# AmZ8160

## Cascadable 16-Bit Error Detection and Correction Unit

## ADVANCED DATA

### DISTINCTIVE CHARACTERISTICS

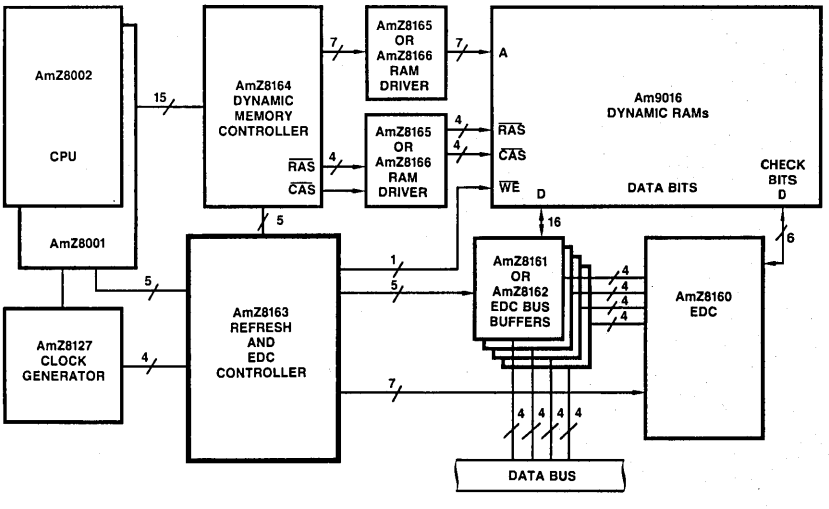
- Modified Hamming Code
  Detects multiple errors and corrects single bit errors in a parallel data word. Ideal for use in dynamic memory systems.
- Syndromes provided
  The AmZ8160 makes available the syndrome bits when an error occurs so the location of memory faults can be logged.
- Microprocessor compatible
  The AmZ8160 is designed to work with AmZ8000 microprocessor systems.
- Advanced circuit and process technologies
  Newest bipolar LSI techniques provide very high performance.
  Data-in to error detection typically 30ns
  Data-in to corrected data out typically 50ns
- Built-in Diagnostics
  Extra logic on the chip provides diagnostic functions to be used during device test and for system diagnostics.

### GENERAL DESCRIPTION

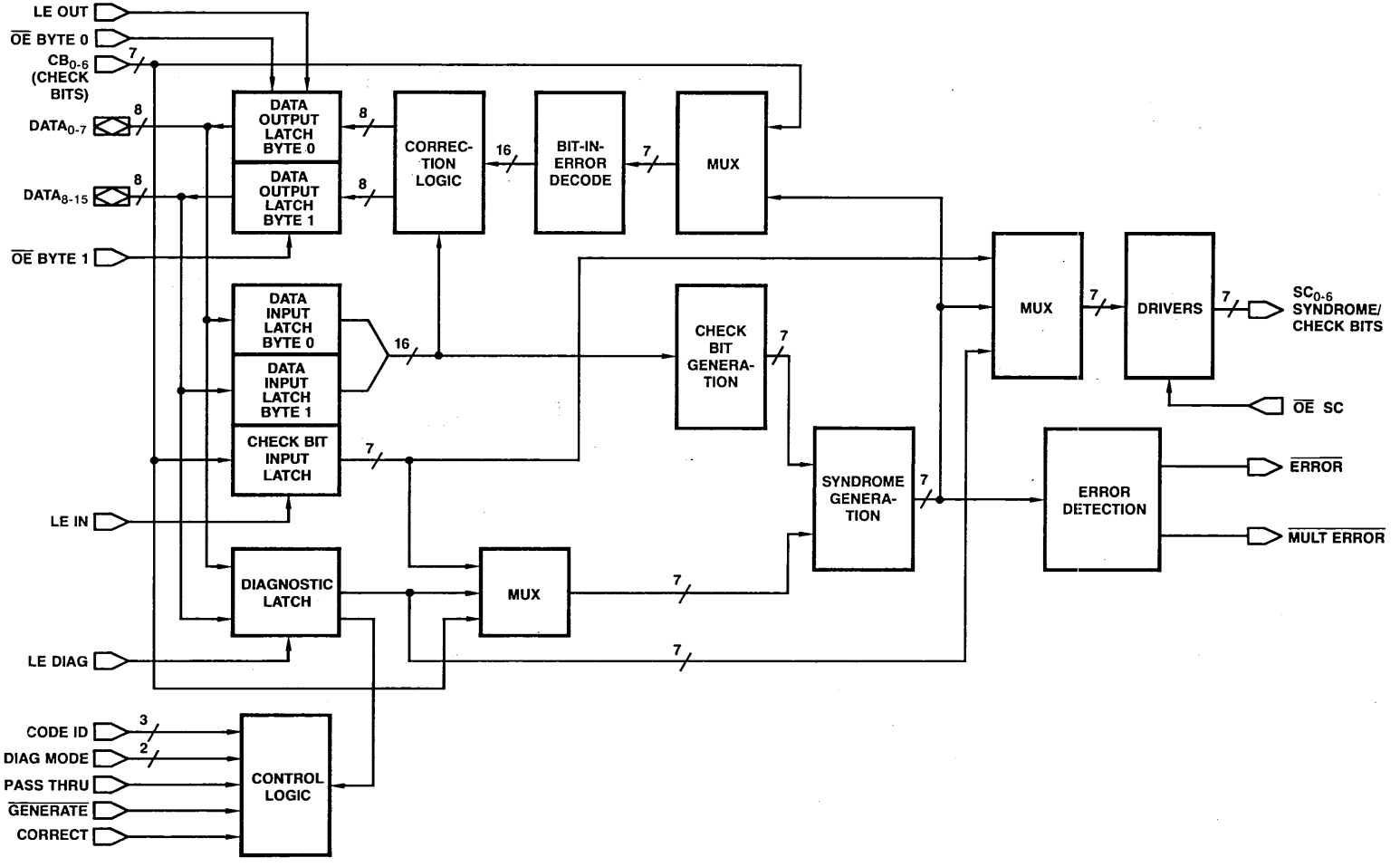
The AmZ8160 Error Detection and Correction Unit (EDC) contains the logic necessary to generate 6 check bits on a 16-bit data field according to a modified Hamming Code, and to correct the data word when check bits are supplied. Operating on data read from memory, the AmZ8160 will correct any single bit error and will detect all double on some triple bit errors. The AmZ8160 is expandable to operate on 32-bit words (7 check bits) and 64-bit words (8 check bits). In all configurations, the device makes the error syndrome available on separate outputs for data logging.

The AmZ8160 also features two diagnostic modes, in which diagnostic data can be forced into portions of the chip to simplify device testing and to execute system diagnostic functions. The product is supplied in a 48 lead hermetic DIP package.

### SYSTEM EXAMPLE




BLI-200




BLOCK DIAGRAM

### EDC Architecture

The EDC Unit is a powerful 16-bit cascadable slice used for check bit generation, error detection, error correction and diagnostics.

As shown in the block diagram, the device consists of the following:

- Data Input Latch
- Check Bit Input Latch
- Check Bit Generation Logic
- Syndrome Generation Logic
- Error Detection Logic
- Error Correction Logic
- Data Output Latch
- Diagnostic Latch
- Control Logic

### Data Input Latch

16 bits of data are loaded from the bidirectional DATA lines under control of the Latch Enable input, LE IN. Depending on the control mode the input data is either used for check bit generation or error detection/correction.

### Check Bit Input Latch

Seven check bits are loaded under control of LE IN. Check bits are used in the Error Detection and Error Correction modes.

### Check Bit Generation Logic

This block generates the appropriate check bits for the 16 bits of data in the Data Input Latch. The check bits are generated according to a modified Hamming code.

### Syndrome Generation Logic

In both Error Detection and Error Correction modes, this logic block compares the check bits read from memory against a newly generated set of check bits produced for the data read in from memory. If both sets of check bits match, then there are no errors. If there is a mismatch, then one or more of the data or check bits is in error.

The syndrome bits are produced by an exclusive-OR of the two sets of check bits. If the two sets of check bits are identical (meaning there are no errors) the syndrome bits will be all zeroes. If there are errors, the syndrome bits can be decoded to determine the number of errors and the bit-in-error.

### Error Detection Logic

This section decodes the syndrome bits generated by the Syndrome Generation Logic. If there are no errors in either the input data or check bits, the $\overline{\text{ERROR}}$ and $\overline{\text{MULTI ERROR}}$ outputs remain HIGH. If one or more errors are detected, $\overline{\text{ERROR}}$ goes LOW. If two or more errors are detected, both $\overline{\text{ERROR}}$ and $\overline{\text{MULTI ERROR}}$ go LOW.

### Error Correction Logic

For single errors, the Error Correction Logic complements (corrects) the single data bit in error. This corrected data is loadable into the Data Output Latch, which can then be read onto the bidirectional data lines. If the single error is one of the check bits, the correction logic does not place corrected check bits on the syndrome/check bit outputs. If the corrected check bits are needed the EDC must be switched to Generate Mode.

### Data Output Latch

The Data Output Latch is used for storing the result of an error correction operation. The latch is loaded from the correction logic under control of the Data Output Latch Enable, LE OUT. The Data Output Latch may also be loaded directly from the Data Input Latch under control of the PASS THRU control input.

The Data Output Latch is split into two 8-bit (byte) latches which may be enabled independently for reading onto the bidirectional data lines.

### Diagnostic Latch

This is a 16-bit latch loadable from the bidirectional data lines under control of the Diagnostic Latch Enable, LE DIAG. The Diagnostic Latch contains check bit information in one byte and control information in the other byte. The Diagnostic Latch is used for driving the device when in Internal Control Mode, or for supplying check bits when in one of the Diagnostic Modes.

### Control Logic

The control logic determines the specific mode the device operates in. Normally the control logic is driven by external control inputs. However, in Internal Control Mode, the control signals are instead read from the Diagnostic Latch.

## PIN DEFINITIONS

**$DATA_{0-15}$** 16 bidirectional data lines. They provide input to the Data Input Latch and Diagnostic Latch, and receive output from the Data Output Latch. $DATA_0$ is the least significant bit; $DATA_{15}$ the most significant.

**$CB_{0-6}$** Seven Check Bit input lines. The check bit lines are used to input check bits for error detection. Also used to input syndrome bits for error correction in 32 and 64-bit configurations.

**LE IN** Latch Enable – Data Input Latch. Controls latching of the input data. When HIGH the Data Input Latch and Check Bit Input Latch follow the input data and input check bits. When LOW, the Data Input Latch and Check Bit Input Latch are latched to their previous state.

**$\overline{\text{GENERATE}}$** Generate Check Bits input. When this input is LOW the EDC is in the Check Bit Generate Mode. When HIGH the EDC is in the Detect Mode or Correct Mode.

In the Generate Mode the circuit generates the check bits or partial check bits specific to the data in the Data Input Latch. The generated check bits are placed on the SC outputs.

In the Detect or Correct Modes the EDC detects single and multiple errors, and generates syndrome bits based upon the contents of the Data Input Latch and Check Bit Input Latch. In Correct Mode, single-bit errors are also automatically corrected – corrected data is placed at the inputs of the Data Output Latch. The syndrome result is placed on the SC outputs and indicates in a coded form the number of errors and the bit-in-error.

**$SC_{0-6}$** Syndrome/Check Bit outputs. These seven lines hold the check/partial-check bits when the EDC is in Generate Mode, and will hold the syndrome/partial syndrome bits when the device is in Detect or Correct Modes. These are 3-state outputs.

**$\overline{\text{OE SC}}$** Output Enable – Syndrome/Check Bits. When LOW, the 3-state output lines $SC_{0-6}$ are enabled. When HIGH, the SC outputs are in the high impedance state.

**$\overline{\text{ERROR}}$** Error Detected output. When the EDC is in Detect or Correct Mode, this output will go LOW if one or more syndrome bits are asserted, meaning there are one or more bit errors in the data or check bits. If no syndrome bits are asserted, there are no errors detected and the output will be HIGH. In Generate Mode, $\overline{\text{ERROR}}$ is forced HIGH. (In a 64-bit configuration, $\overline{\text{ERROR}}$ must be externally implemented.)

**$\overline{\text{MULTI ERROR}}$** Multiple Errors Detected output. When the EDC is in Detect or Correct Mode, this output if LOW indicates that there are two or more bit errors that have been detected. If HIGH this indicates that either one or no errors have been detected. In Generate Mode, $\overline{\text{MULTI ERROR}}$ is forced HIGH. (In a 64-bit configuration, $\overline{\text{MULTI ERROR}}$ must be externally implemented.)

**CORRECT** Correct input. When HIGH this signal allows the correction network to correct any single-bit error in the Data Input Latch (by complementing the bit-in-error) before putting it onto the Data Output Latch. When LOW the EDC will drive data directly from the Data Input Latch to the Data Output Latch without correction.

**LE OUT** Latch Enable – Data Output Latch. Controls the latching of the Data Output Latch. When LOW the Data Output Latch is latched to its previous state. When HIGH the Data Output Latch follows the output of the Data Input Latch as modified by the correction logic network. In Correct Mode, single-bit errors are corrected by the network before loading into the Data Output Latch. In Detect Mode, the contents of the Data Input Latch are passed through the correction network unchanged into the Data Output Latch. The inputs to the Data Output Latch are unspecified if the EDC is in Generate Mode.

**$\overline{\text{OE}}$ BYTE 0, $\overline{\text{OE}}$ BYTE 1** Output Enable – Bytes 0 and 1, Data Output Latch. These lines control the 3-state outputs for each of the two bytes of the Data Output Latch. When LOW these lines enable the Data Output Latch and when HIGH these lines force the Data Output Latch into the high impedance state. The two enable lines can be separately activated to enable only one byte of the Data Output Latch at a time.

**PASS THRU** Pass Thru input. This line when HIGH forces the contents of the Check Bit Input Latch onto the Syndrome/Check Bit outputs ($SC_{0-6}$) and the unmodified contents of the Data Input Latch onto the inputs of the Data Output Latch.

**DIAG $MODE_{0-1}$** Diagnostic Mode Select. These two lines control the initialization and diagnostic operation of the EDC.

**CODE $ID_{0-2}$** Code Identification inputs. These three bits identify the size of the total data word to be processed and which 16-bit slice of larger data words a particular EDC is processing. The three allowable data word sizes are 16, 32 and 64 bits and their respective modified Hamming codes are designated 16/22, 32/39 and 64/72. Special CODE ID input 001 ($ID_2$, $ID_1$, $ID_0$) is also used to instruct the EDC that the signals CODE $ID_{0-2}$, DIAG $MODE_{0-1}$, CORRECT and PASS THRU are to be taken from the Diagnostic Latch, rather than from the input control lines.

**LE DIAG** Latch Enable – Diagnostic Latch. When HIGH the Diagnostic Latch follows the 16-bit data on the input lines. When LOW the outputs of the Diagnostic Latch are latched to their previous states. The Diagnostic Latch holds diagnostic check bits, and internal control signals for CODE $ID_{0-2}$, DIAG $MODE_{0-1}$, CORRECT and PASS THRU.

## FUNCTIONAL DESCRIPTION

The EDC contains the logic necessary to generate check bits on a 16-bit data field according to a modified Hamming code. Operating on data read from memory, the EDC will correct any single-bit error, and will detect all double and some triple-bit errors. It may be configured to operate on 16-bit data words (with 6 check bits), 32-bit data words (with 7 check bits) and 64-bit data words (with 8 check bits). In all configurations, the device makes the error syndrome bits available on separate outputs for error data logging.

### Code and Byte Specification

The EDC may be configured in several different ways and operates differently in each configuration. It is necessary to indicate to the device what size data word is involved and which bytes of the data word it is processing. This is done with input lines CODE $ID_{0-2}$, as shown in Table I. The three modified Hamming codes referred to in Table I are:

- 16/22 code – 16 data bits
  – 6 check bits
  – 22 bits in total.
- 32/39 code – 32 data bits
  – 7 check bits
  – 39 bits in total.
- 64/72 code – 64 data bits
  – 8 check bits
  – 72 bits in total.

CODE ID input 001 ($ID_2$, $ID_1$, $ID_0$) is a special code used to operate the device in Internal Control Mode (described later in this section).

**TABLE I. HAMMING CODE AND SLICE IDENTIFICATION.**

| CODE $ID_2$ | CODE $ID_1$ | CODE $ID_0$ | Hamming Code and Slice Selected |
|---|---|---|---|
| 0 | 0 | 0 | Code 16/22 |
| 0 | 0 | 1 | Internal Control Mode |
| 0 | 1 | 0 | Code 32/39, Bytes 0 and 1 |
| 0 | 1 | 1 | Code 32/39, Bytes 2 and 3 |
| 1 | 0 | 0 | Code 64/72, Bytes 0 and 1 |
| 1 | 0 | 1 | Code 64/72, Bytes 2 and 3 |
| 1 | 1 | 0 | Code 64/72, Bytes 4 and 5 |
| 1 | 1 | 1 | Code 64/72, Bytes 6 and 7 |

### Control Mode Selection

The device control lines are $\overline{\text{GENERATE}}$, CORRECT, PASS THRU, DIAG $MODE_{0-1}$ and CODE $ID_{0-2}$. Table II indicates the control modes selected by various combinations of the control line inputs.

### Diagnostics

Table III shows specifically how DIAG $MODE_{0-1}$ select between normal operation, initialization and one of two diagnostic modes.

The Diagnostic Modes allow the user to operate the EDC under software control in order to verify proper functioning of the device.

### Check and Syndrome Bit Labeling

The check bits generated in the EDC are designated as follows:

- 16-bit configuration – CX C0, C1, C2, C4, C8;
- 32-bit configuration – CX, C0, C1, C2, C4, C8, C16;
- 64-bit configuration – CX, C0, C1, C2, C4, C8, C16, C32.

Syndrome bits are similarly labeled SX through S32. There are only 6 syndrome bits in the 16-bit configuration, 7 for 32 bits and 8 syndrome bits in the 64-bit configuration.

## FUNCTIONAL DESCRIPTION – 16-BIT DATA WORD CONFIGURATION

The 16-bit format consists of 16 data bits, 6 check bits and is referred to as 16/22 code (see Figure 1).

The 16-bit configuration is shown in Figure 2.

### Generate Mode

In this mode check bits will be generated that correspond to the contents of the Data Input Latch. The check bits generated are placed on the outputs $SC_{0-5}$ ($SC_6$ is unspecified for 16-bit operation).

Check bits are generated according to a modified Hamming code. Details of the code for check bit generation are contained in Table IV. Each check bit is generated as either an XOR or XNOR of eight of the 16 data bits as indicated in the table. The XOR function results in an even parity check bit, the XNOR is an odd parity check bit.

### Detect Mode

In this mode the device examines the contents of the Data Input Latch against the Check Bit Input Latch, and will detect all single-bit errors, all double-bit errors and some triple-bit errors. If one or more errors are detected, $\overline{\text{ERROR}}$ goes LOW. If two or more errors are detected, $\overline{\text{MULTI ERROR}}$ goes LOW. Both error indicators are HIGH if there are no errors.

Also available on device outputs $SC_{0-5}$ are the syndrome bits generated by the error detection step. The syndrome bits may be decoded to determine if a bit error was detected and, for single-bit errors, which of the data or check bits is in error. Table V gives the chart for decoding the syndrome bits generated by the 16-bit configuration (as an example, if the syndrome bits SX/S0/S1/S2/S4/S8 were 101001 this would be decoded to indicate that there is a single-bit error at data bit 9). If no error is detected the syndrome bits will all be zeroes.

In Detect Mode, the contents of the Data Input Latch are driven directly to the inputs of the Data Output Latch without correction.

### Correct Mode

In this mode, the EDC functions the same as in Detect Mode except that the correction network is allowed to correct (complement) any single-bit error of the Data Input Latch before putting it onto the inputs of the Data Output Latch. If multiple errors are detected, the output of the correction network is unspecified. If the single-bit error is a check bit there is no automatic correction. If check bit correction is desired, this can be done by placing the device in Generate Mode to produce a correct check bit sequence for the data in the Data Input Latch.

### Pass Thru Mode

In this mode, the unmodified contents of the Data Input Latch are placed on the inputs of the Data Output Latch and the contents of the Check Bit Input Latch are placed on outputs $SC_{0-5}$.

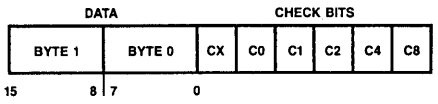
8

**TABLE II. EDC CONTROL MODE SELECTION.**

| $\overline{\text{GENERATE}}$ | CORRECT | PASS THRU | DIAG $\text{MODE}_{0\text{-}1}$ ($DM_1$, $DM_0$) | CODE $\text{ID}_{0\text{-}2}$ ($ID_2$, $ID_1$, $ID_0$) | Control Mode Selected |
|---|---|---|---|---|---|
| LOW | LOW | LOW | 00 | Not 001 | Generate |
| LOW | LOW | LOW | 01 | Not 001 | Generate Using Diagnostic Latch |
| LOW | LOW | LOW | 10 | Not 001 | Generate |
| LOW | LOW | LOW | 11 | Not 001 | Initialize |
| LOW | LOW | HIGH | 00 | Not 001 | Pass Thru |
| LOW | LOW | HIGH | 01 | Not 001 | Pass Thru |
| LOW | LOW | HIGH | 10 | Not 001 | Pass Thru |
| LOW | LOW | HIGH | 11 | Not 001 | Undefined |
| LOW | HIGH | LOW | 00 | Not 001 | Generate |
| LOW | HIGH | LOW | 01 | Not 001 | Generate Using Diagnostic Latch |
| LOW | HIGH | LOW | 10 | Not 001 | Generate |
| LOW | HIGH | LOW | 11 | Not 001 | Initialize |
| LOW | HIGH | HIGH | 00 | Not 001 | Pass Thru |
| LOW | HIGH | HIGH | 01 | Not 001 | Pass Thru |
| LOW | HIGH | HIGH | 10 | Not 001 | Pass Thru |
| LOW | HIGH | HIGH | 11 | Not 001 | Undefined |
| HIGH | LOW | LOW | 00 | Not 001 | Detect |
| HIGH | LOW | LOW | 01 | Not 001 | Detect |
| HIGH | LOW | LOW | 10 | Not 001 | Detect Using Diagnostic Latch |
| HIGH | LOW | LOW | 11 | Not 001 | Initialize |
| HIGH | LOW | HIGH | 00 | Not 001 | Pass Thru |
| HIGH | LOW | HIGH | 01 | Not 001 | Pass Thru |
| HIGH | LOW | HIGH | 10 | Not 001 | Pass Thru |
| HIGH | LOW | HIGH | 11 | Not 001 | Undefined |
| HIGH | HIGH | LOW | 00 | Not 001 | Correct |
| HIGH | HIGH | LOW | 01 | Not 001 | Correct |
| HIGH | HIGH | LOW | 10 | Not 001 | Correct Using Diagnostic Latch |
| HIGH | HIGH | LOW | 11 | Not 001 | Initialize |
| HIGH | HIGH | HIGH | 00 | Not 001 | Pass Thru |
| HIGH | HIGH | HIGH | 01 | Not 001 | Pass Thru |
| HIGH | HIGH | HIGH | 10 | Not 001 | Pass Thru |
| HIGH | HIGH | HIGH | 11 | Not 001 | Undefined |
| Any | Any | Any | Any | 001 | Internal Control Using Diagnostic Latch |

**TABLE III. DIAGNOSTIC MODE CONTROL.**

| DIAG $\text{MODE}_1$ | DIAG $\text{MODE}_0$ | Diagnostic Mode Selected |
|---|---|---|
| 0 | 0 | **Non-diagnostic mode.** The EDC functions normally in all modes. |
| 0 | 1 | **Diagnostic Mode A.** The contents of the Diagnostic Latch are substituted for the normally generated check bits when in the Generate Mode. The EDC functions normally in the Detect or Correct modes. |
| 1 | 0 | **Diagnostic Mode B.** In the Detect or Correct Mode, the contents of the Diagnostic Latch are substituted for the check bits normally read from the Check Bit Input Latch. The EDC functions normally in the Generate Mode. |
| 1 | 1 | **Initialize.** The inputs of the Data Output Latch are forced to zeroes and the check bits generated correspond to the all-zero data. |




Uses Modified Hamming Code 16/22
- 16 data bits
- 6 check bits
- 22 bits in total

BLI-202

**Figure 1. 16 Bit Data Format.**

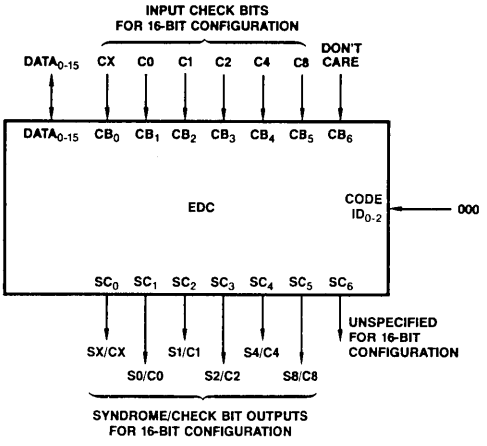



**Figure 2. 16 Bit Configuration.**

BLI-203

**TABLE IV. 16-BIT MODIFIED HAMMING CODE – CHECK BIT ENCODE CHART.**

| Generated Check Bits | Parity | Participating Data Bits | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| CX | Even (XOR) | | X | X | X | | X | | | X | X | | X | | | X | |
| C0 | Even (XOR) | X | X | X | | X | | X | | X | | X | | X | | | |
| C1 | Odd (XNOR) | X | | | X | X | | | X | | X | X | | | X | | X |
| C2 | Odd (XNOR) | X | X | | | | X | X | X | | | | X | X | X | | |
| C4 | Even (XOR) | | | X | X | X | X | X | X | | | | | | | X | X |
| C8 | Even (XOR) | | | | | | | | | X | X | X | X | X | X | X | X |

The check bit is generated as either an XOR or XNOR of the eight data bits noted by an "X" in the table.

**TABLE V. SYNDROME DECODE TO BIT-IN-ERROR.**

| Syndrome Bits | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | S8 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| | | | S4 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| | | | S2 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| SX | S0 | S1 | | | | | | | | | |
| 0 | 0 | 0 | | * | C8 | C4 | T | C2 | T | T | M |
| 0 | 0 | 1 | | C1 | T | T | 15 | T | 13 | 7 | T |
| 0 | 1 | 0 | | C0 | T | T | M | T | 12 | 6 | T |
| 0 | 1 | 1 | | T | 10 | 4 | T | 0 | T | T | M |
| 1 | 0 | 0 | | CX | T | T | 14 | T | 11 | 5 | T |
| 1 | 0 | 1 | | T | 9 | 3 | T | M | T | T | M |
| 1 | 1 | 0 | | T | 8 | 2 | T | 1 | T | T | M |
| 1 | 1 | 1 | | M | T | T | M | T | M | M | T |

* – no errors detected
Number – the location of the single bit-in-error
T – two errors detected
M – three or more errors detected

**TABLE VI. DIAGNOSTIC LATCH LOADING – 16-BIT FORMAT.**

| Data Bit | Internal Function |
|---|---|
| 0 | Diagnostic Check Bit X |
| 1 | Diagnostic Check Bit 0 |
| 2 | Diagnostic Check Bit 1 |
| 3 | Diagnostic Check Bit 2 |
| 4 | Diagnostic Check Bit 4 |
| 5 | Diagnostic Check Bit 8 |
| 6, 7 | Don't Care |
| 8 | CODE ID 0 |
| 9 | CODE ID 1 |
| 10 | CODE ID 2 |
| 11 | DIAG MODE 0 |
| 12 | DIAG MODE 1 |
| 13 | CORRECT |
| 14 | PASS THRU |
| 15 | Don't Care |

8

### Diagnostic Latch

The Diagnostic Latch serves both for diagnostic uses and internal control uses. It is loaded from the DATA lines under the control of LE DIAG. Table VI shows the loading definitions for the DATA lines.

**Generate Using Diagnostic Latch (Diagnostic Mode A)**

**Detect Using Diagnostic Latch (Diagnostic Mode B)**

**Correct Using Diagnostic Latch (Diagnostic Mode B)**

These are special diagnostic modes selected by DIAG $MODE_{0-1}$ where either normal check bit inputs or outputs are substituted for by check bits loaded into the Diagnostic Latch. See Table III for details.

### Internal Control Mode

This mode is selected by CODE $ID_{0-2}$ input 001 ($ID_2$, $ID_1$, $ID_0$).

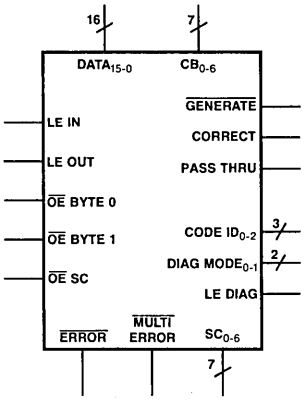
When in Internal Control Mode, the EDC takes the CODE $ID_{0-2}$, DIAG $MODE_{0-1}$, CORRECT and PASS THRU control signals from the internal Diagnostic Latch rather than from the external input lines.

Table VI gives the format for loading the Diagnostic Latch.

### 32 and 64-Bit Operation

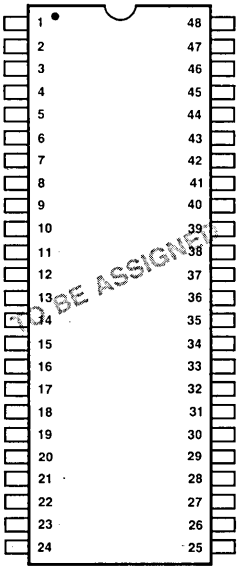
The EDC can easily be cascaded to operate on 32 and 64-bit data words. Since this is unlikely to occur in a Z8000 system, it is not discussed in this data sheet. Interested users should refer to the Am2960 data sheet for more information.

**LOGIC SYMBOL**




BLI-204

**CONNECTION DIAGRAM**
**Top View**




Note: Pin 1 is marked for orientation.

BLI-205