

Computer Mathematics

Week 6 Examples

1. Add one more bit p to each byte, to ensure that it has even parity or odd parity.

<i>even parity</i>		<i>odd parity</i>	
<u>word</u>	<u>p</u>	<u>word</u>	<u>p</u>
00000000	_____	11111111	_____
10010010	_____	10111101	_____
11000101	_____	10111001	_____
10011101	_____	10011001	_____
11111111	_____	00000000	_____

2. The following message was sent with even lateral parity bits and an even longitudinal parity word. Identify the corrupted bit.

	<u>message</u>	
1	01010000 0	
2	01100001 1	
3	01111010 0	incorrect row (number): _____
4	01101001 0	
5	01100100 1	incorrect column (letter): _____
6	01111001 1	
7	00110111 1	
	ABCDEFGH I	

3. Calculate a 3-bit CRC code for the 8-bit data word 00101010 using the 4-bit divisor 1101. Verify that repeating the calculation on an uncorrupted message, sent using your CRC code, produces a remainder of zero.

<u>message</u>	<u>CRC</u>
0 0 1 0 1 0 1 0	0 0 0
- 1 1 0 1	
=	
-	
=	
-	
=	
-	
=	
-	
=	← CRC
<u>message</u>	<u>CRC</u>
0 0 1 0 1 0 1 0	_____ ← CRC
- 1 1 0 1	
=	
-	
=	
-	
=	
-	
=	
-	
=	0 0 0 ← remainder

4. A seven-bit Hamming code for transmitting four-bit messages $d_0d_1d_2d_3$ can be constructed using three parity bits $p_0, p_1,$ and p_2 as follows.

<i>role</i>	p_0	p_1	d_0	p_2	d_1	d_2	d_3	
<i>bit number</i>	1	2	3	4	5	6	7	
<i>binary</i>	1	0	1	0	1	0	1	$\leftarrow p_0$ bits
	0	1	1	0	0	1	1	$\leftarrow p_1$ bits
	0	0	0	1	1	1	1	$\leftarrow p_2$ bits

Construct the seven-bit message that encodes the four-bit data 0101.

<i>role</i>	p_0	p_1	d_0	p_2	d_1	d_2	d_3
<i>value</i>	_____	_____	0	_____	1	0	1

Verify that the parity bits are correct, if the message is received with no error.

Copy your message seven times into the table below, introducing a single-bit error each time. (For each bit position, from p_0 to d_3 , create a single-bit error in that position only.) Check the validity of the parity bits, and verify that they correctly identify the corrupted bit position.

	1	2	3	4	5	6	7		<i>incorrect bit number according to parities</i>
<i>role</i>	p_0	p_1	d_0	p_2	d_1	d_2	d_3		
<i>value</i>	_____	_____	0	_____	1	0	1	(p_0 corrupted)	_____
<i>role</i>	p_0	p_1	d_0	p_2	d_1	d_2	d_3		
<i>value</i>	_____	_____	0	_____	1	0	1	(p_1 corrupted)	_____
<i>role</i>	p_0	p_1	d_0	p_2	d_1	d_2	d_3		
<i>value</i>	_____	_____	1	_____	1	0	1	(d_0 corrupted)	_____
<i>role</i>	p_0	p_1	d_0	p_2	d_1	d_2	d_3		
<i>value</i>	_____	_____	0	_____	1	0	1	(p_2 corrupted)	_____
<i>role</i>	p_0	p_1	d_0	p_2	d_1	d_2	d_3		
<i>value</i>	_____	_____	0	_____	0	0	1	(d_1 corrupted)	_____
<i>role</i>	p_0	p_1	d_0	p_2	d_1	d_2	d_3		
<i>value</i>	_____	_____	0	_____	1	1	1	(d_2 corrupted)	_____
<i>role</i>	p_0	p_1	d_0	p_2	d_1	d_2	d_3		
<i>value</i>	_____	_____	0	_____	1	0	0	(d_3 corrupted)	_____