Product Specification

AHA4011C

10 MBytes/sec Reed-Solomon Error Correction Device

PS4011C_1205

A subsidiary of Comtech Telecommunications Corporation

1126 Alturas Drive ■ Moscow ID 83843 ■ tel: 208.892.5600 ■ fax: 208.892.5601 ■ www.aha.com

Table of Contents

List of Figures

List of Tables

1.0 INTRODUCTION

The AHA4011C is a single chip integrated circuit that implements a high speed Reed-Solomon Forward Error Correction algorithm. The AHA4011C is a member of the AHA PerFEC family of high speed forward error correction (FEC) devices conforming to the Intelsat IESS-308 specification.

The device supports several programmable parameters including block size, error threshold, number of check bytes, order of output and mode of operations. Shortened blocks are supported without requirement of zero padding typically required in Reed Solomon decoders. The data input port is used to initialize the programmable parameters and the two on-chip buffers are used to input and output data. Discontinuities in data flow may be controlled by dedicated control pins.

High operating frequency, input and output data rate flexibility, low processing latency and various programmable parameters make this device ideal for many applications including: DTV, DBS, ADSL, Satellite Communications, ISDN, High Performance Modems and networks.

This specification provides full electrical and mechanical information to help a system engineer develop a system using AHA4011C. This document contains descriptions on correction terms, pinout, functions and features, DC and AC characteristics, package and mechanical specifications, and ordering information. Software simulation of the RS code as implemented in the device is also available.

1.1 FEATURES

HIGH PERFORMANCE

- Polynomial complies to Intelsat IESS-308; RTCA DO-217 Appendix F, Revision D and proposed ITU-TS SG-18 (Formerly CCITT SG-18) standards
- 40 MBytes/sec burst transfer rate with a 40 MHz clock for all block lengths
- Maximum channel rate of 10 MBytes/sec continuous for block lengths from 54 bytes through 255 bytes using a 40 MHz clock
- Processing latency time less than 15.2 µsec in continuous operation for block lengths of 100 bytes

FLEXIBILITY

- Programmable to correct from 1 to 10 error bytes or 20 erasure bytes per block
- Block lengths programmable from 3 to 255 bytes
- Encode, decode or pass-through capability in-line with data flow
- Outputs corrected data or correction vectors in forward or reverse order
- Continuous or burst data transfer
- Programmable error threshold to help determine channel performance

SYSTEM INTERFACE

- Byte wide synchronous I/O ports with internal buffering on both ports
- Dedicated control pins permit discontinuities in system data flow

OTHERS

- 44 pin PLCC; 50 mil lead pitch
- Pin and program compatible with the AHA4012
- Pin compatible with the AHA4013
- Software emulation of the algorithm available
- RoHS compliant

1.2 CONVENTIONS, NOTATIONS AND DEFINITIONS

– Certain signals are logically true at a voltage defined as “low” in the data sheet. All such signals have an “N” appended to the end of the signal name. For example, RSTN and DSON.
– “Signal assertion” means the output signal is logically true.
– Hex values are defined with a prefix of “0x”, such as “0x10”.
– A range of signal names is denoted by a set of colons between the numbers. Most significant bit is always shown first, followed by least significant bit. For example, DI[7:0] represents Data Input Bus 7 through 0.
– A product of two variables is expressed with an “×”, for example, $N \times C_i$ represents Codeword Length multiplied by Input clocks/byte.
– Mega Bytes per second is referred to as MBytes/sec or MB/sec.
– Channel Rate is defined as transfer rate including user data and error correction check bytes.

1.2.1 DEFINITION OF CORRECTION TERMS

TERM	*NAME (other references)*	*DEFINITION*	*RANGE (number of bytes)*
K	Message Length (user data or message bytes)	Number of user data symbols in one message block. Size of a symbol in AHA4011C is 8-bits. Message length is K = N − R. The first message byte is referred to as X_{K-1}; the last message byte is X_0.	1 through 253 (1, 2, 3, 4 . . . 253)
R	Check Symbols (parity or redundancy)	Symbols appended to the user data to detect and correct errors. The number of check symbols required in a system is $R \geq E + 2e$.* The first check symbol is referred to as Y_{R-1}; the last check symbol is Y_0.	2 through 20 in increments of 1 (2, 3, 4 . . . 20)
N	Codeword Length (block length)	Sum of message and check symbols. N = K + R.	3 through 255 (3, 4, 5, 6 . . . 255)
t	Error Corrections	Maximum number of error corrections performed by the device. The value is t = Integer $\frac{N-K}{2}$.	1 through 10 (1, 2, 3 . . . 10)
P	Error Threshold	The threshold limit to determine uncorrectability of a Codeword and the number of check bytes allocated for correction-only purposes (not for detection).	2 through 20 (2, 3, 4 . . . 20)
e	Number of Errors	An error is defined as an erroneous byte whose correct value and position within the message block are both unknown.	0 through N
E	Number of Erasures	An erasure is defined as an error whose position is known within the message block.**	0 through N
G	Burden of Correction	A measure of the burden of correction being placed on the capabilities of the device for that message block. The value G = 2e + E.	0 through R

* *For every 2 check bytes, the AHA4011C can correct either 2 erasures or 1 error.*

** *An erasure is detected by a parity detector or a signal dropout detector. The presence of an erasure is indicated by asserting the ERASE signal when the erased byte is clocked into the AHA4011C.*

2.0 FUNCTIONAL DESCRIPTION

This section describes an architectural overview of the chip and its many functions, features and operations. The block diagram for the chip shows the Reed-Solomon ECC module, the Input and Output Buffers, and their associated control. All input and output data are clocked on the rising edge of CLK.

2.1 FUNCTIONAL OVERVIEW

The AHA4011C Reed-Solomon codec (encoder/decoder) is a member of the AHA PerFEC family of high speed forward error correction (FEC) devices. This single chip, three-layer metal, CMOS device can operate in encode, decode or pass-through modes.

The ECC core implements a full error correcting Reed-Solomon decoder. This code is capable of correcting up to 10 (t=10) byte-errors or 20 (t=10) erasures in a RS block.

The ECC core has three phases of operation: Data In, Calculation and Data Out. Data to be processed is first input into a single ported Input Buffer using a control signal DSIN. ECC core arbitrates for the input data out of the Input Buffer. ECC core has access to the Input Buffer on clock edges where DSIN is not asserted.

Each block is processed within the ECC core and calculations are made. The entire block is processed through the ECC core, and transferred into the Output Buffer. The device asserts RDYON signal and holds active until the Output Buffer is completely emptied.

The ECC core loads the Output Buffer in reverse order for either operation. Data may be strobed out of the device in forward or reverse order. If forward order is desired, output data cannot be strobed out of the device until the entire block has been loaded into the Output Buffer.

The use of internal buffers is restricted per the rules defined in Section 2.9 *Data Rates and Latencies*.

Maximum delay required for each block of a given length to pass through the device is fixed, and does not vary with the location or the number of errors received. This delay (or latency), expressed in the number of clocks is discussed in a later section.

2.2 CORRECTING CAPABILITY AND POLYNOMIALS

Compared with other codes, RS codes require relatively few "overhead" check bytes to be added to the data stream to achieve a high degree of error detection and correction. Since the AHA4011C deals with bytes (or symbols) rather than with individual bits, when a byte is in error it does not matter how many bits within the byte are corrupted; it is counted as one error.

The Reed-Solomon code is defined over the finite field GF(2^8). The field defining primitive polynomial is:

$$P(x) = x^8 + x^7 + x^2 + x + 1$$

and the generator polynomial, dependent on the variable R, is given by:

$$G(x) = \prod_{i=120}^{119+R} (x - \alpha^i)$$

where R ∈ {2, 3, 4, 5,. . . 20} for the AHA4011C. This polynomial is specified in international standards, Intelsat IESS 308; RTCA DO-217 Appendix F (Rev D) and the proposed CCITT SG-18.

For every 2 check bytes, the decoder corrects either 2 erasures or 1 error. An erasure can be determined with a parity detector or a signal dropout detector external to the chip. An erasure is indicated by the ERASE signal when the erased byte is clocked in the device.

Correcting "erasures" takes only half as much of the correction capability of the RS code as it takes to correct "errors", since the position information is already known for "erasures". The correction ability of the code is bounded as:

$$R \geq \#\text{ erasures} + 2(\#\text{ errors})$$

Valid block length (N) is defined by the relationship:

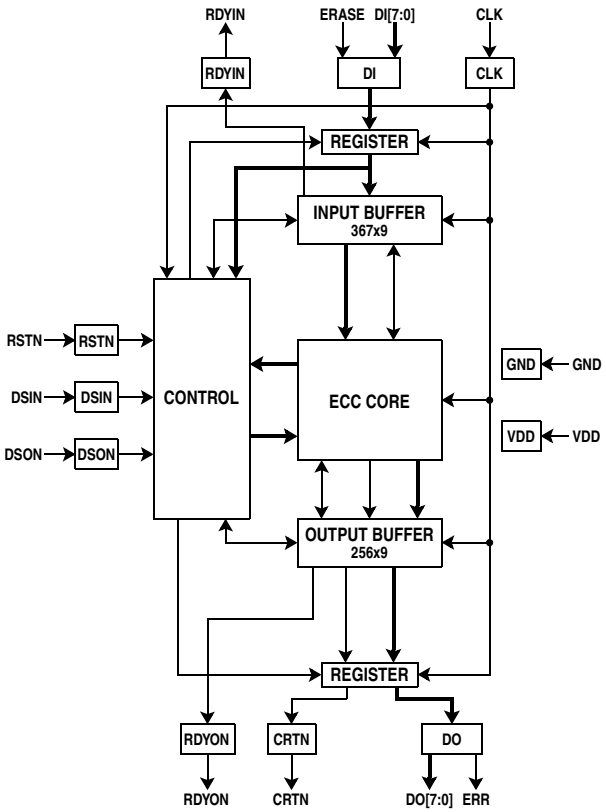
$$R + 1 \leq N \leq 255$$

where R ranges from 2 to 20.

A complete codeword can therefore range from a minimum of 3 bytes to a maximum of 255 bytes.

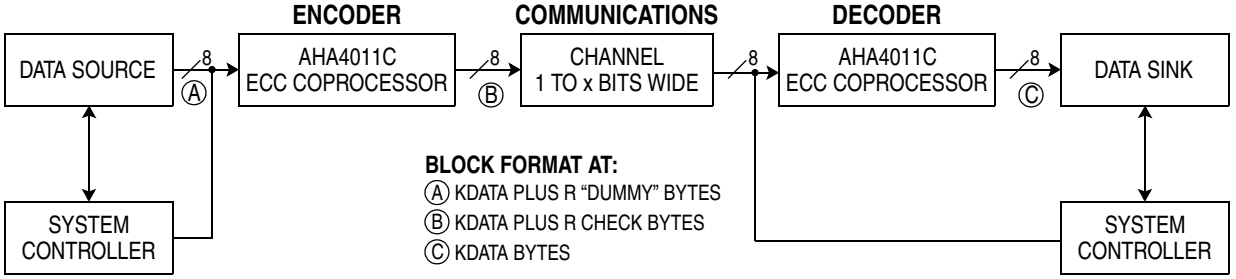
For further discussion on error rate performance, refer to Section 2.10 *Reed-Solomon (ECC) Module and Error Rate Performance*.

Figure 1: Block Diagram



A typical system block diagram is shown in the following figure.

Figure 2: Typical Applications Diagram



2.3 SIGNAL DESCRIPTIONS

Input Pins

DI[7:0] Data Input Bus. The input byte and ERASE are latched on the rising edge of the clock when both DSIN and RDYIN are active. If either DSIN or RDYIN are inactive, the DI and ERASE are ignored.

DSIN Data Input Strobe. Enables data from DI to be loaded into the chip. When RDYIN is active, DSIN being active on the rising edge of the clock loads the input data in the device. DSIN must be active for one clock edge only per each input byte. DSIN is ignored if RDYIN is inactive. Signal is active low.

DSON Data Output Strobe. This input strobe acknowledges to the chip that data available on the Output Bus, DO, has been received by the system. The device uses this strobe to increment its internal address counter to the next data location. DSON must be active for one clock edge only per each output byte. DSON is ignored if RDYON is inactive. Active low.

ERASE Erasure input flag for symbol currently on DI. Signal is active high. ERASE signal is used for marking all check Bytes as erasures (dummy check Bytes) during encode operation. It is also used to mark input symbols that contain errors during decoding. If not used, connect this signal to ground.

RSTN Reset. Input pin. When RSTN is active and DSIN and DSON are inactive, the device forces all internal control circuitry into a known state and initializes all data path elements. RSTN is active during Initialization Phase. In this phase, internal registers are programmed by using DI and DSIN. Signal is active low.

CLK Clock. System clock input. Refer to Section 4.4 *AC Electrical Characteristics* for clock requirements.

Output Pins

RDYIN Ready Input. Indicates the chip's ability to accept data input on DI. If active, DSIN is allowed to enable the loading of input data on DI. When inactive, DSIN is ignored. Signal is active low.

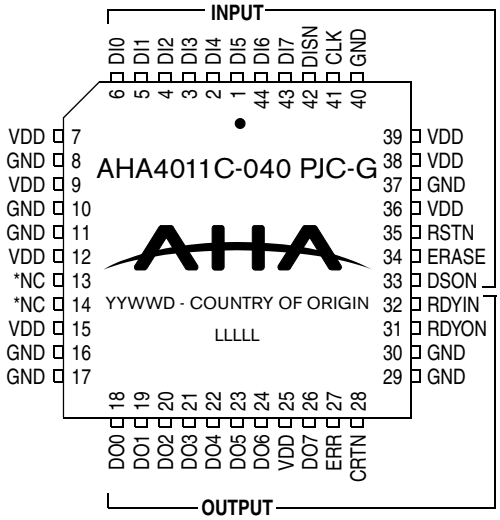
DO[7:0] Data Output. The output byte is available on this bus. The value of the output byte is undefined if RDYON is inactive. Requires an acknowledge strobe, DSON, at a rising edge of the clock to increment internal address counter and output the next location in the buffer. DO bus is always driven and is not tristated by the device.

RDYON Ready Output. This output pin indicates the chip's ability to generate output data. If active, DSON is allowed to increment the internal address counter for the next data byte. When inactive DSON is ignored and DO is undefined. Signal is active low.

CRTN Correctable. The output pin when active indicates the block did not exceed the error threshold programmed by P. Error threshold must be programmed with the same value as the number of check symbols R if erasures are not used. This signal is valid when the first message byte, X_{K-1}, of the block is available out of the chip. During all other times the signal is undefined. Signal is valid for at least one clock. Active low.

ERR Error. Output pin indicates the current value on DO[7:0] is a corrected byte. Active high.

2.4 PINOUT



*NC = No connect, reserved for future considerations.

YYWWD = DATE CODE; LLLLL = LOT NUMBER

2.5 DATA FLOW

The device is first initialized for various programmable parameters including: Erasure Multiplier, Error Threshold, Number of Check bytes, Number of Message bytes per block, Block Length and a Control byte. Following this six-byte initialization, the device may be used to encode, decode or pass-through multiple blocks of data. The device requires reinitialization when the parameters are changed or a reset is required.

The device processes data as "blocks" containing Message and Check Bytes. Order of input bytes must be first message byte X_{K-1} through last message byte X_0, followed by first check byte Y_{R-1} through last check byte Y_0. The device processes the block in this manner:

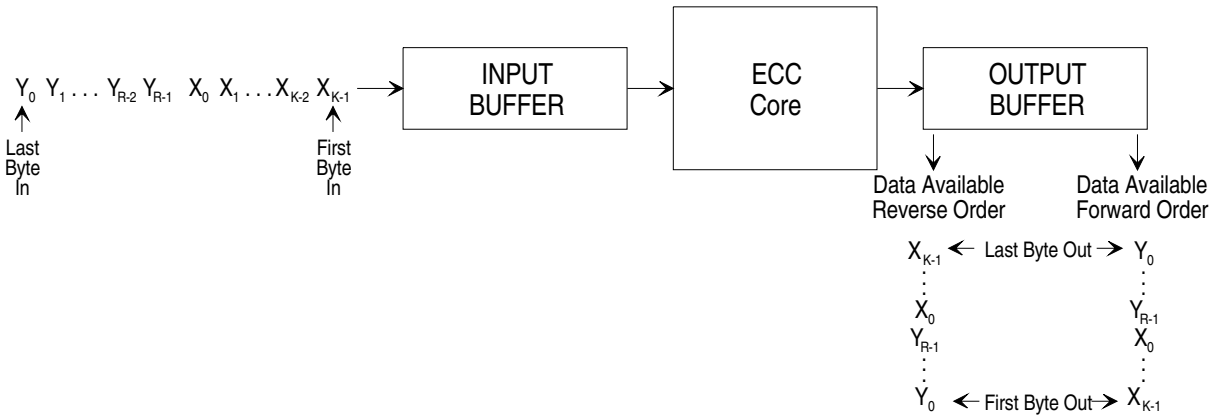
- a block is clocked into the Input Buffer;
- transferred into the ECC module;
- passed to the Output Buffer in the reverse order from what was received at the Input Port; and
- clocked out through the Output Port via the Output Buffer. Consecutive blocks may be input into the Input Buffer while the Output Buffer is being emptied.

Data is available through the Output Port in forward or reverse order. Forward order clocks out the block the same as input and reverse order clocks the check byte Y_0 through check bytes Y_{R-1} followed by message byte X_0 through message byte X_{K-1}.

2.5.1 SHORTENED BLOCKS

This device allows for shortened RS blocks, thus not requiring zero padding when decoding. During encoding, conversely, zero padding is not performed. When the device is programmed to decode a block of less than 255 Bytes, only the message Bytes followed by check Bytes are sent. Prepending with zero value Bytes to fill out the block to 255 Bytes is not required.

Figure 3: Data Input and Output Order



2.6 RESET AND INITIALIZATION SEQUENCE

Reset and initialization first requires pulling the RSTN low signal for at least two clocks while the DSIN and DSON signals are held inactive, i.e., high.

Following this sequence, the six internal registers, referred to as "Initialization Registers" are strobed by DSIN. These bytes are loaded in order of 1 through 6.

The RSTN must be active low for at least two clocks before the first initialization byte is strobed in and remain active for at least one clock after the final byte. RSTN must be high for at least two clocks before the first message byte can be strobed into the device. For a detailed timing diagram, see Figure 7*: Initialization and Reset Timing*.

The chip must be reset and initialized any time a reset is necessary.

Caveat: All six registers must be initialized correctly for proper operation of the chip. The device has no provisions for reading back Initialization Register settings. This sequence must be used if the device needs to be reset or any one register needs updating, i.e., all registers must be reinitialized for a change to any one register.

2.6.1 INITIALIZATION REGISTERS

BYTE 1, ERASURE MULTIPLIER:

[7:0] Multiplier value that must be programmed as shown in Appendix A. The table shows a value to be programmed corresponding to the block length selected.

BYTE 2, ERROR THRESHOLD:

[4:0] The threshold for determining uncorrectability of a data block, and the number of check bytes allocated for correction only purposes. When not using erasures, set to the same value as BYTE 3, CHECK BYTES. Minimum value of 0x02 sets the Threshold to 2 and 0x14 sets to the maximum, 20.

[6:5] Reserved. Set to 0.

[7] Not used. Don't care.

BYTE 3, CHECK BYTES:

[4:0] Number of check bytes in RS code, R. Minimum setting of 0x02 indicates two check bytes for R = 2 and 0x14 indicates the maximum of 20.

[6:5] Reserved. Set to 0.

[7] Not used. Don't care.

BYTE 4, MESSAGE BYTES:

[7:0] Number of message bytes in code, K. Minimum setting of 0x01 indicates 1 byte, setting to 0xFD indicates the maximum 253 message bytes.

BYTE 5, BLOCK LENGTH:

[7:0] Number of bytes in block, N. Setting to 0x03 indicates 3 bytes, setting to 0xFF indicates 255 bytes.

BYTE 6, CONTROL BYTE:

[0] RES Reserved. Set to 0.

[1] NOPAR Parity Symbol Control
- 0 Check bytes are output following the message bytes.
- 1 Check bytes are not output following the message bytes. Correction will be done regardless depending upon the bit 4, RAW, setting.

[2] CRCTS Correction Control
- 0 Outputs correction vectors; to obtain corrected data, externally XOR the correction vector with the corresponding message or check byte.
- 1 Outputs corrected data

[3] FOR Forward Order Control
- 0 Outputs the block in reverse order
- 1 Outputs the block in forward order

[4] RAW Raw Data
- 0 Outputs corrections or corrected data per the CRCTS bit
- 1 Outputs uncorrected, raw input data or 0's depending upon the CRCTS bit setting (See table below). NOPAR bit and CHECK BYTE register settings are ignored.

[5] ERC Erasure Rejection Control. This bit is only used by the device when the Erasures exceed the ERROR THRESHOLD or R settings. This bit is ignored when the Erasures are less than or equal to ERROR THRESHOLD or R.
- 0 If Erasures are greater than the ERROR THRESHOLD or R then erasures are discarded and full correction is performed. The block is flagged uncorrectable and the output CRTN will be high during the last output byte of the block.
- 1 If Erasures are greater than ERROR THRESHOLD or R then erasures are discarded and full correction is performed. The output CRTN will be high only when the block is uncorrectable.

[7:6] Reserved, Set to 0.

RAW	CRCTS	Output
0	0	Correction vectors
0	1	Corrected data
1	0	Zero
1	1	Uncorrected raw input data

2.7 ENCODE, DECODE OR PASS-THROUGH OPERATIONS

The device performs three functions: encoding, decoding and pass-through. As an encoder the device outputs the message block followed by "corrected" check bytes. As a decoder, the device outputs the corrected message bytes or correction vectors with or without check bytes following the message. In pass-through operation, the device passes the input data as it is received. In all three operations, the input block flows through the Input Buffer into the ECC module and out of the Output Buffer. Latencies for all three operations are the same.

The device is initialized for the three operations as shown in the table below.

Table 1: Initialization Register Settings for Encode, Decode and Pass-Through Operations

INITIALIZATION REGISTER	*BIT(S)*	*ENCODE*	*DECODE*	*PASS-THROUGH*
ERASURE MULTIPLIER	[7:0]	Appendix A value	Appendix A value	Appendix A value
ERROR THRESHOLD	[7:0]	Set to R	R or less	R
CHECK BYTES	[7:0]	Set to R	R	R
MESSAGE BYTES	[7:0]	Set to the Number of Message Bytes in block, K	K	K
BLOCK LENGTH	[7:0]	Set to the total of Message and Check bytes, N	N	N
CONTROL BYTE	0 (RESV)	0	0	0
	1 (NOPAR)	0	System specific	0
	2 (CRCTS)	1	System specific	1
	3 (FOR)	System specific	System specific	System specific
	4 (RAW)	0	0	1
	5 (ERC)	0	0	0
	[7:6] Reserved	0	0	0

As an encoder, the device is used with the Erasures feature enabled in the following sequence. (Asserting the ERASE signal high enables the Erasure feature.)

1) After initialization, the device receives the message data followed by "dummy" check bytes. "Dummy" check bytes are clocked into the device with the ERASE signal asserted. The number of "dummy" check bytes must equal R.
2) The ECC core processes the block by "correcting" the check bytes and feeding the codeword into the Output Buffer in reverse order.
3) The block is then made available on the output bus, DO. The state of the output RDYON determines the availability of data. ERR signal is asserted while the "corrected check bytes" are output on the output bus, DO. CRTN is asserted low during the last byte out of the chip indicating that the previous block did not exceed the error threshold.

As a decoder, the device works similar to the encode operation in the following sequence.

1) Following initialization, the system clocks the message data and the check bytes into the Input Buffer. ERASE signal may be asserted as desired by the system. State of the output signal, RDYIN determines the chip's ability to accept data input on the DI bus.
2) The ECC Core processes the block by performing necessary corrections, and feeds the codeword into the Output Buffer in reverse order.
3) The data is available on the output port. The state of the output signal, RDYON determines the availability of valid data. An output byte which has been corrected is indicated by the device asserting ERR. CRTN may be high or low depending upon the THRESHOLD Register and ERC bit programmed and the errors encountered.

In pass-through operation, data flows through the device similar to the encode and decode operations. During initialization the device is programmed as shown above. Check Bytes are programmed in the range of 0x02 to 0x14.5. The Block length here is the sum of Message Bytes and Check Bytes like encode and decode modes of operation even though the device passes through the block of data unchanged.

1) Following initialization, the system clocks the codeword into the Input Buffer.
2) The codeword is processed by the ECC module and passed on to the Output Buffer without correction.
3) The uncorrected codeword is available at the output port. State of the RDYON determines the availability of valid data. The ERASE input is ignored during the Input phase and ERR and CRTN outputs are not valid.

Caveat: The device has no provisions for indicating the start and/or end of message or check bytes. It is the system designers responsibility to keep track of message and check bytes transitions, if required.

2.8 BUFFERS

The Input Port contains a single-ported 367x9 buffer. The Output Port contains a single-ported 256x9 buffer. These buffers store input and output data during the correction process and help maintain the desired system data rate. A Reset operation as described in the Initialization Sequence section clears the buffers.

The use of internal buffers is restricted per the rules defined in Section 2.9 *Data Rates and Latencies*. These rules define the limitations of using the buffers to temporarily store more than one block. It is highly recommended that the system designer clearly understand these rules prior to designing the system.

The Input Buffer receives input data on the DI bus when the ECC module is in the calculation or in data-out phases at the desired system rate. The ability of the Input Buffer to accept data is indicated by RDYIN.

The Output Buffer accepts corrected data from the ECC during the data-out phase. RDYON is asserted low when the Output Buffer is able to output data.

Data flow through the device may occur in burst or continuous rates. The number of clocks per byte used to input or output determines burst or continuous operating conditions. Figure 4 shows the two operations.

Burst operation permits data to be clocked in and out of the device at the maximum rate, i.e., 1 clock per byte. In burst operation, consecutive data blocks are clocked into the device following a processing latency period. Data is input into the Input Buffer and processed through the ECC core. After a processing latency period the entire block of data is transferred to the Output Buffer. While the Output Buffer is being emptied, the Input Buffer is simultaneously filled with the following block at the maximum rate. Input and output rates are controlled by the clock speed and clocks/byte.

Continuous operation requires a minimum of 4 clocks/byte depending upon the block size. Maximum data transfer rates for continuous rate vary accordingly. Blocks may be processed continuously through the device. If the chip is operated with continuous data streams, the RDYIN and RDYON pins will always be active (after the initial latency). Therefore, they need not be used.

Caveat: System designer should be aware that data is put into the Output Buffer in reverse order. Therefore, RDYON may become inactive between blocks in forward order if data is output faster than Output Buffer is filled.

2.9 DATA RATES AND LATENCIES

This section describes data rates and processing latencies for burst and continuous operations. Processing latencies are the same in encode, decode or pass-through operations. The number of clocks used to clock in and out of the device determines the operation. The input and output rates need not be the same. No registers are required to program the device for either operation.

Continuous block flow is achieved by using the appropriate number of clocks per byte and block length. Alternatively, data flow into and out of the device is controlled using control signals, DSIN and DSON.

2.9.1 BURST OPERATION

Maximum processing latency, in forward order, expressed in number of clocks, for burst operation is determined by: $N \times C_i + R + 60 + N$

Definitions:

C_i = input clock rate per byte. If $C_i = 1$, use a value for C_i of 2 in the latency equation

N = block length

R = number of check bytes

Processing Latency = Delay from first input byte to first output byte

In reverse order, processing latency is approximately N clocks less than above.

For a 40 MHz system using 1 clock per byte, latencies and data rates for forward order output are shown in the table for burst operation. Input and Output Burst Rates in all cases will be 40 MBytes/sec. Note: Other frequency operations may be derived similarly.

Output Buffer may be used to hold data from one block while the Input Buffer is being filled with the following block. Two rules listed in the caveats are required to accomplish this. These are illustrated in Figure 4.

Caveats:

1. Output of block i must start coincident with or before the input of block i + 1.
2. Output of block i must be complete: Processing Latency – N – 8 clocks after the start of block i + l on the input.

Table 2: Burst Operation Using 40 MHz Clock and 1 Clock/Byte, Forward Order Output

	CHECK BYTES 'R' = 20			*CHECK BYTES 'R' = 2*		
BLOCK LENGTHS 'N'	***MAXIMUM LATENCY (# of clocks)***	***MAXIMUM LATENCY (μsecs)***	***AVERAGE RATE (MBytes/sec)***	***MAXIMUM LATENCY (# of clocks)***	***MAXIMUM LATENCY (μsecs)***	***AVERAGE RATE (MBytes/sec)***
25	155	3.88	6.45	137	3.43	7.30
50	230	5.75	8.70	212	5.30	9.43
100	380	9.50	10.50	362	9.05	11.00
150	530	13.30	11.30	512	12.80	11.70
200	680	17.00	11.80	662	16.60	12.10
255	845	21.10	12.10	827	20.70	12.30

$$\text{Average Rate} = \frac{N}{\text{Maximum Latency } (\mu\text{sec})}$$

2.9.2 CONTINUOUS OPERATION

Multiple blocks of data may be processed through the device continuously as shown in Figure 4. Consecutive blocks are input into the device at the rate of C_i clocks/byte. The output data stream may or may not be continuous depending on whether parity is being output (controlled by NOPAR) and the choice of C_o. Continuous operation is described by several equations. The following terms are used in these equations:

C_i - Input clock rate per byte: $C_i \geq 4$ for continuous operation

C_o - Output clock rate per byte: $C_o \geq 2$

C_m - Minimum of C_i and C_o: If $C_i < C_o$ then $C_m = C_i$ else $C_m = C_o$

N - Reed-Solomon block length

K - Reed-Solomon message length

R - Reed-Solomon parity length (R = N – K)

L - Output data length: If parity is being output from the chip (NOPAR = 0), L = N; else if the parity is not being output (NOPAR = 1) L = K

A. Conditions for Continuous Operation

The allowable input and output data rates are related to the Reed-Solomon block length by the following two inequalities. C_i, C_o, N and K must be chosen so that these equations are satisfied.

Equation 1:

$$\frac{R + 60 + \dfrac{N \times C_m}{C_m - 1}}{C_i} + N \leq 367$$

Equation 2:

$$(N-1) \times C_i \geq R + 48 + \frac{N \times C_i}{C_i - 1} + \frac{N \times C_m}{C_m - 1}$$

B. Processing Latency

Processing latency is the time from the beginning of a block on the input to the block being ready for output. Maximum processing latency, expressed in number of clocks, for continuous operation is:

Equation 3:

$$\text{Latency} = (N-1) \times C_i + 60 + R + \frac{N \times C_m}{C_m - 1}$$

C. Start and End of Output

Similar to the burst operation, Output Buffer may be used to temporarily "hold" data from one block while the Input Buffer is being filled. However, these conditions must be satisfied: the output of a data block must start after the latency equation (Equation 3) is satisfied, but before the maximum delay is reached. The maximum delay is:

Data of one block must be fully emptied $L \times C_o$ clocks after the start of empty process.

All of the conditions on the maximum delay given in Equation 4 must be satisfied. If any are not, the output data stream will begin to inhibit ECC processing. Eventually this will cause the input buffer to over fill and RDYIN to become inactive.

Equation 4:

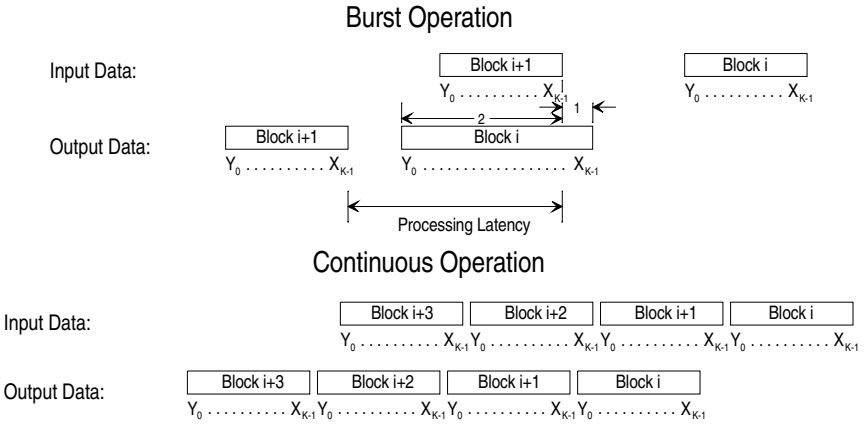
$$\text{maximum\_delay} = 3 \times N \times C_i - L \times C_o - \frac{N \times C_i}{C_i - 1}$$

$$\frac{\text{if maximum\_delay}}{C_i} \geq 367, \text{ then maximum\_delay} = 367 \times C_i$$

$$\frac{\text{if maximum\_delay}}{C_i} > 2 \times N, \text{ then maximum\_delay} = 2 \times N \times C_i$$

Figure 4: Burst and Continuous Operations

(Note: Blocks are shown from right to left as they are input into and output from the chip in Forward Order. Block i is the first input block, block i + 1 is second input block. X_{K-1} is the first input message byte of a block. Y_o is the last input check symbol of a block. Notes 1 and 2 in burst operation are described in Section 2.9.1 Burst Operation - Caveats.)



For a 40 MHz system using the required clocks per byte, maximum latencies and data rates for forward order output are shown in the table for continuous operation. Input and Output rates are assumed the same in this table. Note: Other frequency operations are also possible.

Table 3: Continuous Operation Using 40 MHz Clock and Specified Clocks/Byte, Forward Output Order

	CHECK BYTES 'R' = 20			CHECK BYTES 'R' = 2		
BLOCK LENGTHS 'N'	MINIMUM REQUIRED (clocks/byte)	MAXIMUM DATA RATE (MBytes/sec)	MAXIMUM LATENCY (μsecs)	MINIMUM REQUIRED (clocks/byte)	MAXIMUM DATA RATE (MBytes/sec)	MAXIMUM LATENCY (μsecs)
25	6	6.67	6.35	5	8	5.33
50	5	8	9.69	5	8	9.24
100	4	10	15.23	4	10	14.78
150	4	10	21.90	4	10	21.45
200	4	10	28.57	4	10	28.12
225	4	10	31.90	4	10	31.45
255	4	10	35.90	4	10	35.45

For Intelsat IESS-308, Rev F, Inner FEC Rates, use Table 4 for a system with 40 MHz clock.

Note: Other frequency operations are also possible.

Table 4: Continuous Operation for IESS-308 Codes Using 40 MHz Clock and Specified Clocks/Byte, Forward Output Order

BLOCK LENGTHS 'N'	MESSAGE LENGTH 'K'	ERROR CAPABILITY 't'	MINIMUM REQUIRED (clocks/byte)	MAXIMUM DATA RATE (MBytes/sec)	MAXIMUM LATENCY (# of clocks)	MAXIMUM LATENCY (μsecs)
126	112	7	4	10	742	18.55
194	178	8	4	10	1107	27.67
208	192	8	4	10	1181	29.53
219	201	9	4	10	1242	31.05
225	205	10	4	10	1276	31.90

Appendix B shows a spreadsheet table of block lengths vs. latencies for a 40 MHz clock system.

2.10 REED-SOLOMON (ECC) MODULE AND ERROR RATE PERFORMANCE

The module implements a full error correcting Reed-Solomon (RS) decoder whose function is to perform the necessary corrections on the input blocks. The code used by the decoder is capable of generating corrections for up to 10 (t = 10) byte-errors in an RS block over the block length between R + 1 to 255 bytes. The number of message bytes in an RS block, K, is equal to the RS block length minus R (K = N – R). The RS code implemented uses the primitive polynomial

$$P(x) = x^8 + x^7 + x^2 + x + 1$$

to generate GF(256). The generator polynomial for the code is:

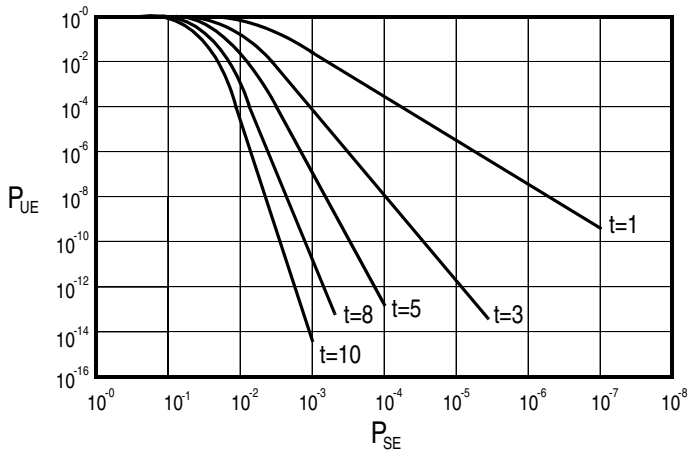
$$G(x) = \prod_{i=120}^{119+R} (x - \alpha^i)$$

An RS block consists of message and redundancy bytes. The number of message bytes in the block, K, is programmable during initialization. The number of check bytes is R and can be programmed during initialization to be 2 through 20 in increments of 1.

The ECC Module may be programmed to output corrections or corrected data. If "corrections" is selected, to obtain corrected data, externally XOR the output correction vector with the corresponding message or check byte. For example, if "corrections" is selected for a block of 200 bytes with errors in locations 100, 123, 153, 176 and 199; output block will be 0's for all locations except for those positions. The bytes output at these positions are referred to as correction vectors and are XOR'd externally with the message bytes to obtain the correct value. If the output of the AHA4011C is programmed to output corrected data, the correction vector is applied internally and the corrected data is output.

The Symbol Error Rate Performance of the Reed Solomon code used is shown in Figure 5.

Figure 5: Symbol (Byte) Error Rate Performance Curves for Codeword Length = 255 Bytes



The most common measures of performance for Reed-Solomon code are P_{UE}, P_{SE}, and C_{BER}. P_{SE} is the probability of symbol errors and is the ratio of the number of received symbol errors to the total number of received symbols. In the AHA4011C device the symbol length, m, is equal to 8 bits. P_{UE} is the probability of an uncorrectable error and is the ratio of the number of uncorrectable code blocks to the total number of received code blocks. An uncorrectable error occurs when more than t received symbols are in error. C_{BER} is the Corrected Bit Error Rate. The C_{BER} is the reciprocal of expected number of correct bits between errors.

If input noise is random, $C_{BER} = \frac{P_{UE}}{m \times N}$.

If $P_{SE} = 8 \times 10^{-4}$ with $t = 5$, $P_{UE} = 10^{-7}$ and $C_{BER} = \frac{10^{-7}}{8 \times 255} = 4.9 \times 10^{-11}$.

The figure shows probability of symbol error and uncorrectable error for block size (N) of 255. It shows the ability of various levels of Reed-Solomon error correction to restore the integrity of the corrupted data. For example, using 255 byte blocks, if 1 out of 1000 of the received bytes have one or more bit errors, RS correction with t = 5 will restore the data to 1 error in 2 million blocks (510 million bytes).

For a detailed discussion on error rate performance of Reed-Solomon code, refer to AHA Application Note, *Primer: Reed-Solomon Error Correction Codes (ECC)*, (ANRS01).

2.11 DETERMINING DECODER PERFORMANCE BOUNDARIES

AHA4011C supports a programmable feature that allows a system designer to determine the channel performance. This programmable feature, referred to as error threshold, P, sets a number of errors to be allowed by the chip prior to flagging the block uncorrectable. Erasure Rejection Control bit of the Control Byte register determines the condition of CRTN output pin.

P and R are both independently selectable by the user during the Initialization Control Sequence. The various configurations of P and R are described as follows:

P > R This is not a sensible choice since this implies that more check bytes are allocated for (correction-only) purposes than there are total check bytes (for both correction and detection). The device will work as if P was set equal to R.

P = R This configuration maximizes the ability to correct errors, particularly if R itself has been chosen to be its maximum value of 20. This is the usual choice. This situation causes the CRTN output to flag a message block as uncorrectable at an error level exceeding that of which the device is capable.

P < R This increases the level of error detection capability. This situation causes the CRTN output to flag a message block as uncorrectable at an error level below that of which the device is capable. This mode only works with erasures.

Caveat: Output block may be corrupted if a block exceeds the correction ability of the ECC module.

2.12 ERASURES

The chip is capable of utilizing erasure information. R erasures may be corrected in any block assuming there are no unmarked errors.

The correction capability is: $E + 2e \leq R$

Where E = number of erasures (marked errors)
e = number of unmarked errors
R = number of check symbols

If there are more than P or R erasures the erasure information is discarded, and full error correction is attempted. The chip can be programmed to either call such a block uncorrectable or not. If programmed not to call the block uncorrectable (ERC bit set to 1), the ECC will utilize the full error correction capability to decide if the block is correctable.

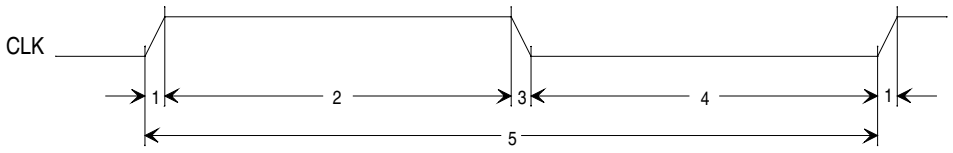
3.0 OPERATIONAL DESCRIPTION

This section describes the relationship of associated signals for various functions of the chip.

3.1 CLOCK

The clock input to the chip must meet the timing requirements shown in Figure 6. The chip is entirely static thus allowing the clock to stop in either the active or inactive state for an indefinite period without loss of stored information.

Figure 6: CLK Characteristics



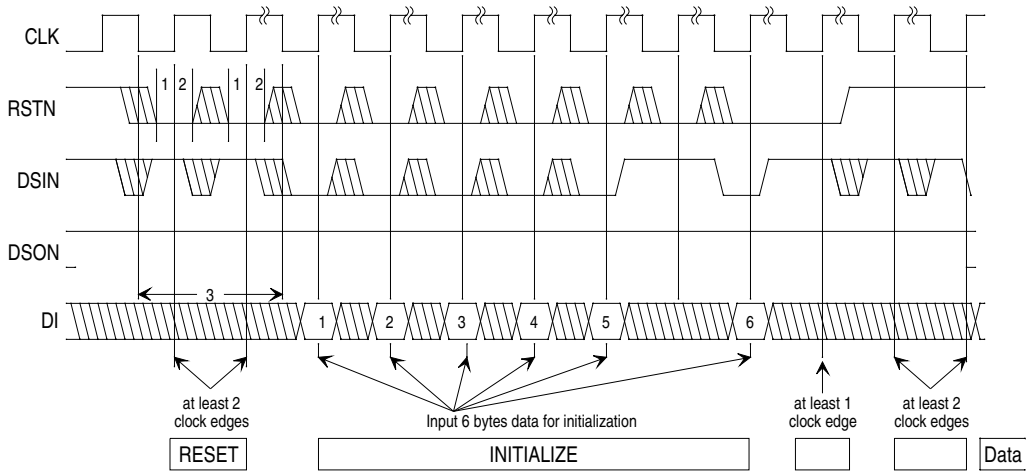
NUMBER	DESCRIPTION	MINIMUM	MAXIMUM	UNITS
1	CLK rise time		5	nsec
2	CLK high time	8		nsec
3	CLK fall time		5	nsec
4	CLK low time	8		nsec
5	CLK period	25		nsec

All timing diagrams in this specification use the clock at the CLK pin as the reference point.

3.2 INITIALIZATION

This section describes the Reset and Initialization Sequence timing. For a detailed discussion on these sequences, refer to Section 2.6 *Reset and Initialization Sequence*.

Figure 7: Initialization and Reset Timing



NUMBER	DESCRIPTION	MINIMUM	MAXIMUM	UNITS
1	RSTN and DSIN setup time	10		nsec
2	RSTN and DSIN hold time	0		nsec
3	RSTN and DSIN assertion	2		Clock edges

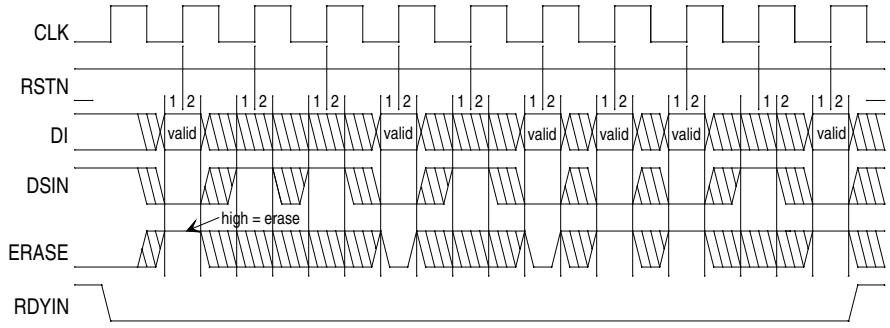
Initialization bytes are strobed into the device while RSTN and DSIN are low during rising edges of CLK. The RSTN must be active low for at least two clocks before the first initialization byte is strobed in and remain active for at least one clock after the final byte. Initialization register data may be strobed at a minimum of 1 clock per byte. After power-on the initializing registers' contents are undefined.

For a detailed description of the Initialization Registers, refer to Section 2.6 *Reset and Initialization Sequence*.

3.3 DATA INPUT

The chip latches the input data on the DI pins on the rising edge of the CLK when DSIN and RDYIN are both active. The two figures below show the timing diagrams for buffer Ready and buffer Not Ready conditions.

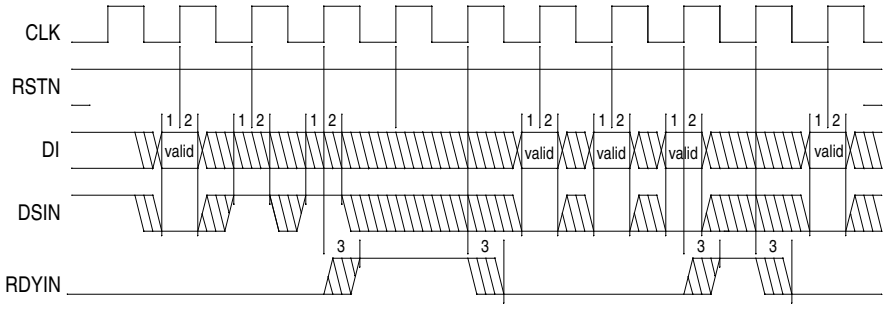
Figure 8: Data Input - Buffer Always Ready



If RSTN is low during write, message bytes are treated as being part of the initialization sequence. If RSTN is high, the data is treated as being part of RS block. In the example above ERASE is asserted high in four sample clocks.

NUMBER	DESCRIPTION	MINIMUM	MAXIMUM	UNITS
1	DI, ERASE and DSIN setup time	10		nsec
2	DI, ERASE and DSIN hold time	0		nsec

Figure 9: Data Input - Buffer Not Ready



NUMBER	DESCRIPTION	MINIMUM	MAXIMUM	UNITS
1	DI, ERASE and DSIN setup time	10		nsec
2	DI, ERASE and DSIN hold time	0		nsec
3	RDYIN output delay		15	nsec

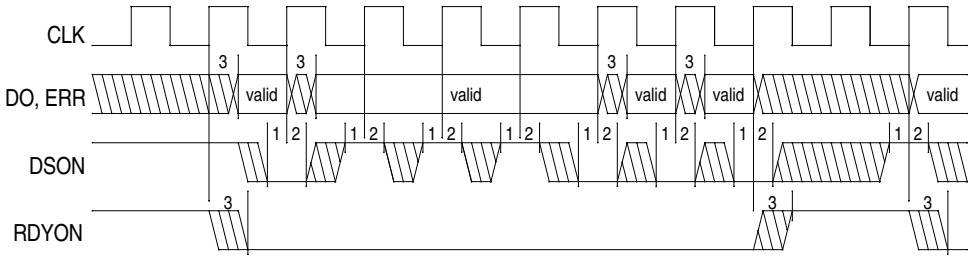
Any input data clocked while RDYIN is inactive are ignored. This is shown in Figure 9.

3.4 DATA OUTPUT

The DO pins are driven from a register clocked on the rising edge of CLK.

Valid data on the DO pins is indicated by RDYON being active. When RDYON is inactive, data on the DO pins is undefined, and DSON is ignored. The DSON signal acknowledges receiving the data and is used by the device to internally increment the address counter and output the next location in the buffer. This data output timing is shown in Figure 10.

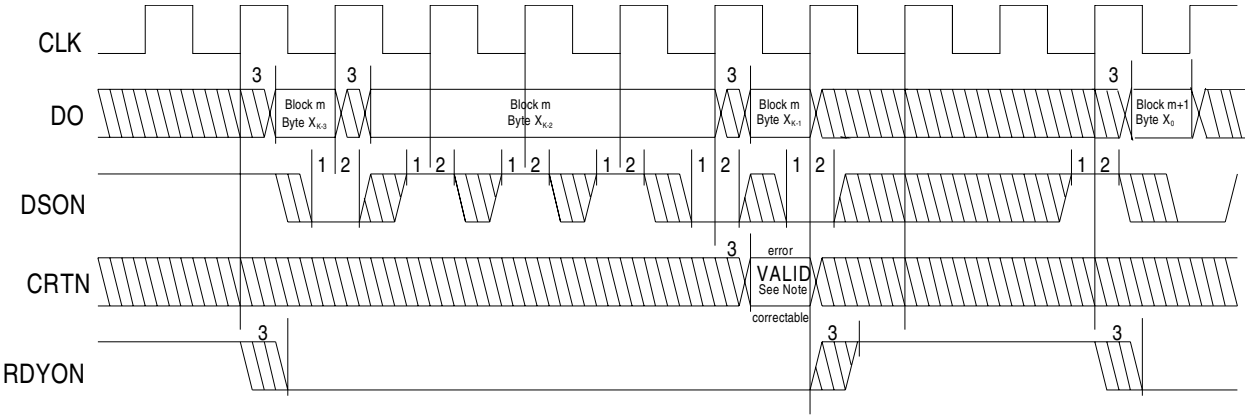
Figure 10: Data Output



NUMBER	DESCRIPTION	MINIMUM	MAXIMUM	UNITS
1	DSON setup time	10		nsec
2	DSON hold time	2		nsec
3	DO and RDYON output delay		15	nsec

CRTN is valid for an RS block when the first message byte, X_{K-1}, is strobed out of the chip. Figure 11 shows Reverse Order output. In this operation, CRTN is valid on the last byte of the block from the Output Buffer. In this example only message bytes are output, no check bytes.

Figure 11: CRTN Timing - Reverse Order Output



Note: CRTN is active (low) if RS block m is correctable. If the number of errors detected in block m exceeds the error threshold, P, CRTN is inactive (high).

NUMBER	DESCRIPTION	MINIMUM	MAXIMUM	UNITS
1	DSON setup time	10		nsec
2	DSON hold time	2		nsec
3	DO and RDYON output delay		15	nsec

4.0 SIGNAL SPECIFICATIONS

4.1 INPUT SPECIFICATIONS

PIN NUMBER	SIGNAL NAME	SELF LOAD (maximum in pF)	TSETUP (min in nsec)	THOLD (min in nsec)	STROBE
43	DI[7]	10	10	0	CLK
44	DI[6]	10	10	0	CLK
1	DI[5]	10	10	0	CLK
2	DI[4]	10	10	0	CLK
3	DI[3]	10	10	0	CLK
4	DI[2]	10	10	0	CLK
5	DI[1]	10	10	0	CLK
6	DI[0]	10	10	0	CLK
42	DSIN	10	10	0	CLK
33	DSON	10	10	2	CLK
35	RSTN	10	10	0	CLK
41	CLK	10	N/A	N/A	N/A
34	ERASE	10	10	0	CLK

N/A = Not Applicable

(Refer to Section 4.5 *DC Electrical Characteristics* for pad specifications)

4.2 OUTPUT SPECIFICATIONS

PIN NUMBER	SIGNAL NAME	LOAD CAP (maximum in pF)	TDEL (min in nsec)	TDEL (max in nsec)	STROBE REF
26	DO[7]	60	0	15	CLK
24	DO[6]	60	0	15	CLK
23	DO[5]	60	0	15	CLK
22	DO[4]	60	0	15	CLK
21	DO[3]	60	0	15	CLK
20	DO[2]	60	0	15	CLK
19	DO[1]	60	0	15	CLK
18	DO[0]	60	0	15	CLK
31	RDYON	60	0	15	CLK
32	RDYIN	60	0	15	CLK
28	CRTN	60	0	15	CLK
27	ERR	60	0	15	CLK

(Refer to Section 4.5 *DC Electrical Characteristics* for pad specifications)

4.3 POWER & GROUND PINS

PIN NUMBER	SIGNAL NAME
8, 10, 11, 16, 17, 29, 30, 37, 40	GND
7, 9, 12, 15, 25, 36, 38, 39	VDD

4.4 AC ELECTRICAL CHARACTERISTICS

CLOCK RATE					
Symbol	**Characteristic**	**Min**	**Max**	**Units**	**Test Conditions**
Fclock	Clock frequency	0	40	MHz	
Tlow	Clock low time	8		nsec	
Thigh	Clock high time	8		nsec	
Trise	Clock rise time		5	nsec	Vil to Vih
Tfall	Clock fall time		5	nsec	Vil to Vih

INPUTS					
Symbol	**Characteristic**	**Min**	**Max**	**Units**	**Test Conditions**
Tsetup	Input setup time	10		nsec	See *Note 1*
Thold	Input hold time	0		nsec	See *Notes 1 and 2*
Notes: *1) Setup and hold times measured from a valid high [2.4V] on the clock input pin.* *2) DSON has a 2 nsec hold time.*					

OUTPUTS					
Symbol	**Characteristic**	**Min**	**Max**	**Units**	**Test Conditions**
Tout	Output delay	0	15	nsec	See *Note*
Note: Output delay measured from valid high [2.4V] on the clock input pad. The output loads for the AC test are given in Section 4.2 Output Specifications.					

4.5 DC ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM STRESS RATINGS					
Symbol	**Characteristic**	**Min**	**Max**	**Units**	**Test Conditions**
Tstg	Storage temperature	-55	150	deg C	
Vdd	Supply voltage	-0.5	6.0	V	
Vin	Input voltage	Vss-0.5	Vdd+0.5	V	
Package: 44-pin PLCC (JEDEC Standard)					

OPERATING CONDITIONS					
Symbol	**Characteristic**	**Min**	**Max**	**Units**	**Test Conditions**
Vdd	Supply voltage	4.75	5.25	V	
Idd	Supply current		1.0	mA	Static; Clock stopped externally
Idd	Supply current		135	mA	Vdd=5V
Ta	Operating temperature (commercial part marking)	0	70	deg C	
P	Power		0.71	W	Vdd=5V

INPUTS					
Symbol	**Characteristic**	**Min**	**Max**	**Units**	**Test Conditions**
Vih	Input high voltage	2.0	Vdd	V	
Vil	Input low voltage	Vss	0.8	V	40 MHz
Iil	Input leakage	-10	10	µA	0<Vin<Vdd
Cin	Capacitance		10	pF	Not 100% tested

OUTPUTS					
Symbol	**Characteristic**	**Min**	**Max**	**Units**	**Test Conditions**
Voh	Output high voltage	2.4	Vdd	V	Ioh=8mA
Vol	Output low voltage	Vss	0.4	V	Iol=8mA
Ioh	Output high current	-8		mA	Voh=2.4V
Iol	Output low current		8	mA	Vol=0.4V

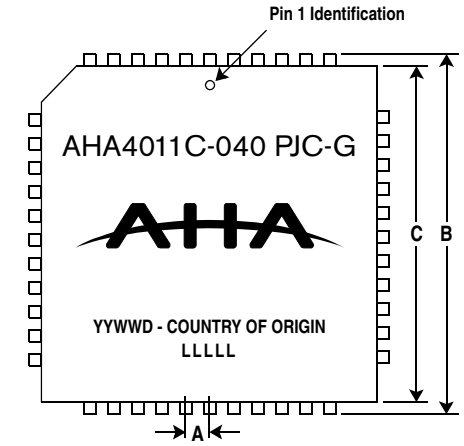
5.0 PACKAGING

PLCC Dimensions

Inches
(Millimeters)

A	*B* *min/max*	*C* *min/max*	*D* *min/max*	*E* *min*	*F* ±	*G* ±
.050 (1.27)	.685/.695 (17.40/17.65)	.650/.656 (16.51/16.66)	.165/.180 (4.19/4.57)	.020 (0.51)	.002 (0.051)	.0035 (0.089)

Packaging



F = Lead Planarity G = Lead Skew

Note: YYWWD = Date Code
LLLLL = Lot Number

Complete Package Drawing Available Upon Request.

6.0 ORDERING INFORMATION

6.1 AVAILABLE PARTS

PART NUMBER	*DESCRIPTION*
AHA4011C-040 PJC	Reed-Solomon ECC Integrated Circuit
AHA4011C-040 PJC-G	Reed-Solomon ECC Integrated Circuit - RoHS Compliant

6.2 PART NUMBERING

AHA	*4011*	*C-*	*040*	*P*	*J*	*C*	*-G*
Manufacturer	Device Number	Revision Level	Speed Designation	Package Material	Package Type	Temperature Specification	RoHS Compliant

Device Number:

4011

Revision Letter

C

Package Material Codes:

P - Plastic

Package Type Codes:

J - Leaded Chip Carrier

Test Specifications:

C - Commercial 0°C to +70°C

Environmental Compliance:

G - RoHS Compliant

7.0ABOUT AHA

Comtech AHA Corporation (AHA) develops and markets superior integrated circuits, boards, and intellectual property core technology for communications systems architects worldwide. AHA has been setting the standard in Forward Error Correction and Lossless Data Compression technology for many years and provides flexible, cost-effective solutions for today's growing bandwidth and reliability challenges. Comtech AHA Corporation is a wholly owned subsidiary of Comtech Telecommuncations Corp. (NASDAQ: CMTL). For more information, visit www.aha.com.

APPENDIX A

Table of Elements

BLOCK SIZE 'N'	HEX VALUE	BLOCK SIZE 'N'	HEX VALUE	BLOCK SIZE 'N'	HEX VALUE	BLOCK SIZE 'N'	HEX VALUE
1	1	2	2	3	4	4	8
5	10	6	20	7	40	8	80
9	87	10	89	11	95	12	ad
13	dd	14	3d	15	7a	16	f4
17	6f	18	de	19	3b	20	76
21	ec	22	5f	23	be	24	fb
25	71	26	e2	27	43	28	86
29	8b	30	91	31	a5	32	cd
33	1d	34	3a	35	74	36	e8
37	57	38	ae	39	db	40	31
41	62	42	c4	43	f	44	1e
45	3c	46	78	47	f0	48	67
49	ce	50	1b	51	36	52	6c
53	d8	54	37	55	6e	56	dc
57	3f	58	7e	59	fc	60	7f
61	fe	62	7b	63	f6	64	6b
65	d6	66	2b	67	56	68	ac
69	df	70	39	71	72	72	e4
73	4f	74	9e	75	bb	76	f1
77	65	78	ca	79	13	80	26
81	4c	82	98	83	b7	84	e9
85	55	86	aa	87	d3	88	21
89	42	90	84	91	8f	92	99
93	b5	94	ed	95	5d	96	ba
97	f3	98	61	99	c2	100	3
101	6	102	c	103	18	104	30
105	60	106	c0	107	7	108	e
109	1c	110	38	111	70	112	e0
113	47	114	8e	115	9b	116	b1
117	e5	118	4d	119	9a	120	b3
121	e1	122	45	123	8a	124	93
125	a1	126	c5	127	d	128	1a
129	34	130	68	131	d0	132	27
133	4e	134	9c	135	bf	136	f9
137	75	138	ea	139	53	140	a6
141	cb	142	11	143	22	144	44
145	88	146	97	147	a9	148	d5
149	2d	150	5a	151	b4	152	ef
153	59	154	b2	155	e3	156	41
157	82	158	83	159	81	160	85
161	8d	162	9d	163	bd	164	fd
165	7d	166	fa	167	73	168	e6
169	4b	170	96	171	ab	172	d1
173	25	174	4a	175	94	176	af
177	d9	178	35	179	6a	180	d4
181	2f	182	5e	183	bc	184	ff
185	79	186	f2	187	63	188	c6

BLOCK SIZE 'N'	HEX VALUE	BLOCK SIZE 'N'	HEX VALUE	BLOCK SIZE 'N'	HEX VALUE	BLOCK SIZE 'N'	HEX VALUE
189	b	190	16	191	2c	192	58
193	b0	194	e7	195	49	196	92
197	a3	198	c1	199	5	200	a
201	14	202	28	203	50	204	a0
205	c7	206	9	207	12	208	24
209	48	210	90	211	a7	212	c9
213	15	214	2a	215	54	216	a8
217	d7	218	29	219	52	220	a4
221	cf	222	19	223	32	224	64
225	c8	226	17	227	2e	228	5c
229	b8	230	f7	231	69	232	d2
233	23	234	46	235	8c	236	9f
237	b9	238	f5	239	6d	240	da
241	33	242	66	243	cc	244	1f
245	3e	246	7c	247	f8	248	77
249	ee	250	5b	251	b6	252	eb
253	51	254	a2	255	c3		

For example, for a block size of 205, the value to be programed in Byte 1 of the Initialization Register is 0xc7.

/*This is a C program to generate Table of Elements. Pass a value of block length, N in decimal to this, and obtain the Element value in hex.*/

```
int alpha(n)
int n;
{
int i,b,c;
    c=01;
    for (i=1;i<n;i++) {
        b=c<<1;
        if (b>0377)
            b=b^0607;
        c=b;
        }
    return c;
}

main()
{
    int i;
    printf("Enter N--> ");
    scanf("%d",&i);
    if(i<1 || i>255)
        printf("1<=N<=255");
    else
        printf("\nN = %d\tALPHA = %2x\n\n", i, alpha(i));
}
```

APPENDIX B

AHA4011C Data Rate Calculations in Continuous Operation

Assumptions and Equations:

1) 40 MHz Clock is used.
2) Input Rate (C_i) = Output Rate (C_o)
3) Latency = $C_i \times (N-1) + (R+60) + N \times \frac{C_i}{C_i - 1}$
4) Data Rate = 40 MHz/C_i clocks/byte
5) GOOD or BAD based on inequality equation:

$$\frac{R + 60 + N \times \frac{C_m}{C_m - 1}}{C_i} + N \leq 367 \tag{5}$$

6) GOOD or BAD based on inequality equation:

$$(N-1) \times C_i \geq R + 48 + N \times \frac{C_i}{C_i - 1} + N \times \frac{C_m}{C_m - 1} \tag{6}$$

7) Check symbols are input into and output from the chip along with message symbols.

Note: The following tables show examples of Data Rates and Latencies for various block sizes. Other block sizes are also possible.

CLOCKS /BYTE	N	T	MAXIMUM LATENCY		DATA RATE (MB/sec)	EQUATION 5	EQUATION 6
			CLOCKS	μSECONDS			
4	25	10	209	5.23	10.00	GOOD	BAD
4	50	10	343	8.57	10.00	GOOD	BAD
4	53	10	359	8.97	10.00	GOOD	BAD
4	75	10	476	11.9	10.00	GOOD	GOOD
4	100	10	609	15.2	10.00	GOOD	GOOD
4	126	7	742	18.6	10.00	GOOD	GOOD
4	194	8	1107	27.7	10.00	GOOD	GOOD
4	208	8	1181	29.5	10.00	GOOD	GOOD
4	219	9	1242	31.1	10.00	GOOD	GOOD
4	200	10	1143	28.6	10.00	GOOD	GOOD
4	225	10	1276	31.9	10.00	GOOD	GOOD
4	250	10	1409	35.2	10.00	GOOD	GOOD
4	255	10	1436	35.9	10.00	GOOD	GOOD

CLOCKS /BYTE	N	T	MAXIMUM LATENCY		DATA RATE (MB/sec)	EQUATION 5	EQUATION 6
			CLOCKS	μSECONDS			
4	25	5	199	4.98	10.00	GOOD	BAD
4	50	5	333	8.32	10.00	GOOD	GOOD
4	75	5	466	11.7	10.00	GOOD	GOOD
4	100	5	599	15.0	10.00	GOOD	GOOD
4	125	5	733	18.3	10.00	GOOD	GOOD
4	150	5	866	21.7	10.00	GOOD	GOOD
4	175	5	999	25.0	10.00	GOOD	GOOD
4	200	5	1133	28.3	10.00	GOOD	GOOD
4	225	5	1266	31.7	10.00	GOOD	GOOD
4	250	5	1399	35.0	10.00	GOOD	GOOD
4	255	5	1426	35.7	10.00	GOOD	GOOD

CLOCKS /BYTE	N	T	MAXIMUM LATENCY		DATA RATE (MB/sec)	EQUATION 5	EQUATION 6
			CLOCKS	μSECONDS			
4	25	3	195	4.88	10.00	GOOD	BAD
4	50	3	329	8.22	10.00	GOOD	GOOD
4	75	3	462	11.6	10.00	GOOD	GOOD
4	100	3	595	14.9	10.00	GOOD	GOOD
4	125	3	729	18.2	10.00	GOOD	GOOD
4	150	3	862	21.6	10.00	GOOD	GOOD
4	175	3	995	24.9	10.00	GOOD	GOOD
4	200	3	1129	28.2	10.00	GOOD	GOOD
4	225	3	1262	31.6	10.00	GOOD	GOOD
4	250	3	1395	34.9	10.00	GOOD	GOOD
4	255	3	1422	35.6	10.00	GOOD	GOOD

CLOCKS /BYTE	N	T	MAXIMUM LATENCY		DATA RATE (MB/sec)	EQUATION 5	EQUATION 6
			CLOCKS	μSECONDS			
4	25	1	191	4.78	10.00	GOOD	BAD
4	50	1	325	8.12	10.00	GOOD	GOOD
4	75	1	458	11.5	10.00	GOOD	GOOD
4	100	1	591	14.8	10.00	GOOD	GOOD
4	125	1	725	18.1	10.00	GOOD	GOOD
4	150	1	858	21.5	10.00	GOOD	GOOD
4	175	1	991	24.8	10.00	GOOD	GOOD
4	200	1	1125	28.1	10.00	GOOD	GOOD
4	225	1	1258	31.5	10.00	GOOD	GOOD
4	250	1	1391	34.8	10.00	GOOD	GOOD
4	255	1	1418	35.5	10.00	GOOD	GOOD