



# Am2960/Am2960-1/Am2960A

## Cascadable 16-Bit Error Detection and Correction Unit

### DISTINCTIVE CHARACTERISTICS

- **Boosts Memory Reliability**  
Corrects all single-bit errors. Detects all double and some triple-bit errors.
- **Very High Speed**  
Perfect for MOS microprocessor, minicomputer, and main-frame systems.  
  
High performance systems can use the Am2960 EDC in check-only mode to avoid memory system slowdown.
- **Handles Data Words From 8 Bits to 64 Bits**  
The Am2960 EDC cascades: 1 EDC for 8 bits or 16 bits, 2 for 32 bits, 4 for 64 bits.
- **Easy Byte Operations**  
Separate byte enables on the data out latch simplify the steps and cut the time required for byte writes.
- **Diagnostics Built-In**  
The processor may completely exercise the EDC under software control to check for proper operation of the EDC.

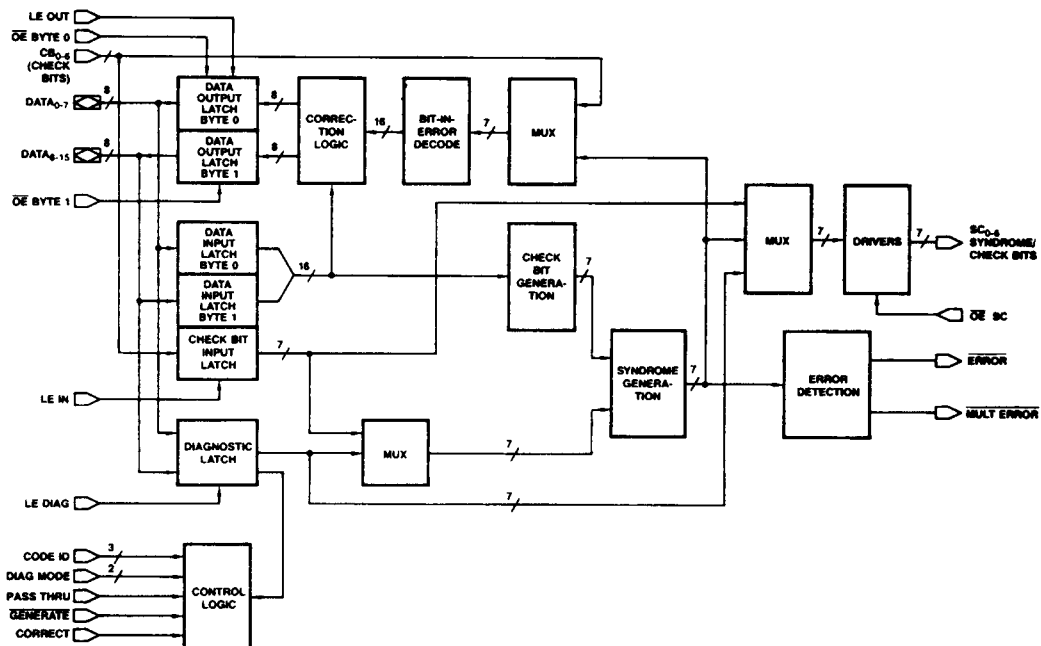
### GENERAL DESCRIPTION

The Am2960 Error Detection and Correction Unit (EDC) contains the logic necessary to generate 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 Am2960 will correct any single-bit error and will detect all double and some triple-bit errors. For 16-bit words, 6 check bits are used. The Am2960 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 Am2960 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.

### BLOCK DIAGRAM

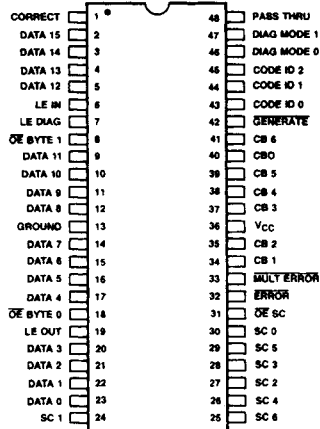


BD001261

# CONNECTION DIAGRAMS

## Top View

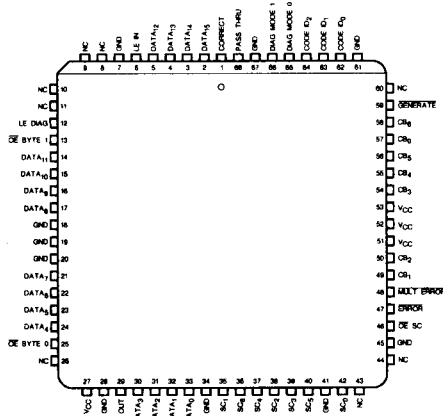
### DIPs\*



CD001421

\*Also available in Flatpack. Pinout is identical to DIPs.

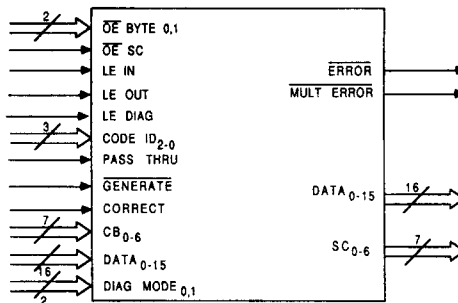
### PLCC



CD010232

Note: Pin 1 is marked for orientation.

### LOGIC SYMBOL



LS002831

## RELATED AMD PRODUCTS

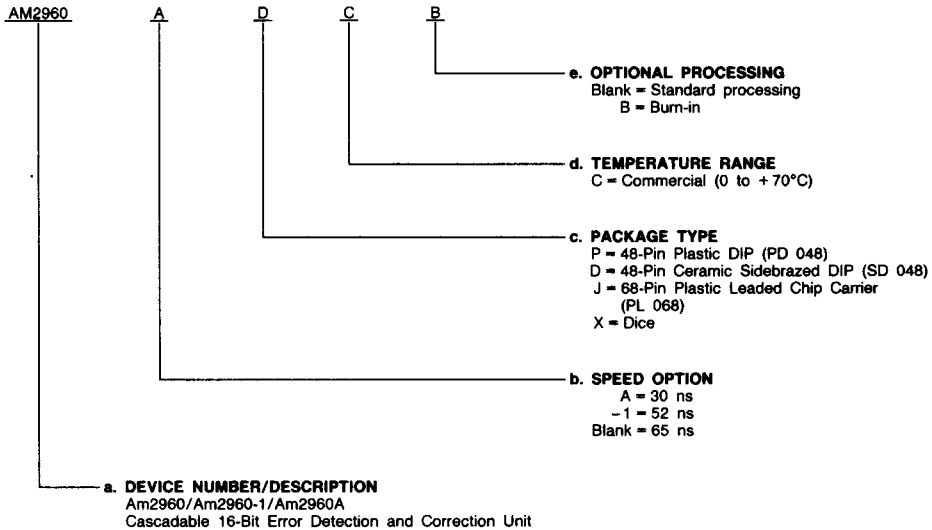
Part No.	Description
Am29368	1M Dynamic Memory Controller
Am29C60A	16-Bit CMOS EDC
Am2968A	256K Dynamic Memory Controller
Am2969	Memory Timing Controller w/EDC Control
Am2971A	Programmable Event Generator
Am29C660	32-Bit CMOS EDC
Am29C668	4M - 64K Configurable Dynamic Memory Controller/Driver
Am29C983	9-Bit x 4-Port Multiple Bus Exchange

## ORDERING INFORMATION

### Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

- a. **Device Number**
- b. **Speed Option** (if applicable)
- c. **Package Type**
- d. **Temperature Range**
- e. **Optional Processing**



### Valid Combinations

Valid Combinations	
AM2960	DC, DCB, PC, PCB, XC, JC, JCB
AM2960A	
AM2960-1	

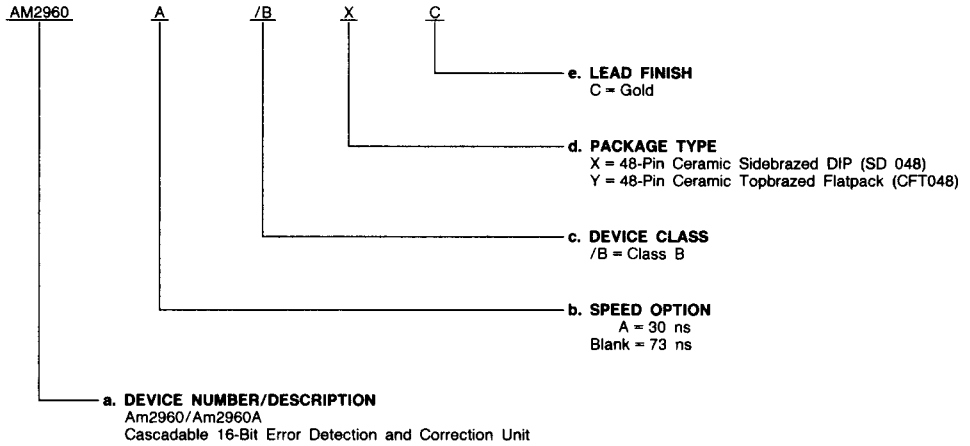
Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

# MILITARY ORDERING INFORMATION

## APL Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. APL (Approved Products List) products are fully compliant with MIL-STD-883C requirements. The order number (Valid Combination) for APL products is formed by a combination of:

- a. **Device Number**
- b. **Speed Option** (if applicable)
- c. **Device Class**
- d. **Package Type**
- e. **Lead Finish**



### Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations or to check for newly released valid combinations.

Valid Combinations	
AM2960	/BXC, /BYC
AM2960A	

### Group A Tests

Group A tests consists of Subgroups:

- 1, 2, 3, 7, 8, 9, 10, 11

## PIN DESCRIPTION

### **DATA<sub>0-15</sub> Data (Input/Output (16))**

These bidirectional data lines provide input to the Data Input Latch and Diagnostic Latch, and receive output from the Data Output Latch. DATA<sub>0</sub> is the least significant bit; DATA<sub>15</sub> the most significant.

### **CB<sub>0-6</sub> Check Bits (Input (7))**

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 (Input)**

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.

### **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<sub>0-6</sub> Syndrome/Check Bits (Output (7))**

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.

### **OE SC Output Enable, Syndrome/Check Bits (Input)**

Syndrome/Check Bits. When LOW, the 3-state output lines SC<sub>0-6</sub> are enabled. When HIGH, the SC outputs are in the high impedance state.

### **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, ERROR is forced HIGH. (In a 64-bit configuration, ERROR must be externally implemented.)

### **MULT 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, MULT

ERROR is forced HIGH. (In a 64-bit configuration, MULT 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 into 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 (Input)**

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.

### **OE BYTE 0, 1 Output Enable Bytes 0, 1 (Input)**

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<sub>0-6</sub>) and the unmodified contents of the Data Input Latch onto the inputs of the Data Output Latch.

### **DIAG MODE<sub>0-1</sub> Diagnostic Mode Select (Input)**

These two lines control the initialization and diagnostic operation of the EDC.

### **CODE ID<sub>2-0</sub> Code Identification (Input)**

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<sub>2</sub>, ID<sub>1</sub>, ID<sub>0</sub>) is also used to instruct the EDC that the signals CODE ID<sub>2-0</sub>, DIAG MODE<sub>0-1</sub>, CORRECT and PASS THRU are to be taken from the Diagnostic Latch, rather than from the input control lines.

### **LE DIAG Diagnostic Latch Enable (Input)**

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<sub>2-0</sub>, DIAG MODE<sub>0-1</sub>, CORRECT and PASS THRU.

## FUNCTIONAL DESCRIPTION

### 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 ERROR and MULT ERROR outputs remain HIGH. If one or more errors are detected, ERROR goes LOW. If two or more errors are detected, both ERROR and MULT 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 appears as 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.

## GENERAL OPERATION

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. The Am2960 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 fact the EDC can be configured to work on data words from 8 to 64 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<sub>2-0</sub>, as shown in Table 1. The three modified Hamming codes referred to in Table 1 are:

- 16/22
  - 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<sub>2</sub>, ID<sub>1</sub>, ID<sub>0</sub>) is a special code used to operate the device in Internal Control Mode (described later in this section).

### Control Mode Selection

The device control lines are GENERATE, CORRECT, PASS THRU, DIAG MODE<sub>0-1</sub> and CODE ID<sub>2-0</sub>. Table 3 indicates the operating modes selected by various combinations of the control line inputs.

### Diagnostics

Table 2 shows specifically how DIAG MODE<sub>0-1</sub> 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.

### Initialize Mode

The outputs of the Data Input Latch are forced to zero (and the Data Input Latches are latched in zero upon removal of the initialized mode).

The inputs of the Data Output Latch are forced to zeroes. The check bit outputs (SC) are generated to correspond to the all-zero data. ERROR and MULT ERROR are forced HIGH in the Initialize Mode.

Initialize Mode is useful after power up when RAM contents are random. The EDC may be placed in initialize mode and its' outputs written in to all memory locations by the processor.

**TABLE 1. HAMMING CODE AND SLICE IDENTIFICATION**

CODE ID <sub>2</sub>	CODE ID <sub>1</sub>	CODE ID <sub>0</sub>	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

**TABLE 2. DIAGNOSTIC MODE CONTROL**

DIAG MODE <sub>1</sub>	DIAG MODE <sub>0</sub>	Diagnostic Mode Selected
0	0	<b>Non-diagnostic mode.</b> The EDC functions normally in all modes.
0	1	<b>Diagnostic Generate.</b> 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	<b>Diagnostic Detect/Correct.</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	<b>Initialize.</b> The outputs of the Data Input Latch are forced to zeroes (and the Data Input Latches are latched in zero upon removal of the Initialize Mode), inputs of the Data Output Latches are forced to zeroes, and the check bits generated correspond to the all-zero data.

### HAMMING CODE SELECTION

The Am2960 EDC uses a modified Hamming Code that allows 1) the EDC to be cascaded; 2) all double errors to be detected; 3) the gross error conditions of all 0s or 1s to be detected.

The error correction code can be selected independently of the processor with the exception of diagnostics software.

Diagnostic software run by a processor to check out the EDC system must know specifically which code is being used. This

is only a problem when the EDC replaces an existing MSI implementation on an existing computer. In this case, the computer's software must first determine which of two codes (the old one used by the MSI implementation or the new one used by the EDC) is used by the computer's memory system.

This is easily determined by writing a test data word into memory and then examining whether the generated check bits are typical of the old or the new code. From then on the software runs only the diagnostic appropriate for the code used on that particular computer's memory system.

**TABLE 3. OPERATING MODES**

Operating Mode	Diagnostic Mode**		GENERATE	
	DM <sub>1</sub>	DM <sub>0</sub>	0	1
Normal	0	0	Generate	Correct*
Diagnostic Generate	0	1	Diagnostic Generate	Correct*
Diagnostic Correct	1	0	Generate	Diagnostic Correct*
Initialize	1	1	Initialize	Initialize
Pass Thru	When PASS THRU is asserted, the Operating Mode is defaulted to the Pass Thru Mode.			

\*Correct if the CORRECT Input is HIGH, Detect if the CORRECT Input is LOW.

\*\*In Code ID<sub>2-0001</sub>(ID<sub>2</sub>, ID<sub>1</sub>, ID<sub>0</sub>) DM<sub>1</sub> and DM<sub>0</sub> are taken from the Diagnostic Latch.

**FUNCTIONAL EQUATIONS**

The following equations and tables describe in detail how the output values of the AM2960 EDC are determined as a function of the value of the inputs and the internal states. Be sure to carefully read the following definitions of symbols before examining the tables.

**Definitions**

- D<sub>i</sub> ← (DATA<sub>i</sub> if LE IN is HIGH or the output of bit i of the Data Input Latch if LE IN is LOW)
- C<sub>i</sub> ← (CB<sub>i</sub> if LE IN is HIGH or the output of bit i of the Check Bit Latch if LE IN is LOW)
- DL<sub>i</sub> ← Output of bit i of the Diagnostic Latch
- S<sub>i</sub> ← Internally generated syndromes (same as outputs of SC<sub>i</sub> if outputs enabled)
- PA ← D<sub>0</sub>⊕D<sub>1</sub>⊕D<sub>2</sub>⊕D<sub>4</sub>⊕D<sub>6</sub>⊕D<sub>8</sub>⊕D<sub>10</sub>⊕D<sub>12</sub>
- PB ← D<sub>0</sub>⊕D<sub>1</sub>⊕D<sub>2</sub>⊕D<sub>3</sub>⊕D<sub>4</sub>⊕D<sub>5</sub>⊕D<sub>6</sub>⊕D<sub>7</sub>
- PC ← D<sub>8</sub>⊕D<sub>9</sub>⊕D<sub>10</sub>⊕D<sub>11</sub>⊕D<sub>12</sub>⊕D<sub>13</sub>⊕D<sub>14</sub>⊕D<sub>15</sub>
- PD ← D<sub>0</sub>⊕D<sub>3</sub>⊕D<sub>4</sub>⊕D<sub>7</sub>⊕D<sub>9</sub>⊕D<sub>10</sub>⊕D<sub>13</sub>⊕D<sub>15</sub>
- PE ← D<sub>0</sub>⊕D<sub>1</sub>⊕D<sub>5</sub>⊕D<sub>6</sub>⊕D<sub>7</sub>⊕D<sub>11</sub>⊕D<sub>12</sub>⊕D<sub>13</sub>
- PF ← D<sub>2</sub>⊕D<sub>3</sub>⊕D<sub>4</sub>⊕D<sub>5</sub>⊕D<sub>6</sub>⊕D<sub>7</sub>⊕D<sub>14</sub>⊕D<sub>15</sub>
- PG<sub>1</sub> ← D<sub>0</sub>⊕D<sub>4</sub>⊕D<sub>6</sub>⊕D<sub>7</sub>
- PG<sub>2</sub> ← D<sub>1</sub>⊕D<sub>2</sub>⊕D<sub>3</sub>⊕D<sub>5</sub>
- PG<sub>3</sub> ← D<sub>8</sub>⊕D<sub>9</sub>⊕D<sub>11</sub>⊕D<sub>14</sub>
- PG<sub>4</sub> ← D<sub>10</sub>⊕D<sub>12</sub>⊕D<sub>13</sub>⊕D<sub>15</sub>

**Error Signals**

$$\text{ERROR} = (S_6 \cdot (ID_1 + ID_2)) \cdot S_5 \cdot S_4 \cdot S_3 \cdot S_2 \cdot S_1 \cdot S_0 + \text{GENERATE} + \text{INITIALIZE} + \text{PASSTHRU}$$

$$\text{MULT ERROR (16 and 32-Bit Modes)} \leftarrow \frac{((S_6 \cdot ID_1) \oplus S_5 \oplus S_4 \oplus S_3 \oplus S_2 \oplus S_1 \oplus S_0) (\text{ERROR}) + \text{TOME} + \text{GENERATE} + \text{PASSTHRU} + \text{INITIALIZE}}$$

$$\text{MULT ERROR (64-Bit Modes)} = \text{TOME} + \text{GENERATE} + \text{PASSTHRU} + \text{INITIALIZE}$$



**TABLE 4. TOME (Three or More Errors)\***

			S0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
			**S6	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
			S5	0	0	0	0	1	1	0	1	1	0	0	1	1	0	0	1
			S4	0	0	0	0	0	1	1	0	1	1	0	1	1	0	0	1
S1	S2	S3		0	0	0	1	0	1	1	1	0	1	1	1	0	0	0	0
0	0	0		0	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
0	1	0		0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1		1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
1	0	0		0	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1
1	0	1		0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
1	1	0		1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1
1	1	1		0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

\*S6, S5, ...S0 are internal syndromes except in Modes 010, 100, 101, 110, 111 (CODE ID2, ID1, ID0). In these modes the syndromes are input over the Check-Bit lines. S6-C6, S5-C5, ...S1-C1, S0-C0. \*\*The S6 internal syndrome is always forced to 0 in CODE ID 000.

**SC Outputs**

Tables 5 to 9 show how outputs SC0-6 are generated in each control mode for various CODE IDs (internal control mode not applicable).

**TABLE 5. GENERATE MODE (Check Bits)**

GENERATE Mode (Check Bits)	CODE ID2-0						
	000	010	011	100	101	110	111
SC0-	PG2⊕PG3	PG1⊕PG3	PG2⊕PG4⊕CB0	PG2⊕PG3	PG2⊕PG3	PG1⊕PG4	PG1⊕PG4
SC1-	PA	PA	PA⊕CB1	PA	PA	PA	PA
SC2-	PD	PD	PD⊕CB2	PD	PD	PD	PD
SC3-	PE	PE	PE⊕CB3	PE	PE	PE	PE
SC4-	PF	PF	PF⊕CB4	PF	PF	PF	PF
SC5-	PC	PC	PC⊕CB5	PC	PC	PC	PC
SC6-	1	PB	PC⊕CB6	PB	PB	PB	PB

**TABLE 6. DETECT AND CORRECT MODES (Syndromes)**

Detect and Correct Modes (Syndromes)	CODE ID2-0						
	000	010	011*	100	101	110	111
SC0-	PG2⊕PG3⊕C0	PG1⊕PG3⊕C0	PG2⊕PG4⊕CB0	PG2⊕PG3⊕C0	PG2⊕PG3	PG1⊕PG4	PG1⊕PG4
SC1-	PA⊕C1	PA⊕C1	PA⊕CB1	PA⊕C1	PA	PA	PA
SC2-	PD⊕C2	PD⊕C2	PD⊕CB2	PD⊕C2	PD	PD	PD
SC3-	PE⊕C3	PE⊕C3	PE⊕CB3	PE⊕C3	PE	PE	PE
SC4-	PF⊕C4	PF⊕C4	PF⊕CB4	PF⊕C4	PF	PF	PF
SC5-	PC⊕C5	PC⊕C5	PC⊕CB5	PC⊕C5	PC	PC	PC
SC6-	1	PB⊕C6	PC⊕CB6	PB	PB	PB⊕C6	PB⊕C6

\*In CODE ID2-0 011 the Check-Bit Latch is forced transparent; the Data Latch operates normally.

**TABLE 7. DIAGNOSTIC READ MODE**

Diagnostic Read Mode	CODE ID2-0						
	000	010	011*	100	101	110	111
SC0-	PG2⊕PG3⊕DL0	PG1⊕PG3⊕DL0	PG2⊕PG4⊕DL0	PG2⊕PG3⊕DL0	PG2⊕PG3	PG1⊕PG4	PG1⊕PG4
SC1-	PA⊕DL1	PA⊕DL1	PA⊕CB1	PA⊕DL1	PA	PA	PA
SC2-	PD⊕DL2	PD⊕DL2	PD⊕CB2	PD⊕DL2	PD	PD	PD
SC3-	PE⊕DL3	PE⊕DL3	PE⊕CB3	PE⊕DL3	PE	PE	PE
SC4-	PF⊕DL4	PF⊕DL4	PF⊕CB4	PF⊕DL4	PF	PF	PF
SC5-	PC⊕DL5	PC⊕DL5	PC⊕CB5	PC⊕DL5	PC	PC	PC
SC6-	1	PB⊕DL6	PC⊕CB6	PB	PB	PB⊕DL6	PB⊕DL7

\*In CODE ID2-0 011 the Check-Bit Latch is forced transparent; the Data Latch operates normally.

**TABLE 8. DIAGNOSTIC WRITE MODE**

Diagnostic Write Mode	CODE ID <sub>2-0</sub>						
	000	010	011*	100	101	110	111
SC <sub>0</sub> --	DL <sub>0</sub>	DL <sub>0</sub>	CB <sub>0</sub>	DL <sub>0</sub>	1	1	1
SC <sub>1</sub> --	DL <sub>1</sub>	DL <sub>1</sub>	CB <sub>1</sub>	DL <sub>1</sub>	1	1	1
SC <sub>2</sub> --	DL <sub>2</sub>	DL <sub>2</sub>	CB <sub>2</sub>	DL <sub>2</sub>	1	1	1
SC <sub>3</sub> --	DL <sub>3</sub>	DL <sub>3</sub>	CB <sub>3</sub>	DL <sub>3</sub>	1	1	1
SC <sub>4</sub> --	DL <sub>4</sub>	DL <sub>4</sub>	CB <sub>4</sub>	DL <sub>4</sub>	1	1	1
SC <sub>5</sub> --	DL <sub>5</sub>	DL <sub>5</sub>	CB <sub>5</sub>	DL <sub>5</sub>	1	1	1
SC <sub>6</sub> --	1	DL <sub>6</sub>	CB <sub>6</sub>	1	1	DL <sub>6</sub>	DL <sub>7</sub>

\*In CODE ID<sub>2-0</sub> 011 the Check-Bit Latch is forced transparent; the Data Input Latch operates normally.

**TABLE 9. PASS THRU MODE**

PASS THRU Mode	CODE ID <sub>2-0</sub>						
	000	010	011*	100	101	110	111
SC <sub>0</sub> --	C0	C0	CB <sub>0</sub>	C0	1	1	1
SC <sub>1</sub> --	C1	C1	CB <sub>1</sub>	C1	1	1	1
SC <sub>2</sub> --	C2	C2	CB <sub>2</sub>	C2	1	1	1
SC <sub>3</sub> --	C3	C3	CB <sub>3</sub>	C3	1	1	1
SC <sub>4</sub> --	C4	C4	CB <sub>4</sub>	C4	1	1	1
SC <sub>5</sub> --	C5	C5	CB <sub>5</sub>	C5	1	1	1
SC <sub>6</sub> --	1	C6	CB <sub>6</sub>	1	1	C6	C6

\*In CODE ID<sub>2-0</sub> 011 the Check-Bit Latch is forced transparent; the Data Input Latch operates normally.

**Data Correction**

Tables 10 to 16 shows which data output bits are corrected (inverted) depending upon the syndromes and the CODE ID position. Note that the syndromes that determine data correction are in some cases syndromes input externally via the CB

inputs and in some cases syndromes generated internally by that EDC (S<sub>i</sub> are the internal syndromes and are the same as the value of the SC<sub>i</sub> output of that EDC if enabled).

The tables show the number of data bit inverted (corrected) if any for the CODE ID and syndrome combination.

**TABLE 10. CODE ID<sub>2-0</sub> = 000\***

		S5	0			1		
		S4	0			1		
		S3	0			1		
S2	S1							
0	0	-	-	-	5	-	11	14
0	1	-	1	2	6	8	12	-
1	0	-	-	3	7	9	13	15
1	1	-	0	4	-	10	-	-

\*Unlisted S combinations are no correction.

**TABLE 11. CODE ID<sub>2-0</sub> = 010\***

		CB <sub>6</sub>	0			1		
		CB <sub>5</sub>	1			0		
		CB <sub>4</sub>	0			1		
		CB <sub>3</sub>	0			1		
CB <sub>2</sub>	CB <sub>1</sub>							
0	0	-	11	14	-	-	-	5
0	1	8	12	-	-	-	1	2
1	0	9	13	15	-	-	-	3
1	1	10	-	-	-	-	0	4

\*Unlisted CB combinations are no correction.

**TABLE 12. CODE ID<sub>2-0</sub> = 011\***

		<b>S6</b>	0	0	0	0	1	1	1	1
		<b>S5</b>	0	0	0	0	1	1	1	1
		<b>S4</b>	0	0	1	1	0	0	1	1
		<b>S3</b>	0	1	0	1	0	1	0	1
<b>S2</b>	<b>S1</b>									
0	0		-	-	-	5	-	11	14	-
0	1		-	1	2	6	8	12	-	-
1	0		-	-	3	7	9	13	15	-
1	1		-	0	4	-	10	-	-	-

\*Unlisted S combinations are no correction.

**TABLE 13. CODE ID<sub>2-0</sub> = 100\***

		<b>CB<sub>0</sub></b>	0	0	0	0	1	1	1	1
		<b>CB<sub>6</sub></b>	0	0	0	0	1	1	1	1
		<b>CB<sub>5</sub></b>	1	1	1	1	0	0	0	0
		<b>CB<sub>4</sub></b>	0	0	1	1	0	0	1	1
		<b>CB<sub>3</sub></b>	0	1	0	1	0	1	0	1
<b>CB<sub>2</sub></b>	<b>CB<sub>1</sub></b>									
0	0		-	11	14	-	-	-	-	5
0	1		8	12	-	-	-	1	2	6
1	0		9	13	15	-	-	-	3	7
1	1		10	-	-	-	-	0	4	-

\*Unlisted CB combinations are no correction.

**TABLE 14. CODE ID<sub>2-0</sub> = 101\***

		<b>CB<sub>0</sub></b>	0	0	0	0	1	1	1	1
		<b>CB<sub>6</sub></b>	0	0	0	0	1	1	1	1
		<b>CB<sub>5</sub></b>	0	0	0	0	1	1	1	1
		<b>CB<sub>4</sub></b>	0	0	1	1	0	0	1	1
		<b>CB<sub>3</sub></b>	0	1	0	1	0	1	0	1
<b>CB<sub>2</sub></b>	<b>CB<sub>1</sub></b>									
0	0		-	-	-	5	-	11	14	-
0	1		-	1	2	6	8	12	-	-
1	0		-	-	3	7	9	13	15	-
1	1		-	0	4	-	10	-	-	-

\*Unlisted CB combinations are no correction.

**TABLE 15. CODE ID<sub>2-0</sub> = 110\***

		<b>CB<sub>0</sub></b>	0	0	0	0	1	1	1	1
		<b>CB<sub>6</sub></b>	1	1	1	1	0	0	0	0
		<b>CB<sub>5</sub></b>	0	0	0	0	1	1	1	1
		<b>CB<sub>4</sub></b>	0	0	1	1	0	0	1	1
		<b>CB<sub>3</sub></b>	0	1	0	1	0	1	0	1
<b>CB<sub>2</sub></b>	<b>CB<sub>1</sub></b>									
0	0		-	-	-	5	-	11	14	-
0	1		-	1	2	6	8	12	-	-
1	0		-	-	3	7	9	13	15	-
1	1		-	0	4	-	10	-	-	-

\*Unlisted CB combinations are no correction.

**TABLE 16. CODE ID<sub>2-0</sub> = 111\***

		<b>CB<sub>0</sub></b>	0	0	0	0	1	1	1	1
		<b>CB<sub>6</sub></b>	1	1	1	1	0	0	0	0
		<b>CB<sub>5</sub></b>	1	1	1	1	0	0	0	0
		<b>CB<sub>4</sub></b>	0	0	1	1	0	0	1	1
		<b>CB<sub>3</sub></b>	0	1	0	1	0	1	0	1
<b>CB<sub>2</sub></b>	<b>CB<sub>1</sub></b>									
0	0		-	11	14	-	-	-	-	5
0	1		8	12	-	-	-	1	2	6
1	0		9	13	15	-	-	-	3	7
1	1		10	-	-	-	-	0	4	-

\*Unlisted CB combinations are no correction.

## 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 5.)

The 16-bit configuration is shown in Figure 6.

### 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<sub>0-5</sub> (SC<sub>6</sub> is a logical one, or high).

Check bits are generated according to a modified Hamming code. Details of the code for check bit generation are contained in Table 17. 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.

Figure 1 shows the data flow in the Generate Mode.

### 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, ERROR goes LOW. If two or more errors are detected, MULT ERROR goes LOW. Both error indicators are HIGH if there are no errors.

Also available on device outputs SC<sub>0-5</sub> 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 18 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. (See Figure 2.) 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<sub>0-5</sub>. ERROR and MULT ERROR are forced HIGH in this mode.

### 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 19 shows the loading definitions for the DATA lines.

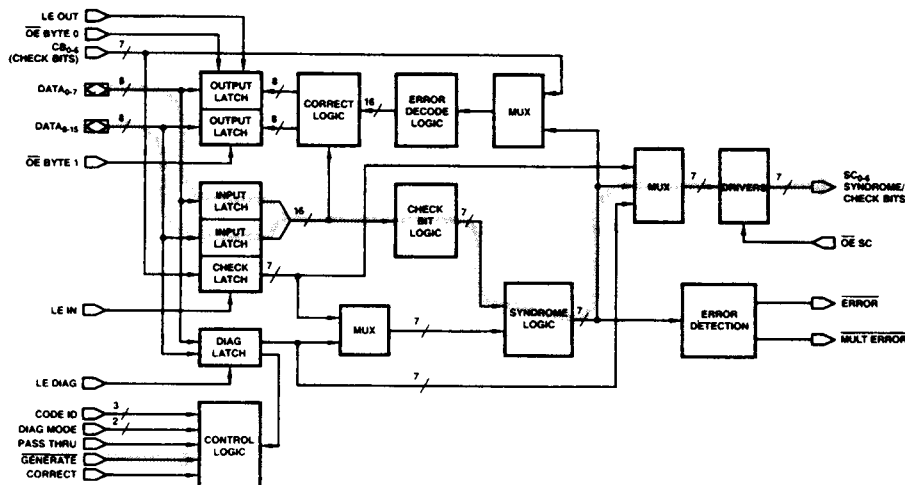
### Diagnostic Generate/Diagnostic Detect/Diagnostic Correct

These are special diagnostic modes selected by DIAG MODE<sub>0-1</sub> where either normal check bit inputs or outputs are substituted for by check bits loaded into the Diagnostic Latch (see Table 3 for details). Figures 3 and 4 illustrate the flow of data during the two diagnostic modes.

### Internal Control Mode

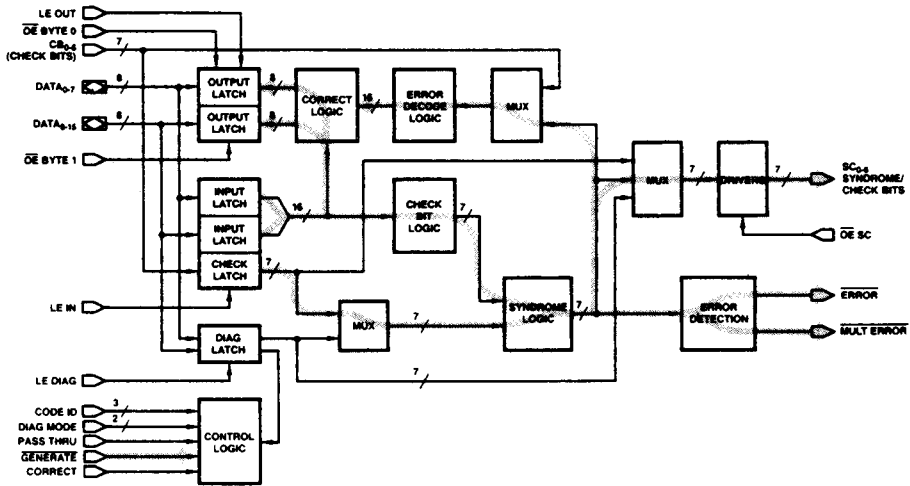
This mode is selected by CODE ID<sub>2-0</sub> input 001 (ID<sub>2</sub>, ID<sub>1</sub>, ID<sub>0</sub>). When in Internal Control Mode, the EDC takes the CODE ID<sub>2-0</sub>, DIAG MODE<sub>0-1</sub>, CORRECT and PASS THRU control signals from the internal Diagnostic Latch rather than from the external input lines.

Table 19 gives the format for loading the Diagnostic Latch.



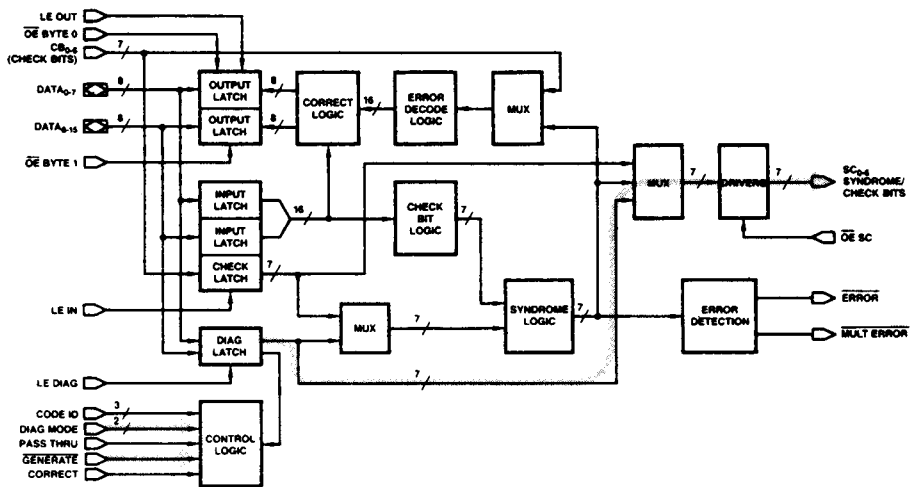
DF000280

Figure 1. Check Bit Generation Data Path



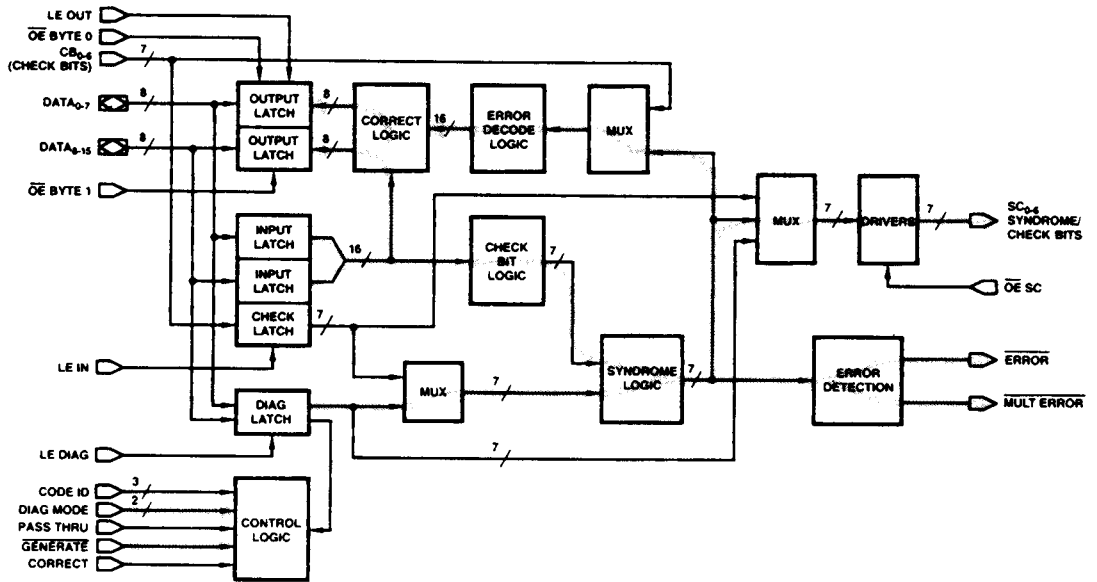
DF000280

Figure 2. Error Detection and Correction Data Path



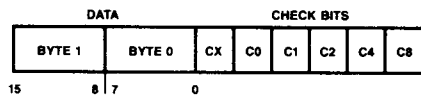
DF000280

Figure 3. Diagnostic Check Bit Generation Data Path



DF000280

Figure 4. Diagnostic Detect and Correct Data Path

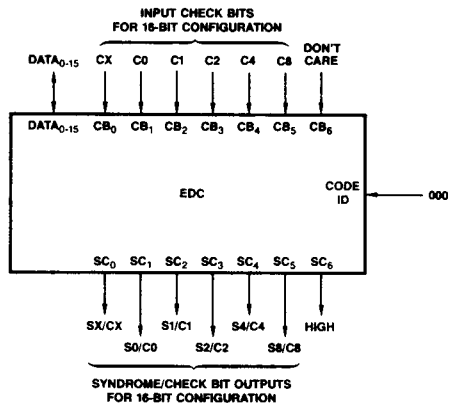


DF000220

Uses Modified Hamming Code 16/22

- 16 data bits
- 6 check bits
- 22 bits in total

Figure 5. 16-Bit Data Format



DF000210

Figure 6. 16-Bit Configuration

**SYNDROME DECODE  
TO BIT-IN-ERROR  
8-BIT MODE**

Syndrom Bits	S4 S0	S4 S2	0	1	0	1
0	0		*	C4	C2	5
0	1		C1	3	TM	7
1	0		C0	2	1	6
1	1		TM	4	0	TM

\* - no errors detected  
TM - two or more errors

**TABLE 17. 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	

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

**TABLE 18. SYNDROME DECODE TO BIT-IN-ERROR**

Syndrome Bits	S8	0	1	0	1	0	1	0	1
SX S0 S1	S4	0	0	1	1	0	0	1	1
	S2	0	0	0	0	1	1	1	1
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 - location of the single bit-in-error  
 T - two errors detected  
 M - three or more errors detected

**TABLE 19. 16-BIT DIAGNOSTIC LATCH LOADING 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

**32-BIT DATA WORD CONFIGURATION**

The 32-bit format consists of 32 data bits and 7 check bits and is referred to as 32/39 code (see Figure 7).

The 32-bit configuration is shown in Figure 8.

The upper EDC (Slice 0/1) handles the least significant bytes 0 and 1 - the external DATA lines 0 to 15 are connected to the same numbered inputs of the upper device. The lower EDC (Slice 2/3) handles the most significant bytes 2 and 3-the external DATA lines for bits 16 to 31 are connected to inputs DATA<sub>0</sub> through DATA<sub>15</sub> respectively.

The valid syndrome and check bit outputs are those of Slice 2/3 as shown in the diagram. In Correct Mode these must be read into Slice 0/1 via the CB inputs and are selected by the MUX as inputs to the bit-in-error decoder (see block diagram). This requires external buffering and output enabling of the check bit lines, as shown. The OE SC signal can be used to control enabling of check bit inputs - when syndrome outputs are enabled, the external check bit inputs will be disabled.

The valid ERROR and MULT ERROR outputs are those of the Slice 2/3. The ERROR and MULT ERROR outputs of Slice 0/1 are unspecified. All of the latch enables and control signals must be input to both of the devices.

**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<sub>0-6</sub> of Slice 2/3.

Check bits are generated according to a modified Hamming code. Details of the code for check bit generation are contained in Table 23. Check bits are generated as either an XOR or XNOR of 16 of the 32 data bits as indicated in the table. The XOR function results in an even parity check bit, the XNOR in 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, ERROR goes LOW. If two or more errors are detected, MULT ERROR goes LOW. Both error indicators are HIGH if there are no errors. The valid ERROR and MULT ERROR signals are those of Slice 2/3 - those of Slice 0/1 are undefined.

Also available on Slice 2/3 outputs SC<sub>0-6</sub> 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,



Also available on Slice 2/3 outputs SC<sub>0-6</sub> 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 20 gives the chart for decoding the syndrome bits generated for the 32-bit configuration (as an example, if the syndrome bits SX/S0/S1/S2/S4/S8/S16 were 0010011, this would be decoded to indicate that there is a single-bit error at data bit 25). If no error is detected, the syndrome bits will be all zeroes.

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

### 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 desired, this would be done by placing the device in Generate Mode to produce a correct check bit sequence for the data in the Data Input Latch.

For data correction, both Slices 0/1 and 2/3 require access to the syndrome bits on Slice 2/3's outputs SC<sub>0-6</sub>. Slice 2/3 has access to these syndrome bits through internal data paths, but for Slice 0/1 they must be read through the inputs CB<sub>0-6</sub>. The device connections for this are shown in Figure 8. When in Correct Mode, the SC outputs must be enabled so that they are available for reading in through the CB inputs.

### 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<sub>0-6</sub> of Slice 2/3. ERROR and MULT ERROR are forced HIGH in this mode.

### Internal Control Mode

This mode is selected by CODE ID<sub>2-0</sub> input 001 (ID<sub>2</sub>, ID<sub>1</sub>, ID<sub>0</sub>). When in Internal Control Mode, the EDC takes the CODE ID<sub>2-0</sub> DIAG MODE<sub>0-1</sub>, CORRECT and PASS THRU control signals from the Internal Diagnostic Latch rather than from the external input lines.

Table 22 gives the format for loading the Diagnostic Latch.

**TABLE 20.**  
**SYNDROME DECODE TO BIT-IN-ERROR**

Syndrome Bits	S16	S8	S4	0	1	0	1	0	1	0	1	0	1
SX	S0	S1	S2										
0	0	0	0	*	C16	C8	T	C4	T	T	T	30	
0	0	0	1	C2	T	T	27	T	5	M	T		
0	0	1	0	C1	T	T	25	T	3	15	T		
0	0	1	1	T	M	13	T	23	T	T	M		
0	1	0	0	C0	T	T	24	T	2	M	T		
0	1	0	1	T	1	12	T	22	T	T	M		
0	1	1	0	T	M	10	T	20	T	T	M		
0	1	1	1	16	T	T	M	T	M	M	T		
1	0	0	0	CX	T	T	M	T	M	14	T		
1	0	0	1	T	M	11	T	21	T	T	M		
1	0	1	0	T	M	9	T	19	T	T	31		
1	0	1	1	M	T	T	29	T	7	M	T		
1	1	0	0	T	M	8	T	18	T	T	M		
1	1	0	1	17	T	T	28	T	6	M	T		
1	1	1	0	M	T	T	26	T	4	M	T		
1	1	1	1	T	0	M	T	M	T	T	M		

\* - no errors detected

Number - number of the single bit-in-error

T - two errors detected

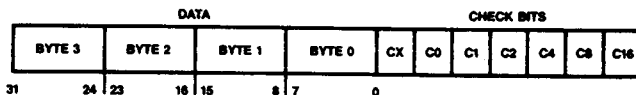
M - three or more errors detected

### 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 22 shows the loading definitions for the DATA lines.

### Diagnostic Generate/Diagnostic Detect/ Diagnostic Correct

These are special diagnostic modes selected by DIAG MODE<sub>0-1</sub> where either normal check bit inputs or outputs are substituted for by check bits from the Diagnostic Latch (see Table 2 for details). Figures 3 and 4 illustrate the flow of data during the two diagnostic modes.



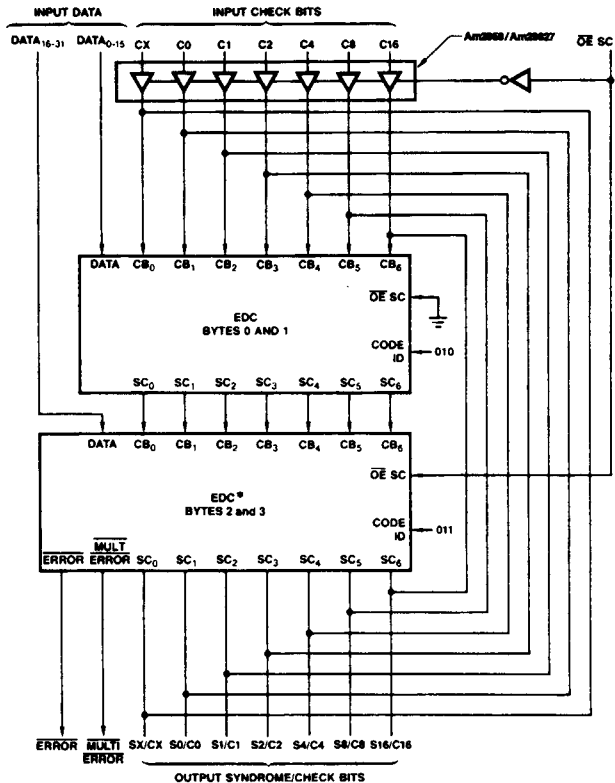
DF000460

- 32 data bits

Uses Modified Hamming Code 32/39  
- 7 check bits

- 39 bits in total

**Figure 7. 32-Bit Data Format**



DF000111

\*Check Bit Latch is Forced Transparent in this Code ID Combination for this Slice.

Figure 8. 32-Bit Configuration

TABLE 21.  
KEY AC CALCULATIONS FOR THE 32-BIT CONFIGURATION

32-Bit Propagation Delay		Component Delay from Am2960 AC Specifications
From	To	
DATA	Check Bits Out	(Data to SC) + (CB to SC, CODE ID 011)
DATA In	Corrected DATA Out	(DATA to SC) + (CB to SC, CODE ID 011) + (CB to DATA, CODE ID 010)
DATA	Syndromes Out	(DATA to SC) + (CB to SC, CODE ID 011)
DATA	ERROR for 32 Bits	(DATA to SC) + (CB to ERROR, CODE ID 011)
DATA	MULT ERROR for 32 Bits	(DATA to SC) + (CB to MULT ERROR, CODE ID 011)

**TABLE 22.**  
**32-BIT DIAGNOSTIC LATCH LOADING 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	Diagnostic Check Bit 16
7	Don't Care
8	Slice 0/1 - CODE ID 0
9	Slice 0/1 - CODE ID 1
10	Slice 0/1 - CODE ID 2
11	Slice 0/1 - DIAG MODE 0
12	Slice 0/1 - DIAG MODE 1
13	Slice 0/1 - CORRECT
14	Slice 0/1 - PASS THRU
15	Don't Care
16-23	Don't Care
24	Slice 2/3 - CODE ID 0
25	Slice 2/3 - CODE ID 1
26	Slice 2/3 - CODE ID 2
27	Slice 2/3 - DIAG MODE 0
28	Slice 2/3 - DIAG MODE 1
29	Slice 2/3 - CORRECT
30	Slice 2/3 - PASS THRU
31	Don't Care

**TABLE 23.**  
**32-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
C8	Even (XOR)									X	X	X	X		X	X	X
C16	Even (XOR)	X	X	X	X	X	X	X									

Generated Check Bits	Parity	Participating Data Bits															
		16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
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
C8	Even (XOR)									X	X	X	X		X	X	X
C16	Even (XOR)	X	X	X	X	X	X	X						X	X	X	X

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

## 64-BIT DATA WORD CONFIGURATION

The 64-bit format consists of 64 data bits, 8 check bits and is referred to as 64/72 code (see Figure 9.).

The configuration to process 64-bit format is shown in Figure 6. In this configuration, a portion of the syndrome generation and error detection is implemented externally of the EDCs in MSI. For error correction, the syndrome bits generated must be read back into all four EDCs through the CB inputs. This necessitates the check bit buffering shown in the connection diagram of Figure 10. The  $\overline{OE}$  SC signal can control the check bit enabling; when syndrome bit outputs are enabled, the external check bit lines will be disabled so that the syndrome bits may be read onto the CB inputs.

The error detection signals for the 64-bit configuration differ from the 16- and 32-bit configurations. The **ERROR** signal functions the same: it is LOW if one or more errors are detected, and HIGH if no errors are detected. The **DOUBLE ERROR** signal is HIGH if and only if a double-bit error is detected; it is LOW otherwise. All of the **MULT ERROR** outputs of the four devices are valid. **MULT ERROR** is LOW for all three **ERROR** cases and some **DOUBLE ERROR** combinations. (See TOME definition in Functional Equations section.) It is HIGH if either zero or one errors are detected.

This is a different meaning for **MULT ERROR** than in other configurations.

### Generate Mode

In this mode, check bits will be generated that correspond to the contents of the Data Input Latch. The check bits generated appear at the outputs of the XOR gates as indicated in Figure 10.

Check bits are generated according to a modified Hamming code. Details of the code for check bit generation are contained in Table 25. Check bits are generated as either an XOR or XNOR of 32 of the 64 bits as indicated in the table. The XOR function results in an even parity check bit, the XNOR in 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, **ERROR** goes LOW. If exactly two errors are detected, **DOUBLE ERROR** goes HIGH. If three or more errors are detected, **MULT ERROR** goes LOW – the **MULT ERROR** output of any of the four EDCs may be used.

Available as XOR gate outputs are the generated syndrome bits (see Figure 10). 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 26 gives the chart for encoding the syndrome bits generated for the 64-bit configuration (as an example, if the syndrome bits **SX/S1/S2/**

**S4/S8/S16/S32** were 00100101, this would be decoded to indicate that there is a single-bit error at data bit 41). 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 corrections.

### 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. Check bit correction 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.

To perform the correction step, all four slices require access to the syndrome bits which are generated externally of the devices. This access is provided by reading the syndrome bits in through the CB inputs, where they are selected as inputs to the bit-in-error decoder by the multiplexer (see Block Diagram). The device connections for this are shown in Figure 10. When in Correct Mode, the SC outputs must be enabled so that the syndrome bits are available at the CB inputs.

### 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 Check Bit Input Latch are passed through the external XOR network and appear inverted at the XOR gate outputs labeled CX to C32 (see Figure 10).

### 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 27 shows the loading definitions for the DATA lines.

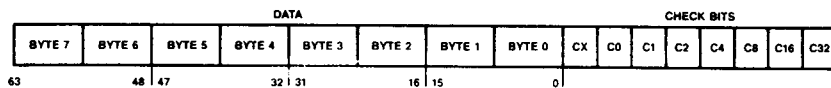
### Diagnostic Generate/Diagnostic Detect/ Diagnostic Correct

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 from the Diagnostic Latch. See Table 2 for details.

### Internal Control Mode

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

When in Internal Control Mode, the EDC takes the CODE  $ID_{2-0}$ , DIAG  $MODE_{0-1}$ , **CORRECT** and **PASS THRU** signals from the internal Diagnostic Latch rather than from the external control lines. Table 27 gives format for loading the Diagnostic Latch.



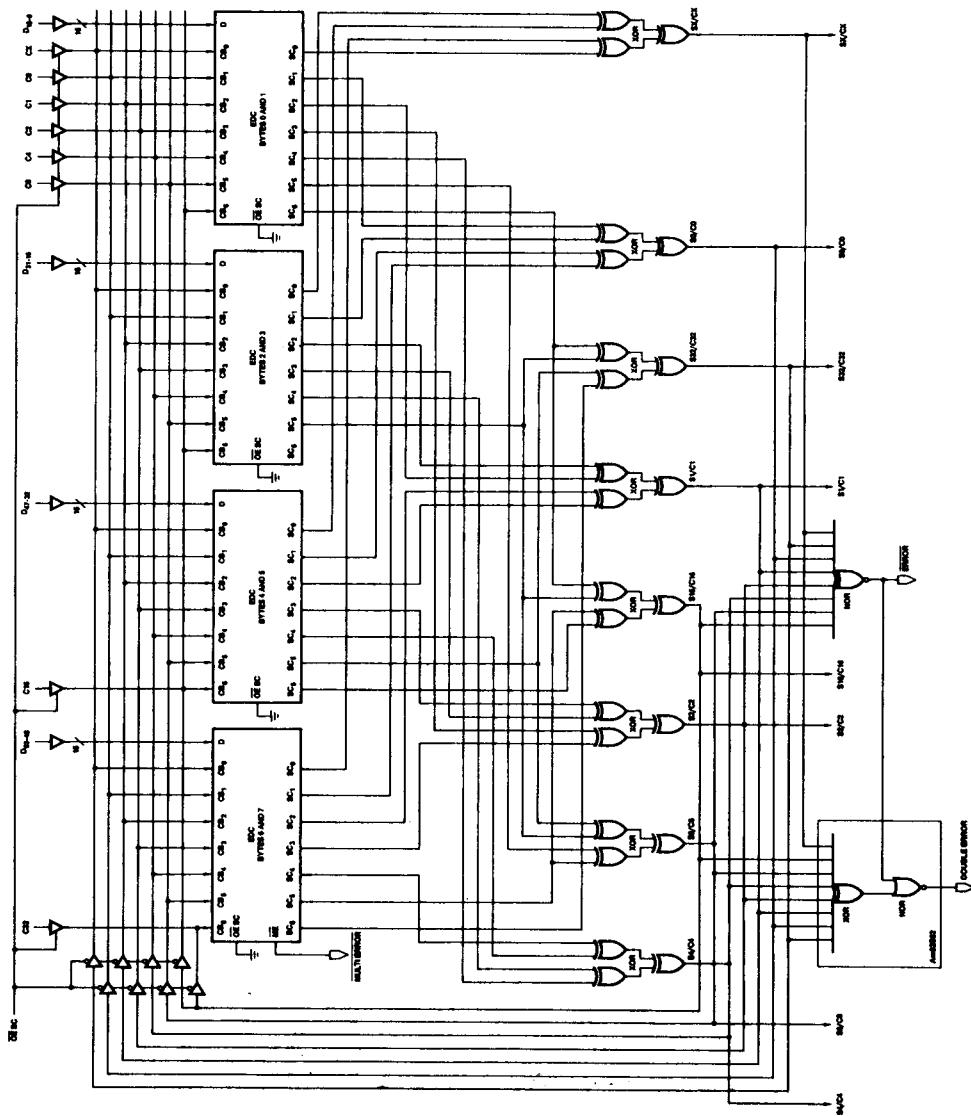
DF000920

– 64 data bits

Uses Modified Hamming Code 64/72  
– 8 check bits

– 72 bits in total

Figure 9. 64-Bit Data Format



BD001250

Notes: 1. In Pass Thru Mode the Contents of the Check Latch Appear on the XOR Outputs Inverted.  
 2. In Diagnostic Generate Mode the Contents of the Diagnostic Latch Appear on the XOR Outputs Inverted.

Figure 10. 64-Bit Data Configuration

**TABLE 24.  
KEY AC CALCULATIONS FOR THE 64-BIT CONFIGURATION**

64-Bit Propagation Delay		Component Delays from Am2960 AC Specifications, plus MSI
From	To	
DATA	Check Bits Out	(DATA to SC) + (XOR Delay)
DATA In	Corrected DATA Out	(DATA to SC) + (XOR Delay) + (Buffer Delay) + (CB to DATA, CODE ID 1xx)
DATA	Syndromes	(DATA to SC) + (XOR Delay)
DATA	ERROR for 64 Bits	(DATA to SC) + (XOR Delay) + (NOR Delay)
DATA	MULT ERROR for 64 Bits	(DATA to SC) + (XOR Delay) + (Buffer Delay) + (CB to MULT ERROR, CODE ID 1xx)
DATA	DOUBLE ERROR for 64 Bits	(DATA to SC) + (XOR Delay) + (XOR/NOR Delay)

**TABLE 25. 64-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		X	
C1	Odd (XNOR)	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
C8	Even (XOR)									X	X	X	X	X	X	X	X
C16	Even (XOR)	X	X	X	X	X	X	X	X								
C32	Even (XOR)	X	X	X	X	X	X	X	X								

Generated Check Bits	Parity	Participating Data Bits															
		16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
CX	Even (XOR)		X	X	X		X			X	X		X			X	
C0	Even (XOR)	X	X	X		X		X		X		X		X		X	
C1	Odd (XNOR)	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
C8	Even (XOR)									X	X	X	X	X	X	X	X
C16	Even (XOR)									X	X	X	X	X	X	X	X
C32	Even (XOR)									X	X	X	X	X	X	X	X

**TABLE 25. (Cont'd.)  
64-BIT MODIFIED HAMMING CODE - CHECK BIT ENCODE CHART**

Generated Check Bits	Parity	Participating Data Bits															
		32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
CX C0	Even (XOR)	X				X	X	X			X		X	X		X	
	Even (XOR)	X	X	X		X		X		X	X		X				
C1 C2	Odd (XNOR)	X			X	X		X		X	X			X		X	
	Odd (XNOR)	X	X				X	X	X			X		X	X		
C4 C8	Even (XOR)			X	X	X	X	X							X	X	
	Even (XOR)									X	X	X	X	X	X	X	
C16 C32	Even (XOR)	X	X	X	X	X	X	X									
	Even (XOR)	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	

Generated Check Bits	Parity	Participating Data Bits															
		48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
CX C0	Even (XOR)	X				X	X	X			X		X	X		X	
	Even (XOR)	X	X	X		X		X		X	X		X				
C1 C2	Odd (XNOR)	X			X	X		X		X	X			X		X	
	Odd (XNOR)	X	X				X	X	X			X		X	X		
C4 C8	Even (XOR)			X	X	X	X	X							X	X	
	Even (XOR)									X	X	X	X	X	X	X	
C16 C32	Even (XOR)									X	X	X	X	X	X	X	
	Even (XOR)	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	

The check bit is generated as either an XOR or XNOR of the 32 data bits noted by an "X" in the table.

**TABLE 26. 64-BIT SYNDROME DECODE TO BIT-IN-ERROR**

Syndrome Bits				S32	S16	S8	S4																						
SX	S0	S1	S2	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0	0	0	*	C32	C16	T	C8	T	T	M	C4	T	T	M	T	46	62	T										
0	0	0	1	C2	T	T	M	T	43	59	T	T	53	37	T	M	T	T	M										
0	0	1	0	C1	T	T	M	T	41	57	T	T	51	35	T	15	T	T	31										
0	0	1	1	T	M	M	T	13	T	T	29	23	T	T	7	T	M	M	T										
0	1	0	0	C0	T	T	M	T	40	56	T	T	50	34	T	M	T	T	M										
0	1	0	1	T	49	33	T	12	T	T	28	22	T	T	6	T	M	M	T										
0	1	1	0	T	M	M	T	10	T	T	26	20	T	T	4	T	M	M	T										
0	1	1	1	16	T	T	0	T	M	M	T	T	M	M	T	M	T	T	M										
1	0	0	0	CX	T	T	M	T	M	M	T	T	M	M	T	14	T	T	30										
1	0	0	1	T	M	M	T	11	T	T	27	21	T	T	5	T	M	M	T										
1	0	1	0	T	M	M	T	9	T	T	25	19	T	T	3	T	47	63	T										
1	0	1	1	M	T	T	M	T	45	61	T	T	55	39	T	M	T	T	M										
1	1	0	0	T	M	M	T	8	T	T	24	18	T	T	2	T	M	M	T										
1	1	0	1	17	T	T	1	T	44	60	T	T	54	38	T	M	T	T	M										
1	1	1	0	M	T	T	M	T	42	58	T	T	52	36	T	M	T	T	M										
1	1	1	1	T	48	32	T	M	T	T	M	M	T	T	M	T	M	M	T										

\* - no errors detected

Number - the number of the single bit-in-error

T - two errors detected

M - more than two errors detected

**TABLE 27. 64-BIT DIAGNOSTIC LATCH LOADING 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	Slice 0/1 - CODE ID 0
9	Slice 0/1 - CODE ID 1
10	Slice 0/1 - CODE ID 2
11	Slice 0/1 - DIAG MODE 0
12	Slice 0/1 - DIAG MODE 1
13	Slice 0/1 - CORRECT
14	Slice 0/1 - PASS THRU
15	Don't Care
16-23	Don't Care
24	Slice 2/3 - CODE ID 0
25	Slice 2/3 - CODE ID 1
26	Slice 2/3 - CODE ID 2
27	Slice 2/3 - DIAG MODE 0
28	Slice 2/3 - DIAG MODE 1
29	Slice 2/3 - CORRECT
30	Slice 2/3 - PASS THRU

Data Bit	Internal Function
31	Don't Care
32-37	Don't Care
38	Diagnostic Check Bit 16
39	Don't Care
40	Slice 4/5 - CODE ID 0
41	Slice 4/5 - CODE ID 1
42	Slice 4/5 - CODE ID 2
43	Slice 4/5 - DIAG MODE 0
44	Slice 4/5 - DIAG MODE 1
45	Slice 4/5 - CORRECT
46	Slice 4/5 - PASS THRU
47	Don't Care
48-54	Don't Care
55	Diagnostic Check Bit 32
56	Slice 6/7 - CODE ID 0
57	Slice 6/7 - CODE ID 1
58	Slice 6/7 - CODE ID 2
59	Slice 6/7 - DIAG MODE 0
60	Slice 6/7 - DIAG MODE 1
61	Slice 6/7 - CORRECT
62	Slice 6/7 - PASS THRU
63	Don't Care

**APPLICATIONS**

**System Design Considerations**

**High Performance Parallel Operation**

For maximum memory system performance, the EDC should be used in the Check-Only configuration shown in Figure 11. With this configuration, the memory system operates as fast with EDC as it would without.

On reads from memory, data is read out from the RAMs directly to the data bus (same as in a non-EDC system). At the same time, the data is read into the EDC to check for errors. If an error exists, the EDC's error flags are used to interrupt the CPU and/or to stretch the memory cycle. If no error is detected, no slowdown is required.

If an error is detected, the EDC generates corrected data for the processor. At the designer's option the correct data may be written back into memory; error logging and diagnostic routines may also be run under processor control.

The Check-Only configuration allows data reads to proceed as fast with EDC as without. Only if an error is detected is there any slowdown. But even if the memory system had an error every hour this would mean only one error every 3-4 billion memory cycles. So even with a very high error rate, EDC in a Check-Only configuration has essentially zero impact on memory system speed.

On writes to memory, check bits must be generated before the full memory word can be written into memory. But using the Am2961/62 Data Bus Buffers allows the data word to be buffered on the memory board while check bits are generated.

This makes the check bit generate time transparent to the processor.

**EDC in the Data Path**

The simplest configuration for EDC is to have the EDC directly in the data path, as shown in Figure 12 (Correct-Always Configuration). In this configuration, data read from memory is always corrected prior to putting the data on the data bus. The advantages are simpler operation and no need for mid-cycle interrupts. The disadvantage is that memory system speed is slowed by the amount of time it takes for error correction on every cycle.

Usually the Correct-Always Configuration will be used with MOS microprocessors which have ample memory timing budgets. Most high performance processors will use the high performance parallel configuration shown in Figure 11 (Check-Only Configuration).

**Scrubbing Avoids Double Errors**

Single-bit errors are by far the most common in a memory system, and are always correctable by the EDC.

Double-bit memory errors are far less frequent than single-bit (50 to 1, or 100 to 1) and are always detected by the EDC, but not corrected.

In a memory system, soft errors occur only one at a time. A double-bit error in a data word occurs when a single soft error is left uncorrected and is followed by another error in the data word hours, days, or weeks after the first.



"Scrubbing" memory periodically avoids almost all double-bit errors. In the scrubbing operation, every data word in memory is periodically checked by the EDC for single-bit errors. If one is found, it is corrected and the data word written back into memory. Errors are not allowed to pile up, and most double-bit errors are avoided.

The scrubbing operation is generally done as a background routine when the memory is not being used by the processor. If memory is scrubbed frequently, errors that are detected and corrected during processor accesses need not be immediately written back into memory. Instead, the error will be corrected

in memory during scrubbing. This reduces the time delay involved in a processor access of an incorrect memory word.

### Correction of Double-Bit Errors

In some cases, double-bit memory errors can be corrected! This is possible when one of the two bit errors is a hard error.

When a double bit error is detected, the data word should be checked to determine if one of the errors is a hard error. If so, the hard error bit may be corrected by inverting it, leaving only a single, correctable error. The time for this operation is negligible, since it will occur infrequently.

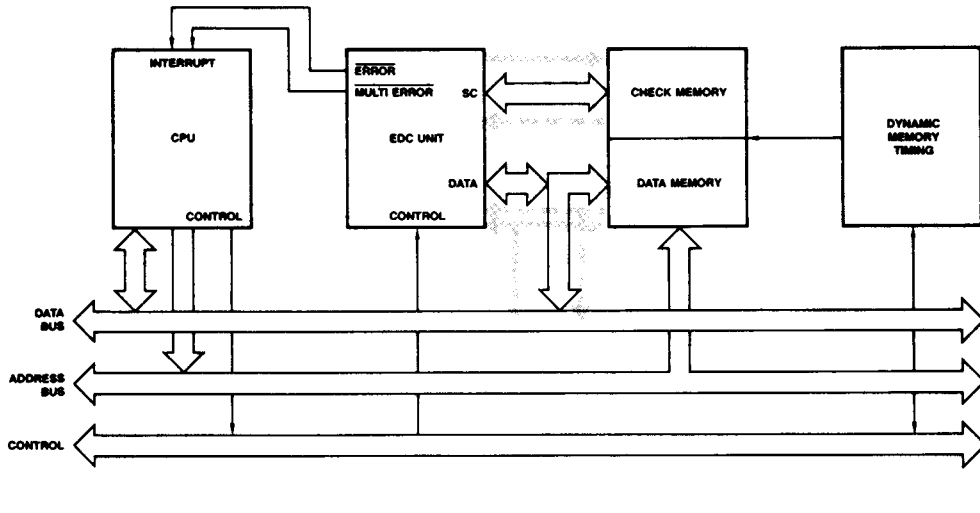


Figure 11. Check-Only Configuration

DF000291

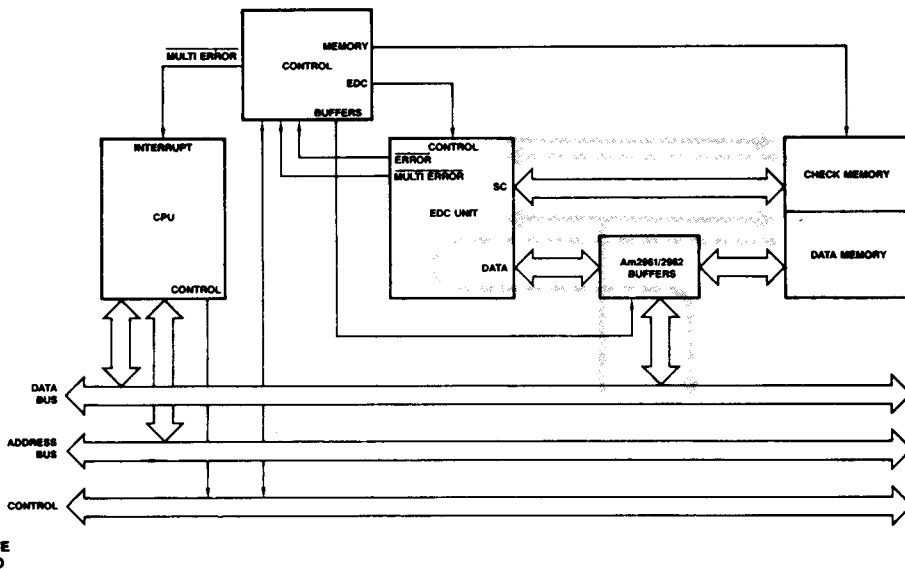


Figure 12. Correct-Always Configuration

DF000910

The procedure after detection of a double error is as follows:

- Invert the data bits read from memory.
- Write the inverted data back into the same memory word.
- Re-read the memory location and XOR the newly read out value with the old. If there is no hard error, then the XOR result will be all 1's. If there is a hard error, it will have the same bit value regardless of what was written in. So it will show as a 0 after the XOR operation.
- Invert the hard error bit (this will "correct" it) leaving only one error in the data.
- The EDC can then correct the single bit error.
- Rewrite the correct data word into memory. This does not change the hard error, but does eliminate the soft error. So the next memory access will find only a single-bit, correctable error.

### Example of Double-Bit Error Correction When One is a Hard Error

- |   |                                  |                        |
|---|----------------------------------|------------------------|
| 1. Data Read from Memory (D <sub>1</sub> )  | 16 data bits<br>1111111100000011 | 6 check bits<br>011010 |
| 2. EDC detects a multiple error.<br>Syndromes: 011000   |                                  |                        |
| 3. Syndrome decode indicates a double-bit error.  |                                  |                        |
| 4. Invert the bits read from memory (D <sub>1</sub> )   | 0000000011111100                 | 100101                 |
| 5. Write D <sub>1</sub> back to the same memory location  |                                  |                        |
| 6. Read back the memory location (D <sub>2</sub> )  | 0000000011111101                 | 100101                 |
| 7. XOR D <sub>1</sub> and D <sub>2</sub>  | 1111111111111110                 | 111111                 |
| 8. So the last data bit is the hard error. Use this to modify D <sub>1</sub>  |                                  |                        |
| 9. Pass the modified D <sub>1</sub> through the EDC. The EDC detects a single bit correctable error and outputs corrected data: | 1111111100000000                 | 011010                 |
| 10. Write the corrected data back to memory to fix the soft error.  |                                  |                        |

### Error Logging and Preventative Maintenance

The effectiveness of preventative maintenance can be increased by logging information on errors detected by the EDC. This is called error logging.

The EDC provides syndromes when errors are detected. The syndromes indicate which bit is in error. In most memory systems, each individual RAM supplies only one bit of the memory word. So the syndrome and data word address specify which RAM was in error.

Typically a permanent/hard RAM failure is preceded by a period of time where the RAM displays an increasing frequency of intermittent, soft errors. Error logging statistic can be used to detect an increasing intermittent error frequency so that the RAM can be replaced before a permanent failure occurs.

Error logging also records the location of already hard failed RAMs. With EDC a hard failure will not halt system operation. EDC can always correct single-bit errors even if it is a hard error. EDC can also correct double-bit errors where one is hard and one soft (see "Correction of Double Bit Errors" Section). The ability to continue operation despite hard errors can greatly reduce the need for emergency field maintenance. The hard-failed RAMs can be instead replaced at low cost during a regularly-scheduled preventative maintenance session.

### Reducing Check-Bit Overhead

Memory word widths need not be the same as the data word width of the processor. There is a substantial reduction in check bit overhead if wider memory words are used:

Memory Word		Check Bit Overhead
# Data Bits	# Check Bits	
8	5	38%
16	6	27%
32	7	14%
64	8	11%

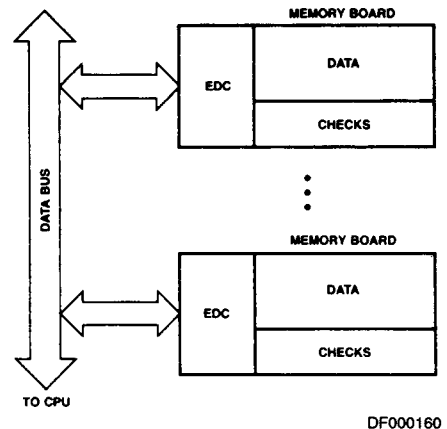


Figure 13. EDC Per Board

This reduction in check bit overhead lowers cost and increases the amount of data that can be packed on to each board.

The trade-off is that when writing data pieces into memory that are narrower than the memory word width, more steps are

required. These steps are exactly the same as those described in Byte Write in the Applications section. No penalty exists for reads from memory.

### EDC Per Board vs. EDC Per System

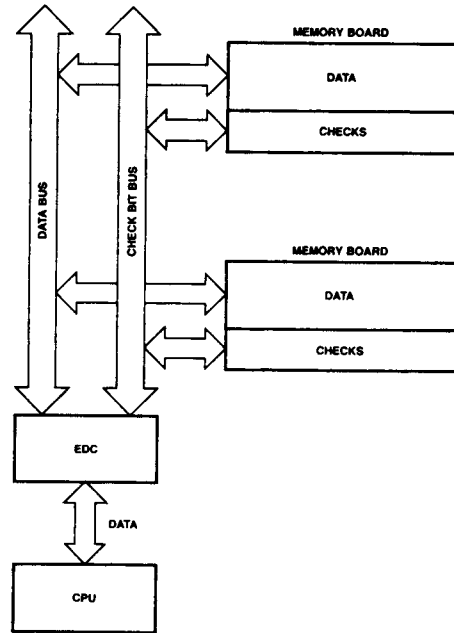
The choice of an EDC per system or per board depends on the economics and the architecture of the system.

Certainly the cheaper approach is to have only one EDC per system and this is a viable solution if only one memory location is accessed at a time.

This solution does require that the system has both data and check bit lines (see Figure 14). This makes retrofitting a system difficult and creates complications if static or ROM memory, which do not require check bits, are mixed in with dynamic RAM.

If the system has an advanced architecture, it is quite likely that it is necessary to simultaneously access memory locations on different memory boards (see Figure 13). Architectural features that require this are interleaved memory, cache memory, and DMA that is done simultaneously with processor memory accesses. EDC per board is a simpler system from a design standpoint.

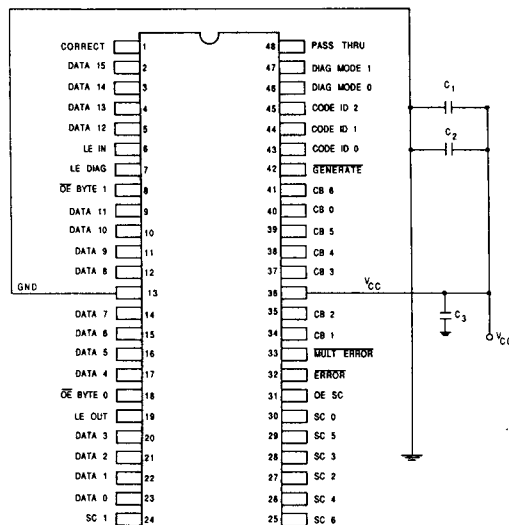
The EDC is designed to work efficiently in either the per system or per board configurations.



DF000150

Figure 14. EDC Per System

### DEVICE DECOUPLING VCC AND GROUND PIN CONNECTIONS



TC003940

Note: 1.  $C_1 = 1.0 \mu\text{F}$ ,  $C_2 = C_3 = 0.01 \mu\text{F}$

The  $C_1$ ,  $C_2$ , and  $C_3$  capacitors should be used to shunt Low - and high-frequency noise from  $V_{CC}$ . Do not replace with one capacitor. Place capacitors as close to the device as possible.

## DESIGNING FOR A FASTER SYSTEM ENVIRONMENT

### Am2960A Designs

When using the Am2960A in a high performance memory system, certain MSI glue logic will not be fast enough. The following device conversions are therefore recommended:

Am2960/Am2960-1 Design	Am2960A Design/Upgrade
EDC Bus Buffer: Am2961/Am2962	EDC Bus Buffer: Am29845/Am29846
Cascaded Design, CB <sub>0-6</sub> Driver: Am2958	Cascaded Design, CB <sub>0-6</sub> Driver: Am29827/Am29828

### Am2960/Am2960-1 to Am2960A Design Upgrades

When upgrading a system from the Am2960/Am2960-1 to the Am2960A special care must be taken to ensure that MSI glue logic specifications are not violated. For example, if latches are used in the data path, set up and hold times of the latches must also be upgraded to match the performance of the Am2960A. This may require replacement of currently designed-in latches to faster versions or alternate solutions.

A set up and hold time specification of 2.5 ns or less is recommended for these latches.

#### Byte Write

Byte operations are increasingly common for 16-bit and 32-bit processors. These complicate memory operations because check bits are generated for a complete 16-bit or 32-bit or 64-bit memory word – not for a single byte.

To write a byte into memory with EDC requires the following steps:

- Latch the byte into the Am2961/62 bus buffers (Figure 15)
- Read the complete data word from memory (Figure 15)
- Correct the complete data word if necessary (Figure 15)
- Insert the byte to be written into the data word (Figure 16)
- Generate new check bits for the entire data word (Figure 16)
- Store the data word back into memory (Figure 16)

(In fact, these steps must be taken for any piece of data being written into memory that is not as wide as a full memory word.)

The Am2960 EDC is designed with the intent of keeping byte operations simple in EDC systems. The EDC has separate output enables for each byte in the Data Output Latch. As shown in Figures 15 and 16, this allows the data word to be read from memory, the new byte to be inserted among the old, and new check bits to be generated using less time and less hardware than if separate byte enables were not available.

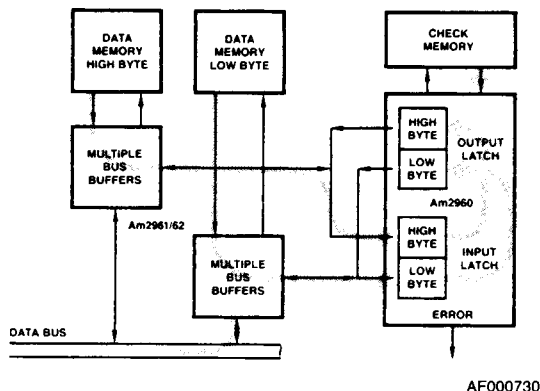


Figure 15. Byte Write, Phase 1: Read Out the Old Word and Correct

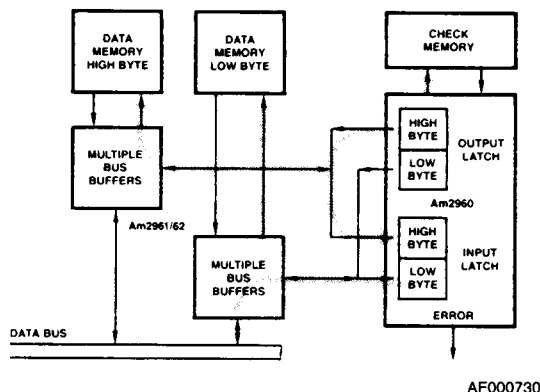


Figure 16. Byte Write, Phase 2: Insert the New Byte, Generate Checks and Write into Memory

## Diagnostics

EDC is used to boost the reliability of the overall system. It is necessary to also be able to check the operation of the EDC itself. For this reason the EDC has an internal control mode, a diagnostic latch, and two diagnostic modes.

To check that the EDC is functioning properly, the processor can put the EDC under software control by setting CODE ID<sub>2-0</sub> to 001. This puts the EDC into Internal Control Mode. In Internal Control Mode the EDC is controlled by the contents of the Diagnostic Latch which is loaded from the DATA inputs under processor control.

The EDC is set into CORRECT Mode. The processor loads in a known set of check bits into the Diagnostic Latch, a known set of data bits into the Data In Latch, and forces data errors. The output of the EDC (syndromes, error flags, corrected data) is then compared against the expected responses. By exercising the EDC with a string of data/check combinations and comparing the output against the expected responses, the EDC can be fully checked out.

### Eight Bit Data Word

Eight bit MOS microprocessors can use EDC too. Only five check bits are required. The EDC configuration for eight bits is shown in Figure 17. It operates as does the normal 16-bit configuration with the upper byte fixed at 0.

Check bit overhead for 8-bit data words can be reduced two ways. See the sections "Single Error Correction Only" and "Reducing Check Bit Overhead."

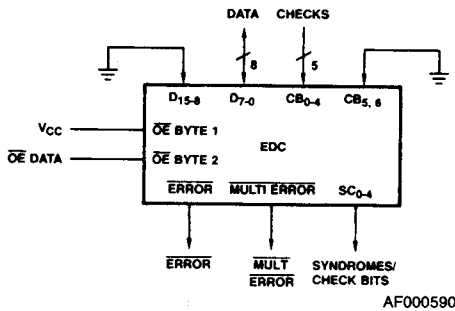


Figure 17. 8-Bit Configuration

### Other Word Widths

EDC on data words other than 8, 16, 32, or 64 bits can be accomplished with the AM2960. In most cases the extra data bits can be forced to a constant, and EDC will proceed as normal. For example, a 24-bit data word is shown in Figure 18.

### Single Error Correction Only

The EDC normally corrects all single-bit errors and detects all double-bit and some triple-bit errors. To save one check bit per word, the ability to detect double bit errors can be sacrificed – single errors are still detected and corrected.

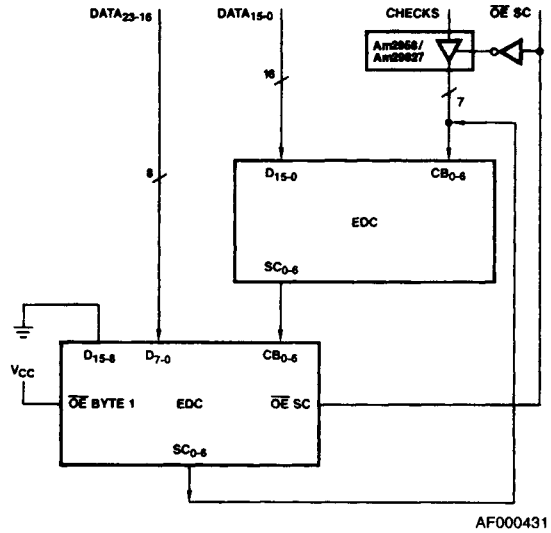


Figure 18. 24-Bit Configuration

Data Bits	Check Bits Required	
	Single Error Correction Only	Single Error Correct & Double Error Detect
8	4	5
16	5	6
32	6	7
64	7	8

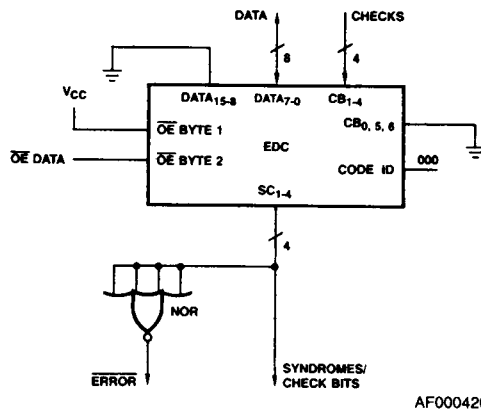


Figure 19. 8-Bit Single Error Correction Only

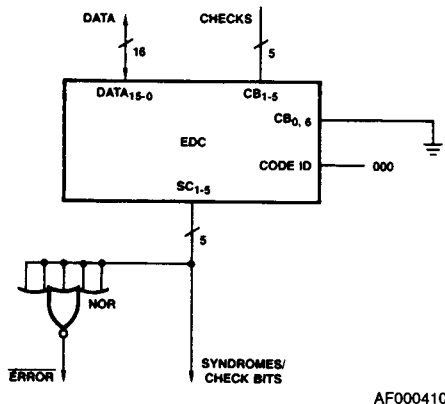
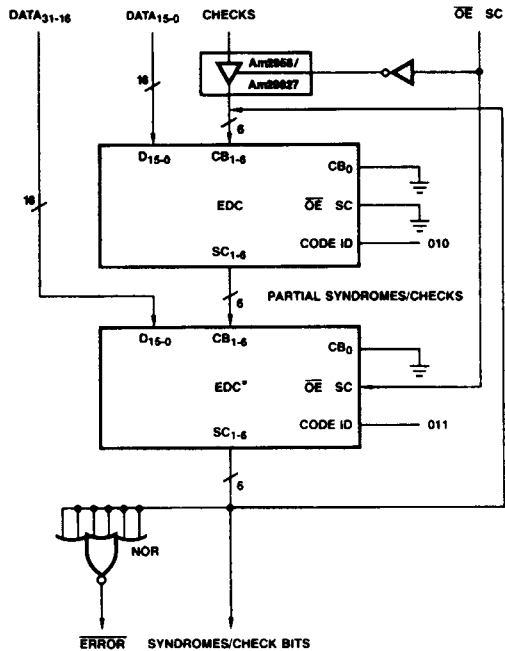


Figure 20. 16-Bit Single Error Correction Only



\* The Code ID Combination for this Slice Forces the Check Bit Latch Transparent.

Figure 21. 32-Bit Single Correction Only

Figures 19, 20, 21, 22 show single error correction only configurations for 8, 16, 32, and 64-bit data words respectively.

### Check Bit Correction

The EDC detects single bit errors whether the error is a data bit or a check bit. Data bit errors are automatically corrected by the EDC. To generate corrected check bits once a single check bit error is detected, the EDC need only be switched to GENERATE mode (data in the DATA INPUT LATCH is valid).

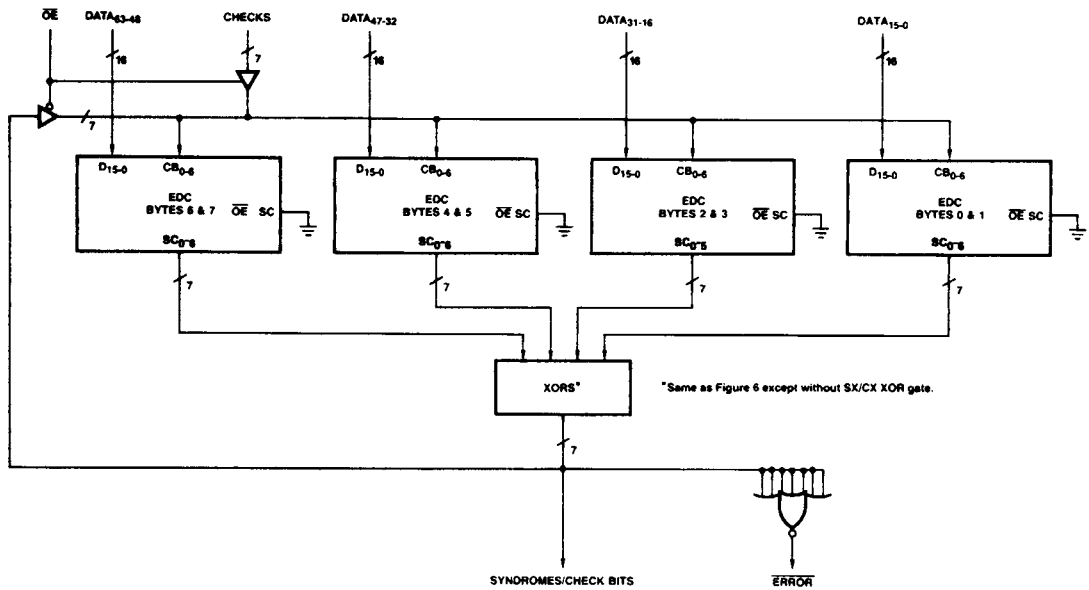
The syndromes generated by the EDC may be decoded to determine whether the single bit error is a check bit.

In many memory systems, a check bit error will be ignored on the memory read and corrected during a periodic "scrubbing" of memory (see section in System Design Considerations).

### Multiple Errors

The bit-in-error decode logic uses syndrome bits S0 through S32 to correct errors, SX is only used in developing the multiple error signal. This means that some multiple errors will cause a data bit to be inverted.

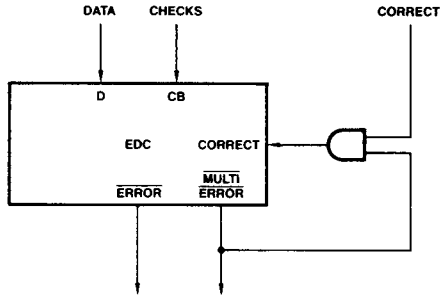
For example, in the 16-bit mode if data bits 8 and 13 are in error the syndrome 111100 (SX, S0, S1, S2, S4, S8) is produced. This is flagged a double error by the error detection logic, but the bit-in-error decoder only receives syndrome 11100 (S0, S1, S2, S4, S8), which it decodes as a single error in data bit 0 and inverts that bit. If it is desired to inhibit this inversion, the multiple error output may be connected to the correct input as in Figure 23. This will inhibit correction when a multiple error occurs. Extra time delay may be introduced in the data to correct data path when this is done.



AF000602

- Notes: 1. In Pass Thru Mode the Contents of the Check Latch Appear on the XOR Outputs Inverted.  
 2. In Diagnostic Generate Mode the Contents of the Diagnostic Latch appear on the XOR Outputs Inverted.

**Figure 22. 64-bit Single Error Correction Only**



AF000530

**Figure 23. Inhibition of Data Modification**

## ABSOLUTE MAXIMUM RATINGS

Storage Temperature .....	-65°C to +150°C
Temperature (Case)	
Under Bias .....	-55°C to +125°C
Supply Voltage to Ground Potential	
Continuous .....	-0.5 V to +7.0 V
DC Voltage Applied to Outputs For	
High Output State .....	-0.5 V to V <sub>CC</sub> Max.
DC Input Voltage .....	-0.5 V to +5.5 V
DC Input Current .....	-30 mA to +5.0 mA

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

## OPERATING RANGES

Commercial (C) Devices	
Ambient Temperature (T <sub>A</sub> ) .....	0°C to +70°C
Supply Voltage .....	+4.75 V to +5.25 V
Military (M) Devices	
Case Temperature (T <sub>C</sub> ) .....	-55°C to +125°C
Supply Voltage .....	+4.5 V to +5.5 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

**DC CHARACTERISTICS** over operating ranges unless otherwise specified (for APL Products, Group A, Subgroups 1, 2, 3 are tested unless otherwise noted)

Parameter Symbol	Parameter Descriptions	Test Conditions (Note 1)		Min.	Max.	Unit
V <sub>OH</sub>	Output HIGH Voltage	V <sub>CC</sub> = Min., V <sub>IN</sub> = V <sub>IH</sub> or V <sub>IL</sub>	I <sub>OH</sub> = -0.8 mA	COM'L	2.4	V
				MIL	2.4	
V <sub>OL</sub>	Output LOW Voltage	V <sub>CC</sub> = Min., V <sub>IN</sub> = V <sub>IH</sub> or V <sub>IL</sub>	I <sub>OL</sub> = 8 mA		0.5	V
V <sub>IH</sub>	Input HIGH Voltage	Guaranteed Input Logical HIGH Voltage for all Inputs (Note 5)		2.0		V
V <sub>IL</sub>	Input LOW Voltage	Guaranteed Input Logical LOW Voltage for all Inputs (Note 5)			0.8	V
V <sub>I</sub>	Input Clamp Voltage	V <sub>CC</sub> = Min., I <sub>IN</sub> = -18 mA			-1.5	V
I <sub>IL</sub>	Input LOW Current	V <sub>CC</sub> = Max. V <sub>IN</sub> = 0.5 V	DATA <sub>0-15</sub> (Note 3)		-410	μA
			All Other Inputs		-360	
I <sub>IH</sub>	Input HIGH Current	V <sub>CC</sub> = Max. V <sub>IN</sub> = 2.7V	DATA <sub>0-15</sub> (Note 3)		70	μA
			All Other Inputs		50	
I <sub>I</sub>	Input HIGH Current	V <sub>CC</sub> = Max., V <sub>IN</sub> = 5.5 V			1.0	mA
I <sub>OZH</sub> I <sub>OZL</sub>	Off State (High Impedance) Output Current	V <sub>CC</sub> Max.	DATA <sub>0-15</sub>	V <sub>O</sub> = 2.4	70	μA
				V <sub>O</sub> = 0.5	-410	
			SC <sub>0-6</sub>	V <sub>O</sub> = 2.4	50	
				V <sub>O</sub> = 0.5	-50	
I <sub>OS</sub>	Output Short-Circuit Current (Note 2)	V <sub>CC</sub> = V <sub>CC</sub> Max. + 0.5 V, V <sub>O</sub> = 0.5 V	Am2960/2960-1	-25	-85	mA
			Am2960A	-20		
I <sub>CC</sub>	Power Supply Current (Note 4)	V <sub>CC</sub> = Max.	C Devices	Am2960/2960-1	400	mA
				Am2960A	365	
			M Devices	Am2960	400	mA
				Am2960A	400	
		V <sub>CC</sub> = 5.0 V	T <sub>A</sub> = 25°C	Am2960/2960-1	290	mA
Am2960A	275					

- Notes: 1. For conditions shown as Min. or Max., use the appropriate value specified under Operating Range for the applicable device type.  
 2. Not more than one output should be shorted at a time. Duration of the short-circuit test should not exceed one second.  
 3. These are three-state outputs internally connected to TTL inputs. Input Characteristics are measured with outputs disabled (High Impedance).  
 4. Worst case I<sub>CC</sub> is at minimum temperature. Typical I<sub>CC</sub> (V<sub>CC</sub> = 5.0 V, T<sub>A</sub> = 25°C) represents nominal units and is not tested.  
 5. These input levels provide zero noise immunity and should only be tested in a static, noise-free environment.



## SWITCHING CHARACTERISTICS over COMMERCIAL operating range (Notes 1 and 2)

The following table specifies the guaranteed device performance over the commercial operating range of 0°C to +70°C (ambient), with  $V_{CC}$  4.75 to 5.25 V. All inputs switching is between 0 V and 3 V at 1 V/ns and measurements are made at 1.5 V. All outputs have maximum DC load. All units are in nanoseconds (ns).

No.	Param.	Data Path Description		Am2960		Am2960-1		Am2960A	
		From Input	To Output	Min.	Max.	Min.	Max.	Min.	Max.
1	t <sub>PD</sub>	DATA <sub>0-15</sub> (Note 3)	SC <sub>0-6</sub>		32		28		25
			DATA <sub>0-15</sub>		65		52		30
			ERROR		32		25		18
			MULT ERROR		50		50		23
2	t <sub>PD</sub>	CB <sub>0-6</sub> (CODE ID <sub>2-0</sub> 000, 011)	SC <sub>0-6</sub>		28		23		18
			DATA <sub>0-15</sub>		56		50		25
			ERROR		29		23		15
			MULT ERROR		47		47		20
3	t <sub>PD</sub>	CB <sub>0-6</sub> (CODE ID <sub>2-0</sub> 010, 100, 101, 110, 111)	SC <sub>0-6</sub>		28		28		18
			DATA <sub>0-15</sub>		45		34		23
			ERROR		29		29		15
			MULT ERROR		34		34		20
4	t <sub>PD</sub>	GENERATE	SC <sub>0-6</sub>		35		35		27
			DATA <sub>0-15</sub>		63		63		33
			ERROR		36		36		18
			MULT ERROR		55		55		23
5	t <sub>PD</sub>	CORRECT (Not Internal Control Mode)	SC <sub>0-6</sub>		-		-		-
			DATA <sub>0-15</sub>		45		45		24
			ERROR		-		-		-
			MULT ERROR		-		-		-
6	t <sub>PD</sub>	DIAG MODE (Not Internal Control Mode)	SC <sub>0-6</sub>		50		50		26
			DATA <sub>0-15</sub>		78		78		31
			ERROR		59		59		17
			MULT ERROR		75		75		23
7	t <sub>PD</sub>	PASS THRU (Not Internal Control Mode)	SC <sub>0-6</sub>		36		36		19
			DATA <sub>0-15</sub>		44		44		23
			ERROR		29		29		15
			MULT ERROR		46		46		18
8	t <sub>PD</sub>	CODE ID <sub>2-0</sub>	SC <sub>0-6</sub>		61		61		29
			DATA <sub>0-15</sub>		90		90		34
			ERROR		60		60		24
			MULT ERROR		80		80		27

Notes: See notes following end of tables continued on next page.

**SWITCHING CHARACTERISTICS** over **COMMERCIAL** operating range (Notes 1 and 2) (Cont'd.)

No.	Param.	Data Path Description		Am2960		Am2960-1		Am2960A	
		From Input	To Output	Min.	Max.	Min.	Max.	Min.	Max.
9	t <sub>PD</sub>	LE IN (From latched to transparent)	SC <sub>0-6</sub>		39		39		21
			DATA <sub>0-15</sub>		72		72		27
			ERROR		39		39		18
			MULT ERROR		59		59		21
10	t <sub>PD</sub>	LE OUT (From latched to transparent)	SC <sub>0-6</sub>		-		-		-
			DATA <sub>0-15</sub>		31		31		19
			ERROR		-		-		-
			MULT ERROR		-		-		-
11	t <sub>PD</sub>	LE DIAG (From latched to transparent; Not Internal Control Mode)	SC <sub>0-6</sub>		45		45		20
			DATA <sub>0-15</sub>		78		78		27
			ERROR		45		45		18
			MULT ERROR		65		65		21
12	t <sub>PD</sub>	Internal Control Mode: LE DIAG (From latched to transparent)	SC <sub>0-6</sub>		67		67		29
			DATA <sub>0-15</sub>		96		96		30
			ERROR		66		66		24
			MULT ERROR		86		86		27
13	t <sub>PD</sub>	Internal Control Mode: DATA <sub>0-15</sub> (Via Diagnostic latch)	SC <sub>0-6</sub>		67		67		28
			DATA <sub>0-15</sub>		96		96		35
			ERROR		66		66		24
			MULT ERROR		86		86		27
14	t <sub>SET</sub>	DATA <sub>0-15</sub> (Notes 4, 5)	LE IN		6		6		3
15	t <sub>HOLD</sub>				7		7		5
16	t <sub>SET</sub>	CB <sub>0-6</sub> (Notes 4, 5)			5		5		3
17	t <sub>HOLD</sub>				6		6		4
18	t <sub>SET</sub>	DATA <sub>0-15</sub> (Notes 4, 5)	LE OUT		44		34		20
19	t <sub>HOLD</sub>				5		5		4
20	t <sub>SET</sub>	CB <sub>0-6</sub> (Notes 4, 5) (CODE ID 000, 011)			35		35		15
21	t <sub>HOLD</sub>				0		0		0
22	t <sub>SET</sub>	CB <sub>0-6</sub> (Notes 4, 5) (CODE ID 010, 100, 101, 110, 111)			27		27		15
23	t <sub>HOLD</sub>				0		0		0
24	t <sub>SET</sub>	GENERATE (Notes 4, 5)			42		42		20
25	t <sub>HOLD</sub>				0		0		0
26	t <sub>SET</sub>	CORRECT (Notes 4, 5)			26		26		15
27	t <sub>HOLD</sub>				1		1		1
28	t <sub>SET</sub>	DIAG MODE (Notes 4, 5)			69		69		20
29	t <sub>HOLD</sub>				0		0		0
30	t <sub>SET</sub>	PASS THRU (Notes 4, 5)			26		26		7
31	t <sub>HOLD</sub>				0		0		0
32	t <sub>SET</sub>	CODE ID <sub>2-0</sub> (Notes 4, 5)			81		81		22
33	t <sub>HOLD</sub>				0		0		0

Notes: See notes following table continued on next page.

**SWITCHING CHARACTERISTICS** over **COMMERCIAL** operating range (Notes 1 and 2) (Cont'd.)

No.	Param.	Data Path Description		Am2960		Am2960-1		Am2960A	
		From Input	To Output	Min.	Max.	Min.	Max.	Min.	Max.
34	t <sub>SET</sub>	LE IN (Notes 4, 5)	LE OUT	51		51		24	
35	t <sub>HOLD</sub>			5		5		4	
36	t <sub>SET</sub>	DATA <sub>0-15</sub> (Notes 4, 5)	LE DIAG	6		6		3	
37	t <sub>HOLD</sub>			8		8		6	
38	t <sub>EN</sub>	OE BYTE 0,1 ENABLE (Note 6)	DATA <sub>0-15</sub>		30		30		14
39	t <sub>DIS</sub>				30		30		23
40	t <sub>EN</sub>	OE SC DISABLE (Note 6)	SC <sub>0-6</sub>		30		30		7
41	t <sub>DIS</sub>				30		30		21
42	t <sub>PW</sub>	MINIMUM PULSE WIDTH: LE IN, LE OUT, LE DIAG		15		15		12	

- Notes: 1. C<sub>L</sub> = 50 pF.  
 2. Certain parameters are combinational propagation delay calculations.  
 3. Data In or LE In to Correct Data Out measurement requires timing as shown in Figure 26.  
 4. Set-up and Hold times relative to Latch Enables (Latching up data).  
 5. Set-up and Hold times are not tested, but are guaranteed by characterization.  
 6. Output disable tests specified with C<sub>L</sub> = 5 pF and measured to 0.5 V change of output voltage level. Testing is performed at C<sub>L</sub> = 50 pF and correlated to C<sub>L</sub> = 5 pF.

**SWITCHING CHARACTERISTICS** over **MILITARY** operating range (for APL Products, Group A, Subgroups 7, 8, 9, 10, 11 are tested unless otherwise noted) (Notes 1 and 2)

The following table specifies the guaranteed device performance over the Military operating range of -55°C to +125°C (case), with V<sub>CC</sub> 4.5 to 5.5 V. All inputs switching is between 0 V and 3 V at 1 V/ns and measurements are made at 1.5 V. All outputs have maximum DC load. All units are in nanoseconds (ns).

No.	Param.	Data Path Description		Am2960		Am2960A	
		From Input	To Output	Min.	Max.	Min.	Max.
1	t <sub>PD</sub>	DATA <sub>0-15</sub> (Note 3)	SC <sub>0-6</sub>			35	25
			DATA <sub>0-15</sub>			73	30
			ERROR			36	20
			MULT ERROR			56	23
2	t <sub>PD</sub>	CB <sub>0-6</sub> (CODE ID <sub>2-0</sub> 000, 011)	SC <sub>0-6</sub>			30	20
			DATA <sub>0-15</sub>			61	28
			ERROR			31	16
			MULT ERROR			50	20
3	t <sub>PD</sub>	CB <sub>0-6</sub> (CODE ID <sub>2-0</sub> 010, 100, 101, 110, 111)	SC <sub>0-6</sub>			30	22
			DATA <sub>0-15</sub>			50	25
			ERROR			31	16
			MULT ERROR			37	20
4	t <sub>PD</sub>	GENERATE	SC <sub>0-6</sub>			38	27
			DATA <sub>0-15</sub>			69	33
			ERROR			41	19
			MULT ERROR			62	24
5	t <sub>PD</sub>	CORRECT (Not Internal Control Mode)	SC <sub>0-6</sub>			-	-
			DATA <sub>0-15</sub>			49	25
			ERROR			-	-
			MULT ERROR			-	-
6	t <sub>PD</sub>	DIAG MODE (Not Internal Control Mode)	SC <sub>0-6</sub>			58	26
			DATA <sub>0-15</sub>			89	32
			ERROR			65	20
			MULT ERROR			90	24
7	t <sub>PD</sub>	PASS THRU (Not Internal Control Mode)	SC <sub>0-6</sub>			39	22
			DATA <sub>0-15</sub>			51	25
			ERROR			34	16
			MULT ERROR			54	20
8	t <sub>PD</sub>	CODE ID <sub>2-0</sub>	SC <sub>0-6</sub>			69	31
			DATA <sub>0-15</sub>			100	35
			ERROR			68	26
			MULT ERROR			90	29

Notes: See notes following tables continued on next pages.

**SWITCHING CHARACTERISTICS** over **MILITARY** operating range (Cont'd.)

No.	Param.	Data Path Description		Am2960		Am2960A	
		From Input	To Output	Min.	Max.	Min.	Max.
9	t <sub>PD</sub>	LE IN (From latched to transparent)	SC <sub>0-6</sub>		44		24
			DATA <sub>0-15</sub>		82		27
			ERROR		43		19
			MULT ERROR		66		23
10	t <sub>PD</sub>	LE OUT (From latched to transparent)	SC <sub>0-6</sub>		-		-
			DATA <sub>0-15</sub>		33		19
			ERROR		-		-
			MULT ERROR		-		-
11	t <sub>PD</sub>	LE DIAG (From latched to transparent; Not Internal Control Mode)	SC <sub>0-6</sub>		50		27
			DATA <sub>0-15</sub>		88		27
			ERROR		49		19
			MULT ERROR		72		23
12	t <sub>PD</sub>	Internal Control Mode: LE DIAG (From latched to transparent)	SC <sub>0-6</sub>		75		31
			DATA <sub>0-15</sub>		106		35
			ERROR		74		26
			MULT ERROR		96		29
13	t <sub>PD</sub>	Internal Control Mode: DATA <sub>0-15</sub> (Via Diagnostic latch)	SC <sub>0-6</sub>		75		32
			DATA <sub>0-15</sub>		106		35
			ERROR		74		24
			MULT ERROR		96		28
14	t <sub>SET</sub>	DATA <sub>0-15</sub> (Notes 4, 5)	LE IN	7		3	
†15	t <sub>HOLD</sub>			7		5	
16	t <sub>SET</sub>	CB <sub>0-6</sub> (Notes 4, 5)		5		3	
†17	t <sub>HOLD</sub>			7		4	
18	t <sub>SET</sub>	DATA <sub>0-15</sub> (Notes 4, 5)	50		20		
†19	t <sub>HOLD</sub>		5		4		
20	t <sub>SET</sub>	CB <sub>0-6</sub> (Notes 4, 5) (CODE ID 000, 011)	38		15		
†21	t <sub>HOLD</sub>		0		0		
22	t <sub>SET</sub>	CB <sub>0-6</sub> (Notes 4, 5) (CODE ID 010, 100, 101, 110, 111)	30		15		
†23	t <sub>HOLD</sub>		0		0		
24	t <sub>SET</sub>	GENERATE (Notes 4, 5)	46		20		
†25	t <sub>HOLD</sub>		0		0		
26	t <sub>SET</sub>	CORRECT (Notes 4, 5)	28		15		
†27	t <sub>HOLD</sub>		1		1		
28	t <sub>SET</sub>	DIAG MODE (Notes 4, 5)	84		20		
†29	t <sub>HOLD</sub>		0		0		
30	t <sub>SET</sub>	PASS THRU (Notes 4, 5)	30		7		
†31	t <sub>HOLD</sub>		0		0		
32	t <sub>SET</sub>	CODE ID <sub>2-0</sub> (Notes 4, 5)	89		22		
†33	t <sub>HOLD</sub>		0		0		
			LE OUT				

Notes: See notes continued on next page.

**SWITCHING CHARACTERISTICS** over **MILITARY** operating range (Cont'd.)

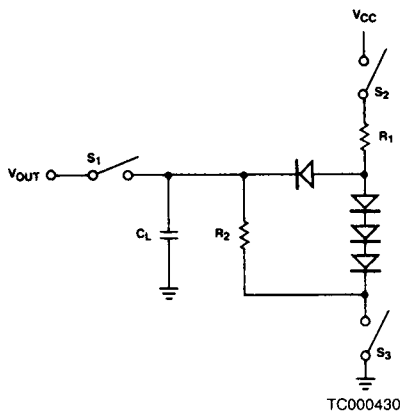
No.	Param.	Data Path Description		Am2960		Am2960A	
		From Input	To Output	Min.	Max.	Min.	Max.
34	t <sub>SET</sub>	LE IN (Notes 4, 5)	LE OUT	59		24	
†35	t <sub>HOLD</sub>			5		4	
36	t <sub>SET</sub>	DATA <sub>0-15</sub> (Notes 4, 5)	LE DIAG	7		3	
†37	t <sub>HOLD</sub>			9		6	
38	t <sub>EN</sub>	OE BYTE 0,1 ENABLE (Note 6)	DATA <sub>0-15</sub>		35		14
39	t <sub>DIS</sub>				35		23
40	t <sub>EN</sub>	OE SC DISABLE (Note 6)	SC <sub>0-6</sub>		35		7
41	t <sub>DIS</sub>				35		21
42	t <sub>PW</sub>	MINIMUM PULSE WIDTH: LE IN, LE OUT, LE DIAG		15		15	

Notes: 1. C<sub>L</sub> = 50 pF.

2. Certain parameters are combinational propagation delay calculations.
3. Data In or LE In to Correct Data Out measurement requires timing as shown in Figure 26.
4. Set-up and Hold times relative to Latch Enables (Latching up data).
5. Setup and Hold times are not tested, but are guaranteed by characterization.
6. Output disable tests specified with C<sub>L</sub> = 5 pF and measured to 0.5 V change of output voltage level. Testing is performed at C<sub>L</sub> = 50 pF and correlated to C<sub>L</sub> = 5 pF.

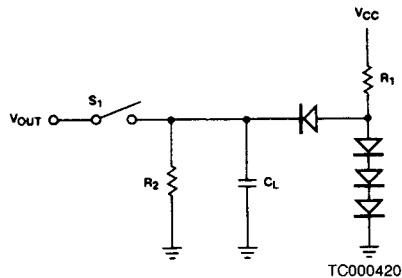
† = Not Included in Group A Tests

## SWITCHING TEST CIRCUITS



**Figure 24. Three-State Outputs**

- Notes:
1.  $C_L = 50\text{pF}$  includes scope probe, wiring and stray capacitances without device in test fixture.
  2.  $S_1, S_2, S_3$  are closed during function test and all A.C. tests, except output enable tests.
  3.  $S_1$  and  $S_3$  are closed while  $S_2$  is open for  $t_{pZH}$  test.  
 $S_1$  and  $S_2$  are closed while  $S_3$  is open for  $t_{pZL}$  test.
  4.  $R_2 = 1\text{K}$  for three-state output.  
 $R_2$  is determined by the  $I_{OH}$  at  $V_{OH} = 2.4\text{V}$  for non-three-state outputs.
  5.  $R_1$  is determined by  $I_{OL}$  (MIL) with  $V_{CC} = 5.0\text{V}$  minus the current to ground through  $R_2$ .
  6.  $C_L = 5.0\text{pF}$  for output disable tests.



**Figure 25. Normal Outputs**

## TEST OUTPUT LOADS

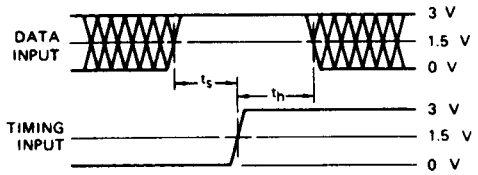
Pin #	Pin Label	Test Circuit	$R_1$	$R_2$
-	D <sub>0</sub> -D <sub>15</sub>	Fig. 24	430Ω	1KΩ
24-30	SC <sub>0</sub> -SC <sub>6</sub>	Fig. 24	430Ω	1KΩ
32	ERROR	Fig. 25	470Ω	3kΩ
33	MULTERROR	Fig. 25	470Ω	3KΩ

### Notes on Testing

Incoming test procedures on this device should be carefully planned, taking into account the complexity and power levels of the part. The following notes may be useful.

1. Insure the part is adequately decoupled at the test head. Large changes in  $V_{CC}$  current as the device switches may cause erroneous function failures due to  $V_{CC}$  changes.
2. Do not leave inputs floating during any tests, as they may start to oscillate at high frequency.
3. Do not attempt to perform threshold tests at high speed. Following an input transition, ground current may change by as much as 400mA in 5-8ns. Inductance in the ground cable may allow the ground pin at the device to rise by 100's of millivolts momentarily.
4. Use extreme care in defining input levels for AC tests. Many inputs may be changed at once, so there will be significant noise at the device pins and they may not actually reach  $V_{IL}$  or  $V_{IH}$  until the noise has settled. AMD recommends using  $V_{IL} \leq 0\text{V}$  and  $V_{IH} \geq 3\text{V}$  for AC tests.
5. To simplify failure analysis, programs should be designed to perform DC, Function, and AC tests as three distinct groups of tests.
6. Changing the CODE ID inputs can cause loss of data in some of the Am2960 internal latches. Specifically, the entire checkbit latch and bits 6 and 7 of the diagnostic latch are indeterminate after a change in CODE ID inputs.  
Logic simulations should store "x" (i.e., "don't care") in these bits after CODE ID change. Test programs should reload these registers before they are used.

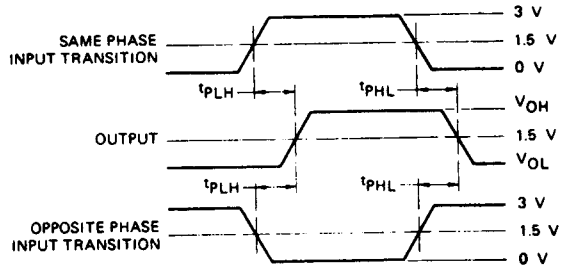
## SWITCHING TEST WAVEFORMS



WF021191

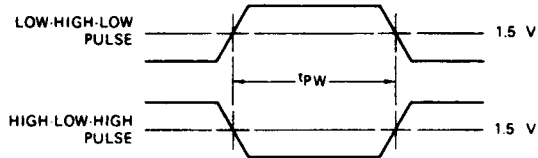
- Notes: 1. Diagram show for HIGH data only. Output transition may be opposite sense.  
 2. Cross-hatched are is don't care condition.

### Setup and Hold Times



WF021200

### Propagation Delay








WF021210

### Pulse Width

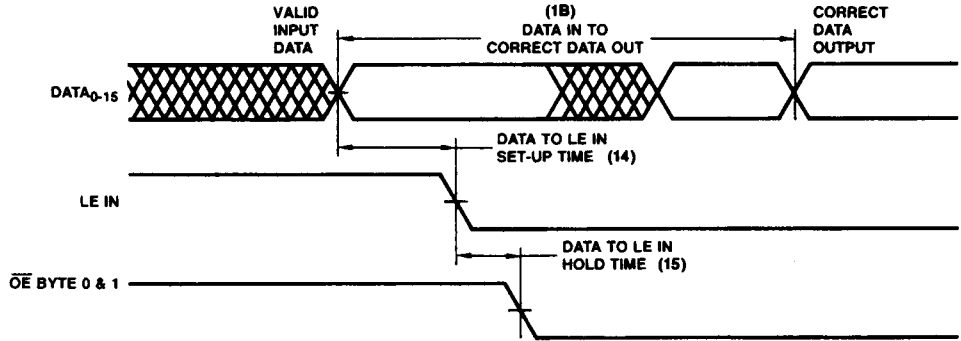


# SWITCHING WAVEFORMS

## KEY TO SWITCHING WAVEFORMS

WAVEFORM	INPUTS	OUTPUTS
	MUST BE STEADY	WILL BE STEADY
	MAY CHANGE FROM H TO L	WILL BE CHANGING FROM H TO L
	MAY CHANGE FROM L TO H	WILL BE CHANGING FROM L TO H
	DON'T CARE; ANY CHANGE PERMITTED	CHANGING; STATE UNKNOWN
	DOES NOT APPLY	CENTER LINE IS HIGH IMPEDANCE "OFF" STATE

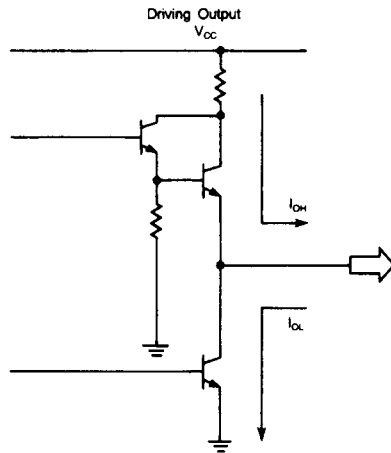
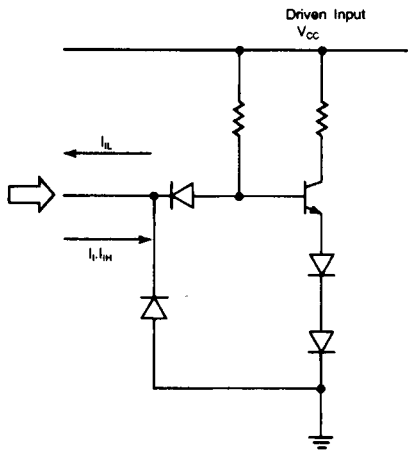
KS000010



WF001521

**Figure 26. DATA IN/LE IN to Correct DATA OUT**

# INPUT/OUTPUT CIRCUIT DIAGRAMS



IC000883