Product Specification

AHA4501 Astro

36 Mbits/sec Turbo Product Code Encoder/Decoder, 3.3V

2365 NE Hopkins Court
Pullman, WA 99163-5601
tel: 509.334.1000
fax: 509.334.9000
e-mail: sales@aha.com
www.aha.com

AHA
advancedhardwarearchitectures

This product and the algorithm are covered under multiple patents pending.

Table of Contents

Figures

Tables

1.0 INTRODUCTION

The AHA4501 is the first single-chip Forward Error Correction LSI device using Turbo Product Codes (TPC). The device operates as a block code encoder at the input or a block code decoder at the output of a communication channel. The device supports various programmable features, such as block size, codes and code rates, to optimize various communication channel needs for error performance and data throughput. Turbo Product Codes offer a higher performance alternative to Reed-Solomon or Reed-Solomon concatenated with Viterbi error correction methods.

When encoding, the device appends the Error Correction Code (ECC) bits to the blocks, and outputs the encoded blocks. When decoding, the device accepts soft decision values and stores the data as a block in its internal RAM. The block is then decoded iteratively by running it through the device's soft in/soft out (SISO) decoder. The device iterates to the maximum programmed iteration limit. The decoded block is then output through the device output data port.

This specification describes the functional operation, programming, timing and ordering information. Please contact AHA for additional support material, including evaluation software and relevant technical publications; or visit our website at http://www.aha.com.

1.1 CONVENTIONS, NOTATIONS AND DEFINITIONS

- Code block – A data stream to be encoded or decoded is segmented into blocks for processing by the AHA4501. Data in a code block is configured as a 2D or 3D array.
- Axis iteration – Decoding one axis of an array (all x rows, all y columns, or all z columns).
- Full iteration – Decoding all axes of an array (all rows and columns).
- Soft value – Input to the decoder from either an Analog/Digital Converter(ADC) or digital demodulator.
- Code rate – Ratio of the number of data bits to the number of data and ECC bits.
- Data rate – The rate at which unencoded data is input to the device when encoding or output from the device when decoding.
- Channel Rate – The rate at which encoded data is output from the device when encoding or input to the device when decoding. Note that system channel rate may be different due to external synchronization marks or other overhead.
- Original Array (OA) – The soft decision input data array. Data is stored as a 6 bit soft value per location to support the maximum 6 bit input quantization.
- Intermediate Storage Array (ISA) – The storage array for data between iterating.
- Hard Decision Array (HDA) – The hard decision output. Data is stored as one bit per location.
- $(n_1,k_1)x(n_2,k_2)$ – A general representation of a 2D block code for use in the descriptions to follow in this specification. For example, in a (64,57)x(64,57) code; n_1,n_2=64 represents the length of the data + ECC bits, and k_1,k_2=57 represents the length of only the data bits. 3D codes are represented as $(n_1,k_1)x(n_2,k_2)x(n_3,k_3)$
- Vector – One row or column of data in a block.
- Latency – The time from the first bit of a block in to first bit of the same block out.
- Active low signals have an "N" appended to the end of the signal name. For example, MCSN and RESETN.
- Hex values are represented with a prefix of "0x", such as register "0x00". Binary values do not contain a prefix.

1.2 FEATURES

PERFORMANCE:

- Maximum 50 Mbits/sec channel rate encoding
- 36.5 Mbits/sec channel rate decoding for a 64x57 square code at two iterations
- Two or more devices can be used in parallel to increase throughput
- Optional "helical" interleaving (encoding) and deinterleaving (decoding)

FLEXIBILITY:

- Internal buffering allows continuous data streaming
- Programmable block size from 256 to 4096 bits
- Two or three dimensional blocks
- Programmable number of iterations per block up to 32
- Programmable quantization up to 6-bits for soft or hard decision input data (decoding)
- Support for external synchronization

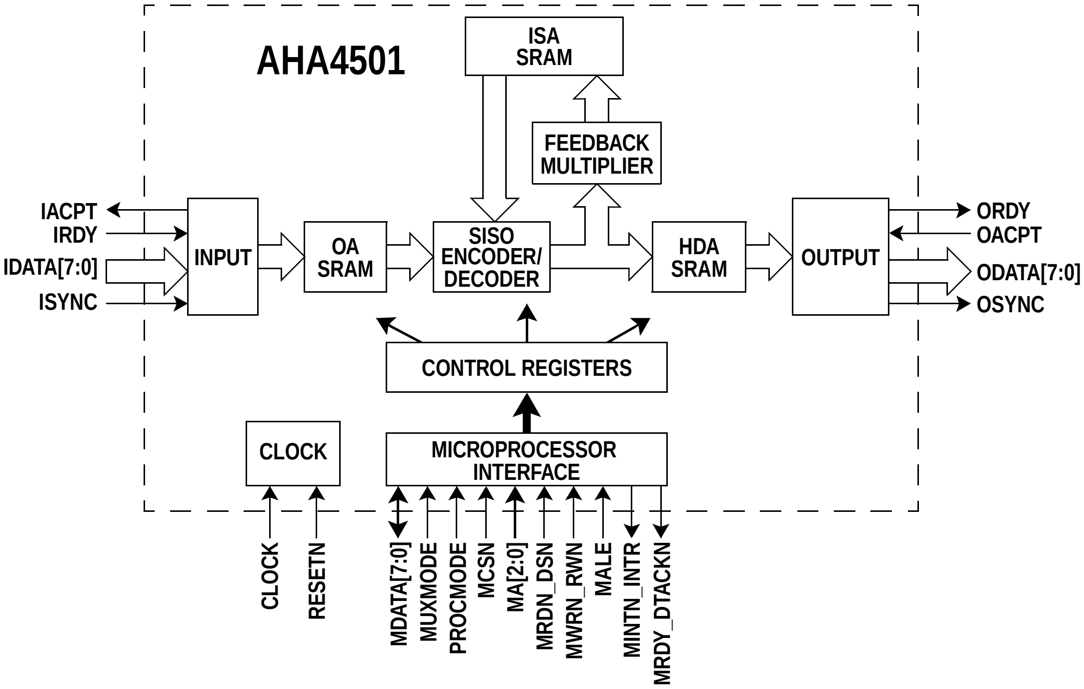
SYSTEM INTERFACE:

- Serial or 8-bit parallel input and output data ports
- Selectable microprocessor interface for Intel or Motorola processors
- Control Commands for: Decode, Encode, Soft Reset, Resynchronize and Dump Current Block
- System Interrupts include Block Decode Complete, Block Correction Incomplete, Sync Mark Mismatch
- Number of corrections per block accumulated in an internal register

OTHERS:

- 3.3 Volt operation
- 100 pin quad flat package
- Output signals may be tristated to facilitate board level testing

Figure 1: Functional Block Diagram



2.0 FUNCTIONAL OVERVIEW

The sections below describe the various configurations, programming and other special considerations for developing an error correction system using the AHA4501.

Refer to Figure 1*: Functional Block Diagram* for the data flow while reading the remainder of this section.

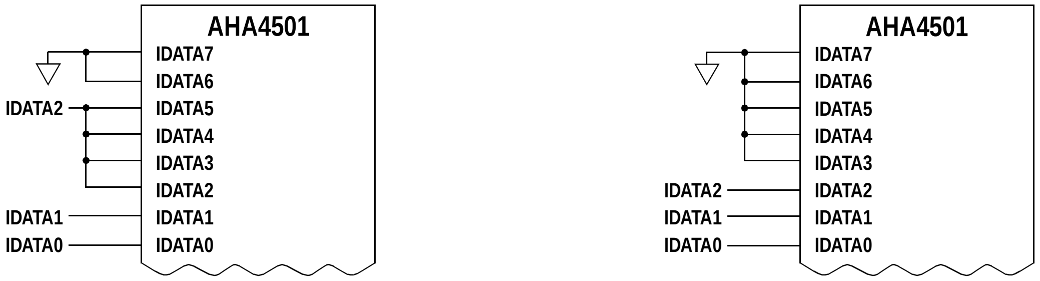
2.1 DATA INPUT

Data is input to the IDATA port via a fully synchronous ready/accept handshake. Data is registered internally on the rising edge of clock when both the ready input (IRDY) and the accept output (IACPT) are asserted. Refer to Section 8.0 Figure 18 for data input timing details.

When encoding, data is input either serially (one bit per handshake) on IDATA[0], or in parallel (one byte per handshake) on IDATA[7:0]. When parallel loading is used, IACPT toggles at most once every eight clocks since the data is serialized internally.

When decoding, data is input one quantization value per handshake on IDATA[(q-1):0] where q is the input quantization size. The quantization size is configurable based on the setting of QSIZE[1:0] within the *Quant* register. The QMODE[1:0] bits within the *Quant* register determine the type of input data. The input data may be 2’s complement, sign/magnitude, or unsigned. When QMODE = “00”, all unused IDATA inputs should be tied to IDATA[q-1]. When QMODE = “01” or “10”, all unused IDATA inputs should be tied to ground. Figure 2 shows example connections when QSIZE = “01” (3 bits).

Figure 2: IDATA Interface



For QMODE = "00" connect IDATA as shown

For QMODE = "01" or "10" connect IDATA as shown

The value of QSHIFT[1:0] in the *Quant* register sets the number of bit positions the input data is shifted left internally before decoding begins. The IDATA bits should be shifted to fill the internal resolution, allowing for higher precision for internal processing. The increased precision results in the best possible decoding performance. Throughput and latency are not affected by the quantization size or shift values. The following equations should be used to determine how to set QSHIFT[1:0] in relation to QSIZE[1:0]. In each equation, qsize and qshift are the values represented by the programmed value, instead of the programmed value itself. For example, if QSIZE[1:0] = "01", qsize in the following equations would be 3 bits.

QMODE[1:0] = "01" or "10":

$$qsize + qshift < 6$$

or

$$qshift < 6 - qsize$$

QMODE[1:0] = "00":

$$qsize + qshift < 7$$

or

$$qshift < 7 - qsize$$

For best performance, do not shift the data beyond the internal resolution (7 bits). The above equations guarantee that this does not occur. For example, if a particular system has three quantization bits using QMODE[1:0] = "00", the following shows the values to program for each register.

QSIZE[1:0] = "01" (3 bit input values)
qshift < 7 - qsize
qshift < 7 - 3
qshift < 4
QSHIFT[1:0] = "11" (shift left of 3)

For best performance, do not shift the data beyond the internal resolution (7 bits). The above equations guarantee that this does not occur.

The following table shows recommended QSHIFT values for each QMODE and QSIZE.

Table 1: Recommended QSHIFT Values

QSIZE (bits)	*2s COMPLEMENT*	*SIGN/MAGNITUDE*	*UNSIGNED*
1	NA	3	NA
2	3	3	3
3	3	2	2
4	2	1	1
5	1	0	0
6	0	NA	NA

2.2 ENCODING

When encoding a 2D block for a $(n_1,k_1)x(n_2,k_2)$ code, $k_1 \times k_2$ data bits constitute one block. When encoding a 3D block for a $(n_1,k_1)x(n_2,k_2)x(n_3,k_3)$ code, $k_1 \times k_2 \times k_3$ data bits constitute one block. The input data is loaded into the OA SRAM input buffer as an array with 1 bit per SRAM location. Encoding begins once the entire block is in the OA SRAM buffer. The device's OA SRAM can accommodate another block while the device encodes the first block in the buffer. ECC bits are generated for each x-axis row of the block array and are appended to the end of the vector. Each y-axis column and each z-axis column (if applicable) are then encoded in the same fashion. The encoded block is loaded into the HDA SRAM output buffer and then transferred out of the device through the ODATA port.

The following figure shows one block of a (8,4)x(8,4) product code. 'D' represents data and 'E' represents ECC bits. Organizing data blocks into arrays and interleaving are performed by the device automatically without any system intervention.

D	D	D	D	E	E	E	E
D	D	D	D	E	E	E	E
D	D	D	D	E	E	E	E
D	D	D	D	E	E	E	E
E	E	E	E	E	E	E	E
E	E	E	E	E	E	E	E
E	E	E	E	E	E	E	E
E	E	E	E	E	E	E	E

The following list of register settings shows an example program to encode data. In this example, the outgoing data is not interleaved, the incoming data loads in parallel on the IDATA[7:0] bus, the block code is a 2D (64,57)x(64,57) product code, and the output asserts OSYNC with the first bit of every third block. OSYNC usage is discussed further in Section 2.6.

- Program *Config0* register
  - INTER 0
  - PAR_SER 1
- Program *Config1* register
  - XCODE[2:0] 111
  - Reserved bits 00011
- Program *Config2* register
  - YCODE[2:0] 111
  - ZCODE[2:0] 000
- Program *Sync* register
  - SYNC MARK FREQUENCY[3:0] 0011
- Program *Control* register
  - ENCODE 1

After these registers are programmed, the AHA4501 asserts IACPT to allow IACPT-IRDY handshakes.

2.3 DECODING

Decoding is done in an iterative fashion. Decoding begins once a complete received data block is available in the OA SRAM. The device's OA SRAM can accommodate another block while the device decodes the first block. Each full iteration begins by passing an x-row from the OA SRAM into the Soft Input Soft Output (SISO) decoder. The SISO output is multiplied by a programmable XFEEDBACK[2:0] value, and stored in the ISA SRAM. The completion of all x-rows constitutes one axis iteration. Refer to Section 2.4 for an explanation of the feedback multipliers.

Next, each y-axis column from the OA SRAM is passed into the SISO decoder. The SISO output is multiplied by the programmable YFEEDBACK[2:0] value and stored in the ISA SRAM. The completion of all y-columns constitutes one axis iteration.

If a 3D code is being decoded, each z-axis column from the ISA SRAM is passed into the SISO decoder. The SISO output is multiplied by the programmable ZFEEDBACK[2:0] value and stored in the ISA SRAM. The completion of all z-columns constitutes one axis iteration.

One full iteration is completed when one X and one Y axis iteration is complete for a 2D code; or one X, one Y, and one Z axis iteration is complete for a 3D code. The iterations continue until the iteration counter equals the number programmed in ITER[4:0] within the *Config0* register.

The following sequence may be used to program the AHA4501 for decoding data. This configuration decodes the same blocks of data encoded using the configuration shown in the encoding section.

- Program *Config0* register
  - PAR_SER 1
    When decoding, the PAR_SER bit configures the output on ODATA[7:0]. In this case, parallel output is selected.
  - STITER 1
    The STITER bit causes the AHA4501 to stop decoding when a full iteration completes with no corrections.
  - ITER[4:0] 00100
    Set for 4 iterations. Refer to Section 4.0 *Performance Curves.*
- Program *Config1* register
  - XCODE[2:0] 111
    (64,57) code type
  - Reserved bits 00011

- Program *Config2* register
  - OECC 0
    No ECC bits will be output with the data.
  - XFBCK[2] 1
    Usually set for a feedback multiplier of 1/2 with square codes.
  - YCODE[2:0] 111
    (64,57) code type
- Program *Feedback* register
  - YFEEDBACK[2:0] 100
- Program *Quant* register
  - QMODE[1:0] 01
    This depends on the type of ADC selected to recover the transmitted data. For this example sign/magnitude is selected.
  - QSIZE[1:0] 10
    This also depends on the type of ADC selected. For this example, 4 bit quantization is selected.
  - QSHIFT[1:0] 01
    This should be set for a 1 bit left shift for the best possible internal precision with 7 bit internal resolution and 4 bit quantization. Refer to Section 2.1 *Data Input* for guidelines on setting QSHIFT[1:0].
- Program *Sync* register
  - SYNC MARK LENGTH[3:0] 0101
    The length of the sync mark is selected by the system designer. For this example, the sync mark length is 20 bits long.
  - SYNC MARK FREQUENCY[3:0] 0011
- Program *Control* register
  - Decode 1

After these registers are programmed, the AHA4501 asserts IACPT and discards the input data until ISYNC is asserted.

2.4 FEEDBACK

The TPC algorithm uses feedback, or weighting, values for performance timing. After each axis iteration, the output of the SISO Decoder is multiplied by the feedback constant for that axis. These values are then fed back into the SISO for future iterations.

The feedback multiplier values used for each code axis vary from 1/4 to 11/16 depending on the number of iterations and system parameters (soft input bits, resolution). The feedback multipliers must be tuned to give optimum decoder performance in a given system. The following describes the tuning process. The choice of feedback multiplier has no effect on throughput or latency.

For 2D square (XCODE[2:0] = YCODE[2:0]) codes, a typical feedback multiplier value for both axes at 3 or 4 iterations is 8/16. For 3D cubic (XCODE[2:0] = YCODE[2:0] = ZCODE[2:0]) codes, a typical feedback multiplier value at 6 iterations is 7/16.

When using non-square or cubic codes, the following general rules should be applied. Parity codes should have their feedback multiplier values set higher than Hamming codes when mixed. For example, in a (32,26)x(32,26)x(4,3) code, the X and Y feedback multipliers should be set to 6/16 while the Z feedback should be set to 9/16 or 10/16. When mixing Hamming codes with shorter Hamming codes, the feedback multiplier should be set slightly higher for the shorter code. For example, in a (64,57)x(32,26) code, the X feedback multiplier could be set to 8/16, while the Y feedback multiplier could be set to 9/16.

The feedback values must be tuned for the number of iterations allowed in a system. For less iterations than the above guidelines, the feedback values should be increased. For more iterations, the values should be decreased. For example, when using a (64,57)x(64,57) code with only 2 iterations, the feedback multiplier for both axes should be set to 10/16. Conversely, in a system that allows 12 or more iterations, the value for the feedback should be set to 7/16.

The feedback may also need to be tuned depending on the number of soft input bits (QSIZE[1:0]). This parameter will only affect the optimum feedback multiplier value slightly, meaning that it should be adjusted by only 1/16 or 2/16 to allow for these differences.

Since systems vary widely, the system designer should experiment with various feedback multiplier values to obtain the best performance. Recommended starting values for feedback are listed in Table 4.

2.5 DATA OUTPUT

Data is output through the ODATA port via a fully synchronous ready/accept handshake. Data is transferred on the rising edge of clock when both the accept input (OACPT) and the ready output (ORDY) are asserted. Refer to Section 8.0 Figure 19 for data output timing details.

When encoding, data is always output serially on ODATA[0]. When decoding, data can be output either serially on ODATA[0] or in 8-bit parallel on ODATA[7:0].

2.6 SYNCHRONIZATION

Since the TPC is a block code, data synchronization is required to correctly decode each block. External synchronization circuitry is required to insert and detect synchronization marks. The AHA4501 provides features to remove the synchronization marks and indicate correct synchronization mark placement.

The ISYNC and OSYNC signals are handled in the same fashion as IDATA[7:0] and ODATA[7:0]. ISYNC is registered internally on the rising edge of clock when both IRDY and IACPT are asserted. OSYNC is only valid when the ORDY output signal is asserted.

2.6.1 ENCODE SYNCHRONIZATION

When encoding, the AHA4501 provides the output signal OSYNC to indicate when a synchronization mark should be inserted in the data stream. OSYNC is asserted with the first data bit of each *x* output blocks, where *x* is the value of SYNC MARK FREQUENCY[3:0] programmed within the *Sync* register. The system can then use OSYNC to insert a synchronization mark in the encoded data stream before transmitting.

2.6.2 DECODE SYNCHRONIZATION

When decoding, the synchronization circuitry depends on the channel and demodulation method. For the first block after reset, the AHA4501 discards the input data on IDATA[7:0] until the ISYNC signal is asserted. The first bit of the block is registered on the same clock as ISYNC is asserted.

After the first block, synchronization marks are automatically removed from the data by programming SYNC MARK LENGTH[3:0] and SYNC MARK FREQ[3:0] within the *Sync* register. SYNC MARK LENGTH[3:0] configures the AHA4501 to remove *x* bits from the start of every synchronization block, where *x* is 4 times SYNC MARK LENGTH[3:0]. The AHA4501 expects a synchronization mark at the block interval specified by SYNC MARK FREQUENCY[3:0].

If ISYNC is not asserted with the first bit after SYNC MARK LENGTH[3:0] handshakes, a loss of synchronization is indicated by assertion of the SMMIS bit in the *Interrupt* register. If the system designer chooses not to use the synchronization support logic of the AHA4501, the ISYNC signal must be tied high. The CORRECTIONS[9:0] count and CORINC bit can also be used to indicate a loss of synchronization. The control microprocessor may use this information to send a resynchronize command to cause the AHA4501 to synchronize on the start of the next data block by discarding input data until ISYNC is asserted.

The OSYNC signal is asserted with the first bit of every block when decoding.

2.6.3 RESYNCHRONIZE

Loss in synchronization may be detected using SMMIS, CORINC, NITER[4:0], and CORRECTIONS[9:0]. The control microprocessor may use this information to determine that the AHA4501 is not synchronized and issue a resynchronize command. The AHA4501 does not automatically resynchronize the data stream unless instructed to do so by the microprocessor.

The resynchronize command causes the AHA4501 to stop decoding blocks and discard the input data until ISYNC is asserted. The AHA4501 continues to output any blocks that have been decoded and are waiting to be unloaded from the HDA SRAM.

The AHA4501 registers the input from IDATA[7:0] on the same clock as ISYNC is asserted.

2.7 HELICAL INTERLEAVING

The device can optionally interleave when encoding and deinterleave when decoding. Interleaving data spreads bursts of noise across all axes of the block code for the best error correction performance in burst channel use. Interleaving in the AHA4501 is applied after encoding takes place. Deinterleaving in the AHA4501 takes place before the decoding operation.

Helical interleaving is applied along a diagonal path through the encoded block. Data is output along diagonal lines from the upper left to lower right corner (for a 2D code). The first diagonal output starts with the bit row 1, column 1 followed by the diagonal starting at row 1, column 2. For 3D codes, instead of reading diagonally through the 2D array, interleaving reads diagonally through a cube of data.

The example below shows how interleaving is applied for a 2D (64,57)x(64,57) code.

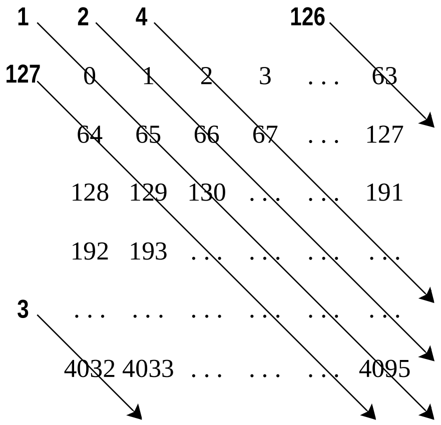
Figure 3: Input Block

0	1	2	3	. . .	63
64	65	66	67	. . .	127
128	129	130	. . .	. . .	191
192	193	. . .	. . .	. . .	. . .
. . .	. . .	. . .	. . .	. . .	. . .
4032	4033	. . .	. . .	. . .	4095

Note: *The number reflects the bit order, including generated ECC bits.*

The encoded, interleaved data output is taken along diagonal lines starting with bit 0 as shown below. The order of the interleaving is noted for each diagonal line.

Figure 4: 2D Interleaving



0	1	2	3	. . .	63
64	65	66	67	. . .	127
128	129	130	. . .	. . .	191
192	193	. . .	. . .	. . .	. . .
. . .	. . .	. . .	. . .	. . .	. . .
4032	4033	. . .	. . .	. . .	4095

For the (64,57)x(64,57) block, the data output from the AHA4501 is: 0, 65, 130, ..., 4095, 1, 66, ..., 4031, 4032, 2, 67, ..., ..., 63, 64, ..., 4094 for a total of 4096 bits output. The AHA4501 operating as a decoder deinterleaves the block to restore it to its orginal order.

Figure 5: Encoded/Interleaved Data Output

0	65	130	. . .	4030	4095
1	66	131	. . .	4031	4032
2	67	132	. . .	3968	4033
3	68	. . .	. . .	. . .	. . .
. . .	. . .	. . .	. . .	. . .	. . .
63	64	129	. . .	4029	4094

Data bits are output from the encoder in row order from left to right. 3D interleaving/ deinterleaving is done by reading/writing cells diagonally through the x, y, and z dimensions. Note that the data rate drops when interleaving and/or deinterleaving as discussed in Section 2.8.

2.8 DATA THROUGHPUT

The AHA4501 contains internal buffering at the input (OA SRAM) and the output (HDA SRAM) to allow the device to maintain a constant input data rate with no external memory. The AHA4501 is capable of loading a code block into 1/2 of the OA SRAM while it is processing a second code block from the other 1/2 of the OA SRAM. The second code block is loaded into the 1/2 of the HDA SRAM while a third code block can be output from the other 1/2 of the HDA SRAM. This ping-pong buffer arrangement on the input and output sides of the AHA4501 allows code blocks to be processed in a continuous stream as long as the overall bandwidth of the device is not exceeded.

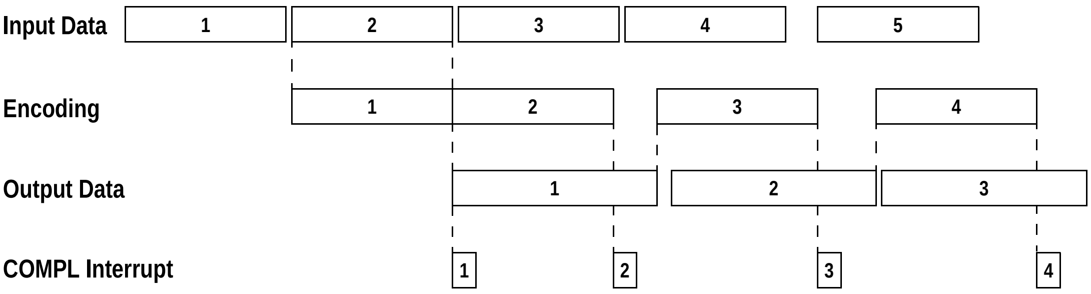
When encoding a data block, the data can be transferred continuously at up to one bit per clock, independent of the code type. A 1 clock delay occurs between blocks.

When decoding a code block and not deinterleaving, the maximum input rate is one soft value per clock, independent of code type. A 1 clock delay occurs between blocks. When decoding a code block and deinterleaving, the maximum input rate is one soft value every 3 clocks, independent of code type. A 1 clock delay also occurs between blocks when deinterleaving.

The following diagrams illustrate how code blocks are processed through the AHA4501 and when the internal status registers update the status of the blocks.

In Figure 6, the data is input at a bit per clock and encoded. The data is encoded by appending ECC bits to the data. Note that the data is always encoded faster than data can be input to the device. Since the output is operating at a bit/clock and there are more output bits than input bits, the output is the limiting factor in the system. After the initial buffer loading, the input must wait for the output to finish unloading a block before accepting another block. The ratio of the input block size to the output block size is the code rate.

Figure 6: Encoding at the Maximum Input Rate (bit/clock) with Maximum Output Rate (bit/clock)



In Figure 7, the data is input at a slower rate than the output rate which allows a constant input data rate.

In Figure 8, the received data is input at a bit every other clock and decoded. When decoding, the decode time is variable depending on the number of iterations used, type of code and the status of the STITER bit. In this example, the iterations are set so that the decode time is approximately equal to the data input time. For continuous data input, set the ITER[4:0] count such that the decoder time is less than the data block input time. Refer to Section 2.9 *Encoding/Decoding Time* for decoding time calculations.

In Figure 9, the received data is input at a bit every other clock and decoded. When decoding, the decode time is variable depending on the number of iterations used, type of code and the status of the STITER bit. In this example, the iteration count in ITER is set to 6 to illustrate a case where the decode time is the limiting factor in the throughput. To achieve maximum decoder performance with burst data, set the iteration count such that the decoder time is equal to or exceeds the data block input time.

Figure 7: Encoding at Less Than Maximum Input Rate with Maximum Output Rate

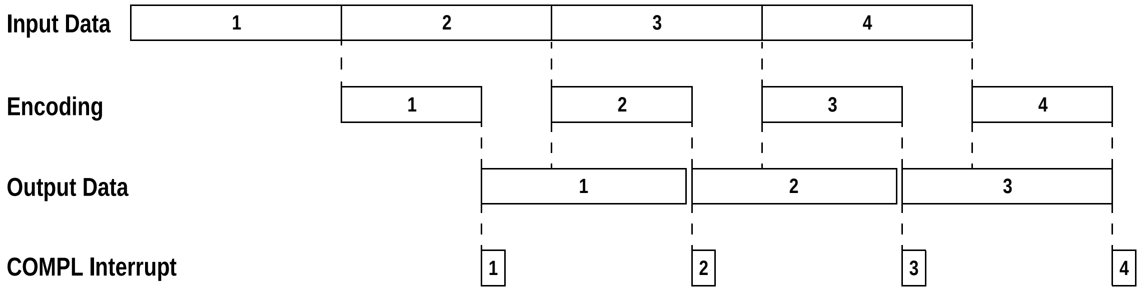


Figure 8: Decoding with Continuous Input Data Rate - STITER not Asserted

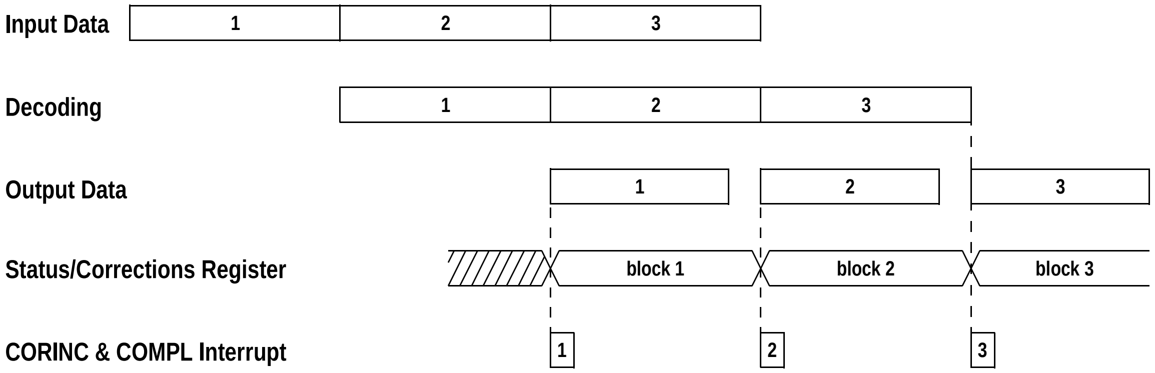
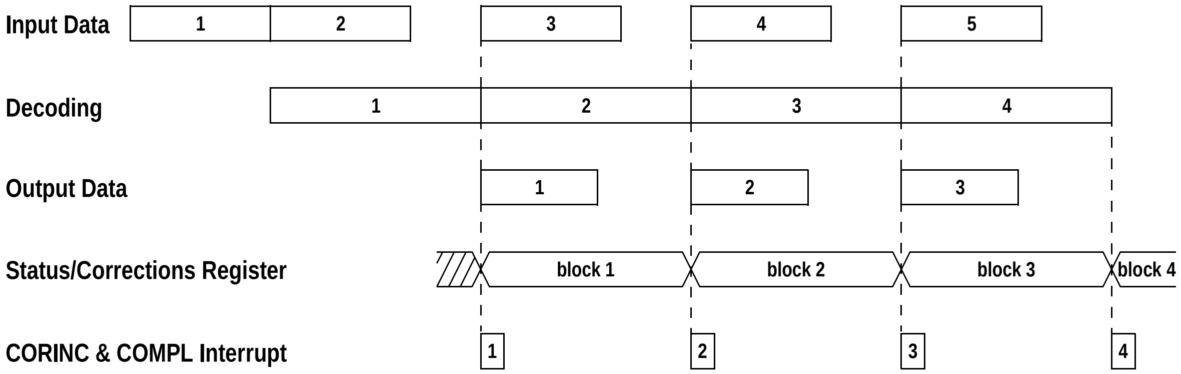


Figure 9: Decoding with Burst Input Data Rate - STITER not Asserted



In Figure 10, the received data is input at a bit per clock and decoded. In this example, the iteration count in ITER is set to 6 and the STITER bit is set to illustrate a case where the decode time is variable depending on the number of iterations required to correct the errors in the code block. In this example, blocks 1, 2, 4, and 5 decode in 2 iterations while block 3 requires 6 iterations to decode.

When decoding, writing a one to the DUMP bit within the *Control* register causes the module to stop iterations on the current block and send it to the output data port. The dump occurs upon completion of the axis iteration following the current axis iteration. The worst case delay is two axis iterations (which is equal to one full iteration for 2-D codes). Decoding then begins on the next (already loaded) data block, and another block can begin loading.

In Figure 11, the example from Figure 10 is shown to illustrate the DUMP feature. In the third decoding block, DUMP is set in the fourth iteration. The decoder finishes the current full iteration before it outputs the block. Decoding on block 4 starts normally.

The DUMP feature is useful when using an input buffer with the STITER configuration bit, and the input buffer becomes full. Note that the DUMP bit does not cause loss of data. Upon completion of the dump, the COMPL interrupt bit is set, and the CORINC bit is set if any corrections were made in the last axis iteration. This bit should not be set when encoding.

Figure 10: Decoding with Burst Input Data Rate - STITER Asserted

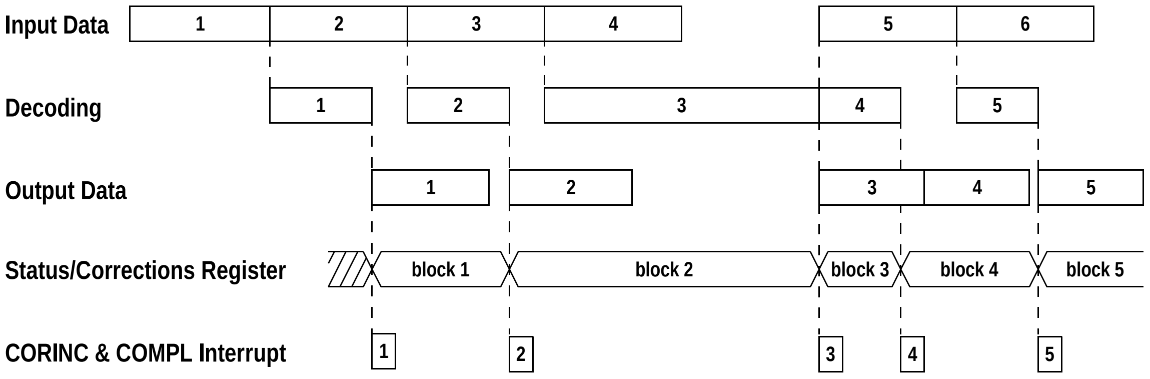
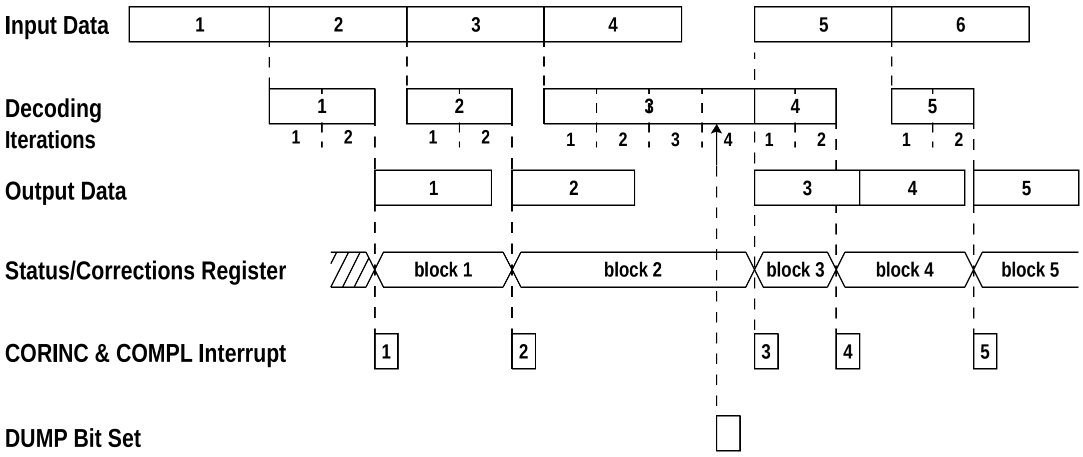


Figure 11: DUMP Feature



2.9 ENCODING/DECODING TIME

The time required to encode a data block is equivalent to the time to decode 1 code block with 1 full iteration. Since the AHA4501 can perform 1 full iteration faster than data can be transferred serially through the encoded data output port, the output rate is the limiting factor for the overall data rate.

The time to decode a code block depends on the input clock frequency, the code type, and the number of iterations. The following equations are used to compute the overall decoding time. The decoding time can be used to compute the data rate and latency.

Note that if the STITER bit is set in the *Config0* register, the AHA4501 stops iterating when there are no corrections. When the STITER bit is set, the number of iterations is unpredictable and the decode time may be shortened or lengthened depending on the error content data stream. The number of iterations will never be more than the number set in ITER[4:0] even when STITER is set.

nx = Length of entire vector (data + ECC) for X axis code
kx = Length of data vector for X axis code
ny = Length of vector (data + ECC) for Y axis code
ky = Length of data vector for Y axis code
nz = Length of vector (data + ECC) for Z axis code (nz=1 for 2D codes)
kz = Length of data vector for Z axis code (kz=1 for 2D codes)
i = Iterations
d = Number of clocks to decode one block
f = Clock frequency (Hz)
r_{ch} = Channel rate (bits/sec)
r_d = Data rate (bits/sec)
C_1,C_0 = decode constants, see table 1.
CR = code rate.

Code rate for a *(nx,kx)* x *(ny,ky)* x *(nz,kz)* code:

$$CR = \left(\frac{kx \times ky \times kz}{nx \times ny \times nz}\right)$$

Clocks to decode an entire block:

$$d = c_1 \times i + c_0$$

Maximum channel rate for a 3D block (for a 2D block, nz = 1):

$$r_{ch} = \frac{nx \times ny \times nz \times f}{d}$$

Maximum data rate for a 3D block (for a 2D block, kz=1):

$$r_d = \frac{kx \times ky \times kz \times f}{d}$$

If interleaving is used, the maximum channel rate will be the lesser of r_{ch} listed above and $f / 3$ and the maximum data rate will be the lesser of r_d listed above and $(f \times CR)/ 3$.

2.10 SUMMARY OF CHANNEL RATES

The channel rates listed in the following table are calculated with a 50 MHz clock frequency. The channel rate changes in proportion with the change in clock frequency. For example, if the clock frequency is 25 Mhz, all channel rates are divided by 2. To compute data rate, multiply these values by the overall code rate.

Note: *This table does not include all codes supported by the device. For decode times and code rates for codes other than those listed here, see the AHA4501 windows evaluation software.*

Table 2: Channel Rates

BLOCK CONFIGURATION *$(n_1,k_1)x(n_2,k_2)x(n_3,k_3)$*		*NUMBER OF ITERATIONS*				
		2	**3**	**6**	C_1	C_0
(64,57)x(64,57)	decode time (clocks)	5612	7963	15016	2351	910
Code Rate = .793	channel rate (Mbits/sec)	36.5	25.7	13.6		
(32,26)x(32,26)x(4,3)	decode time (clocks)	8368	11917	22564	3549	1270
Code Rate = .495	channel rate (Mbits/sec)	24.5	17.2	9.1		
(16,11)x(16,11)x(16,11)	decode time (clocks)	7816	11197	21340	3381	1054
Code Rate = .325	channel rate (Mbits/sec)	26.2	18.3	9.6		
(32,26)x(16,11)x(8,4)	decode time (clocks)	7992	11421	21708	3429	1134
Code Rate = .278	channel rate (Mbits/sec)	25.6	17.9	9.4		
(64,57)x(32,26)	decode time (clocks)	2972	4227	7992	1255	462
Code Rate = .724	channel rate (Mbits/sec)	34.5	24.2	12.8		
(32,26)x(16,11)x(4,3)	decode time (clocks)	4304	6141	11652	1837	630
Code Rate = .419	channel rate (Mbits/sec)	23.8	16.7	8.8		
(64,57)x(8,4)x(4,3)	decode time (clocks)	4448	6357	12084	1909	630
Code Rate = .334	channel rate (Mbits/sec)	23.0	16.1	8.5		
(32,26)x(32,26)	decode time (clocks)	1540	2211	4224	671	198
Code Rate = .660	channel rate (Mbits/sec)	33.2	23.2	12.1		
(16,11)x(16,11)x(4,3)	decode time (clocks)	2224	3181	6052	957	310
Code Rate = .354	channel rate (Mbits/sec)	23.0	16.1	8.5		
(32,26)x(16,11)	decode time (clocks)	860	1239	2376	379	102
Code Rate = .559	channel rate (Mbits/sec)	29.8	20.7	10.8		
(16,11)x(16,11)	decode time (clocks)	464	679	1324	215	34
Code Rate = .473	channel rate (Mbits/sec)	27.6	18.9	9.7		

2.11 LATENCY

Since product codes are block codes, the data for an entire block must be input to the AHA4501 before encoding or decoding can start. If interleaving is not enabled, the latency (in clock cycles) from the first input bit of a block to the first output bit of the same block is:

block input time + block decode(encode) time + 9

If interleaving is enabled the latency for encoding a block (in clock cycles) is:

block input time + block encode time + 17

The latency for decoding an interleaved block (in clock cycles) is:

block input time + block decode time + 19

For example, the block decode time and latency to decode a (64,57)x(64,57) non-interleaved code with 2 iterations and a 50 MHz clock are shown below. The example assumes the input and output operate at maximum speed.

The time to decode a (64,57)x(64,57) block (as shown in Section 2.10) is:

$$5612 \times \left(\frac{1}{50\ MHz}\right) = 112.2\mu s$$

Assuming that the data is input at the decode data rate (i.e., the block input time equals the block decode time) the total latency is:

$$2 \times 112.2\mu s + 9 \times \left(\frac{1}{50MHz}\right) = 224.47\mu s$$

2.12 MICROPROCESSOR INTERFACE

The AHA4501 is capable of interfacing directly to a microprocessor for embedded applications. All register accesses to the AHA4501 are performed on an 8-bit bidirectional bus, using either an Intel® or Motorola® style interface. The interface is in Motorola® mode when the PROCMODE input signal is asserted, otherwise the interface is in Intel® mode.

A MUXMODE input is also provided to allow the data and address to be multiplexed on the MDATA[7:0] bus. The data and address are multiplexed when MUXMODE is asserted, otherwise both MDATA and MA busses are used.

Refer to Section 8.0 for microprocessor interface timing diagrams.

3.0 INTERNAL REGISTER PROGRAMMING

Table 3: Register Summary

ADDRESS	*R/W*	*MNEMONIC*	*REGISTER NAME*	*HARD RESET*	*SOFT RESET*
0x00	R/W	*Config0*	*Configuration0*	0x26	unchanged
0x01	R/W	*Config1*	*Configuration1*	0xE0	unchanged
0x02	R/W	*Config2*	*Configuration2*	0x38	unchanged
0x03	R/W	*Feedback*	*Feedback*	0x24	unchanged
0x04	R/W	*Quant*	*Quantization*	0x0C	unchanged
0x05	R	*Correct*	*Corrections/*	0x00	unchanged
	W	*Sync*	*Synchronization*		
0x06	R	*Status*	*Status*	0x00	unchanged
0x07	R	*Interrupt*	*Interrupt*	0x00	UUUUU000
	W	*Control*	*Control*		
0x08		Reserved	Reserved	0x0C	unchanged

Notes:

1) *U - These bits remain unchanged after a soft reset.*

2) *The reserved bits in Register Address 0x01 must be written to 00011 after any hard reset.*

3.1 CONFIGURATION 0, ADDRESS 0x00 - READ/WRITE

Address	*bit7*	*bit6*	*bit5*	*bit4*	*bit3*	*bit2*	*bit1*	*bit0*
0x00	INTER	PAR_SER	STITER	ITER[4:0]				

This register is initialized to 0x26 after hard reset, unchanged after soft reset.

INTER - Deinterleave incoming data when decoding, and interleave outgoing data when encoding. See Section 2.1 *Data Input* for details about interleaving/deinterleaving.

PAR_SER - Parallel/Serial Mode Select.When PAR_SER is set during encoding, the device will input 8 bits in parallel from the IDATA[7:0] bus. The first encoded bit output on ODATA[0] will be from IDATA[0]. In other words, IDATA[0] is the LSB and IDATA[7] is the MSB. The IRDY pin controls the flow of data into IDATA[7:0]. Typically, the IACPT pin will be active 1 out of 8 clock cycles when in parallel mode. During encoding the output is always bit serial on ODATA[0].

When PAR_SER is not set during encoding, the data is input from IDATA[0] one bit per handshake and output from ODATA[0] one bit per handshake. The unused pins IDATA[7:1] should not be left floating, they should be tied to GND.

When PAR_SER is set during decoding, the device will pack the output decoded data bits into 8-bit bytes on ODATA[7:0]. The output bit on ODATA[0] is generated from the first soft input symbol. ODATA[0] is the LSB and ODATA[7] is the MSB. The ORDY pin controls the flow of data out of ODATA[7:0]. Typically, the ORDY pin is active 1 out of 8 clock cycles when in parallel mode. The input data will be on IDATA[n-1:0] where n is the number of soft input bits. The unused IDATA pins should not be left floating, they should be tied to GND.

When PAR_SER is not set during decoding, the data is input from IDATA[n-1:0] one soft symbol per handshake where n is the number of soft input bits. The decoded data is output from ODATA[0] one bit per handshake.

STITER - Stop iterating when no corrections. Used only when decoding; the AHA4501 can determine if future iterations can be useful. When this bit is set, the module stops iterating when the decoding is completed. When cleared, it always executes the number of full iterations in ITER[4:0]. Note that in either case, the number of full iterations does not exceed ITER[4:0]. At any time after the first iteration, the microprocessor can also write a 1 to the DUMP bit in the *Control* register, in which case all future iterations are cancelled and the current block is output. This bit is ignored when encoding.

ITER[4:0] - Maximum Iterations. Used only when decoding; number of full iterations to perform for each block. One full iteration is defined as decoding the x and y axes for a 2D block or all x, y and z axes for a 3D block. The value of 0x0 indicates 32 full iterations. If STITER is asserted, less iterations than the ITER[4:0] count may be performed. This value is ignored when encoding.

3.2 CONFIGURATION 1, ADDRESS 0x01 - READ/WRITE

Address	*bit7*	*bit6*	*bit5*	*bit4*	*bit3*	*bit2*	*bit1*	*bit0*
0x01	XCODE[2:0]			*res*				

This register is initialized to 0xE0 after hard reset, unchanged after soft reset.

XCODE[2:0] -Code for x axis of product array. See *Config2* register ZCODE [2:0] description.

res - Reserved bits [4:0]. Must be written to 00011.

3.3 CONFIGURATION 2, ADDRESS 0x02 - READ/WRITE

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x02	OECC	XFBCK[2]	YCODE[2:0]			ZCODE[2:0]		

This register is initialized to 0x38 after hard reset, unchanged after soft reset.

OECC - Output ECC Bits. Used only when decoding. When cleared, only the data bits are output; total output bits per block = $k_1 \times k_2$ (2D), or $k_1 \times k_2 \times k_3$ (3D). When set, both the data and ECC bits are output; total output bits per block = $n_1 \times n_2$ (2D), $n_1 \times n_2 \times n_3$ (3D).

XFBCK[2] -x axis feedback bit 2. See description for *Feedback* register (0x03).

YCODE[2:0] -Code for y axis of product array. See ZCODE[2:0] description for code definitions.

ZCODE[2:0] -Code for z axis of product array. Set to 000 for all 2D codes.

Each code axis is defined as follows:

111	(64,57)	-- Extended Hamming Codes
110	(32,26)	
101	(16,11)	
100	(8,4)	
011	(16,15)	-- Parity Only Codes - even parity
010	(8,7)	
* 001	(4,3)	
* 000	no code	* valid only in ZCODE[2:0]

The following rules must be followed when selecting codes:

1) The (4,3) parity code and “n_0 code” is illegal for both the x axis code and the y axis code.
2) 2D codes require that the x dimension product code and the y dimension product code be at least 16 bits.
3) n_1 x n_2 x n_3 must be less than or equal to 4096.

The AHA4501 is designed to allow any combination of x, y and z codes that follow the above rules. The code combinations listed below have been fully verified.

In addition to the following codes, the TPC codes may be shortened with zero padding techniques. Contact AHA Application Engineering for details.

Table 4: Supported Codes with Recommended Feedback Values

BLOCK CONFIGURATION $(n_1,k_1)x(n_2,k_2)x(n_3,k_3)$	BLOCK SIZE (bits)	DATA SIZE (bits)	CODE RATE	FEEDBACK (4 iterations)	FEEDBACK (32 iterations)
(64,57)x(64,57)	4096	3249	0.793	1/2,1/2	7/16,7/16
(32,26)x(32,26)x(4,3)	4096	2028	0.495	3/8,3/8,11/16	5/16,5/16,9/16
(16,11)x(16,11)x(16,11)	4096	1331	0.325	7/16,7/16,7/16	3/8,3/8,3/8
(32,26)x(16,11)x(8,4)	4096	1144	0.278	3/8,3/8,7/16	3/8,3/8,3/8
(64,57)x(32,26)	2048	1482	0.724	1/2,9/16	7/16,1/2
(32,26)x(16,11)x(4,3)	2048	858	0.419	3/8,7/16,11/16	5/16,3/8,9/16
(64,57)x(8,4)x(4,3)	2048	684	0.334	3/8,7/16,11/16	5/16,3/8,9/16
(32,26)x(32,26)	1024	676	0.660	1/2,1/2	7/16,7/16
(16,11)x(16,11)x(4,3)	1024	363	0.354	3/8,3/8,11/16	5/16,5/16,9/16
(32,26)x(16,11)	512	286	0.559	7/16,1/2	3/8,7/16
(16,11)x(16,11)	256	121	0.473	1/2,1/2	7/16,7/16

Note: The feedback values listed in Table 4 are recommended starting values. Depending on the target Bit Error Rate, the user may wish to adjust the feedback values slightly for more optimal performance. The AHA4501 Windows Evaluation software can be used to fine tune the feedback for any selected configuration.

3.4 FEEDBACK, ADDRESS 0x03 - READ/WRITE

Address	*bit7*	*bit6*	*bit5*	*bit4*	*bit3*	*bit2*	*bit1*	*bit0*
0x03	XFEEDBACK[1:0]		YFEEDBACK[2:0]			ZFEEDBACK[2:0]		

This register is initialized to 0x24 after hard reset, unchanged after soft reset.

XFEEDBACK[1:0] - Feedback multiplier for x axis iteration. Used only when decoding.

YFEEDBACK[2:0] - Feedback multiplier for y axis iteration. Used only when decoding.

ZFEEDBACK[2:0] - Feedback multiplier for z axis iteration. Used only when decoding 3D codes.

Each feedback value is defined as follows:

000	-	Multiply feedback by 1/4
001	-	Multiply feedback by 5/16
010	-	Multiply feedback by 3/8
011	-	Multiply feedback by 7/16
100	-	Multiply feedback by 1/2
101	-	Multiply feedback by 9/16
110	-	Multiply feedback by 5/8
111	-	Multiply feedback by 11/16

Refer to the Section 2.3 *Decoding* for a functional description of the feedback values.

3.5 QUANTIZATION, ADDRESS 0x04 - READ/WRITE

Address	*bit7*	*bit6*	*bit5*	*bit4*	*bit3*	*bit2*	*bit1*	*bit0*
0x04	*res*		QSHIFT[1:0]		QSIZE[1:0]		QMODE[1:0]	

This register is initialized to 0x0C after hard reset, unchanged after soft reset.

res - Reserved bits. Must be written to 00.

QSHIFT[1:0] - Quantization Shift. Used only when decoding; must be set to “00” when encoding. The input data can be shifted bitwise left inside the device before decoding begins. This is useful with smaller input quantization sizes. See Section 2.1 *Data Input* for details about how to set this value. Defined as follows:

00	-	No shift
01	-	Shift input data left 1 (multiply by 2)
10	-	Shift input data left 2 (multiply by 4)
11	-	Shift input data left 3 (multiply by 8)

QSIZE[1:0] - Quantization Size for soft input data. Used only when decoding. Specifies the number of bits of data for each soft input value. Soft input data must always be driven on IDATA[QSIZE-1:0]. Defined as follows:

00	-	1 bit, 2 bits
01	-	3 bits
10	-	4 bits
11	-	5 bits
	*	See *Note 1* for 6 bit

Notes:

1) *6 bit quantization is supported when QMODE[1:0] = “00”. Since the data is in 2’s complement notation, the data on IDATA[5:0] is transferred directly to the OA SRAM. The value of QSIZE[1:0] is ignored when QMODE[1:0]= “00”.*
2) *1 bit quantization (hard decision input) is supported by setting QSIZE[1:0] = “00” and QMODE[1:0] = “01”. It is recommended that QSHIFT[1:0] be set to “11” for hard decision input data. The hard decision input is connected to IDATA[1]. IDATA[0] must be tied high.*

QMODE[1:0] - Quantization Mode for soft input data. Used only when decoding; must be set to “00” for encoding. When set to “00”, input data is assumed to be in signed 2's complement notation (mid-tread). When set to “01”, data is assumed to be mid-riser sign/magnitude notation. When set to “10”, data is assumed to be mid-riser unsigned. The confidence mapping for each mode is shown on the next page with four bit quantization.

QMODE[1:0]	***Input Data Type***	***Hard Decision 0 Confidence Range***			***No Confidence***	***Hard Decision 1 Confidence Range***		
		Max	***. . .***	***Min***		***Min***	***. . .***	***Max***
00	2’s Complement	1000	. . .	1111	0000	0001	. . .	0111
01	Sign/Magnitude	0111	. . .	0000	N/A	1000	. . .	1111
10	Unsigned	0000	. . .	0111	N/A	1000	. . .	1111

3.6 CORRECTIONS, ADDRESS 0x05 - READ

Address	***bit7***	***bit6***	***bit5***	***bit4***	***bit3***	***bit2***	***bit1***	***bit0***
0x05	CORRECTIONS[7:0]							

This register is initialized to 0x00 after hard reset, unchanged after soft reset.

CORRECTIONS[7:0] - The eight least significant bits of the number of hard decision corrections made over the entire number of iterations executed on one block. The upper two most significant bits are accessed via the *Status* register. This value is updated each time the COMPL interrupt bit is set. This value contains the count of the number of bits corrected between the input data (assuming hard decision decoding), and the output data. If the count overflows, the CFLOW bit is set, and the value of the count is invalid. Invalid when encoding.

3.7 SYNCHRONIZATION, ADDRESS 0x05 - WRITE

Address	***bit7***	***bit6***	***bit5***	***bit4***	***bit3***	***bit2***	***bit1***	***bit0***
0x05	SYNC MARK LENGTH [3:0]				SYNC MARK FREQUENCY[3:0]			

This register is initialized to 0x00 after hard reset, unchanged after soft reset.

SYNC MARK LENGTH[3:0] - When decoding, the module can automatically remove sync marks from the incoming data stream. The number of bits to be discarded is 4 times the value of the *Sync Mark Length* register. This allows sync marks to be from 4 to 60 bits in length. A value of zero results in no discarded bits.

SYNC MARK FREQUENCY[3:0] - The number of blocks between synchronization marks. When encoding, the OSYNC signal asserts with the first bit of each x output blocks, where x is the value of SYNC MARK FREQUENCY[3:0] in the *Sync* register. When decoding, the ISYNC is checked at the first bit of each *x* input blocks, where *x* is the value of SYNC MARK FREQUENCY[3:0] in the *Sync* register. If the ISYNC signal is not asserted, the Sync Mark Mismatch interrupt bit is set. A value of 1 indicates that the signals are asserted/checked at the start of every block. A value of 0 indicates every 32 blocks.

See Section 2.6 *Synchronization* for more information.

3.8 STATUS, ADDRESS 0x06 - READ ONLY

Address	*bit7*	*bit6*	*bit5*	*bit4*	*bit3*	*bit2*	*bit1*	*bit0*
0x06	NITER[4:0]					CFLOW	CORRECT[9:8]	

This register is initialized to 0x00 after hard reset, unchanged after soft reset.

NOTE: This register must not be written.

NITER[4:0] - Number of iterations. The actual number of iterations performed in the last block decoded. This value is updated each time the COMPL interrupt bit is set. Note that if the STITER bit is cleared, this value will always be equal to the programmed ITER[4:0] value.

The internal iteration value is incremented after a y-axis iteration is completed for 2D codes or after a z-axis iteration is completed for 3D codes. If the STITER bit is set, the iterations may finish after an axis iteration rather than a full iteration. In this case, the NITER[4:0] value will be the number of full iterations completed.

CFLOW - Correction Overflow. Set if the CORRECTIONS[9:0] count exceeded the value of 1024 corrections. When set, the CORRECTIONS[9:0] count value is invalid.

CORRECT[9:8] - The two most significant bits of the number of hard decision corrections made over the entire set of iterations. This value is updated each time the COMPL interrupt bit is set. Invalid when encoding.

3.9 CONTROL/INTERRUPT, ADDRESS 0x07 - READ/WRITE

RD/WR	*bit7*	*bit6*	*bit5*	*bit4*	*bit3*	*bit2*	*bit1*	*bit0*
Read	DECODE	ENCODE	CORINCM	SMMISM	COMPLM	CORINC	SMMIS	COMPL
Write						SRESET	RESYNC	DUMP

This register is initialized to 0x00 after hard reset. The interrupt bits [2:0] are reset, but all other register contents are unchanged after soft reset.

Note: All interrupts and bits 0, 1, 2 are cleared when this register is read.

DECODE (R/W) - When set, the module performs decoding on the input data. The DECODE bit must not be set with the ENCODE bit. If both bits are cleared, the module is idle and does not accept input data. The SRESET bit should be written with or after clearing the DECODE bit to ensure proper operation. The DECODE bit should not be set within 5 clocks of SRESET.

ENCODE (R/W) - When set, the module performs encoding on the input data. The output data is a serial bit stream of both data and ECC code bits. The ENCODE bit must not be set with the DECODE bit. If both bits are cleared, the module is idle, and does not accept input data. The SRESET bit should be written with or after clearing the ENCODE bit to ensure proper operation. The ENCODE bit should not be set within 5 clocks of SRESET.

CORINCM (R/W) - Correction Incomplete Mask. When cleared, the MINTN_INTR interrupt signal is asserted when the CORINC interrupt bit is asserted. When set, the interrupt signal is not asserted with CORINC.

SMMISM (R/W) - Sync Mark Mismatch Mask. When cleared, the MINTN_INTR interrupt signal is asserted when the SMMIS interrupt bit is asserted. When set, the interrupt signal is not asserted with SMMIS.

COMPLM (R/W) - Block Decode Complete Mask. When cleared, the MINTN_INTR interrupt signal is asserted when the COMPL interrupt bit is asserted. When set, the interrupt signal is not asserted with COMPL.

CORINC (R) - Correction Incomplete. The nature of the algorithm allows the module to determine if another iteration would be useful. When the module has reached the maximum number of iterations programmed in ITER[4:0], a check is done to determine if another iteration would be useful. If another iteration would be useful, then the data in the block may or may not be correct, and the CORINC interrupt bit is set. Note that this bit is pessimistic, meaning that it may be asserted when errorless data is output from the device when the last correction happens on the last iteration. This is especially true when the value of ITER[4:0] is 2 or less. Therefore, it must not be used to discard a block of data if set. Its use is more valuable in verifying that a correct synchronization of data has occurred, or that synchronization has been lost. This bit is never set when encoding.

SRESET (W) - Soft Reset. Writing a 1 to this bit causes a reset of the entire data path. The input and output ports immediately stop handshaking data. The *Control/Interrupt* register bits [2:0] are reset, but all other register contents are unaffected by SRESET. Note that all data internal to the module is lost. SRESET should not be issued at the same time as either DECODE or ENCODE.

SMMIS (R) -Sync Mark Mismatch. After the first block of data has been input to the device, each x block is checked for assertion of the ISYNC signal with the first bit of the block, where x is the value of SYNC MARK FREQUENCY[3:0] in the *Sync* register. If the ISYNC signal is not asserted when the module is at the start of a new block, the SMMIS bit is set. Note, however, that when this condition occurs, the module assumes the ISYNC signal is incorrect, and does not set the write pointers to the beginning of the block. If this is necessary, the microprocessor must issue a resynchronize command by setting the RESYNC bit of the *Control* register. This bit will never be set when encoding.

RESYNC (W) - Resynchronize. When decoding, if the microprocessor has determined that synchronization has been lost in the data stream, it can issue a RESYNC by writing a one to this bit. This will cause the module to stop reading input data bits, discard all data read in the current input block, and wait for an ISYNC signal. Note that the block that is being decoded when a RESYNC is issued will be output to the data port. This bit must not be set when encoding.

COMPL (R) - Block Decode Complete Set at the completion of each block encoding or decoding cycle, indicating that the block is ready to be output. When decoding, the value of NITER[4:0], and the CORRECTIONS[9:0] count value in the *Correct* and *Status* registers are updated each time the COMPL bit is set. The CORINC bit will also be set with COMPL if the incomplete condition was detected.

DUMP (W) -Dump Current Block. When decoding, writing a 1 to this bit will cause the module to stop iterations on the current block and send it to the output data port. The dump occurs upon completion of the axis iteration following the current axis iteration. The worst case delay is two axis iterations (which is equal to one full iteration for 2D codes). Decoding then begins on the next (already loaded) data block, and another block can begin loading.

3.10 RESERVED, ADDRESS 0x08 - RESERVED

This register is for production test purposes only.

4.0 PERFORMANCE CURVES

The following figures show a comparison between the (64,57)x(64,57) TPC implementation which has a code rate of 0.793, and two Reed-Solomon/Viterbi implementations with code rates of 0.806 and 0.790. The modulation is Phase Shift Keying (PSK), and the channel model is Additive White Gaussian Noise (AWGN). Note that the TPC implementation consistently outperforms the RS/Viterbi implementations at 2 iterations. Additional iterations increase TPC performance.

Figure 12: Turbo Product Code vs. Reed-Solomon/Viterbi Performance Comparison

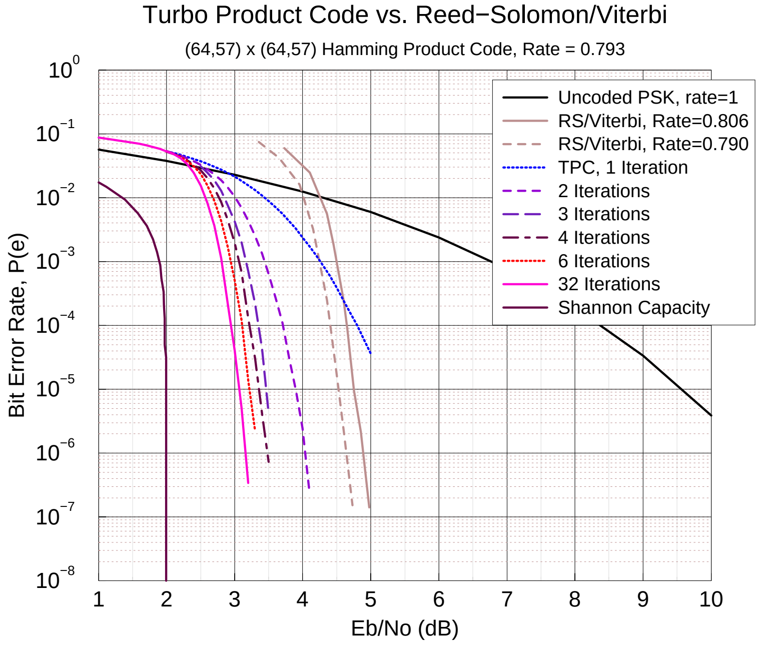


Figure 13 shows the performance of the three 4k block codes in the same channel with 32 iterations.

Figure 13: Comparison of TPC Code Types

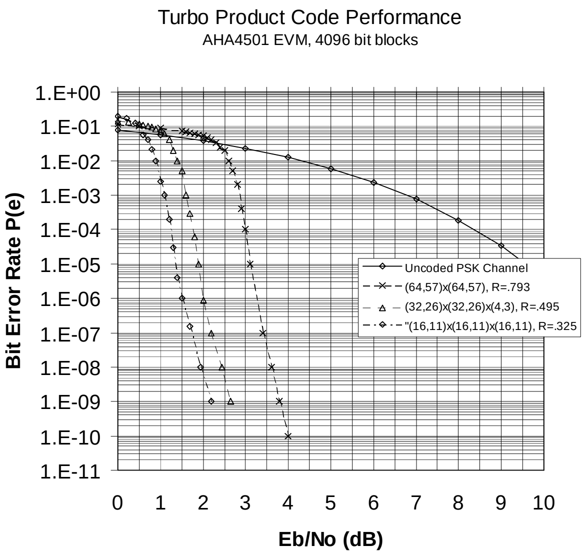
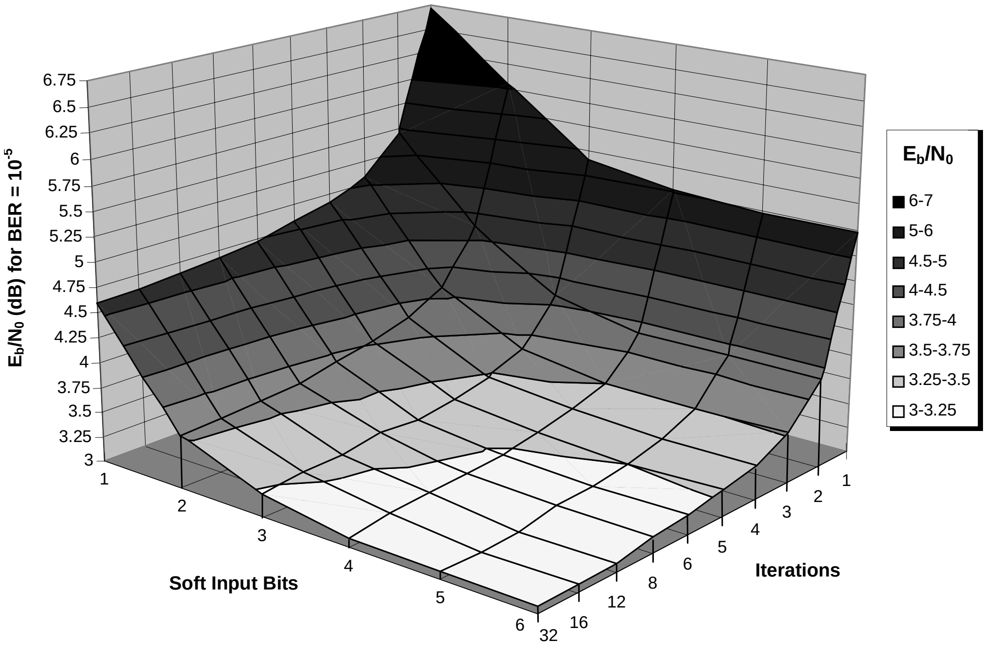


Figure 14 shows the E_b/N_o required to achieve a Bit Error Rate (BER) of $10^{-5.}$ using the (64,57)x(64,57) block code in an AWGN channel. This figure shows the correction performance trade-offs between input quantization bits and iterations (data rate). Optimum performance occurs with 32 iterations and 6 input quantization bits, however, excellent performance can be achieved with only 3 input quantization bits or only 3 iterations.

Figure 14: Performance Curve of E_b/N_0 for BER of 10^{-5}



5.0 SIGNAL DESCRIPTIONS

This section contains descriptions for all the pins. Each signal has a type code associated with it. The type codes are described in the following table.

TYPE CODE	*DESCRIPTION*
I	Input only pin
O	Output only pin
I/O	Input/Output pin
S	Synchronous signal
A	Asynchronous signal

5.1 SYSTEM CONTROL

SIGNAL	*TYPE*	*DESCRIPTION*
CLK	I	System Clock. 50 MHz maximum frequency.
RESETN	I A	Power on Reset. Active low reset signal. RESETN should be a minimum of 4 clock periods. When RESETN is asserted, all registers are reset as defined in Section 3.0 *Internal Register Programming*, all control signals are deasserted, and the data path is cleared.
TRI_STATE	I	Tristate Enable. When this pin is asserted high, the I/O and output signal drivers are tristated. Tied low for normal operation.
TESTMODE	I	Testmode Enable. Tied low for normal operation.

5.2 MICROPROCESSOR INTERFACE

SIGNAL	TYPE	DESCRIPTION
MDATA[7:0]	I/O A	Processor Data. Data for all microprocessor reads and writes of registers within the AHA4501 transfers across this bus.
MA[2:0]	I A	Processor Address Bus. Used to address internal registers within the AHA4501.
MCSN	I A	Processor Chip Select. Selects the AHA4501 as the source or destination of the current microprocessor bus cycle. MCSN needs to be active for a minimum of one clock cycle to start a microprocessor access.
MWRN_RWN	I A	Processor Read/Write Select. When PROCMODE is deasserted, this is the active low write enable signal. When PROCMODE is asserted, this is the active low read/write select signal.
MRDY_DTACKN	O A	Processor Ready/Data Transfer Acknowledge. When PROCMODE is deasserted, this is an active high ready signal. When PROCMODE is asserted, this signal is an active low data transfer acknowledge.
MRDN_DSN	I A	Processor Read Enable/Data Strobe. When PROCMODE is deasserted, this is the active low read enable signal. When PROCMODE is asserted, this is the active low data strobe signal.
MINTN_INTR	O A	Processor Interrupt. When PROCMODE is deasserted, this signal is active high. When PROCMODE is asserted, this signal is active low.
MALE	I A	Processor Address Latch Enable. When PROCMODE is not asserted and MUXMODE is asserted, this signal is the active high address latch enable. Otherwise, this pin is not used and must be tied low.
PROCMODE	I A	Processor Mode. Intel® mode when deasserted, Motorola® mode when asserted.
MUXMODE	I A	Muxed Processor Mode. Deassert for non-muxed address and data bus mode, assert for muxed address and data bus mode.

5.3 INPUT INTERFACE

SIGNAL	TYPE	DESCRIPTION
IDATA[7:0]	I S	Data input bus.
IRDY	I S	Input data ready. Data is registered into the AHA4501 on the rising edge of clock when IRDY and IACPT are asserted.
IACPT	O S	Input data accept. Data is registered into the AHA4501 on the rising edge of clock when IRDY and IACPT are asserted.
ISYNC	I S	Input synchronize. Asserted with IDATA for the first bit of data after the detection of a sync mark in the data stream. ISYNC is ignored when encoding.

5.4 OUTPUT INTERFACE

SIGNAL	TYPE	DESCRIPTION
ODATA[7:0]	O S	Data output bus.
ORDY	O S	Output data ready. Data is registered out of the AHA4501 on the rising edge of clock when ORDY and OACPT are asserted.
OACPT	I S	Output data accept. Data is registered out of the AHA4501 on the rising edge of clock when ORDY and OACPT are asserted.
OSYNC	O S	Output synchronize. Asserted with ODATA for the first bit in every x blocks, as programmed in the SYNC MARK FREQUENCY[3:0] section of the *Sync* register.

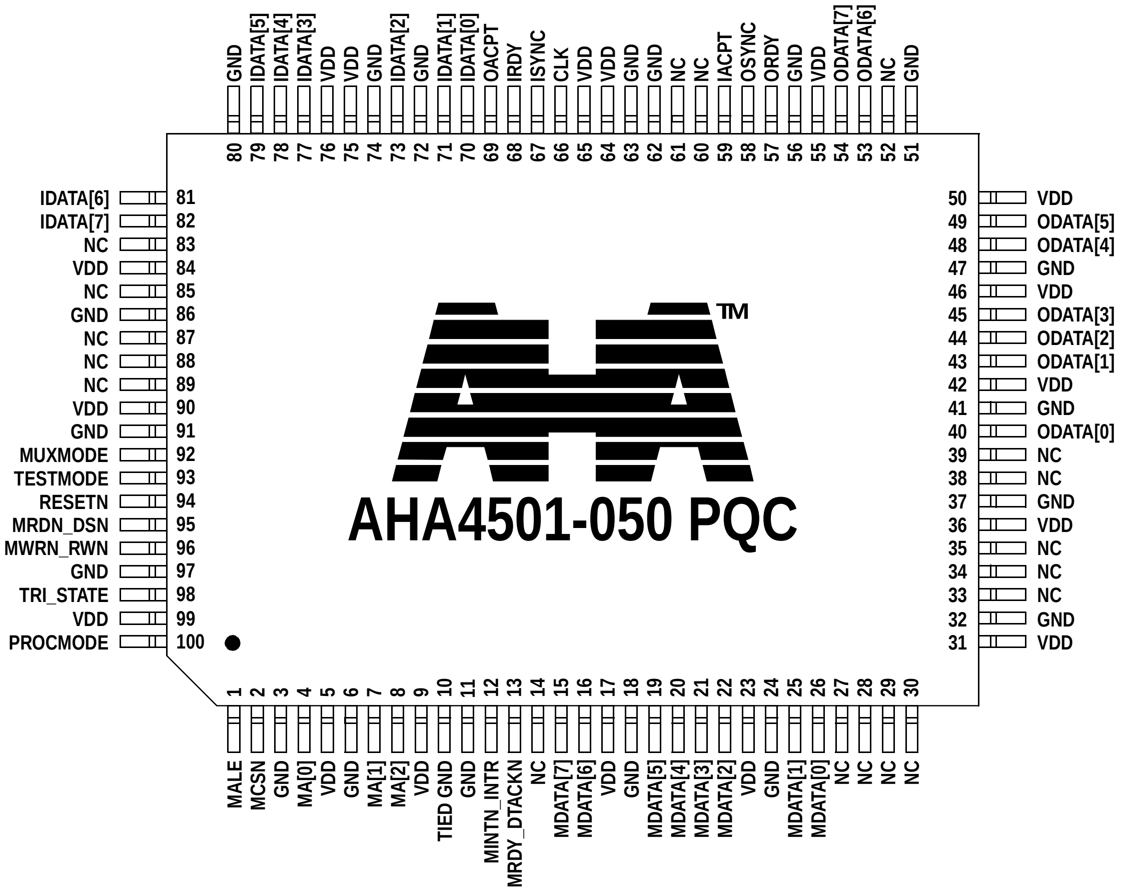
6.0 PINOUT

Table 5: Pin Designation

PIN	SIGNAL
1	MALE
2	MCSN
3	GND
4	MA[0]
5	VDD
6	GND
7	MA[1]
8	MA[2]
9	VDD
10	TIED GND
11	GND
12	MINTN_INTR
13	MRDY_DTACKN
14	NC
15	MDATA[7]
16	MDATA[6]
17	VDD
18	GND
19	MDATA[5]
20	MDATA[4]
21	MDATA[3]
22	MDATA[2]
23	VDD
24	GND
25	MDATA[1]
26	MDATA[0]
27	NC
28	NC
29	NC
30	NC
31	VDD
32	GND
33	NC
34	NC
35	NC
36	VDD
37	GND
38	NC
39	NC
40	ODATA[0]
41	GND
42	VDD
43	ODATA[1]
44	ODATA[2]
45	ODATA[3]
46	VDD
47	GND
48	ODATA[4]
49	ODATA[5]
50	VDD
51	GND
52	NC
53	ODATA[6]
54	ODATA[7]
55	VDD
56	GND
57	ORDY
58	OSYNC
59	IACPT
60	NC
61	NC
62	GND
63	GND
64	VDD
65	VDD
66	CLK
67	ISYNC
68	IRDY
69	OACPT
70	IDATA[0]
71	IDATA[1]
72	GND
73	IDATA[2]
74	GND
75	VDD
76	VDD
77	IDATA[3]
78	IDATA[4]
79	IDATA[5]
80	GND
81	IDATA[6]
82	IDATA[7]
83	NC
84	VDD
85	NC
86	GND
87	NC
88	NC
89	NC
90	VDD
91	GND
92	MUXMODE
93	TESTMODE
94	RESETN
95	MRDN_DSN
96	MWRN_RWN
97	GND
98	TRI_STATE
99	VDD
100	PROCMODE

NC - not connected internally.

Figure 15: Pinout – 100 MQFP



7.0 ELECTRICAL SPECIFICATIONS

7.1 ABSOLUTE MAXIMUM RATINGS

SYMBOL	PARAMETER	MIN	MAX	UNITS
V_{DD}	Power supply voltage		4.6	Volts
V_{PIN}	Voltage applied to any pin	-0.5	4.6	Volts

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

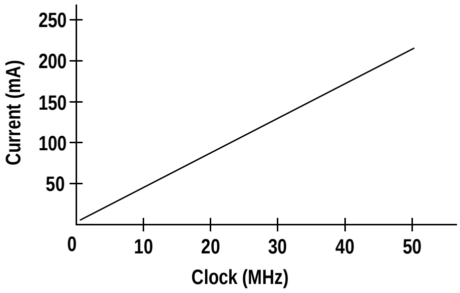
7.2 RECOMMENDED OPERATING CONDITIONS

SYMBOL	PARAMETER	MIN	MAX	UNITS
V_{DD}	Power supply voltage	3.0	3.6	Volts
T_A	Operating temperature	0	70	°C

7.2.1 DC SPECIFICATIONS

SYMBOL	PARAMETER	CONDITIONS	MIN	MAX	UNITS
V_{IL}	Input low voltage		–0.3	0.8	Volts
V_{IH}	Input high voltage		2.0	V_{DD}–0.3	Volts
V_{OL}	Output low voltage	4ma output loads		0.4	Volts
V_{OH}	Output high voltage	4ma output loads	2.4		Volts
I_{IL}	Input low current	V_{IN} = 0 Volts		-5	µAmps
I_{IH}	Input high current	V_{IN} = V_{DD} Volts		5	µAmps
I_{DD}	Active I_{DD} current	50 MHz clock, decoding at maximum data rate, V_{DD}=3.3V, no loads		250	mAmps
I_{DD}	Supply current (static)			1.0	mAmps
I_{DD}	Standby current	Chip idle, 50 MHz clock, V_{DD}=3.3V, no loads		20	mAmps
I_{OL}	Output low current			4	mAmps
I_{OH}	Output high current			4	mAmps

Figure 16: Current vs. Data Rate (typ)

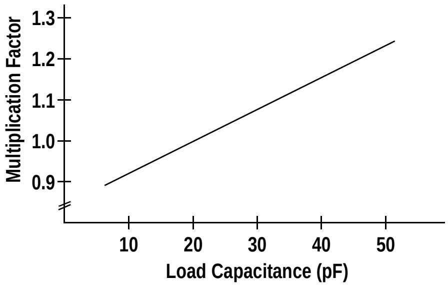


7.2.2 TEST CONDITIONS

PARAMETER	VALUE
AC timing reference	1.4 V

Note: The timing diagrams for these signals assume a capacitive load of 20pF. The specified signal timings must be derated by the factor shown in Figure 17 when operating at loads other than 20pF.

Figure 17: Signal Timing vs. Output Load



LOAD CAPACITANCE	MULTIPLICATION FACTOR
10 pF	0.92
20 pF	1.00
30 pF	1.08
40 pF	1.16
50 pF*	1.25

*Production test conditions

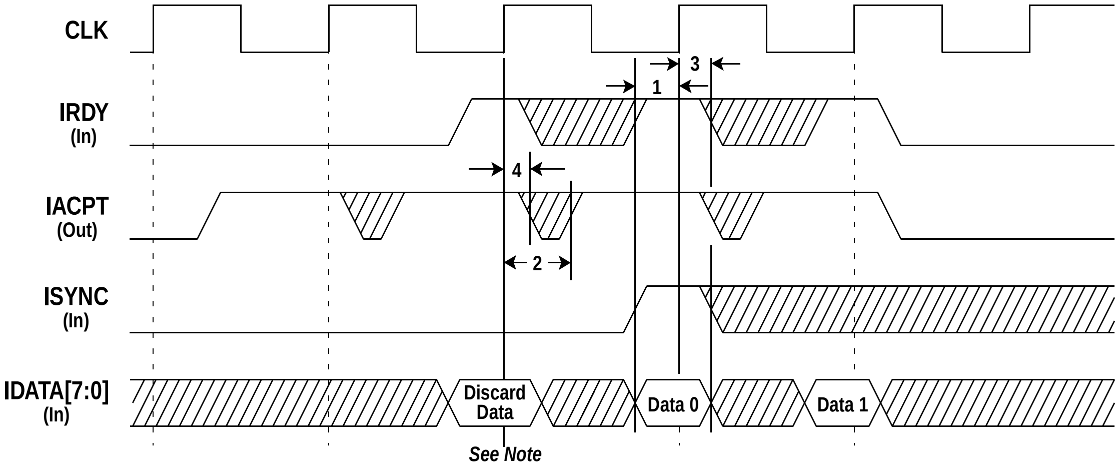
7.2.3 PIN CAPACITANCE

SYMBOL	PARAMETER	MAX	UNITS
C_{IN}	Input capacitance	10	pF
C_{OUT}	Output self load capacitance	10	pF
C_{IO}	I/O self load capacitance	10	pF

Notes: Not tested in production.

8.0 TIMING SPECIFICATIONS

Figure 18: Data Input Timing



Note: For the first block after reset, the input data is discarded until the ISYNC signal is asserted (decoding only).

Table 6: Data Input Timing Requirements

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	IRDY, ISYNC, IDATA setup to CLK rising edge	5		ns	
2	IACPT delay from CLK rising edge		9	ns	
3	IRDY, ISYNC, IDATA hold from CLK rising edge	2		ns	
4	IACPT hold from CLK rising edge	2		ns	

Figure 19: Data Output Timing

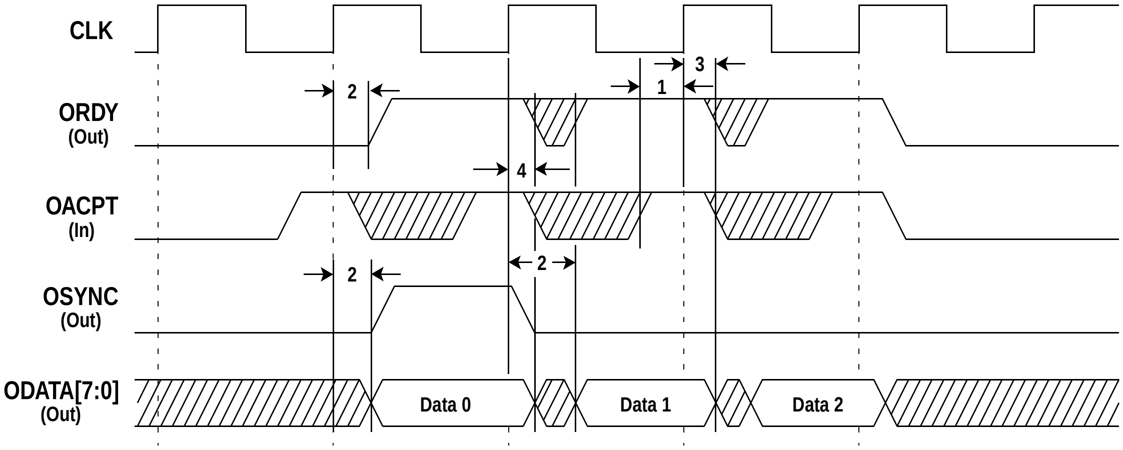


Table 7: Data Output Timing Requirements

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	OACPT setup to CLK rising edge	5		ns	
2	ORDY, OSYNC, ODATA delay from CLK rising edge		9	ns	
3	OACPT hold from CLK rising edge	2		ns	
4	ORDY, OSYNC, ODATA hold from CLK rising edge	2		ns	

Figure 20: Microprocessor Interface Timing (Write); PROCMODE=0, MUXMODE=0

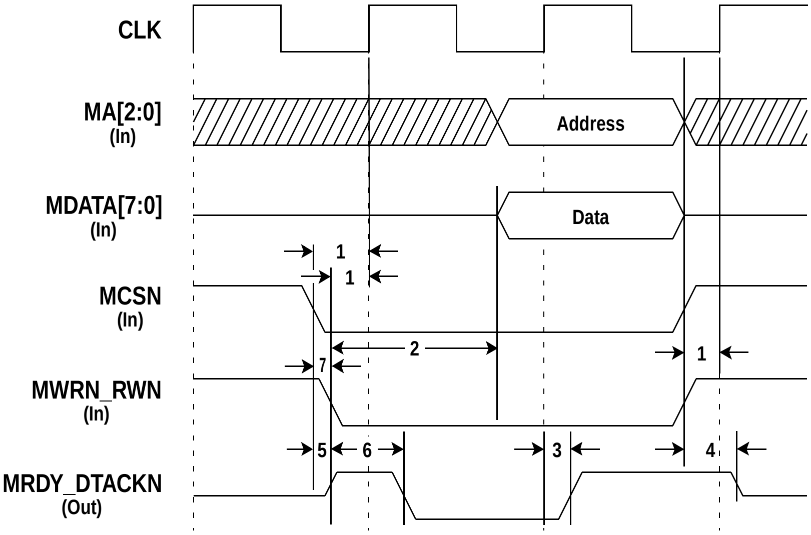


Table 8: Microprocessor Interface Timing Requirements - Write

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	MCSN, MWRN_RWN setup to CLK rising edge	5		ns	1,2,3,4
2	MWRN_RWN low to MDATA[7:0] valid		1 Tcp–3ns	ns	5
3	CLK rising edge to MRDY_DTACKN high		15	ns	
4	MCSN high to MRDY_DTACKN tristate		12	ns	
5	MCSN low to MRDY_DTACKN active		12	ns	
6	MWRN_RWN low to MRDY_DTACKN low		15	ns	
7	MCSN low setup to MWRN_RWN low	2		ns	6

Notes:

1) *The microprocessor interface can be asynchronous to CLK. The above timings indicate the required setup and hold to allow for fastest microprocessor accesses.*
2) *Write cycle begins when both MCSN and MWRN_RWN are low and meet setup to rising edge of CLK.*
3) *MCSN may be held low continuously for back to back accesses.*
4) *Neither MRDN_DSN nor MWRN_RWN may pulse high or low for less than one clock period.*
5) *Tcp = clock period. (ns)*
6) *Asynchronous operation only*

Figure 21: Microprocessor Interface Timing (Read); PROCMODE=0, MUXMODE=0

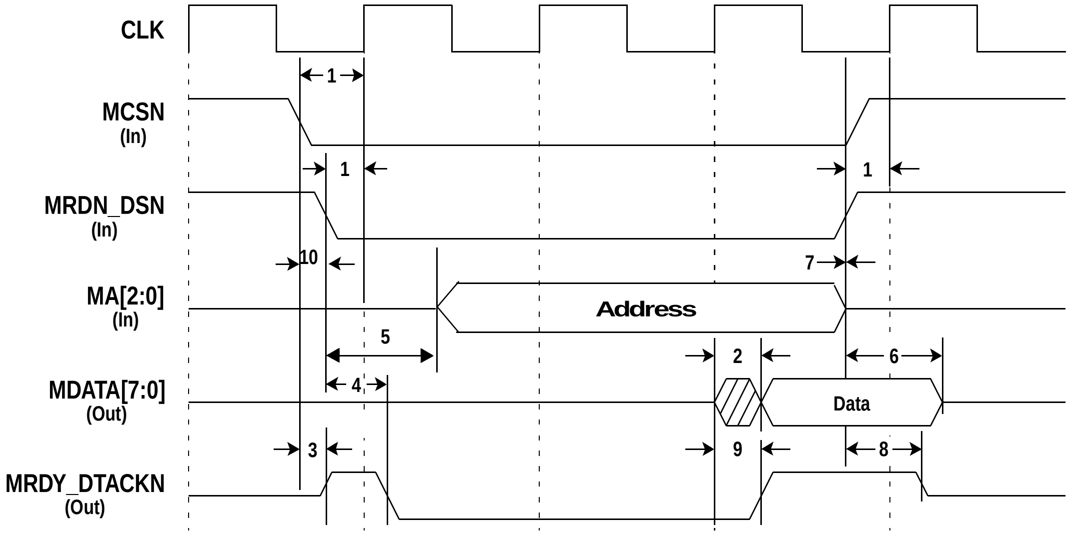


Table 9: Microprocessor Interface Timing Requirements - Read; PROCMODE=0, MUXMODE=0

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	MCSN/MRDN_DSN setup to CLK rising edge	5		ns	1,2,3,4
2	CLK rising edge to MDATA[7:0] valid		11	ns	
3	MCSN low to MRDY_DTACKN active		12	ns	
4	MRDN_DSN low to MRDY_DTACKN low		15	ns	
5	MA[2:0] valid from MRDN_DSN low		1 Tcp–3ns	ns	5
6	MRDN_DSN high to MDATA[7:0] tristate	2	12	ns	
7	MA[2:0] hold from MRDN_DSN	0		ns	
8	MCSN high to MRDY_DTACKN tristate		12	ns	
9	CLK rising edge to MRDY_DTACKN high	3	15	ns	
10	MCSN low setup to MRDN_DSN low	2		ns	6

Notes:

1) *The microprocessor interface can be asynchronous to CLK. The above timings indicate the required setup and hold to allow for fastest microprocessor accesses.*
2) *Write cycle begins when both MCSN and MRDN_DSN are low and meet setup to rising edge of CLK.*
3) *MCSN may be held low continuously for back to back accesses.*
4) *Neither MRDN_DSN nor MWRN_RWN may pulse high or low for less than one CLK period.*
5) *Tcp = clock period. (ns)*
6) *Asynchronous operation only.*

Figure 22: Microprocessor Interface Timing (Write); PROCMODE=0, MUXMODE=1

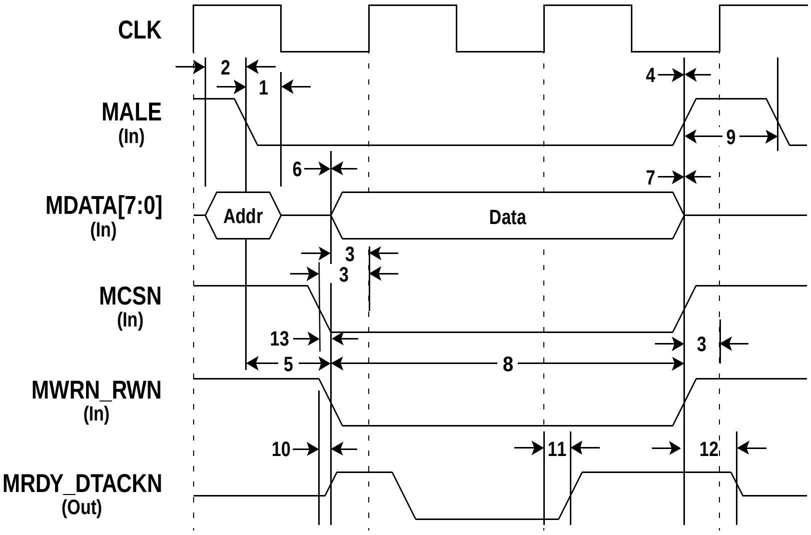


Table 10: Microprocessor Interface Timing Requirements - Write; PROCMODE=0, MUXMODE=1

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	Address hold from MALE falling edge	10		ns	
2	Address setup to MALE falling edge	7		ns	
3	MCSN/MWRN_RWN setup to CLK rising edge	10		ns	1,2,3,4
4	MALE hold from MWRN_RWN rising edge	0		ns	
5	MALE setup to MWRN_RWN falling edge	10		ns	
6	Write data setup to MWRN_RWN falling edge	0		ns	
7	MDATA[7:0] and CSN hold from MWRN_RWN rising edge	0		ns	
8	MWRN_RWN low width	2		clocks	
9	MALE high width	10		ns	
10	MCSN low to MRDY_DTACKN active		12	ns	
11	CLK rising edge to MRDY_DTACKN high		15	ns	
12	MCSN high to MRDY_DTACKN tristate		12	ns	
13	MCSN low setup to MWRN_RWN low	2		ns	

Notes:

1) *The microprocessor interface can be asynchronous to CLK. The above timings indicate the required setup and hold to allow for fastest microprocessor accesses.*
2) *Write cycle begins when both MCSN and MWRN_RWN are low and meet setup to rising edge of CLK.*
3) *MCSN may be held low continuously for back to back accesses.*
4) *Neither MRDN_DSN nor MWRN_RWN may pulse high or low for less than one clock period.*
5) *Asynchronous operation only.*

Figure 23: Microprocessor Interface Timing (Read); PROCMODE=0, MUXMODE=1

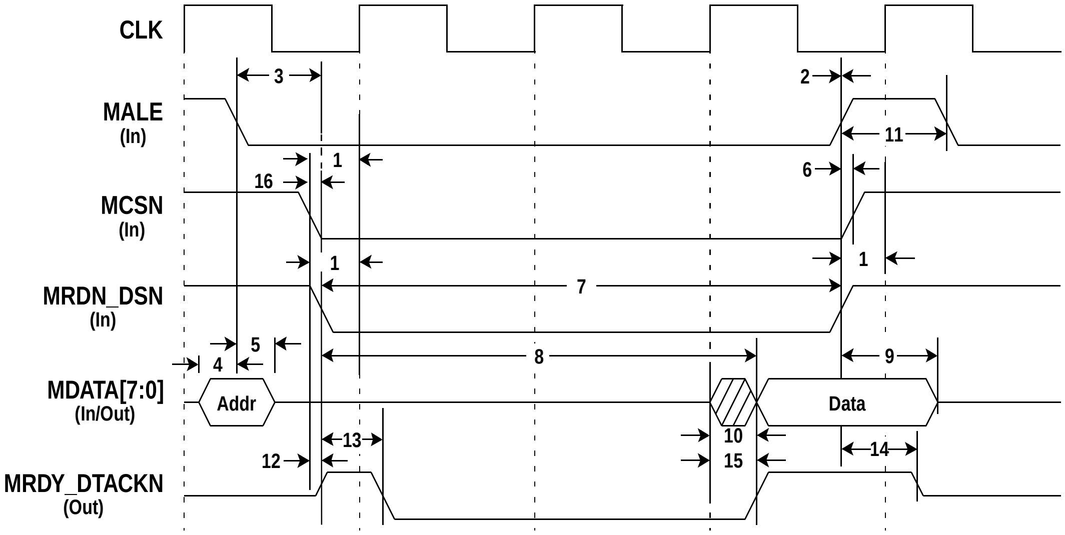


Table 11: Microprocessor Interface Timing Requirements - Read; PROCMODE=0, MUXMODE=1

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	MCSN/MRDN_DSN setup to CLK rising edge	10		ns	1,2,3,4
2	MALE hold from MRDN_DSN rising edge	0		ns	
3	MALE setup to MRDN_DSN falling edge	10		ns	
4	Address setup to MALE falling edge	7		ns	
5	Address hold from MALE falling edge	10		ns	
6	MCSN hold from MRDN_DSN rising edge	0		ns	
7	MRDN_DSN low width	2		clocks	5
8	MRDN_DSN falling edge to MDATA[7:0] valid		2 Tcp+11 ns	ns	5
9	MRDN_DSN high to MDATA[7:0] tristate	2	12	ns	
10	Rising edge of CLK to MDATA[7:0] valid		11	ns	
11	MALE high width	10		ns	
12	MCSN low to MRDY_DTACKN active		12	ns	
13	MRDN_DSN low to MRDY_DTACKN low		15	ns	
14	MCSN high to MRDY_DTACKN tristate		12	ns	
15	CLK rising edge to MRDY_DTACKN high	3	15	ns	
16	MCSN low setup to MRDN_DSN low	2		ns	6

Notes:

1) *The microprocessor interface can be asynchronous to CLK. The above timings indicate the required setup and hold to allow for fastest microprocessor accesses.*
2) *Write cycle begins when both CSN and MRDN_DSN are low and meet setup to rising edge of CLK.*
3) *CSN may be held low continuously for back to back accesses.*
4) *Neither MRDN_DSN nor MWRN_RWN may pulse high or low for less than one CLK period.*
5) *Tcp = clock period. (ns)*
6) *Asynchronous operation only.*

Figure 24: Microprocessor Interface Timing (Write); PROCMODE=1, MUXMODE=0

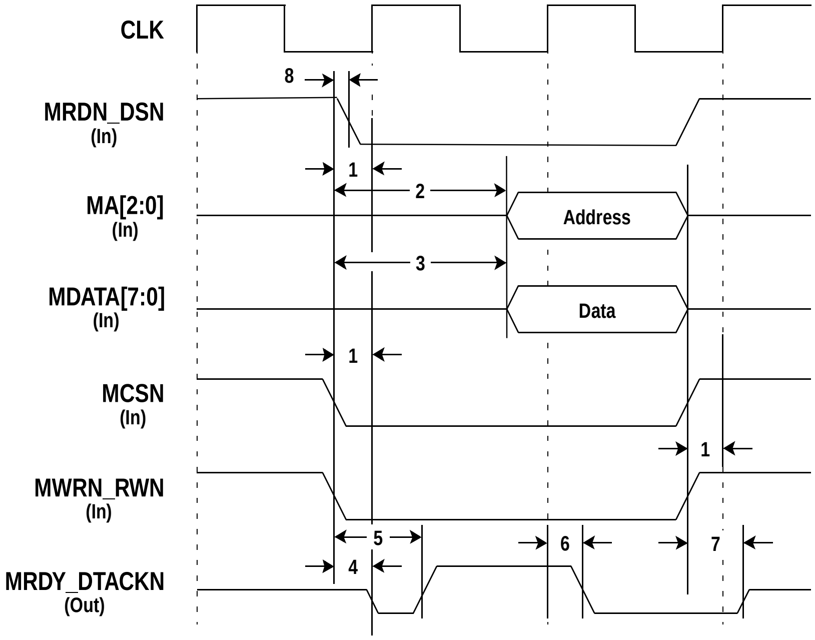


Table 12: Microprocessor Interface Timing Requirements - Write; PROCMODE=1, MUXMODE=0

NUMBER	*PARAMETER*	*MIN*	*MAX*	*UNITS*	*NOTES*
1	MCSN, MRDN_DSN, MWRN_RWN setup to CLK rising edge	5		ns	1,2,3,4
2	MRDN_DSN low to MDATA[7:0] valid		1 Tcp–3ns	ns	5
3	MRDN_DSN low to MA[2:0] valid		1 Tcp–3ns	ns	5
4	MCSN low to MRDY_DTACKN active		12	ns	
5	MRDN_DSN low to MRDY_DTACKN high		15	ns	
6	CLK rising edge to MRDY_DTACKN low		15	ns	
7	MCSN high to MRDY_DTACKN tristate	0		ns	
8	MCSN low setup to MRDN_DSN low	2		ns	6

Notes:

1) *The microprocessor interface can be asynchronous to CLK. The above timings indicate the required setup and hold to allow for fastest microprocessor accesses.*
2) *Write cycle begins when both MCSN and MWRN_RWN are low and meet setup to rising edge of CLK.*
3) *MCSN may be held low continuously for back to back accesses.*
4) *Neither MRDN_DSN nor MWRN_RWN may pulse high or low for less than one clock period. Only required to be recognized on current clock edge. If less than this number, cycle will be recognized on the next clock.*
5) *Tcp = clock period. (ns)*
6) *Asynchronous operation only.*

Figure 25: Microprocessor Interface Timing (Read); PROCMODE=1, MUXMODE=0

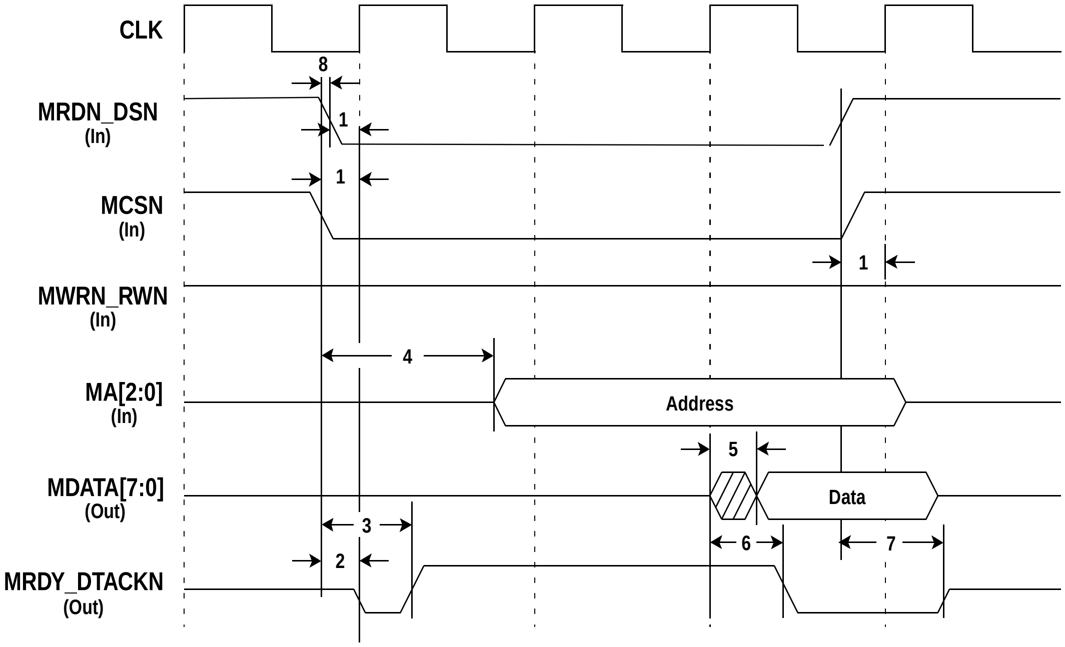


Table 13: Microprocessor Interface Timing Requirements - Read; PROCMODE=1, MUXMODE=0

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	MCSN, MRDN_DSN, MWRN_RWN setup to CLK rising edge	5		ns	1,2,3,4
2	MCSN low to MRDY_DTACKN active		12	ns	
3	MRDN_DSN low to MRDY_DTACKN high		15	ns	
4	Address valid from MRDN_DSN low		1 Tcp–3ns	ns	5
5	CLK rising edge to MDATA[7:0] valid		11	ns	
6	CLK rising edge to MRDY_DTACKN low	3	15	ns	
7	MRDN_DSN high to MRDY_DTACKN tristate		12	ns	
8	MCSN low to MRDN_DSN low	2		ns	6

Notes:

1) *The microprocessor interface can be asynchronous to CLK. The above timings indicate the required setup and hold to allow for fastest microprocessor accesses.*
2) *Write cycle begins when both MCSN and MRDN_DSN are low and meet setup to rising edge of CLK.*
3) *MCSN may be held low continuously for back to back accesses.*
4) *Neither MRDN_DSN nor MWRN_RWN may pulse high or low for less than one CLK period.*
5) *Tcp = clock period. (ns)*
6) *Asynchronous operation only.*

Figure 26: Interrupt Timing

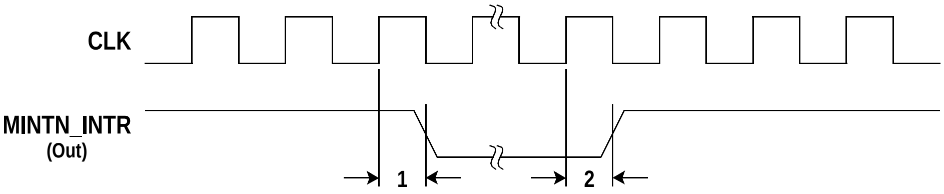


Table 14: Interrupt Timing Requirements

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	MINTN_INTR delay time		12	ns	
2	MINTN_INTR hold time	2		ns	

Figure 27: Clock Timing

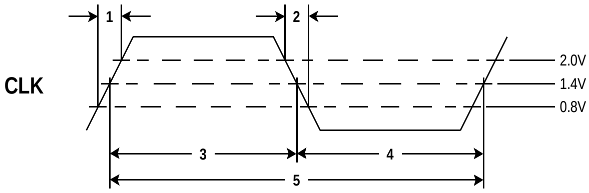


Table 15: Clock Timing Requirements

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	Clock rise time		2	ns	1
2	Clock fall time		2	ns	1
3	Clock high time	8		ns	
4	Clock low time	8		ns	
5	Clock period	20		ns	

Notes:

1) Not tested in production.

Figure 28: Power On Reset Timing

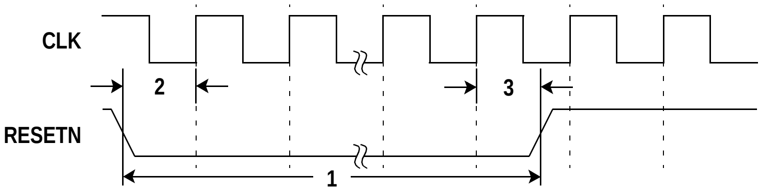


Table 16: Power On Reset Timing Requirements

NUMBER	PARAMETER	MIN	MAX	UNITS	NOTES
1	RESETN low pulsewidth	4		clocks	
2	RESETN setup to clock rise	6		ns	1
3	RESETN hold time	3		ns	1

Notes:

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

9.0 PACKAGING

Figure 29: AHA4501 Package Specifications – 100 MQFP

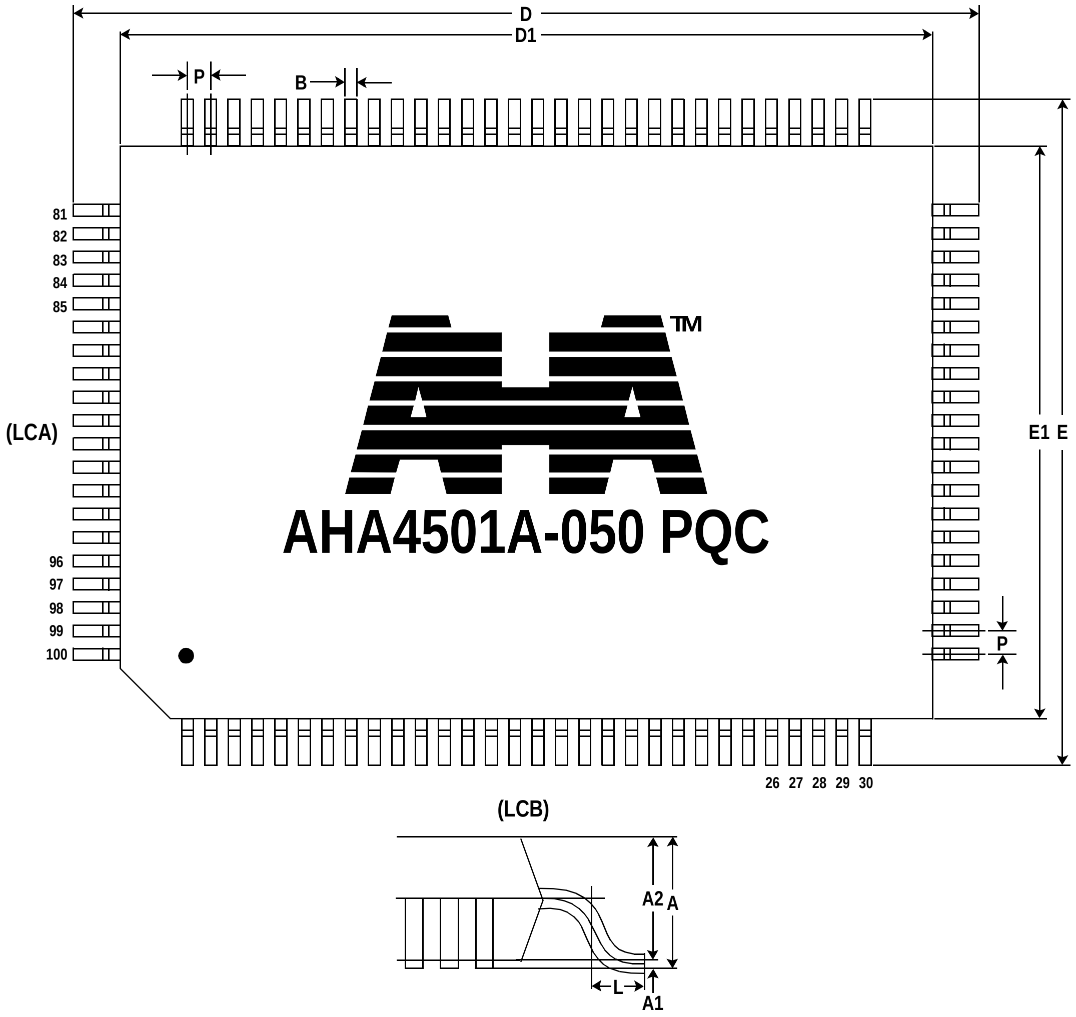


Table 17: PQFP (Plastic Quad Flat Pack) 14 × 20 mm Package Dimensions

(All dimensions are in mm)

SYMBOL	*NUMBER OF PIN AND SPECIFICATION DIMENSION*		
	100		
	RB		
	MIN	NOM	MAX
(LCA)	20		
(LCB)	30		
A			3.1
A1	0.1	0.23	0.36
A2	2.57	2.71	2.87
D	23.65	23.9	24.15
D1	19.9	20	20.1
E	17.65	17.9	18.15
E1	13.9	14	14.1
L	0.73	0.88	1.03
P		0.65	
B	0.22	0.3	0.33

JEDEC Outline MO-112

10.0 ORDERING INFORMATION

10.1 AVAILABLE PARTS

PART NUMBER	*DESCRIPTION*
AHA4501-050 PQC	AHA4501 Astro 36 Mbits/sec Turbo Product Code Encoder/Decoders, 3.3 V

10.2 PART NUMBERING

AHA	*4501*	*A*	*050*	*P*	*Q*	*C*
Manufacturer	Device Number	Revision Level	Speed Designation	Package Material	Package Type	Test Specification

Device Number:

4501

Revision Letter:

A

Speed Designation:

50 MHz

Package Material Codes:

P	Plastic

Package Type Codes:

Q	Quad

Test Specifications:

C	Commercial	0°C to +70°C

11.0 RELATED PUBLICATIONS

PART NUMBER	*DESCRIPTION*
PB4501	AHA Product Brief – AHA4501 Astro 36 Mbits/sec Turbo Product Code Encoder/Decoder
PB4501EVM	AHA Product Brief – AHA4501 TPC EVM ISA Evaluation Module
PB4501EVSW	AHA Product Brief – AHA4501 TPC Windows Evaluation Software
PB4540	AHA Product Brief – AHA4540 Astro OC-3 155 Mbits/sec Turbo Product Code Encoder/Decoder
PBGALAXY	AHA Product Brief – Galaxy Core Generator Turbo Product Code Decoder Cores
PBGALAXY_EVSW	AHA Product Brief – AHA Galaxy TPC Windows Evaluation Software
PBGALAXY_STK	AHA Product Brief – AHA Galaxy Simulation Tool Kit
PS4540	AHA Product Specification – AHA4540 Astro OC-3 155 Mbits/sec Turbo Product Code Encoder/Decoder
ANTPC01	AHA Application Note – Primer: Turbo Product Codes
ANTPC02	AHA Application Note – Use and Performance of Shortened Codes with the AHA4501 Turbo Product Code Encoder/Decoder
ANTPC03	AHA Application Note – Use and Performance of the AHA4501 Turbo Product Code Encoder/Decoder with Quadrature Amplitude Modulation (QAM)
ANTPC04	AHA Application Note – Use and Performance of the AHA4501 Turbo Product Code Encoder/Decoder with Differential Phase Shift Keying (DPSK)
ANTPC05	AHA Application Note – AHA4501 Turbo Product Code Encoder/Decoder Designer’s Guide
ANTPC06	AHA Application Note – AHA4501 Turbo Product Code Encoder/Decoder Frequently Asked Questions (FAQ)
ANTPC07	AHA Application Note – Turbo Product Codes for LMDS
ANTPC08	AHA Application Note – Using Multiple AHA4501 Devices in Parallel for Higher Data Rates
TPCEVAL	AHA Evaluation Software – Turbo Product Codes - Windows Evaluation Software

This product and the algorithm are covered under multiple applied patents.