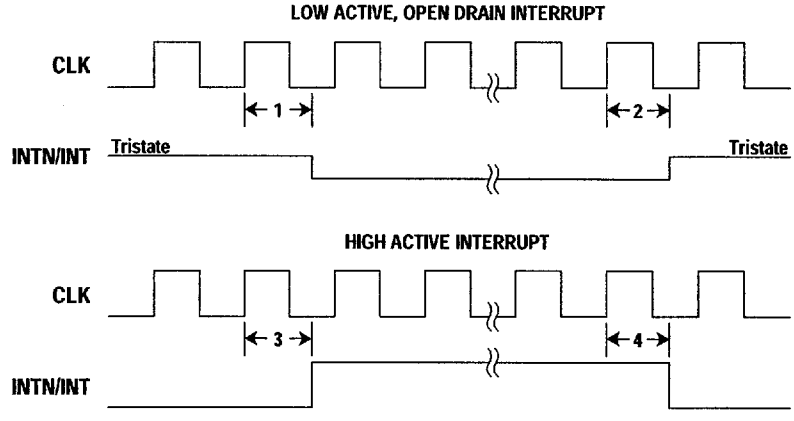


Table 21: Interrupt Timing Specifications

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	CLK rise to INTN/INT low		30	nsec	1
2	CLK rise to INTN/INT tristate		30	nsec	1, 3
3	CLK rise to INTN/INT high		30	nsec	2
4	CLK rise to INTN/INT low		30	nsec	2

Notes:

1) The INTN/INT pin is a low active, open drain interrupt when the PROCMODE input pin is at high voltage.

2) The INTN/INT pin is a high active interrupt when the PROCMODE input pin is at low voltage.

3) This specification has been proven by worst case timing simulations. It is not fully tested in production.

Figure 29: Reset Timing

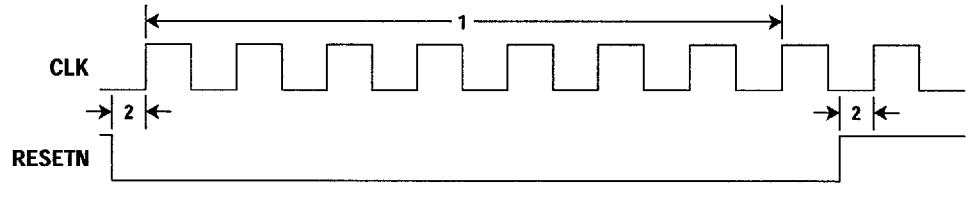


Table 22: Reset Timing Specifications

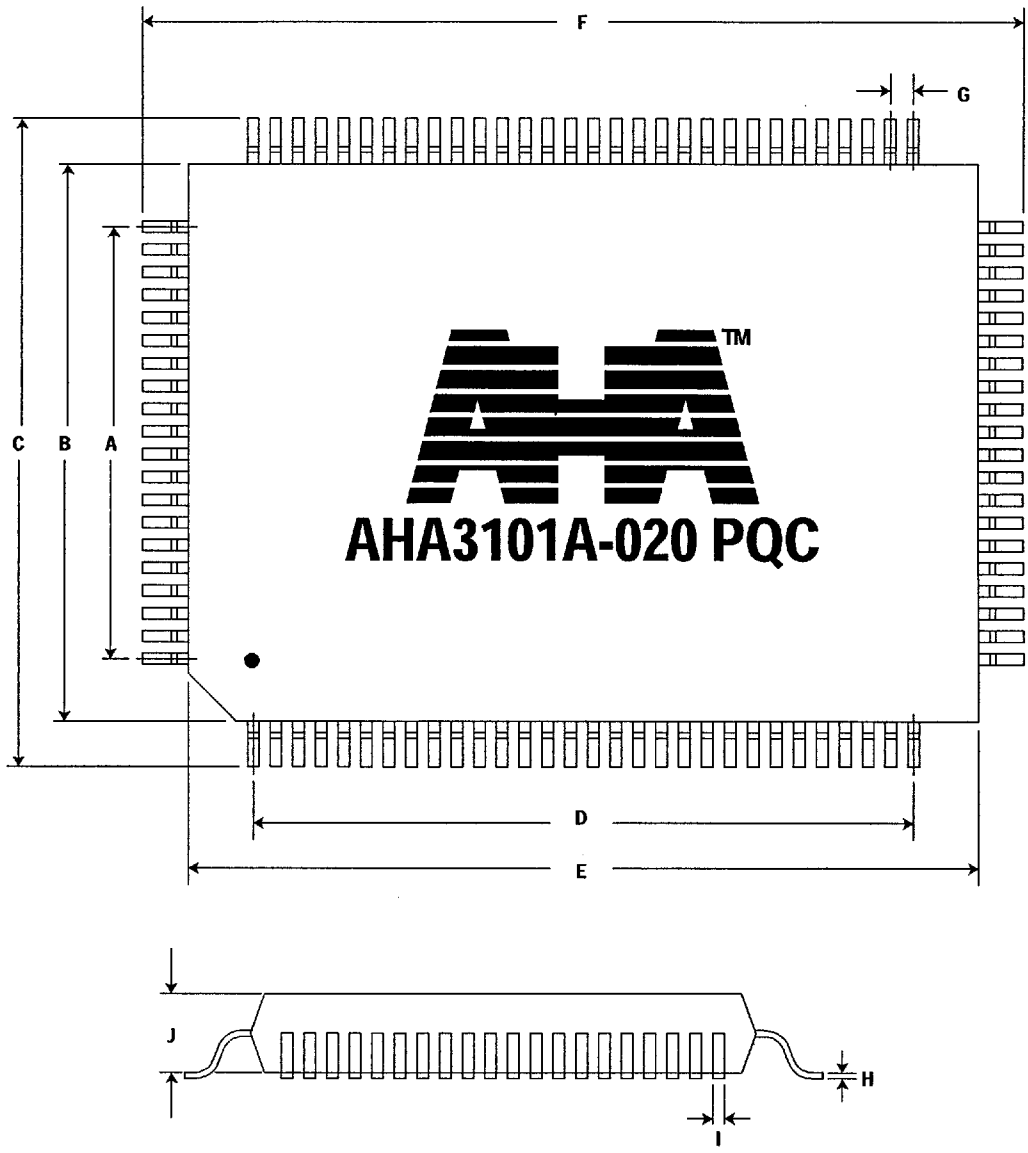
NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	RESETN low pulsewidth	8		clocks	
2	RESETN setup to CLK rise	10		nsec	1

Notes:

1) The RESETN signal can be asynchronous to the CLK signal. It is internally synchronized to the rising edge of CLK.

11.0 AHA3101 PACKAGE SPECIFICATIONS

Figure 30: Package Specifications



AHA3101 CHIP DIMENSIONS									
A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*
12.35	14.0±0.1	17.9±0.4	18.85	20.0±0.1	23.9±0.4	0.65±0.12	0.15±0.050	0.3±0.1	2.75±0.10

Notes: All dimensions are in millimeters
Package type is 100 pin quad flat pack

12.0 ORDERING INFORMATION

PART NUMBER	DESCRIPTION
AHA3101A-020 PQC	Data Compression Coprocessor IC, 20MHz Clock, Commercial Specifications 0°-70°C

EXAMPLE PART NUMBER DESCRIPTION						
AHA	**3101**	**A-**	**020**	**P**	**Q**	**C**
Manufacturer	Device Number	Revision Level	Speed Designation in MHz	Package Material (P-Plastic)	Package Type (Q=Quad Flat Pack)	Test Specification (C-Commercial 0° to 70°)

13.0 RELATED TECHNICAL PUBLICATIONS

DOCUMENT #	DESCRIPTION
PB3101	AHA Product Brief – AHA3101 2.5 MBytes/sec DCLZ Data Compression Coprocessor IC
PB3210B	AHA Product Brief – AHA3210B 10 MBytes/sec DCLZ Data Compression Coprocessor IC
PB3211	AHA Product Brief – AHA3211 20 MBytes/sec DCLZ Data Compression Coprocessor IC
PS3101	AHA Product Specification – AHA3101 2.5 MBytes/sec DCLZ Data Compression Coprocessor IC
PS3210B	AHA Product Specification – AHA3210B 10 MBytes/sec DCLZ Data Compression Coprocessor IC
PS3211	AHA Product Specification – AHA3211 20 MBytes/sec DCLZ Data Compression Coprocessor IC
ABDC02	AHA Application Brief – DCLZ Software Licensing Procedure
ABSTD1	AHA Application Brief – AHA Data Compression and Forward Error Correction Standards
ANDC01	AHA Application Note – Primer: Data Compression Lempel-Ziv (DCLZ)
ANDC02	AHA Application Note – Programming the AHA3101
ANDC03	AHA Application Note – Interfacing the AHA3101 to the NCR SCSI Controller
ANDC04	AHA Application Note – AHA3210 Data Managment Application Note
ANDC05	AHA Application Note – AHA3210 Designer's Guide
ANDC07	AHA Application Note – DCLZ Evaluation Software
ANDC09	AHA Application Note – Error Detection and Recovery in Data Compression System Using AHA3210
ANDC10	AHA Application Note – Compression Performance: DCLZ Algorithm on the Calgary Corpus
GLGEN1	General Glossary of Terms
RAECMA-0791	"DCLZ Emerges as an Open DC Standard," article reprint *Computer Technology Review*, Summer 1991
DCEVAL	DCLZ Evaluation Software (Windows™)