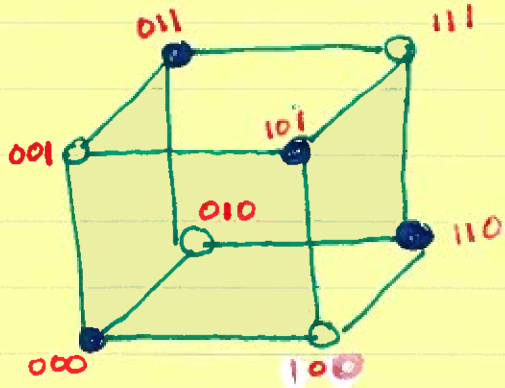
hw 1

Homework - cp. 1

p.17 #1 a) By trial and error, find a set of four binary words of length 3 s.t. each word is at least a distance of 2 from every other word.

Well, consider once again the vertices of a cube in 3-dimensional space.



The vertices that are darkened, i.e. {000, 011, 110 and 101} is one such set.

b). Find a set of 16 binary words of length 7 s.t. each word is at least a distance of 3 from every other word.

We write the 16 distinct binary words of length 4.

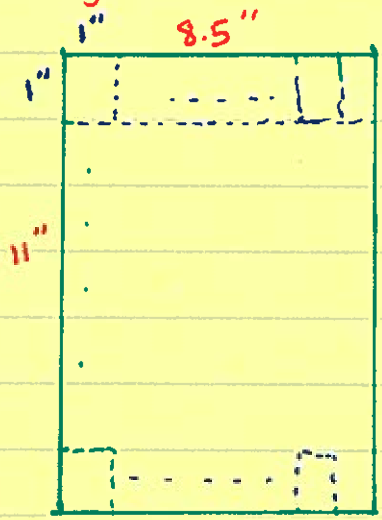
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

and then append three bits to each word by using the check equations of the Hamming (7,4) code.

How do we know this procedure will work?

hw 1

#2) a) Describe how to cut 88 circles of 1" diameter out of a sheet of paper of width 8.5" and length 11".



Begin by cutting the sheet into 88 squares, each of area 1 in. square

How do we know this is possible?

Next, cut a circle of 1" diameter from each of the unit squares.

Prove that it is not possible to cut out more than 119 circles of 1-inch diameter.

→ Area of this rectangular sheet of paper = 8.5" × 11" = 93.5 sq.in

→ Area of a circle = $\pi \cdot r^2 = 3.14 \times (\frac{1}{2} \times \frac{1}{2})$

$= 3.14/4 = .785$ sq. in

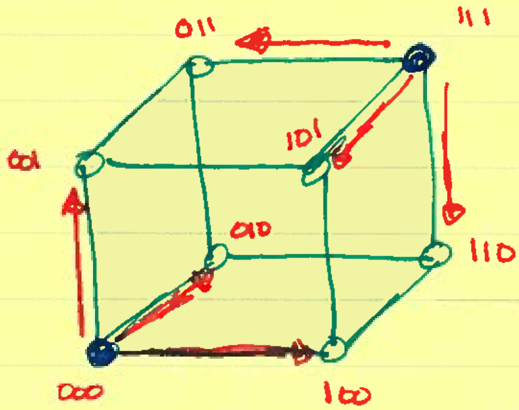
→ Area of 119 such circles is $.785 \times 119 = 93.42$ sq. in

120 circles would exceed the area of the paper

b) Prove that it is not possible to find 32 binary words each of length 8 bits, s.t. every word differs from every other word in at least three places.

hw1

If code words were of length three instead, then we are in 3-dimensional space (see problem 1).



wlog of generality, let our code consist of $\{000, 111\}$

Note that there are three codewords (each with errors) at a distance of one from each of 000 and 111

Now, with 8-bit code words we are in 8-dimensional space. Our code consists of 32 words each with 8 bits. There are 8 codewords (sensewords) at a distance of one from each of these 32 words. Note that $32 \times 8 = 256 = 2^8$ (the total number of 8-bit code words). We have already exhausted the number of distinct 8 bit words!

#3. A single-error-correcting Hamming code has $2^m - 1$ bits of which m bits are check bits.

a) Write (n, k) for the first five nontrivial Hamming codes (starting at $m = 3$).
b) Calculate their rates.

Each Hamming code is an $(2^m - 1, 2^m - 1 - m)$ code

$m = 3$, the code is $(2^3 - 1, 2^3 - 1 - 3)$ or $(7, 4)$
Rate of a block code $R = k/n$; $R = 4/7$

$m = 4$ $(2^4 - 1, 2^4 - 1 - 4) = (15, 11)$
$R = 11/15$

$m = 5$ $(2^5 - 1, 2^5 - 1 - 5) = (31, 26)$
$R = 26/31$

$m = 6$ $(2^6 - 1, 2^6 - 1 - 6) = (63, 57)$
$R = 57/63$

$m = 7$ $(2^7 - 1, 2^7 - 1 - 7) = (127, 120)$
$R = 120/127$

Observe that as code length increases, the rate R is getting closer and closer to one.

3 con't c) Write an expression for the probability of decoding error, pe, when the code is used with a binary channel that makes errors with probability q. How does the probability of error behave with n?

A Hamming code is single error correcting. Hence, it will function incorrectly when more than one error occurs.

Pr(pe) corresponds to the probability of more than one error occuring.

$$Pr(pe) = \sum_{i=2}^{n} \underbrace{\binom{n}{i}}_{\substack{\text{\# ways to choose}\\ \text{i error positions out}\\ \text{of } n}} \overbrace{q^i}^{i \text{ errors}} \overbrace{(1-\underbrace{q}_{\substack{\text{prob. of}\\ \text{no error}}})^{n-i}}^{\text{no error should occur on } n-i \text{ of the } n \text{ bits}}$$

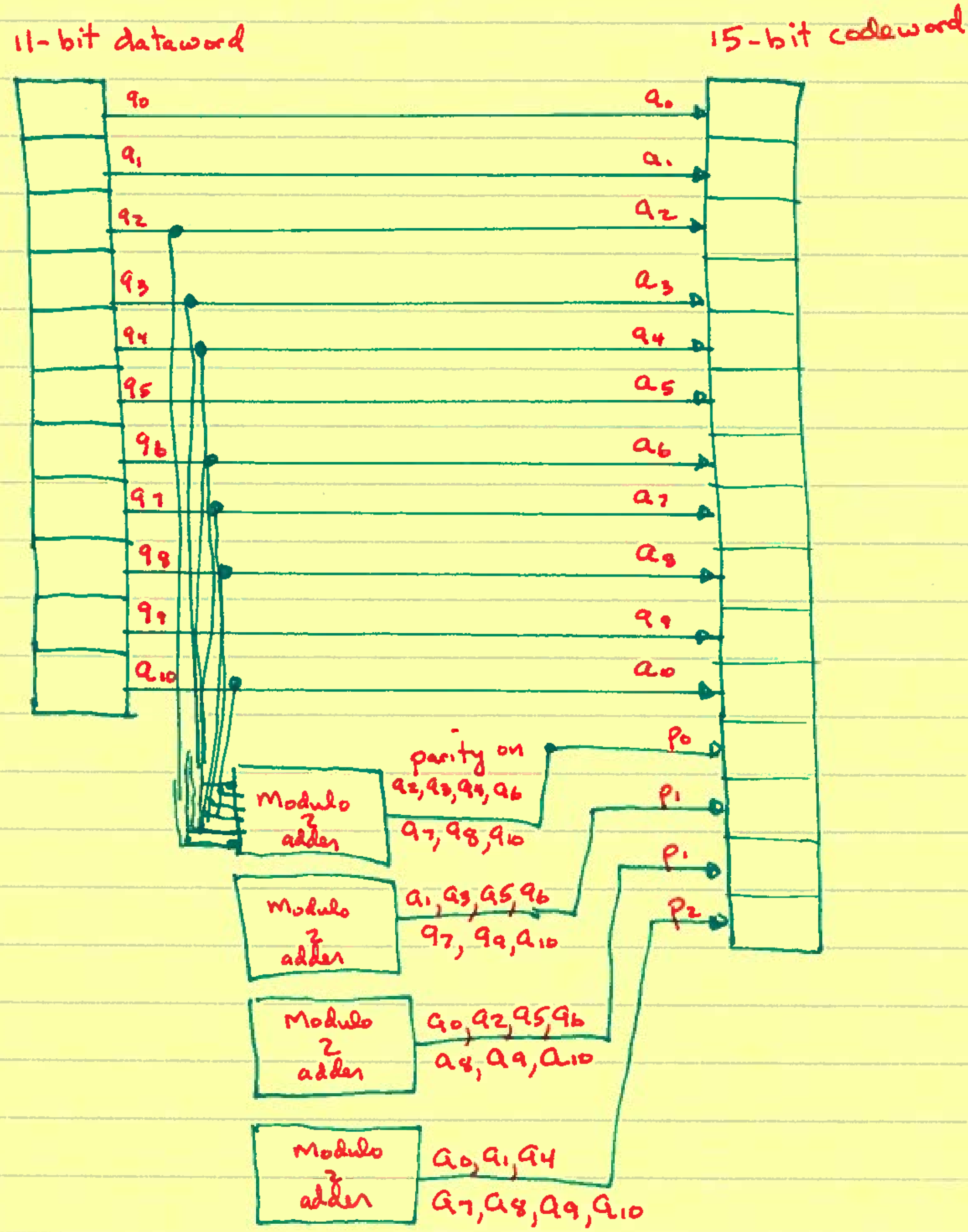
Naturally, this probability corresponds to 1 − prob. that 0 or 1 error occurs

$$Pr(pe) = 1 - \left[q^0 (1-q)^n + \binom{n}{1} q^1 (1-q)^{n-1} \right]$$

$$= 1 - \left[\underbrace{(1-q)^n}_{\text{prob. of no mistakes}} + \underbrace{nq(1-q)^{n-1}}_{\text{prob. of 1 mistake}} \right]$$

As n increases so does this probability of a decoding error

4) Design an encoder/decoder for a (15,11) Hamming code.



Encoder for (15,11) Hamming code

hw 1

Generator matrix for (15,11) Hamming code:

Generator matrix G

these appear 11 rows that ensure dataword appears as a prefix of codeword

$$
\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \\ a_{10} \\ p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix}
=
\begin{bmatrix}
1&0&0&0&0&0&0&0&0&0&0 \\
0&1&0&0&0&0&0&0&0&0&0 \\
0&0&1&0&0&0&0&0&0&0&0 \\
0&0&0&1&0&0&0&0&0&0&0 \\
0&0&0&0&1&0&0&0&0&0&0 \\
0&0&0&0&0&1&0&0&0&0&0 \\
0&0&0&0&0&0&1&0&0&0&0 \\
0&0&0&0&0&0&0&1&0&0&0 \\
0&0&0&0&0&0&0&0&1&0&0 \\
0&0&0&0&0&0&0&0&0&1&0 \\
0&0&0&0&0&0&0&0&0&0&1 \\
\hline
0&0&1&1&1&0&1&1&1&0&1 \\
0&1&0&1&0&1&1&1&0&1&1 \\
1&0&1&0&0&1&1&0&1&1&1 \\
1&1&0&0&1&0&0&1&1&1&1
\end{bmatrix}
\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \\ a_{10} \end{bmatrix}
$$

$\binom{4}{2} + \binom{4}{3} + \binom{4}{4} = \underline{11}$ column vectors with 2, 3, or 4 1's.

G defines an encoding function from $\mathbb{Z}_2^{11} \to \mathbb{Z}_2^{15}$.

We require a 4×11 submatrix to complete G.

$$
\begin{pmatrix}0\\0\\1\\1\end{pmatrix},
\begin{pmatrix}0\\1\\0\\1\end{pmatrix},
\begin{pmatrix}1\\0\\1\\0\end{pmatrix},
\begin{pmatrix}1\\1\\0\\0\end{pmatrix},
\begin{pmatrix}1\\0\\0\\1\end{pmatrix},
\begin{pmatrix}0\\1\\1\\0\end{pmatrix},
\begin{pmatrix}1\\1\\1\\0\end{pmatrix},
\begin{pmatrix}1\\1\\0\\1\end{pmatrix},
\begin{pmatrix}1\\0\\1\\1\end{pmatrix},
\begin{pmatrix}0\\1\\1\\1\end{pmatrix},
\begin{pmatrix}1\\1\\1\\1\end{pmatrix}
$$

By viewing our generator matrix G:
Four check bits p_0, p_1, p_2, p_3 where

$$p_0 = a_2 + a_3 + a_4 + a_6 + a_7 + a_8 + a_{10}$$
$$p_1 = a_1 + a_3 + a_5 + a_6 + a_7 + a_9 + a_{10}$$
$$p_2 = a_0 + a_2 + a_5 + a_6 + a_8 + a_9 + a_{10}$$
$$p_3 = a_0 + a_1 + a_4 + a_7 + a_8 + a_9 + a_{10}$$

The decoder receives a 15-bit senseword

$$v = (a_0', a_1', a_2', a_3', a_4', a_5', a_6', a_7', a_8', a_9', a_{10}', p_0', p_1', p_2', p_3')$$

The syndrome computes:

$$S_0 = p_0' + a_2' + a_3' + a_4' + a_6' + a_7' + a_8' + a_{10}'$$
$$S_1 = p_1' + a_1' + a_3' + a_5' + a_6' + a_7' + a_9' + a_{10}'$$
$$S_2 = p_2' + a_0' + a_2' + a_5' + a_6' + a_8' + a_9' + a_{10}'$$
$$S_3 = p_3' + a_0' + a_1' + a_4' + a_7' + a_8' + a_9' + a_{10}'$$

(S_0, S_1, S_2, S_3) is the syndrome
16 distinct syndromes correspond to the 16 ways in which one error (or none) can occur in a 16 bit senseword.

An example

Let message (or dataword) equal $\bar{a} = a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10}$
$= 1\,1\,1\,1\,0000000$

Note, there are 2^{11} distinct messages

To obtain the codeword, we compute $\vec{G} \times \bar{a}$

$\underbrace{11110000000}_{\text{original message}}\ \underbrace{p_0 p_1 p_2 p_3}_{\text{4 check bits}}$

$p_0 = 1+1+0+0+0+0+0 = 0$
$p_1 = 1+1+0+0+0+0+0 = 0$
$p_2 = 1+1+0+0+0+0+0 = 0$
$p_3 = 1+1+0+0+0+0+0 = 0$

$\therefore$ codeword $= 11110000000\,0000$
syndrome for this codeword equals $(0,0,0,0)$

Next, let us assume that an error occurs in a_7.
Hence we receive senseword 111100010000000

syndrome for this senseword is: $(1,1,0,1)$
This will correspond to the error-tuple
$e = 000000010000000$
Adding this to the senseword corrects for the error in a_7.

Syndrome table will have 16 rows, one for each error pattern.