## Parity Bits: Error Detection

A parity bit is a bit that is added as the leftmost bit of a bit string to ensure that the number of bits that are 1 in the bit string are even or odd. The following table summarize this approach.

|  |  |  |
| --- | --- | --- |
| **Data Bits (7)** | **Add a parity bit to get 8 bits** | |
| **Even Parity Scheme**  **Total number 1s is even** | **Odd Parity Scheme**  **Total number of 1s is odd** |
| 000 0000 (0 1s) | 0 000 0000 | 1 000 0000 |
| 011 0010 (3 1s) | 1 011 0010 | 0 011 0010 |
| 011 0011 (4 1s) | 0 011 0011 | 1 011 0011 |
| 011 0111 (5 1s) | 1 011 0111 | 0 011 0111 |

Circle the bytes (a group of 8 bits) below that have an error in them, assuming an *even* parity scheme.

|  |  |  |  |
| --- | --- | --- | --- |
| 0000 0101 | 0010 1000 | 1000 0100 | 0010 1000 |
| 0001 0001 | 1010 0101 | 1010 0010 | 0111 1111 |
| 1010 0011 | 1001 1001 | 1000 0000 | 1010 1000 |
| 0010 1001 | 0110 1001 | 0110 0011 | 1000 0010 |

For the data bits below, assume that an *odd* parity scheme is being used and add the appropriate parity bit (0 or 1). Remember to add the parity bit on the left side.

|  |  |  |  |
| --- | --- | --- | --- |
| 000 0101 | 010 1000 | 000 0100 | 010 1000 |
| 001 0001 | 010 0101 | 010 0010 | 111 1111 |
| 010 0011 | 001 1001 | 000 0000 | 010 1000 |
| 010 1001 | 110 1001 | 110 0011 | 000 0010 |

## 

## 

## Parity Bits: Error Correction

**Hamming Code:** In a block of 7 bits, 3 of the bits are parity bits and 4 are data bits. The relationship between the data and parity bits is represented by a Venn diagram.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| p1 | d1 | p2 | d2 | p3 | d3 | d4 |
|  | 1 |  | 0 |  | 1 | 1 |

For the green (upper) circle, the parity bit (p1) applies to the data bits d1, d2, and d4. In an even parity scheme, if the data bits are as shown above in the table (101), then the p1 parity bit would be 0. For the blue circle (lower left), the data bits are d1, d3, and d4 (111) and it’s parity bit, p2, would be a 1. For the red circle (lower right), the data bits are d2, d3, and d4 (011) and it’s parity bit, p3, would be a 0. Here is the completed table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| p1 | d1 | p2 | d2 | p3 | d3 | d4 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 |

Try filling in the parity bits on your own for the data bits shown below (using even parity).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| p1 | d1 | p2 | d2 | p3 | d3 | d4 |
|  | 1 |  | 1 |  | 0 | 0 |

If a single error occurs, two or more of the parity bits will be incorrect. For the binary string below, which parity bits are incorrect? Which data bit is found in both circles? That data bit is incorrect and can then be corrected.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| p1 | d1 | p2 | d2 | p3 | d3 | d4 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 |

For the binary strings below, identify which parity bits are incorrect, the corresponding data bit with an error, and then write the corrected binary string.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| p1 | d1 | p2 | d2 | p3 | d3 | d4 | Incorrect Parity Bit(s) | Incorrect Data Bit | Corrected Binary String |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 |  |  |  |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |  |  |  |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 |  |  |  |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 |  |  |  |