Note 38 - Hamming code algorithm

The problem: Data bits are communicated from source to target. Have any of the bits changed?
Solution: Add parity bits to your data according to the Hamming code algorithm.

The Wikipedia page has a nice explanation. Scroll down to “General Algorithm”:
en.wikipedia.org/wiki/Hamming_code

A **codeword** is comprised of data bits and parity bits.
The length of the codeword is: N = M data bits + R parity bits. For Hamming $R = \log_2 M + 1$

The format of our codeword is:

- Number the bits from 1 to M.
- Each bit that is a power of 2 is a parity bit: 1, 2, 4, 8, 16, 32, etc. All other bits are data.
- Parity bits are set based on the data bits that include the parity bit number in their binary representation. For example:
    - Parity bit 1 checks all data bits that include the 1 bit (0001) in their binary representation: 3, 5, 7, 9, 11…
    - Parity bit 2 checks all data bits that include the 2 bit (0010): 3, 6, 7, 10, 11…
    - Parity bit 3 checks all data bits that include the 4 bit (0100): 5, 6, 7, 12, 13…

Example: 4 data bits
Your codeword will have: 4 data bits + 3 ($\log_2 4 + 1$) parity bits = 7 total bits
It looks like this (P=parity bit, d=data bit):

```
bits:  PPdP ddd
count: 1234 567
```

Example: 8 data bits
Your codeword will have: 8 data bits + 4 ($\log_2 8 + 1$) parity bits = 12 total bits

```
bits:  PPdP dddP dddd
count: 1234 5678 9012
```

Example: 16 data bits
Your codeword will have: 16 data bits + 5 ($\log_2 16 + 1$) parity bits = 21 total bits

```
bits:  PPdP dddP dddd dddP dddd d
count: 1234 5678 9012 3456 7890 1
```

If you understand this format, then there are two problems you must be able to solve:

1. Given some data bits, create a codeword
2. Given a codeword, determine if any data bits have changed.

Turn the page over for some examples.

Example #1: For data bits `1101`, create a codeword.
The codeword format is:

```
bits:  PPdP ddd
count: 1234 567
```

Plus in data bits: `PP1P101`.
Calculate parity bits, one at a time:

- P_1 = parity(d_3, d_5, d_7) = parity(1, 1, 1) = 1
- P_2 = parity(d_3, d_6, d_7) = parity(1, 0, 1) = 0
- P_4 = parity(d_5, d_6, d_7) = parity(1, 0, 1) = 0

So, our codeword is: `1010 101`

Example #2: For codeword `1010 111`, what data bits, if any, must be corrected?
The data bits in this codeword are: `1111`.
Recalculate each parity bit for the data bits in the codeword:

- P_1 = parity(d_3, d_5, d_7) = parity(1, 1, 1) = 1
- P_2 = parity(d_3, d_6, d_7) = parity(1, 1, 1) = 1
- P_4 = parity(d_5, d_6, d_7) = parity(1, 1, 1) = 1

An error has been detected because parity bits P_2 and P_4 have changed!
The data bit that has changed is 2 (P_2) + 4 (P_4)= 6… flip bit #6.
The correct data bits are `1101`.

Example #3: For data bits `1001 1000`, create a codeword.
The codeword format is:

```
bits:  PPdP dddP dddd
count: 1234 5678 9012
```

Plug in our data bits: `PP1P 001P 1000`

Calculate parity bits:

- P_1 = parity(d_3, d_5, d_7, d_9, d_{11}) = parity(1, 0, 1, 1, 0) = 1
- P_2 = parity(d_3, d_6, d_7, d_{10}, d_{11}) = parity(1, 0, 1, 0, 0) = 0
- P_4 = parity(d_5, d_6, d_7, d_{12}) = parity(0, 0, 1, 0) = 1
- P_8 = parity(d_9, d_{10}, d_{11}, d_{12}) = parity(1, 0, 0, 0) = 1

So, our codeword is: `1011 0011 1000`

Example #4: For codeword `1001 0011 1000`, which bits, if any has changed?

I’m out of room. Can you show that bit 3 has changed?