ELEC 6131: Error Detecting and Correcting Codes
Lecture 4: Linear Block Codes

Linear block codes:

In a digital communication system, the sequence of bits to be transmitted are arranged as blocks of k bits. So, there are 2^k possible k-tuples to be transmitted. In a block code, the encoder assigns n bits to each k-tuple where $n > k$. For a block code to be useful we require that all of 2^k, n-tuples (called codewords) be distinct. That is there should be a 1-to-1 correspondence between the input $\underline{u}$ and the output $\underline{v}$ of the encoder.

Unless the codewords are structured according to a certain structure, the encoding (and obviously decoding) will be prohibitively complex. That is why we are interested in linear block codes. A code is linear if a linear combination of any two of its codewords is a codeword, or equivalently:

Definition: a block code of length n and 2^k codewords is an (n, k) linear code if and only if its 2^k codewords form the k-dimensional subspace of the vector space of n-tuples over $GF(2)$.

A linear (n, k) code C is a k-dimensional subspace of all the binary n-tuples (V_n). So, we can find k linearly independent members of C, say $\underline{g}_0, \underline{g}_1, \cdots, \underline{g}_{k-1}$ such that any $\underline{v} \in V$ can be written as:

$$\underline{v} = u_0\underline{g}_0 + u_1\underline{g}_1 + \cdots + u_{k-1}\underline{g}_{k-1}.$$

Arranging these k linearly independent in a matrix:

$$G = \begin{bmatrix} \underline{g}_0 \\ \underline{g}_1 \\ \vdots \\ \underline{g}_{k-1} \end{bmatrix} = \begin{bmatrix} g_{00} & g_{01} & \cdots & g_{0,n-1} \\ g_{10} & g_{11} & \cdots & g_{1,n-1} \\ \vdots & \vdots & & \vdots \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{bmatrix}$$

where G is a $k \times n$, binary matrix.

Let $\underline{u} = (u_0, u_1, \cdots, u_{k-1})$ be the message to be sent. Then, the codeword can be given as:

$$\underline{v} = \underline{u} \cdot G = (u_0, u_1, \cdots, u_{k-1}) \begin{bmatrix} \underline{g}_0 \\ \underline{g}_1 \\ \vdots \\ \underline{g}_{k-1} \end{bmatrix} = u_0\underline{g}_0 + u_1\underline{g}_1 + \cdots + u_{k-1}\underline{g}_{k-1}.$$

That is, rows of G, span or generate C. That is why G is called the generator matrix.

Example: (Hamming code)

Consider (7,4) code we saw before:

$$G = \begin{bmatrix} \underline{g}_0 \\ \underline{g}_1 \\ \underline{g}_2 \\ \underline{g}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Let’s message be $\underline{u} = (1\ 1\ 0\ 1)$. Then,

$$\underline{v} = 1 \cdot \underline{g}_0 + 1 \cdot \underline{g}_1 + 0 \cdot \underline{g}_2 + 1 \cdot \underline{g}_3$$

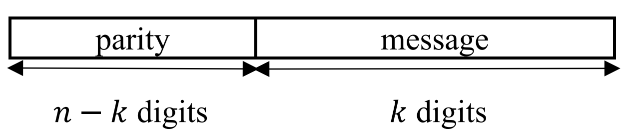
$$= (1101000) + (0110100) + (1010001)$$

$$= (0001101)$$

Example: (7,4) linear block code:

message	codeword
0000	0000000
1000	1101000
0100	0110100
1100	1011100
0010	1110010
1010	0011010
0110	1000110
1110	0101110
0001	1010001
1001	0111001
0101	1100101
1101	0001101
0011	0100011
1011	1001011
0111	0010111
1111	1111111

Definition: a block code is called systematic if its message bits are consecutive and so are its parity bits.



The generator of a systematic code consists of a $k \times k$ identity matrix (to repeat the message bits) and a $k \times (n-k)$ parity matrix to generate parity bits.

$$G = \begin{bmatrix} \underline{g}_0 \\ \underline{g}_1 \\ \vdots \\ \underline{g}_{k-1} \end{bmatrix} = \left[\begin{array}{cccc|cccc} p_{00} & p_{01} & \cdots & p_{0,n-k-1} & 1 & 0 & \cdots & 0 \\ p_{10} & p_{11} & \cdots & p_{1,n-k-1} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots & & & & \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} & 0 & 0 & \cdots & 1 \end{array}\right]$$

So, $G = [P \,\vdots\, I_k]$.

For an input $\underline{u} = (u_0, u_1, \cdots, u_{k-1})$, the output of the encoder is:

$$\underline{v} = (v_0, v_1, \cdots, v_{n-1}) = (u_0, u_1, \cdots, u_{k-1})G.$$

So, $v_i = u_0 p_{0i} + u_1 p_{1i} + \cdots + u_{k-1} p_{k-1,i}$ for $0 \le i < n-k$ and $v_{n-k+i} = u_i$ for $0 \le i < k$.

Going back to our (7,4) example:

$$\underline{v} = (u_0, u_1, \cdots, u_{k-1}) \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Therefore,

$$v_0 = u_0 + u_2 + u_3$$

$$v_1 = u_0 + u_1 + u_2$$

$$v_2 = u_1 + u_2 + u_3$$

$$v_3 = u_0$$

$$v_4 = u_1$$

$$v_5 = u_2$$

$$v_6 = u_3.$$

Parity check matrix:

Let G be the generating polynomial of a code C. Form an $(n-k) \times n$ matrix H whose rows are orthogonal to all rows of G. For a systematic code $G = [P \,\vdots\, I_k]$ and $H = [I_{n-k} \,\vdots\, P^T]$, where P^T is the transpose of P. That is:

$$H = [I_{n-k} \,\vdots\, P^T] = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & p_{00} & \cdots & p_{k-1,0} \\ 0 & 1 & 0 & \cdots & 0 & p_{01} & \cdots & p_{k-1,1} \\ \vdots & \vdots & & & & \vdots & & \\ 0 & 0 & 0 & \cdots & 1 & p_{0,n-k-1} & \cdots & p_{k-1,n-k-1} \end{bmatrix}$$

Then, we have:

$$G \cdot H^T = \underline{0}.$$

Therefore, for any $\underline{v} \in C \Rightarrow \underline{v} = u \cdot G \cdot H^T = \underline{0}$.

For the $(7,4)$ Hamming code:

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

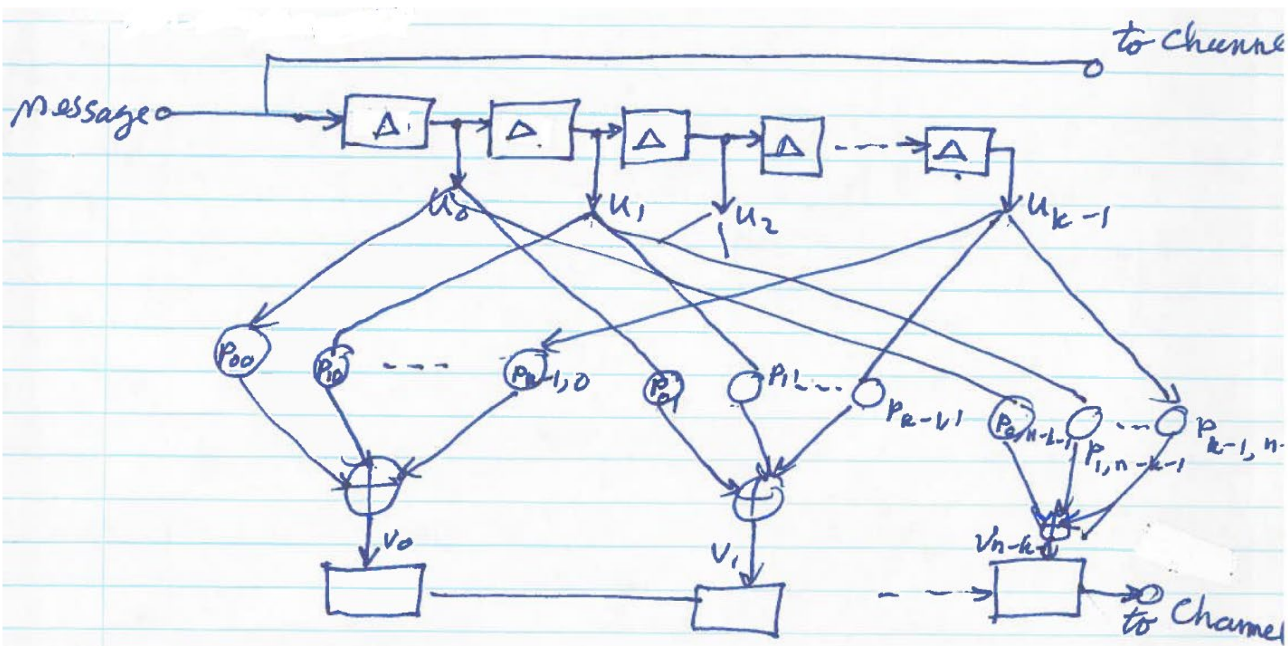
Note that a parity check matrix can generate an $(n, n-k)$ code. Each codeword of this code, C_d is orthogonal to each codeword of C. C_d is called the dual code of C.

To encode a linear block code, we use XOR gates to form parities. Following figure shows how a systematic linear block code is encoded:

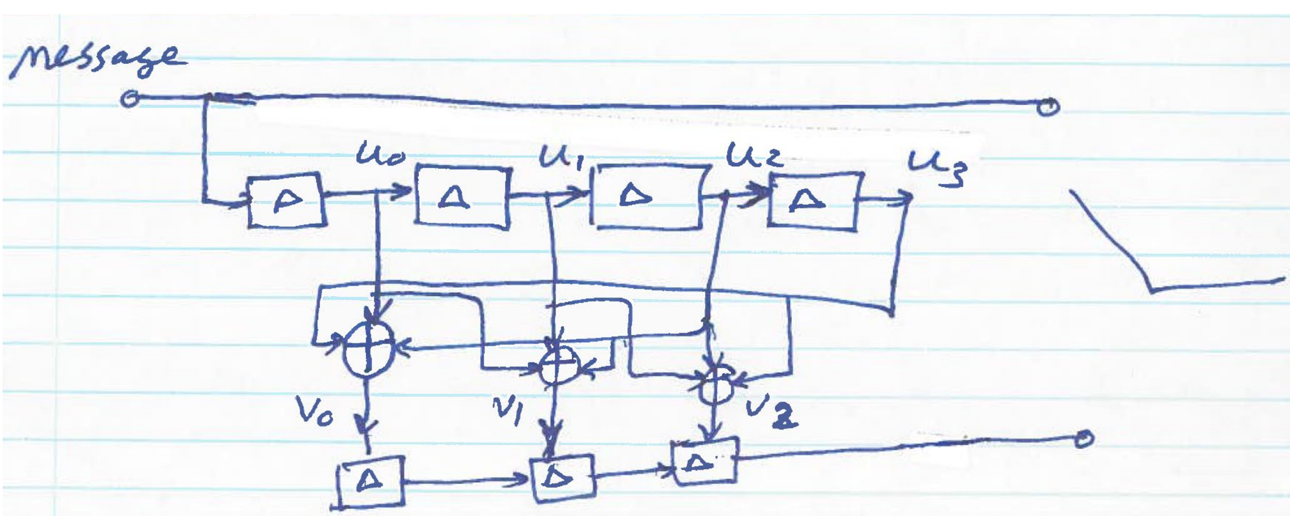
Bits of the message are fed to a shift register and also go to the channel. When they are in the shift register, they are linearly combined according to:

$$\underline{v}_i = u_0 p_{0i} + u_1 p_{1i} + \cdots + u_{k-1} p_{k-1,i},$$

and placed in an output register and fed to channel.



For the $(7, 4)$ code:



Syndrome:

Assume that the message $\underline{u}$ is encoded as $\underline{v} = \underline{u} \cdot G$. If there is no error, at the receiver we have $\underline{r} = \underline{v}$ and no need for error detection and error correction. But if there is an error, we get:

$$\underline{r} = \underline{v} + \underline{e},$$

where $\underline{e} = (e_0, e_1, \cdots, e_n)$ is an error vector. If we multiply $\underline{r}$ by H^T, we get:

$$\underline{r} \cdot H^T = (\underline{v} + \underline{e}) \cdot H^T = \underline{v} \cdot H^T + \underline{e} \cdot H^T = \underline{e} \cdot H^T$$

It is important to note that the result does not depend on the message, but on the error pattern $\underline{e}$. We call the vector $\underline{s} = \underline{r} \cdot H^T$ the syndrome. Since $\underline{r}$ is an n-vector and H^T is $n \times (n-k)$, there are $(n-k)$ bits in vector $\underline{s}$. So, $\underline{s}$ can point to 2^{n-k} patterns (one correct transmission $0, 0, \cdots, 0$ and $2^{n-k} - 1$ error patterns).

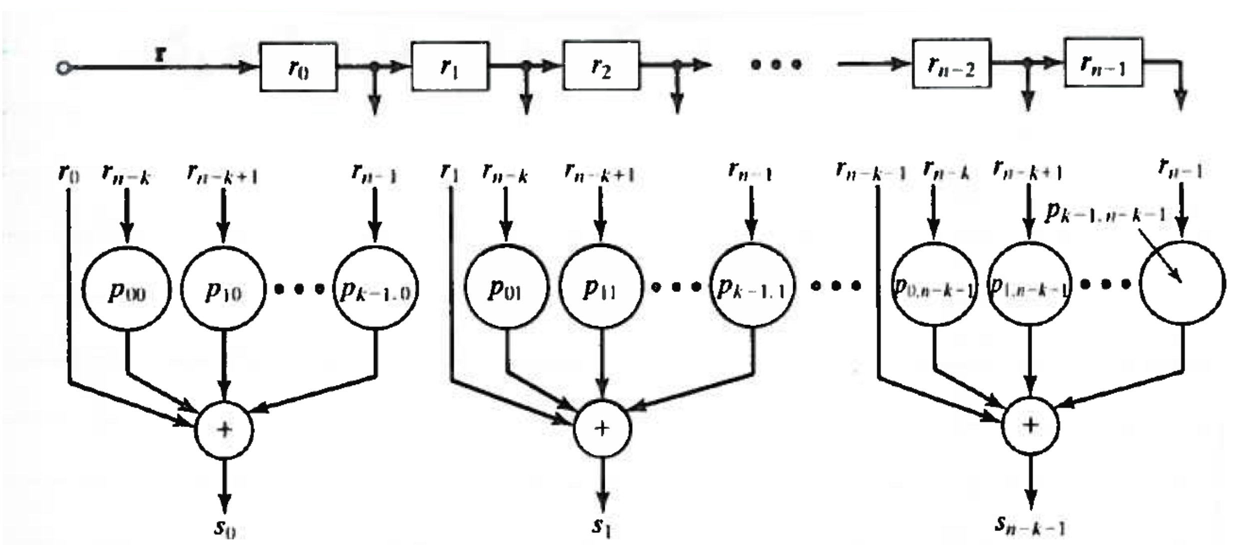
Example: consider the $(7, 4)$ code. Let $\underline{r} = (r_0, r_1, r_2, r_3, r_4, r_5, r_6)$ be the received vector (output of demodulator). Then the syndrome

$$\underline{s} = (s_0, s_1, s_2) = (r_0, r_1, r_2, r_3, r_4, r_5, r_6) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

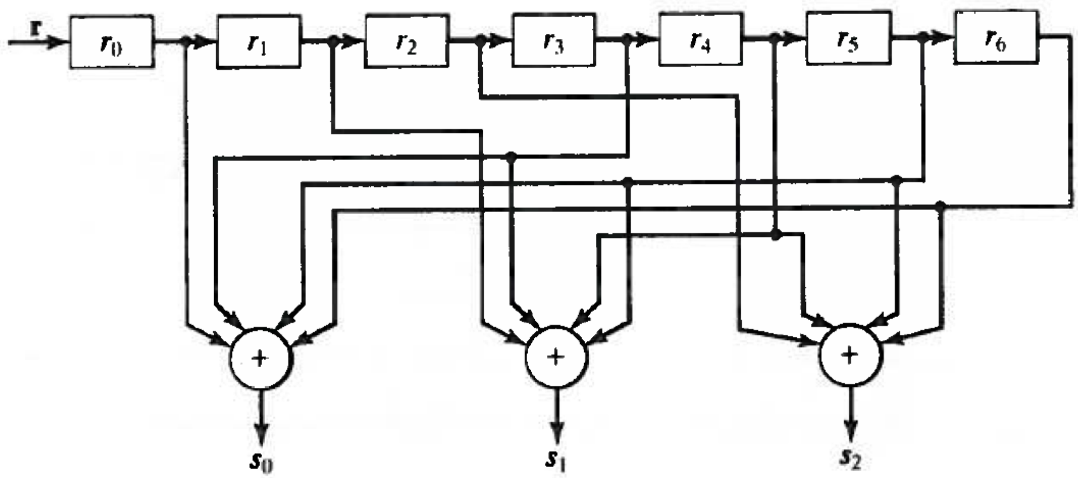
or:

$$s_0 = r_0 + r_3 + r_5 + r_6$$

$$s_1 = r_1 + r_3 + r_4 + r_5$$



Syndrome circuit for a linear systematic (n, k) code



Syndrome circuit for the $(7, 4)$ code

We saw that:

$$\underline{s} = \underline{r} \cdot H^T = \underline{e} \cdot H^T.$$

So, we can write s_i's as:

$$s_i = r_i + r_{n-k}p_{0i} + r_{n-k+1}p_{1i} + \cdots + r_{n-1}p_{k-1,i}, \qquad i = 0, 1, \cdots, n-k-1.$$

Since $\underline{r} = \underline{v} + \underline{e}$, we have:

$$s_i = (v_i + e_i) + (v_{n-k} + e_{n-k})p_{0i} + \cdots + (v_{n-1} + e_{n-1})p_{k-1,i}.$$

But $v_i + v_{n-k}p_{0i} + \cdots + v_{n-1}p_{k-1,i} = 0$ and

$$s_i = e_i + e_{n-k}p_{0i} + e_{n-k+1}p_{1i} + \cdots + e_{n-1}p_{k-1,i}, \qquad i = 0, 1, \cdots, n-k-1.$$

This shows that $n-k$ syndromes provide us with $n-k$ equations about error pattern. There are 2^n error patterns, but we have 2^{n-k} equations. So, we cannot catch all errors.

In fact, there are 2^k error patterns for each syndrome. To put it another way, the code C is a <u>subgroup</u> of the set of n-tuples. The set of n-tuples is partitioned into 2^{n-k} cosets of C. All the n-tuples in one coset result in the same syndrome. So, the syndrome only points us to a coset of C not to a single error pattern. Out of 2^k patterns (n-tuples in the coset), we decide (based on the property of the channel) which error has occurred.

Example: take again the $(7, 4)$ code. Assume that we receive $\underline{r} = (1001001)$. Then,

$$\underline{s} = \underline{r} \cdot H^T = (1, 1, 1).$$

This means that

$$1 = e_0 + e_3 + e_5 + e_6$$

$$1 = e_1 + e_3 + e_4 + e_5$$

$$1 = e_2 + e_4 + e_5 + e_6$$

Any of the following $2^4 = 16$ patterns satisfy these equations:

(0000010),	(1010011),
(1101010),	(0111011),
(0110110),	(1100111),
(1011110),	(0001111),
(1110000),	(0100001),
(0011000),	(1001001),
(1000100),	(0010101),
(0101100),	(1111101),

To decide which error to choose depends on our expectation about the channel behaviours. For example, in a BSC channel, we know that the probability of a single error is more than multiple errors. So, we decide $\underline{e} = (0000010)$ as the error and therefore, the codeword transmitted must have been:

$$\underline{v} = \underline{r} + \underline{e} = (1001001) + (0000010) = (1001011).$$

Minimum distance of a code:

Hamming distance $d(\underline{v}, \underline{w})$ between two vectors $\underline{v}$ and $\underline{w}$ is the number of places they are different. In binary case, the distance $d(\underline{v}, \underline{w})$ is the <u>weight</u> (the number of places a vector is non-zero) of $\underline{v} + \underline{w}$ or

$$d(\underline{v}, \underline{w}) = w(\underline{v}, \underline{w})$$

The <u>minimum distance</u> of a code C is the minimum value of $d(\underline{v}, \underline{w})$ for all non-identical $\underline{v}$ and $\underline{w} \in C$

$$d_{min} = min\{d(\underline{v}, \underline{w}): \underline{v}, \underline{w} \in C, \underline{v} \neq \underline{w}\}.$$

Since for any $\underline{v}$ and $\underline{w} \in C$, $\underline{v} + \underline{w} \in C$ then the minimum distance of a linear block code is equal to minimum weight of its non-zero codewords:

$$d_{min} = min\{w(\underline{v} + \underline{w}): \underline{v}, \underline{w} \in C, \underline{v} \neq \underline{w}\}$$

$$= min\{w(\underline{x}): \underline{x} \in C, \underline{x} \neq \underline{0}\}$$

$$= w_{min}.$$

Therefore, we have:

Theorem 1: the minimum distance of a linear block code is equal to the minimum weight of its non-zero codewords.

Theorem 2: let C be an (n, k) linear block code with parity check matrix H.

- For any codeword $\underline{v} \in C$ of weight l, there are l columns of H such that their vector sum is $\underline{0}$.
- If there are l columns of H whose vector sum is $\underline{0}$, then there is a codeword $\underline{v} \in C$ with weight l.

Proof: let $\underline{v} = (v_0, v_1, \cdots, v_{n-1})$ have l non-zero elements at places $i_1, i_2, \cdots, i_l$. Then,

$$\underline{v} \cdot H^T = \underline{0} \Rightarrow v_0 \underline{h}_0 + v_1 \underline{h}_1 + \cdots + v_{n-1} \underline{h}_{n-1} = \underline{0}$$

$$\Rightarrow v_{i_1} \underline{h}_{i_1} + v_{i_2} \underline{h}_{i_2} + \cdots + v_{i_l} \underline{h}_{i_l} = \underline{0}$$

$$\Rightarrow \underline{h}_{i_1} + \underline{h}_{i_2} + \underline{h}_{i_3} + \cdots + \underline{h}_{i_l} = \underline{0}$$

So, part 1 is proved.

Now assume that:

$$\underline{h}_{i_1} + \underline{h}_{i_2} + \underline{h}_{i_3} + \cdots + \underline{h}_{i_l} = \underline{0}.$$

Take $\underline{x} = (x_0, x_1, \cdots, x_{n-1})$ such that:

$$\begin{cases} x_j = 1 & at\ j = i_1, i_2, \cdots, i_l \\ x_j = 0 & otherwise. \end{cases}$$

Then,

$$\underline{x} \cdot H^T = x_0 \underline{h}_0 + x_1 \underline{h}_1 + \cdots + x_{n-1} \underline{h}_{n-1}$$

$$= x_{i_1} \underline{h}_{i_1} + x_{i_2} \underline{h}_{i_2} + \cdots + x_{i_l} \underline{h}_{i_l}$$

$$= \underline{h}_{i_1} + \underline{h}_{i_2} + \cdots + \underline{h}_{i_l} = \underline{0},$$

so, $\underline{x} \in C$.

Corollary 2.1: let C be a linear block code with parity check matrix H. If no $d-1$ or less columns of H add to $\underline{0}$, then minimum weight of H is at least d.

Corollary 2.2: the minimum distance of a linear block code C is the smallest number of columns of H adding to $\underline{0}$.

Error-detection and error-correction capability of a linear block code:

If the minimum distance of a code is d_{min}, it can detect <u>any error pattern with $d_{min} - 1$</u> or less errors.

Definition: assume that $A_0, A_1, A_2, \cdots, A_n$ are the number of codewords with weight $0, 1, 2, \cdots, n$ in a code C. $A_0, A_1, A_2, \cdots, A_n$ are called <u>weight distribution</u> of the code.

For example, for $(7, 4)$ Hamming code,

$$A_0 = A_7 = 1,\ A_3 = 7,\ A_4 = 7,\ \text{and } A_i = 0 \text{ otherwise.}$$

If we send a codeword $\underline{v}$ and we receive $\underline{r} = \underline{v} + \underline{e}$, we can detect errors unless $\underline{e} \in C$. So, $p_u(E) = \sum_{i=1}^{n} A_i (1-p)^{n-i} p^i$, where $p_u(E)$ is the probability of undetected error and p is the probability of error of modulation-demodulation.

For the $(7,4)$ code, we have:

$$p_u(E) = 7p^3(1-p)^4 + 7p^4(1-p)^3 + p^7.$$

So, if $p = 10^{-2}$, we get $p_u(E) = 7 \times 10^{-6}$. That is if one million bits are transmitted on the average 7 errors go through undetected.

Error correction capability:

A code C with minimum distance d_{min} can correct $t = \left\lfloor \frac{d_{min}-1}{2} \right\rfloor$ and less errors. ($\lfloor i \rfloor$ denotes the floor, i.e., the largest integer number less than i). $t = \left\lfloor \frac{d_{min}-1}{2} \right\rfloor$ means that $d_{min} = 2t+1$ or $d_{min} = 2t+2$ or $2t+1 \le d_{min} \le 2t+2$.

Triangle inequality: $d(\underline{v}, \underline{r}) + d(\underline{w}, \underline{r}) \ge d(\underline{v}, \underline{w})$

But: $d(\underline{v}, \underline{w}) \ge d_{min} \ge 2t+1$.

Let $d(\underline{v}, \underline{r}) = t'$, then: $d(\underline{w}, \underline{r}) \ge 2t + 1 - t'$. If $t' \le t$, then $d(\underline{w}, \underline{r}) \ge t$. This means if the distance between the received vector and the transmitted code is less than or equal to t, the received vector is closer to this codeword, say $\underline{v}$, than any other codeword $\underline{w}$.

A code C with minimum distance d_{min} can correct $t = \left\lfloor \frac{d_{min}-1}{2} \right\rfloor$ errors. It may correct some of the error patterns of weight higher than t, but it cannot correct all of those with $t+1$ errors. Probability of error is upper bounded as

$$p(E) \le \sum_{i=t+1}^{n} \binom{n}{i} p^i (1-p)^{n-i}.$$

Erasures:

Sometimes instead of deciding 0 or 1 at the output of the demodulator, we decide 0 and 1 for those received values far away zero and e or erasure for those close to zero.

A linear block code with d_{min} can correct γ errors and e erasures such that:

$$d_{min} \ge 2\gamma + e + 1.$$

Standard arrays:

We said that a code of length n and dimension k, i.e., and (n,k) code partitions the set V_n of n-tuples into 2^{n-k} cosets of the code C. If we write elements of C in a row and then from $2^n - 2^k$ remaining n-tuples take a vector $\underline{e}_2$, add $\underline{e}_2$ to each element of C and write in the second row, then take an unused element of the n-tuples say $\underline{e}_3$, add it to each codeword and write in the second row and continue this until we have used all n-tuples, we get a standard array.

$$\begin{array}{ccccccc}
\underline{v}_1 = 0 & & \underline{v}_2 & \cdots & \underline{v}_i & \cdots & \underline{v}_{2^k} \\
\underline{e}_2 & & \underline{e}_2 + \underline{v}_2 & \cdots & \underline{e}_2 + \underline{v}_i & \cdots & \underline{e}_2 + \underline{v}_{2^k} \\
\underline{e}_3 & & \underline{e}_3 + \underline{v}_2 & \cdots & \underline{e}_3 + \underline{v}_i & \cdots & \underline{e}_3 + \underline{v}_{2^k} \\
\vdots & & \vdots & & \vdots & & \vdots \\
\underline{e}_l & & \underline{e}_l + \underline{v}_2 & \cdots & \underline{e}_l + \underline{v}_i & \cdots & \underline{e}_l + \underline{v}_{2^k} \\
\vdots & & \vdots & & \vdots & & \vdots \\
\underline{e}_{2^{n-k}} & & \underline{e}_{2^{n-k}} + \underline{v}_2 & \cdots & \underline{e}_{2^{n-k}} + \underline{v}_i & \cdots & \underline{e}_{2^{n-k}} + \underline{v}_{2^k}
\end{array}$$

Theorem 3: no two n-tuples in the same row are identical. Every n-tuple is in one and only one row.

Proof: since C is a subgroup of V_n and each row is a coset of C.

Since a code C with minimum distance d_{min} can correct up to $t = \left\lfloor \frac{d_{min}-1}{2} \right\rfloor$ errors, we can use as the first coset leaders ($\underline{e}_i$'s) the patterns with t and less 1's. this covers for:

$$\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{t} = \sum_{i=0}^{t} \binom{n}{i}$$

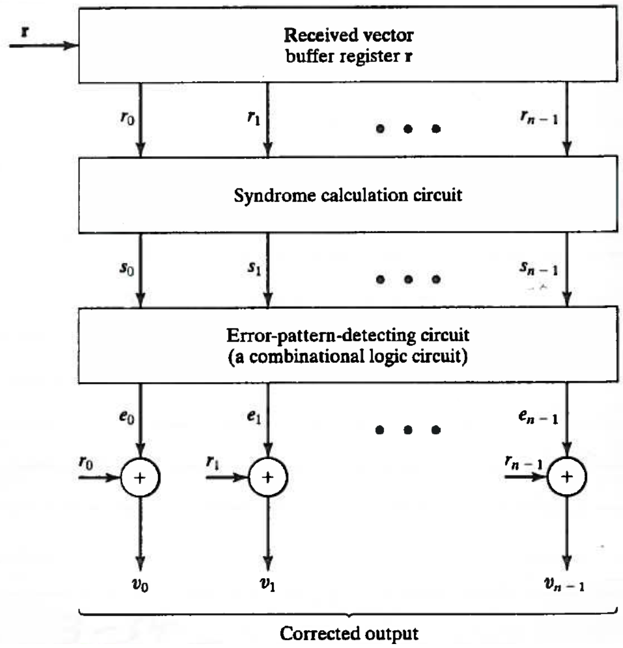
coset leaders, but this sum may not be equal to 2^{n-k}. So, we may add some error patterns with two or more errors.

Definition: if $\sum_{i=0}^{t} \binom{n}{i} = 2^{n-k}$, we say that the (n, k) code is perfect.

$(7, 4)$ code is perfect since it has $d_{min} = 3$ and therefore, $t = 1$ and

$$\sum_{i=0}^{t} \binom{n}{i} = \binom{7}{0} + \binom{7}{1} = 1 + 7 = 8 = 2^3 = 2^{n-k}.$$

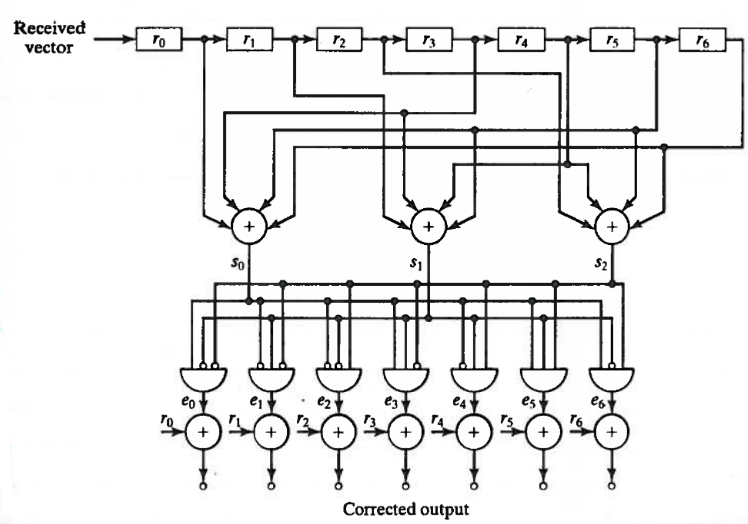
Note that since the elements on each row of the standard array are the 2^k codewords each added to a unique n-tuple (the coset leader), the syndromes of all numbers of a coset are the same. So, by finding the syndrome, we find out in what row of the standard array the received vector and hopefully the transmitted codeword is. We can the output the coset leader. For small codes, a lookup table is feasible. But for longer codes, we need to calculate the error based on the syndrome.



General decoder for a linear block code

Truth table for the error digits of the correctable error patterns of the (7, 4) linear code

Syndromes			Correctable error patterns (coset leaders)						
s_0	s_1	s_2	e_0	e_1	e_2	e_3	e_4	e_5	e_6
0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0
1	1	0	0	0	0	1	0	0	0
0	1	1	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1



Decoding circuit for the $(7, 4)$ code