Channel Coding Theory

Laura Cottatellucci

EURECOM

laura.cottatellucci@eurecom.fr

Outlines

I. Groups, Finite Fields, and Vector Spaces
II. Linear Block Codes
III. Error Detecting and Error Correcting Capabilities
IV. Error Correcting Decoders
V. Single Parity Check Codes
VI. Hamming Codes
VII. Hadamart Codes
VIII. Cyclic Codes
IX. References

Group: Definition

Definition: A group is an algebraic system $\langle G, * \rangle$, where G is a nonempty set and $*$ is an operation on pairs of elements of G such that

(A1) *(axiom of closure)* for every a and b in G, $a * b$ is also in G;

(A2) *(associative law)* for every a, b and c in G, $a * (b * c) = (a * b) * c$;

(A3) *(existence of a neutral element)* there is an element e of G such that $a * e = e * a = a$;

(A4) *(existence of inverses)* for every a in G there is b in G such that $a * b = b * a = e$. The element b is called the inverse of a and it is denoted by a^{-1}.

Group: Examples

Let $G = \mathbb{Z}_m = \{0, 1, \ldots m-1\}$

$\langle \mathbb{Z}_m, \oplus \rangle$ **where** $\oplus$ **is addition modulo** m.

$\langle \mathbb{Z}_p \backslash \{0\}, \odot \rangle$ **where** $\odot$ **is multiplication modulo a prime** p.

$\langle \mathbb{Z}_m^N, \oplus \rangle$ **where** $\mathbb{Z}_m^N$ **is the set of** N**-tuples whose components are in** $\mathbb{Z}_m$ **and** $\oplus$ **is component-by-component addition modulo** m.

Group: Exercises

Verify that the following algebraic systems are groups, i.e. verify closure and additivity, determine neutral element, inverse elements.

Case 1

Let $G = \{0, 1\}$ and

$\boxdot$	0	1
0	0	1
1	1	0

Case 2

Let $G = \{0, 1, 2, 3, 4, 5\}$ and addition modulo 6

$\boxdot$	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	4

Field: Definition

Definition: A field is an algebraic system $\langle \mathbb{F}, +, \cdot \rangle$, such that

(i) $\langle \mathbb{F}, + \rangle$, is a commutative, or Abelian, group (neutral element 0)

(ii) $\langle \mathbb{F} \backslash \{0\}, \cdot \rangle$, is a commutative, or Abelian, group (neutral element 1)

(iii) For every a, b and c in $\mathbb{F}$, the distributive law

$$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

holds, and multiplication[(1)] by zero obeys $a \cdot 0 = 0 \cdot a = 0$

[(1)] Required because $\langle \mathbb{F} \backslash \{0\}, \cdot \rangle$ does not define multiplication by 0

Fields: Examples

The field $\langle \mathbb{Q}, +, \cdot \rangle$ of rational numbers.

The field $\langle \mathbb{R}, +, \cdot \rangle$ of real numbers.

The field $\langle \mathbb{C}, +, \cdot \rangle$ of complex numbers.

Binary field with

$\boxplus$	0	1
0	0	1
1	1	0

and

$\circledast$	0	1
1	0	1

If p is a prime, $\langle \mathbb{G}, +, \cdot \rangle$ is a field with $\mathbb{G} = \{0, 1, \ldots, p-1\}$, $+$ is the modulo p addition and $\cdot$ is the modulo p multiplication.

Subfields and Finite Fields

Definition: If $\langle \mathbb{F}, +, \cdot \rangle$ is a field and $\mathbb{F}$ is a subset of $\mathbb{E}$ such that $\langle \mathbb{E}, +, \cdot \rangle$ is a field, then

- **$\langle \mathbb{F}, +, \cdot \rangle$ is a subfield of $\langle \mathbb{E}, +, \cdot \rangle$**
- **$\langle \mathbb{E}, +, \cdot \rangle$ is an extension field of $\langle \mathbb{F}, +, \cdot \rangle$**

Example: $\langle \mathbb{Q}, +, \cdot \rangle$ is a subfield of $\langle \mathbb{C}, +, \cdot \rangle$.

Definition: A field $\langle \mathbb{F}, +, \cdot \rangle$ such that $\mathbb{F}$ is finite is called Galois field or finite field.
Example: $\langle \mathbb{Z}_p, \oplus, \odot \rangle$ with p prime is a field denoted by $GF(p)$.

Fact: A finite field with q elements exists if and only if $q = p^m$ being p a prime and m a positive integer. This field is unique and denoted by $GF(q)$ or $GF(p^m)$.

Polynomials over a Field

Definition: A polynomial over a field $\langle \mathbb{F}, +, \cdot \rangle$ in the indeterminate X is a sum

$$A(X) = a_0 + a_1 X + a_2 X^2 + \ldots + a_n X^n$$

where the coefficient a_i is in $\mathbb{F}$, $i = 0, 1, 2, \ldots, n$.

Definition: The degree of $A(X)$, with $A(X) \neq 0$, is the greatest i such that $a_i \neq 0$ and it is denoted by $\deg[A(X)]$. The degree of $A(X) = 0$ is taken to be $-\infty$.

The set of all polynomials over the field $\langle \mathbb{F}, +, \cdot \rangle$ is denoted by $\mathbb{F}[X]$

Division Theorem for $\mathbb{F}[X]$: Given $N(X)$ and $D(X)$ in $\mathbb{F}[X]$, with $D(X) \neq 0$, there exist unique polynomials $Q(X)$ and $R(X)$ such that

$$N(X) = Q(X) \cdot D(X) + R(X)$$

and $\deg[R(X)] < \deg[D(X)]$.

Division of Polynomials over a Binary Field

Divide $X^7 + 1$ by $X^3 + X + 1$

$X^3 + X + 1$	X^4	$+X^2$	$+X$	$+1$			
	X^7						$+1$
	X^7	$+X^5$	$+X^4$				
		X^5	$+X^4$				$+1$
		X^5		$+X^3$	$+X^2$		
			X^4	$+X^3$	$+X^2$		$+1$
			X^4		$+X^2$	$+X$	$+1$
				X^3		$+X$	$+1$
				X^3		$+X$	$+1$
							0

Vector Spaces: Definition

Definition: A vector space is an algebraic system $\langle \mathcal{V}, \oplus, \mathbb{F}, +, \cdot, \odot \rangle$ such that for all c_1, c_2 in $\mathbb{F}$ and $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ in $\mathcal{V}$

(i) $\langle \mathbb{F}, +, \cdot \rangle$ is a field

(ii) $\langle \mathcal{V}, \oplus \rangle$ is an Abelian or commutative group (neutral element 0)

(iii) $\odot$ operate on pairs $(c, \boldsymbol{v})$ in $\mathbb{F} \times \mathcal{V}$ such that

- $c_1 \odot \boldsymbol{v}_1 \in \mathcal{V}$
- $1 \odot \boldsymbol{v}_1 = \boldsymbol{v}_1$
- $c_1 \odot (c_2 \odot \boldsymbol{v}_1) = (c_1 \cdot c_2) \odot \boldsymbol{v}_1$
- $(c_1 + c_2) \odot \boldsymbol{v}_1 = (c_1 \odot \boldsymbol{v}_1) \oplus (c_2 \odot \boldsymbol{v}_1)$
- $c_1 \odot (\boldsymbol{v}_1 \oplus \boldsymbol{v}_2) = (c_1 \odot \boldsymbol{v}_1) \oplus (c_1 \odot \boldsymbol{v}_2)$

Vector Spaces

Definition: If $\langle \mathcal{V}, \oplus, \mathbb{F}, +, \cdot \rangle$ is a vector space and $\mathcal{U}$ is a subset of $\mathcal{V}$ such that $\langle \mathcal{U}, \oplus, \mathbb{F}, +, \cdot \rangle$ is a vector space, then $\langle \mathcal{U}, \oplus, \mathbb{F}, +, \cdot \rangle$ is a subspace of $\langle \mathcal{V}, \oplus, \mathbb{F}, +, \cdot \rangle$.

- For vectors $\boldsymbol{v}_i$ in $\mathcal{V}$ and scalars c_i in $\mathbb{F}$ consider linear combinations of vectors

$$c_1\boldsymbol{v}_1 + c_2\boldsymbol{v}_2 + \ldots + c_n\boldsymbol{v}_n.$$

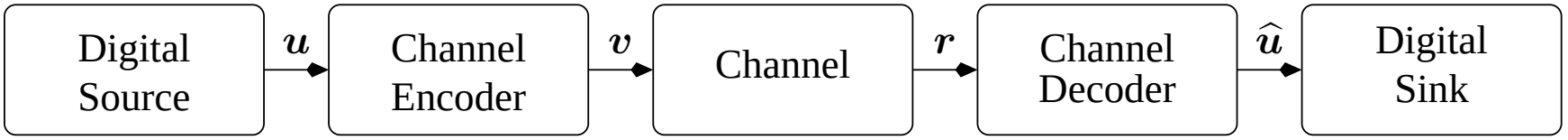
- The subspace spanned by $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ is the set of all linear combinations of these vectors, and it is denoted by $\mathcal{S}(\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n)$.
- The vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ are linearly independent if

$$c_1\boldsymbol{v}_1 + c_2\boldsymbol{v}_2 + \ldots + c_n\boldsymbol{v}_n = \mathbf{0}$$

  implies $c_1 = c_2 = \ldots = c_n = 0$. Otherwise the vectors are linearly dependent.
- If $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots \boldsymbol{v}_n$ are linearly independent and $\mathcal{V} = \mathcal{S}(\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots \boldsymbol{v}_n)$ then $(\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots \boldsymbol{v}_n)$ is called a basis for $\mathcal{V}$.

Block Codes: System Model and Definitions

Communication Model



A message is segmented in blocks of k bits, $\boldsymbol{u}$.

The encoder transforms $\boldsymbol{u}$ in a block of n bits $\boldsymbol{v}$ called codeword.

Definition: A binary block code of length n is a non-empty set $\mathcal{B}$ of binary vectors of length n. Equivalently, $\mathcal{B}$ a non-empty subset of $GF(2^n)$. It is denoted by $\mathcal{C}(n,k)$. The rate of the code is $\frac{\log_2|\mathcal{B}|}{n}$.

The theory for binary ($GF(2)$) block codes can be generalized to codes on any finite field $GF(q)$.

If $q=2^r$, a block code $\mathcal{C}(n,k)$ is equivalent to a block code $\mathcal{C}(nr,kr)$ in $GF(2)$.

Linear Block Code

Definition: An (n,k) binary linear block code is a k dimensional subspace V of $GF(2^n)$. The rate is thus $R = \frac{k}{n}$.

We restrict to binary linear block codes.

Advantages of linear block codes over general block codes:

- **Easier to analyze and understand (e.g. code distance property);**
- **Easier to implement encoder and, sometimes, decoder;**
- **Excellent performance of hard, soft, and iterative decoding.**

Generator Matrix

Let $g_1, g_2, \ldots g_k$ be a basis for V, the k dimensional subspace of the codewords. Then, every codeword can be written as

$$\boldsymbol{v} = u_1\boldsymbol{g}_1 + u_2\boldsymbol{g}_2 + \ldots u_k\boldsymbol{g}_k$$

Equivalently

$$\boldsymbol{v} = (u_1, u_2, \ldots, u_k) \begin{pmatrix} \boldsymbol{g}_1 \\ \boldsymbol{g}_2 \\ \vdots \\ \boldsymbol{g}_k \end{pmatrix} = \boldsymbol{uG}.$$

Any matrix G whose rows are a basis for the linear code V is a generator matrix for V.

Some properties

The encoder needs to memorize only the k row vectors $\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots \boldsymbol{g}_k$.

The properties of a vector space/subspace and the definition of linear block codes yield

Property 1: A linear block code consists of all possible sums of the rows of a generator matrix.

Property 2: The sum of two codewords is still a codeword.

Property 3: The n-tuple of all zeros is always a codeword.

Exercise

- Consider the following codebooks

$$\mathcal{C}_1 = \{(00000), (11001), (01100), (11111)\}$$
$$\mathcal{C}_2 = \{(00000), (11001), (00110), (11111)\}$$
$$\mathcal{C}_3 = \{(00000), (10101), (01010), (11111)\}$$

  which of them is a linear block code? Explain why, eventually, they are not linear block codes.

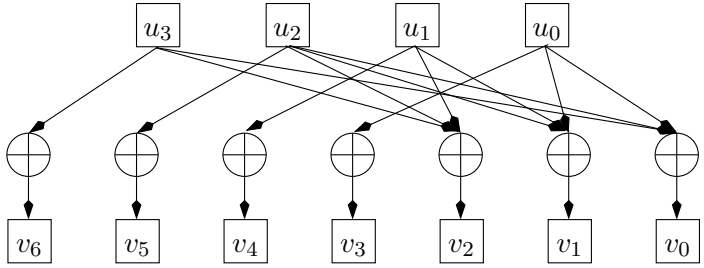
- Provide the generator matrices of the linear block codes in the previous item.
- Given the codebook $\mathcal{C}^* = \{101101, 111111\}$ add one or more codewords such that the resulting codebook is linear.

Examples and Implementation

A parity check code $(7, 4)$

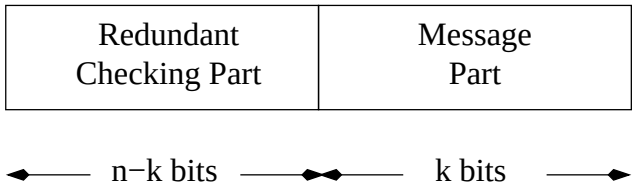
$$
\begin{aligned}
v_6 &= u_3 \\
v_5 &= u_2 \\
v_4 &= u_1 \\
v_3 &= u_0 \\
v_2 &= u_1 + u_2 + u_3 \\
v_1 &= u_0 + u_1 + u_2 \\
v_0 &= u_0 + u_2 + u_3
\end{aligned}
$$

$$
\boldsymbol{v} = (u_0, u_1, u_2, u_3) \underbrace{\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}}_{\text{Generator matrix}}
$$



Systematic Codes

Systematic Structure of a Codeword



Definition: An (n, k) block code is called systematic if it has a generator matrix of the form

$$\boldsymbol{G} = (\boldsymbol{P}, \boldsymbol{I}_k)$$

where $\boldsymbol{P}$ is a $k \times (n-k)$ matrix.

Any generator matrix can be reduced to a systematic generator matrix by linear combination of rows and/or column permutations.

Definition: Two codes whose code generator matrices can be obtained each other by linear combination of rows and/or column permutations are said equivalent.

Two equivalent codes have the same word error probability but not necessarily the same bit error probability.

Parity Check Coding Matrix

Definition: The equations uP are called the parity check equations of the code.

Equivalent representation of a code $\mathcal{C}(n,k)$ generated by $\boldsymbol{G}$:

An n-tuple v is a codeword of the code $\mathcal{C}(n,k)$ generated by $\boldsymbol{G} = (\boldsymbol{P}, \boldsymbol{I}_k)$ if and only if $v\boldsymbol{H}^T = 0$ with

$$\boldsymbol{H} = (\boldsymbol{I}_{n-k}, \boldsymbol{P}^T).$$

Definition: The matrix H is called the parity check matrix.

Property The matrix H is the null space of G, i.e. G and H are orthogonal

$$\boldsymbol{G}\boldsymbol{H}^T = 0$$

Exercise

Consider a linear block code defined by the parity check equations below and determine the parity check coding matrix and the generator matrix

Parity Check Equations

$$
\begin{aligned}
x_1 + x_3 + x_4 &= 0 \\
x_3 + x_5 &= 0 \\
x_1 + x_3 + x_6 &= 0 \\
x_2 + x_3 + x_7 &= 0 \\
x_1 + x_2 + x_8 &= 0 \\
x_2 + x_9 &= 0 \\
x_1 + x_2 + x_{10} &= 0
\end{aligned}
$$

Parity Check and Generator Matrices

$$
\boldsymbol{H} = \begin{pmatrix}
101 & 1000000 \\
001 & 0100000 \\
101 & 0010000 \\
011 & 0001000 \\
110 & 0000100 \\
010 & 0000010 \\
110 & 0000001
\end{pmatrix}
$$

$$
\boldsymbol{G} = \begin{pmatrix}
100 & 1010101 \\
010 & 0001111 \\
001 & 1111000
\end{pmatrix}
$$

Syndrome and Error Detection

Error Vector

$$\boldsymbol{e} = \boldsymbol{r} + \boldsymbol{v}$$

with

$$e_j = \begin{cases} 0 & \text{for } r_j = v_j, \\ 1 & \text{for } r_j \neq v_j. \end{cases}$$

Definition: A syndrome is the $(n-k)$-dimensional vector $\boldsymbol{s} = \boldsymbol{r}\boldsymbol{H}^T$

$$\boldsymbol{s} \neq 0 \quad \Rightarrow \boldsymbol{r},\, \boldsymbol{e} \notin \mathcal{C}(n,k) \qquad \text{Detected Error}$$

$$(\boldsymbol{s} = 0) \wedge (\boldsymbol{e} \neq \boldsymbol{0}) \quad \Rightarrow \quad \boldsymbol{r},\, \boldsymbol{e} \in \mathcal{C}(n,k) \qquad \text{Undetectable Error}$$

($2^k - 1$ undetectable errors)

The syndrome sums up the received parity check digits $(r_0, r_1, \ldots r_{n-k-1})$ and the parity check digits recomputed from the received information digits $(r_{n-k}, r_{n-k+1}, \ldots r_n)$.

Exercise

- Consider the encoder proposed at page 21 and determine its codewords.
- Determine the number of undetectable error patterns of weight 1, 2, 3, 4, 5, 6, 7, 8, 9.
- Determine the percentage of error pattern of weight 1,2,...9 that can be detected by the code.
- Given a BSC with error probability ϵ, determine the probability that an error pattern of weight 5 occurs.

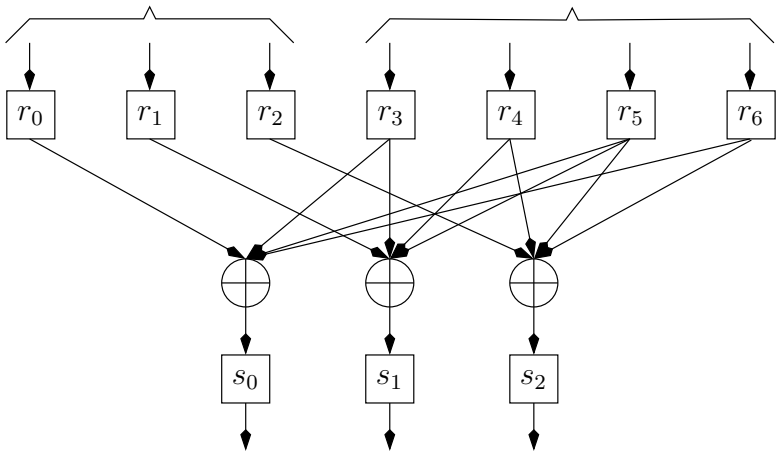
Syndrome Circuit: An Example

Parity Check Equations

$$\boldsymbol{v}_0 = \boldsymbol{u}_0 + \boldsymbol{u}_2 + \boldsymbol{u}_3$$

$$\boldsymbol{v}_1 = \boldsymbol{u}_0 + \boldsymbol{u}_1 + \boldsymbol{u}_2$$

$$\boldsymbol{v}_2 = \boldsymbol{u}_1 + \boldsymbol{u}_2 + \boldsymbol{u}_3$$



$$\boldsymbol{s} = \boldsymbol{r}\boldsymbol{H}^T = (\boldsymbol{v} + \boldsymbol{e})\boldsymbol{H}^T = \boldsymbol{e}\boldsymbol{H}^T$$

Linear system of $n - k$ equations in n unknowns. $\Rightarrow$ There are 2^k possible solutions. $\Rightarrow$ Among all possible solutions we choose the one that minimize the error codeword probability.

An Example

Let us transmit a codeword $\boldsymbol{v} = (1001011)$ of the $(7, 4)$ code proposed in slide 18 and let us receive $\boldsymbol{r} = (1001001)$

$$\boldsymbol{s} = \boldsymbol{r}\boldsymbol{H}^T = \boldsymbol{e}\boldsymbol{H}^T = (1, 1, 1)$$

System of linear equations for the error

$$\begin{aligned} 1 &= e_0 + e_3 + e_5 + e_6 \\ 1 &= e_1 + e_3 + e_4 + e_5 \\ 1 &= e_2 + e_4 + e_5 + e_6 \end{aligned}$$

Solutions

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

$\boldsymbol{e}^* = (0000010)$ is the most probable error vector for memoryless BSC.
We assume $\boldsymbol{e}^*$ to be actual error and reconstruct the transmitted vector

$$\begin{aligned} \widehat{\boldsymbol{v}} &= \boldsymbol{r} + \boldsymbol{e}^* \\ &= (1001001) + (0000010) \\ &= (1001011) \end{aligned}$$

Hamming Distance

Definition: $w(\boldsymbol{v})$, **Hamming weight of** $\boldsymbol{v}$ **is the number of nonzero components of** $\boldsymbol{v}$.

Definition: The minimum weight $w_{\min}$ **of a linear code** $\mathcal{C}(n,k)$ **is the minimum weight of its nonzero codewords, i.e.** $w_{\min} = \min_{\boldsymbol{x}\in\{\boldsymbol{x}\in\mathcal{C},\boldsymbol{x}\neq\mathbf{0}\}} w(\boldsymbol{x})$.

Definition: $d(\boldsymbol{v},\boldsymbol{w})$, **the Hamming distance between the vectors** $\boldsymbol{v}$ **and** $\boldsymbol{w}$ **is the number of elements where** $\boldsymbol{v}$ **and** $\boldsymbol{w}$ **differ, or equivalently,** $d(\boldsymbol{v},\boldsymbol{w}) = w(\boldsymbol{v}+\boldsymbol{w})$.

Definition: The minimum distance of a linear code $\mathcal{C}(n,k)$, **denoted by** $d_{\min}$ **is defined as**

$$d_{\min} = \min\{d(\boldsymbol{v},\boldsymbol{w}) : \boldsymbol{v},\boldsymbol{w} \in \mathcal{C}(n,k), \boldsymbol{v} \neq \boldsymbol{w}\}$$

Theorem: The minimum distance of a linear block code is equal to the minimum weight of its nonzero codewords and viceversa.

$$\begin{aligned} d_{\min} &= \min\{d(\boldsymbol{v},\boldsymbol{w}) : \boldsymbol{v},\boldsymbol{w} \in \mathcal{C}(n,k), \boldsymbol{v} \neq \boldsymbol{w}\} \\ &= \min\{w(\boldsymbol{x}) : \boldsymbol{x} \in \mathcal{C}(n,k), \boldsymbol{x} \neq \mathbf{0}\} \\ &= w_{min} \end{aligned}$$

Minimum Distance

Theorem: Let $\mathcal{C}(n,k)$ be a linear code with parity check matrix H. For each codeword of Hamming weight ℓ, there exist ℓ columns of H such that the vector sum of these columns is equal to the zero vector. Conversely, if there exist ℓ columns of H whose vector sum is the zero vector, there exists a codeword of Hamming weight ℓ.

Corollary: Let $\mathcal{C}(n,k)$ be a linear block code with parity check matrix H. If no $d-1$ columns of H add to zero, the code has minimum distance at least d.

Corollary: Let $\mathcal{C}(n,k)$ be a linear block code with parity check matrix H. The minimum distance of the code is equal to the smallest number of columns of H that sum to 0.

Error Detection Capabilities

- If a code has minimum distance $d_{\min}$ it can detect any error pattern of weight not greater than $d_{\min} - 1$.
- If a code has minimum distance $d_{\min}$ there are $2^n - 2^k$ detectable error patterns even with $d_{\min}$ or more errors .
- If a linear block code has minimum distance $d_{\min}$there are $2^k - 1$ undetectable error patterns.

Definition: Let A_i be the number of codewords of weight i in the linear block code $\mathcal{C}(n, k)$. The numbers $A_1, A_2, \ldots A_n$ are called the weight distribution of $\mathcal{C}(n, k)$.

Probability of Undetected Error

$$P_u(E) = \sum_{i=1}^{n} A_i \varepsilon^i (1 - \varepsilon)^{n-i}$$

** Compute the probability of undetected error for the code at page 21. Make use of the intermediate results obtained for the exercise at page 23.

Error Correction Capabilities

Theorem: If a linear block code $\mathcal{C}(n,k)$ had minimum distance $d_{\min}$, it is capable to correct all error patterns of t or fewer errors with

$$2t+1 \leq d_{\min} \leq 2t+2.$$

Proof: Let transmit $\boldsymbol{v}$ and receive $\boldsymbol{r}$. For another codeword $\boldsymbol{w} \in \mathcal{C}(n,k)$

$$d(\boldsymbol{v},\boldsymbol{r}) + d(\boldsymbol{w},\boldsymbol{r}) \geq d(\boldsymbol{v},\boldsymbol{w})$$

then $d(\boldsymbol{w},\boldsymbol{v}) \geq d_{\min} \geq 2t+1$.

Let an error pattern with t' errors occur $\Rrightarrow d(\boldsymbol{v},\boldsymbol{r}) = t'$. Then

$$\begin{aligned} d(\boldsymbol{w},\boldsymbol{r}) &\geq d(\boldsymbol{w},\boldsymbol{v}) - d(\boldsymbol{v},\boldsymbol{r}) \\ &\geq 2t+1-t' \end{aligned}$$

If $t' < t$ then $d(\boldsymbol{w},\boldsymbol{r}) > t$ and for a BSC channel $\boldsymbol{v}$ is the codeword closest to $\boldsymbol{r}$. Applying maximum likelihood decoding $P(\boldsymbol{r}|\boldsymbol{v})$ is greater than $P(\boldsymbol{r}|\boldsymbol{w})$ for $\boldsymbol{v} \neq \boldsymbol{w}$ and the decoder detects correctly $\boldsymbol{v}$.

In contrast, it can be shown that for $t' > t$ there exists at least an error pattern that can not be reconstructed correctly.

Error Correction Capabilities

Definition: The parameter $t = \left\lfloor \frac{d_{\min}-1}{2} \right\rfloor$ is called the error correcting capability of the code.

Definition: The code is referred to as a t-error correcting code.

Upper bound on the Probability of Erroneous Decoding in BSC

$$P(E) \leq \sum_{i=t+1}^{n} \binom{n}{i} \varepsilon^i (1-\varepsilon)^{n-i}$$

Combined Error Correction and Error Detection Capabilities

Theorem: A code can correct λ or fewer errors and simultaneously can detect ℓ, with $\ell > \lambda$ errors if

$$d_{\min} \geq \lambda + \ell + 1.$$

Erasures

A receiver may declare an erasure when a symbol is received ambiguously or unreliably. The decoders receives in this case a vector $\boldsymbol{r}$ of $0, 1,$ and erasures.

Theorem: A code of minimum distance $d_{\min}$ can correct any pattern of ν errors and e erasures provided that

$$d_{\min} \geq 2\nu + e + 1.$$

Proof: Delete from all the codewords the e components where the receiver has declared erasures and obtain a code $\mathcal{C}'$.

The minimum distance of $\mathcal{C}'$ is at least $d_{\min} - e \geq 2\nu + 1$. For $\mathcal{C}'$ we can correct ν errors.

We correct ν errors in $\mathcal{C}'$ and then we come back to $\mathcal{C}$. Because $d_{\min} \geq e + 1$, there is one and only one codeword in $\mathcal{C}$ that agrees with the unerased components. We can recover the transmitted codeword.

Since $d_{\min}$ plays a key role in the performance of $\mathcal{C}$ often a code is denoted by $\mathcal{C}(n, k, d_{\min})$.

Example of Nonlinear Block Code

Consider the following randomly generated codebook (7,3):

$$\mathcal{C} = \{0010000, 1001011, 0010101, 0001110, 1011001, 1000001, 0001111, 1000100\}\,.$$

The Hamming distances between pairs of codewords are shown in the following table.

index	1	2	3	4	5	6	7	8
1	0	5	2	4	3	3	5	3
2	5	0	5	3	2	2	2	4
3	2	5	0	4	3	3	3	3
4	4	3	4	0	5	5	1	3
5	3	2	3	5	0	2	4	4
6	3	2	3	5	2	0	4	2
7	5	2	3	1	4	4	0	4
8	3	4	3	3	4	2	4	0

1. What is the minimum distance of the code?
2. For equally distributed messages, determine the probability of undetected errors.
3. Provide an error pattern with weight 1 that determines an undetected error.

Error Correcting Decoders

General Approach:

- The 2^n n-tuple over $GF(2)$ are partitioned into 2^k disjoint sets $D_1, D_2, \ldots D_{2^k}$ such that the codeword $\boldsymbol{v}_i$ is contained in D_i, for $1 \leq i \leq 2^k$.
- If the received vector $\boldsymbol{r}$ is found in the subset D_j, $\boldsymbol{r}$ is decoded into $\boldsymbol{v}_j$.
- The detection is correct if and only if $\boldsymbol{r}$ is in the set D_i that corresponds to the transmitted codeword.

Complexity: storage of 2^k sets consisting of a total of 2^n vectors of n bits ($n2^n$ bits) and repeated comparison of the received vector.

If n and k are large the complexity is unacceptable!

Standard Array of a Linear Block Code

A standard array is a rule to partition the 2^n possible received signal into 2^k disjoint sets.

1. Place the 2^k codewords of C in a row with all-zero codeword $\boldsymbol{v}_1 = (00\ldots0)$ in the first (leftmost) position;
2. From the $2^n - 2^k$ remaining elements of $GF(2^n)$ choose an n-tuple $\boldsymbol{e}_2$ and place it under the vector $\boldsymbol{v}_1$;
3. Build the second row by adding $\boldsymbol{e}_2$ to each codeword $\boldsymbol{v}_i$ in the first row;
4. Build the third row by choosing a vector $\boldsymbol{e}_3$ that does not appears in the first two rows and add it to each $\boldsymbol{v}_i$, $1 \leq i \leq 2^k$;
5. Repeat the previous step until all elements of $GF(2^n)$ are in the table.

$\boldsymbol{v}_1 = \boldsymbol{0}$	$\boldsymbol{v}_2$	$\ldots$	$\boldsymbol{v}_j$	$\ldots$	$\boldsymbol{v}_{2^k}$
$\boldsymbol{e}_2$	$\boldsymbol{e}_2 + \boldsymbol{v}_2$	$\ldots$	$\boldsymbol{e}_2 + \boldsymbol{v}_j$	$\ldots$	$\boldsymbol{e}_2 + \boldsymbol{v}_{2^k}$
$\boldsymbol{e}_3$	$\boldsymbol{e}_3 + \boldsymbol{v}_2$	$\ldots$	$\boldsymbol{e}_3 + \boldsymbol{v}_j$	$\ldots$	$\boldsymbol{e}_3 + \boldsymbol{v}_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$\boldsymbol{e}_\ell$	$\boldsymbol{e}_\ell + \boldsymbol{v}_2$	$\ldots$	$\boldsymbol{e}_\ell + \boldsymbol{v}_j$	$\ldots$	$\boldsymbol{e}_\ell + \boldsymbol{v}_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$\boldsymbol{e}_{2^{n-k}}$	$\boldsymbol{e}_{2^{n-k}} + \boldsymbol{v}_2$	$\ldots$	$\boldsymbol{e}_{2^{n-k}} + \boldsymbol{v}_j$	$\ldots$	$\boldsymbol{e}_{2^{n-k}} + \boldsymbol{v}_{2^k}$

Some Definitions

- The 2^{n-k} rows of the standard array are called the cosets of the code C;
- The first n-tuple $\boldsymbol{e}_j$ of each coset is called a coset leader or coset representative;
- Any element of a coset can be used as its coset leader. This does not change the elements of the coset, it simply permutes them.
- The standard array induces a partition$^{(*)}$ of $GF(2^n)$ in 2^k sets

$$D_j = \{\boldsymbol{v}_j, \boldsymbol{e}_2 + \boldsymbol{v}_j, \boldsymbol{e}_3 + \boldsymbol{v}_j, \ldots \boldsymbol{e}_{2^{n-k}} + \boldsymbol{v}_j\}$$

Coset Leader			D_j		
$\boldsymbol{v}_1 = \boldsymbol{0}$	$\boldsymbol{v}_2$	$\ldots$	$\boldsymbol{v}_j$	$\ldots$	$\boldsymbol{v}_{2^k}$
$\boldsymbol{e}_2$	$\boldsymbol{e}_2 + \boldsymbol{v}_2$	$\ldots$	$\boldsymbol{e}_2 + \boldsymbol{v}_j$	$\ldots$	$\boldsymbol{e}_2 + \boldsymbol{v}_{2^k}$
$\boldsymbol{e}_3$	$\boldsymbol{e}_3 + \boldsymbol{v}_2$	$\ldots$	$\boldsymbol{e}_3 + \boldsymbol{v}_j$	$\ldots$	$\boldsymbol{e}_3 + \boldsymbol{v}_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$\boldsymbol{e}_\ell$	$\boldsymbol{e}_\ell + \boldsymbol{v}_2$	$\ldots$	$\boldsymbol{e}_\ell + \boldsymbol{v}_j$	$\ldots$	$\boldsymbol{e}_\ell + \boldsymbol{v}_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$\boldsymbol{e}_{2^{n-k}}$	$\boldsymbol{e}_{2^{n-k}} + \boldsymbol{v}_2$	$\ldots$	$\boldsymbol{e}_{2^{n-k}} + \boldsymbol{v}_j$	$\ldots$	$\boldsymbol{e}_{2^{n-k}} + \boldsymbol{v}_{2^k}$

$^{(*)}$ We will show later that a standard array induces a partition on $GF(2^n)$.

Example

Consider the $(6, 3)$ linear code with generator matrix

$$\boldsymbol{G} = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The standard array is

Coset Leader							
000000	011100	101010	110001	110110	101101	011011	000111
100000	111100	001010	010001	010110	001101	111011	100111
010000	001100	111010	100001	100110	111101	001011	010111
001000	010100	100010	111001	111110	100101	010011	001111
000100	011000	101110	110101	110010	101001	011111	000011
000010	011110	101000	110011	110100	101111	011001	000101
000001	011101	101011	110000	110111	101100	011010	000110
100100	111000	001110	010101	010010	001001	111111	100011

Some Properties of a Standard Array

Theorem: No two n-tuples in the same row of a standard array are identical. Every n-tuple is in one and only one row.

Proof: 1) Let us assume that two n-tuples on the same line are identical. Then,

$$\begin{aligned} & \boldsymbol{e}_\ell + \boldsymbol{v}_j = \boldsymbol{e}_\ell + \boldsymbol{v}_i, i \neq j \\ \Longrightarrow \quad & \boldsymbol{v}_j = \boldsymbol{v}_i \end{aligned}$$

which is impossible.

2) Assume two n-tuples in different rows are identical

$$\begin{aligned} \boldsymbol{e}_m + \boldsymbol{v}_i = \boldsymbol{e}_n + \boldsymbol{v}_j \qquad & \text{with } m < n \\ \boldsymbol{e}_n = \boldsymbol{e}_m + \boldsymbol{v}_i + \boldsymbol{v}_j = \boldsymbol{e}_m + \boldsymbol{v}_k \qquad & \text{with } \boldsymbol{v}_k = \boldsymbol{v}_i + \boldsymbol{v}_j \in C \end{aligned}$$

$\boldsymbol{e}_n$ should be in the coset having as coset leader $\boldsymbol{e}_m$ but this is impossible for the construction of the standard array.

A standard array induces a partition in $GF(2^n)$.

Some Properties of a Standard Array (cntd)

Theorem: Every (n, k) linear block code is capable of correcting 2^{n-k} error patterns.

Proof: It follows from the fact that errors can be corrected only if the error pattern is a coset leader.

In fact, if the error is a coset leader $\boldsymbol{e}_\ell$ and $\boldsymbol{v}_j$ is transmitted $\boldsymbol{r} = \boldsymbol{e}_\ell + \boldsymbol{v}_j$ is in D_j and $\boldsymbol{v}_j$ is detected properly.

If the error $\boldsymbol{x}$ is not a coset leader, then $\boldsymbol{x}$ is in the standard array as $\boldsymbol{x} = \boldsymbol{v}_m + \boldsymbol{e}_\ell$. The received vector is

$$\boldsymbol{r} = \boldsymbol{v}_j + \boldsymbol{x} = \boldsymbol{v}_j + \boldsymbol{v}_m + \boldsymbol{e}_\ell = \boldsymbol{v}_s + \boldsymbol{e}_\ell \text{ with } \boldsymbol{v}_j + \boldsymbol{v}_m = \boldsymbol{v}_s \in C.$$

Then, $\boldsymbol{r} \in D_s$ and erroneously we detect $\boldsymbol{v}_s$ instead of $\boldsymbol{v}_j$.

Standard Array and Maximum Likelihood Detection

- The probability of a decoding error is minimized if the most frequent error patterns are coset leaders;
- In a BSC an error pattern of smaller weight is more probable than an error pattern of larger weight;
- In a BSC, a standard array should be built choosing as coset leader the n-tuple with least weight among the remaining n-tuples.
- The partition induced by such a standard array yields the maximum likelihood decoder.

Definition: **Let α_i denote the number of coset leader with weight i. The numbers $\alpha_0, \alpha_1, \ldots \alpha_n$ are called the weight distribution of the coset leaders.**

Error Probability for a BSC

$$P(E) = 1 - \sum_{i=0}^{n} \alpha_i \varepsilon^i (1-\varepsilon)^{n-i}.$$

Some Properties of a Standard Array (cntd)

Theorem: All the 2^k n-tuples of a coset have the same syndrome. The syndromes for different cosets are different.

Proof:

1) Let us consider a coset with coset leader $\boldsymbol{e}_i$. Any vector $\boldsymbol{v}_j + \boldsymbol{e}_i$ belonging to such a coset has syndrome

$$(\boldsymbol{v}_j + \boldsymbol{e}_i)\boldsymbol{H}^T = \boldsymbol{e}_i\boldsymbol{H}^T.$$

Then, all n-tuples belonging to a coset have the same syndrome as the coset leader.

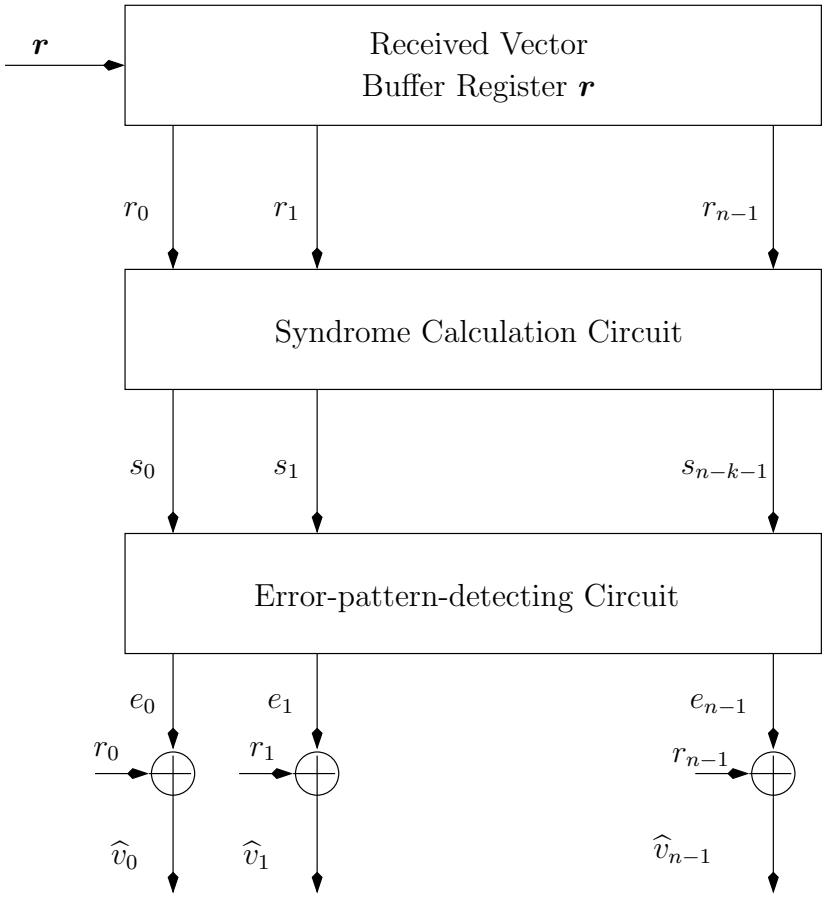
2) Let us consider to different cosets with coset leaders $\boldsymbol{e}_m$ and $\boldsymbol{e}_n$ with $m < n$. If they had had the same syndrome then

$$\begin{aligned} \boldsymbol{e}_m\boldsymbol{H}^T &= \boldsymbol{e}_n\boldsymbol{H}^T \\ (\boldsymbol{e}_m + \boldsymbol{e}_n)\boldsymbol{H}^T &= \boldsymbol{0} \\ \boldsymbol{v}_\ell\boldsymbol{H}^T &= \boldsymbol{0} \qquad \boldsymbol{v}_\ell = \boldsymbol{e}_m + \boldsymbol{e}_n \in C. \end{aligned}$$

Then, $\boldsymbol{e}_n = \boldsymbol{e}_m + \boldsymbol{v}_\ell$ which is impossible because of the construction of the standard array.

Syndrome Decoding or Table Look-up Decoding

1. Compute the syndrome of $\boldsymbol{r}$, $\boldsymbol{r}\boldsymbol{H}^T$;
2. Locate the coset leader $\boldsymbol{e}_\ell$ whose syndrome is equal to $\boldsymbol{r}\boldsymbol{H}^T$.
3. Assume $\boldsymbol{e}_\ell$ to be the actual error pattern;
4. Decode $\boldsymbol{r}$ into the codeword $\boldsymbol{v}^* = \boldsymbol{r} + \boldsymbol{e}_\ell$.



Dual Codes and Weight Distributions

Definition: Given the parity check matrix $\boldsymbol{H}$ of a code $\mathcal{C}$, the $(n, n-k)$ linear block code $\mathcal{C}_d$ with generator matrix $\boldsymbol{H}$ is called the dual code of $\mathcal{C}$.

Mac Williams identity: Let $\{A_0, A_1, \ldots A_n\}$ and $\{B_0, B_1, \ldots B_n\}$ be the weight distributions of $\mathcal{C}$ and $\mathcal{C}_d$, respectively. Let

$$A(z) = A_0 + A_1 z + \ldots + A_n z^n \qquad \text{and} \qquad B(z) = B_0 + B_1 z + \ldots + B_n z^n$$

be the corresponding weight enumerating functions. Then,

$$A(z) = 2^{-(n-k)}(1+z)^n B\left(\frac{1-z}{1+z}\right)$$

- The weight distribution is directly related to the probability of undetected error.
- For large n and k exhaustive computation to determine $A(z)$ is impossible.
- If $n - k \ll k$ it is useful to compute the weight distribution of a code $\mathcal{C}$ via the enumerating function of $\mathcal{C}_d$.

Again on Undetected Error Probability

The undetected error probability $P_u(E) = \sum_{i=1}^{n} A_i \varepsilon^i (1-\varepsilon)^{n-i}$ for a BSC can be expressed in terms of the weight enumerator functions $A(z)$ and $B(z)$ as

$$\begin{aligned} P_u(E) &= (1-\varepsilon)^n \left(A\left(\frac{\varepsilon}{1-\varepsilon}\right) - 1 \right) \\ &= 2^{-(n-k)} B(1-2\varepsilon) - (1-\varepsilon)^n \end{aligned}$$

The weight distributions of many linear codes are still unknown!

Average Undetected Error Probability over All (n, k) Linear Codes

Let us consider the set Γ of all (n, k) linear codes, $\boldsymbol{P}_u(\boldsymbol{E})$ the probability of undetected errors choosing randomly a code in Γ is upper bounded by:

$$\begin{aligned}\boldsymbol{P}_u(\boldsymbol{E}) &= \sum_{j=1}^{|\Gamma|} P(C_j) P_u(E|C_j) \\ &\leq 2^{-(n-k)}[1-(1-\varepsilon)^n].\end{aligned}$$

There exists (n, k) linear codes with $P_u(E)$ decreasing exponentially with the number of parity check digits $n - k$!

Remarks on Linear Block Codes

Decoding	Complexity
Full Search	storage: $n \cdot 2^n$ bits search over 2^n elements
Syndrome-Based	storage (syndrome & coset leader): $(n-k) \cdot 2^{n-k} + n \cdot 2^{n-k}$ bits search over 2^{n-k} elements

Complexity still too high!

Performance analysis based on the weight enumerating function (WEF)...

...but WEF unknown for the most of the codes!

Simplify decoding and performance analysis by introducing structure in codes!

Single-Parity-Check Codes (SPC)

A single-parity-check code is a $(k+1, k)$ linear code with parity check equation

$$p = u_0 + u_1 + \ldots + u_{k-1}$$

Generator Matrix

$$\boldsymbol{G} = \left(\begin{array}{c|cccc} 1 & 1 & 0 & \ldots & 0 \\ 1 & 0 & 1 & \ldots & 0 \\ 1 & 0 & 0 & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 0 & \ldots & 0 & 1 \end{array}\right) = \left(\begin{array}{c|c} 1 & \\ 1 & \\ 1 & \boldsymbol{I}_k \\ \vdots & \\ 1 & \end{array}\right)$$

Parity Check Matrix

$$\boldsymbol{P} = \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \end{pmatrix}$$

- Each codeword has even weight.
- All the error patterns with odd weight are detectable, all error patterns with even weight are undetectable ($d_{min} = 2$).
- The dual code of a single-error-check code is the repetition code.
- SPC codes are used as component codes to construct long, powerful codes.

Hamming Codes

For any positive integer $m \geq 3$ there exists a Hamming code with

code length	$n = 2^m - 1$
number of information bits	$k = 2^m - m - 1$
number of parity-check bits	$n - k = m$

whose parity check matrix $\boldsymbol{H}$ consists of all nonzero n-tuples as its columns.

Systematic Form

$$\boldsymbol{H} = [\, \boldsymbol{I}_m \,|\, \boldsymbol{Q} \,]$$

$\boldsymbol{Q}$ is the $m \times (2^m - m - 1)$ matrix of the m-tuples of weight 2 or more.

Lexicographic Form **The columns appear in lexicographic order. Example**

$$\boldsymbol{H} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Each column identifies the column position. The syndrome expresses in binary digit the error position in r.

Hamming Codes: Properties

The minimum distance of a Hamming code is $d_{min} = 3$.

1. Because two columns of $\boldsymbol{H}$ are nonzero and distinct, no two columns add to zero. Thus, $d_{min} > 0$.
2. Given two columns of $\boldsymbol{H}$, $\boldsymbol{h}_\ell$ and $\boldsymbol{h}_m$ their sum is also a column of $\boldsymbol{H}$, $\boldsymbol{h}_n$. Thus, $\boldsymbol{h}_\ell + \boldsymbol{h}_m + \boldsymbol{h}_n = \boldsymbol{0}$ and $d_{min} = 3$.

A Hamming code can correct all error patterns with unit weight.

A Hamming code can detect all error patterns with two or fewer errors.

The rate of a Hamming code $R = \frac{2^m - m - 1}{2^m - 1} \to 1$ as $m \to \infty$.

Hamming Codes: Properties Cntd

In a Hamming code each received sequence is at unit distance from a code-word. A code with such a property is called perfect code.

The weight enumerating function of a Hamming code is known

$$A(z) = \frac{1}{n+1}\left((1+z)^n + n(1-z)(1-z^2)^{(n-1)/2}\right)$$

The dual code of a $(2^m-1, 2^m-m-1)$ Hamming code is a $(2^m-1, m)$ linear code with weight distribution function

$$B(z) = 1 + (2^m-1)z^{2^{m-1}}$$

The nonzero codewords of a dual Hamming code $(2^m-1, m)$ have all identical weight 2^{m-1}. The code distance is $d_{min} = 2^{m-1}$. The dual code of a Hamming code is also called maximal-length or equidistant or simplex code.

Single-Error-Correcting and Double-Error-Detecting Code

SEC-DED codes are fast in encoding and error-correction decoding.

Construction of SEC-DED Codes

Given a parity check matrix H of a Hamming code delete columns from H to obtain a parity check matrix H_0 that satisfies the following requirements:

1. **Every column should have an odd number of 1's.**
2. **The total number of 1's in H_0 should be minimum.**
3. **The total number of 1's in each row of H_0 should be made equal, or as close as possible to the average number (i.e. the total number of 1's in H_0 divided the number of rows.)**

Requirement 1 implies that the code with parity check matrix H_0 has minimum distance at least $d_{min} = 4$.

Requirements 2 and 3 yield minimum logic levels in forming parity check and syndrome bits ($\Rightarrow$ syndrome bits formed simultaneously and received bits corrected in parallel.)

Interleaved Codes

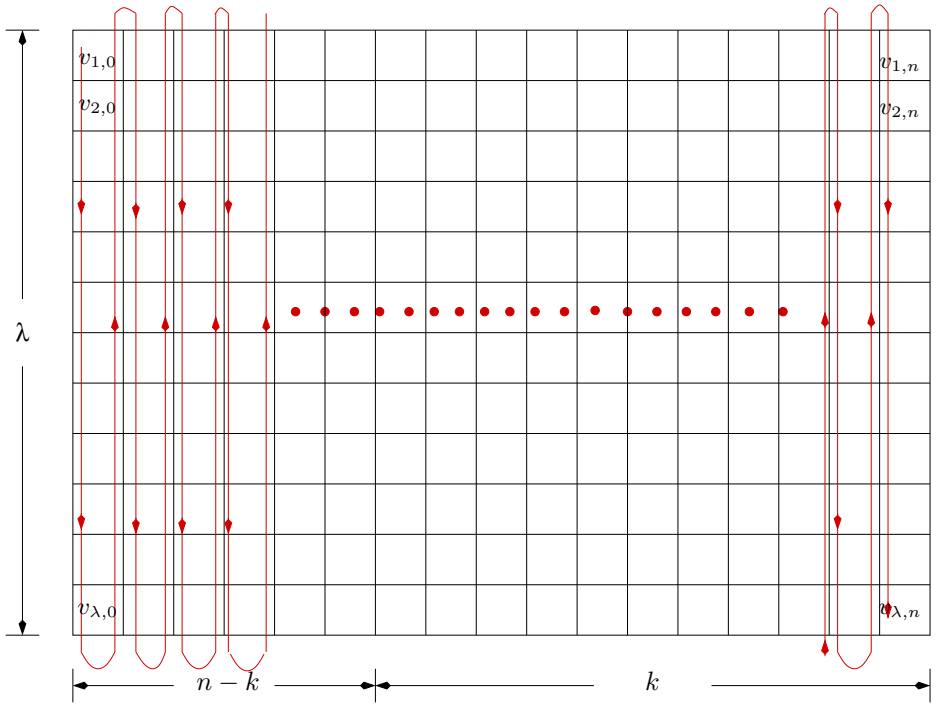
A $(\lambda n, \lambda k)$ code can be obtained from an (n, k) code $\mathcal{C}$ by arranging λ codewords as row of a rectangular array and then transmitting the array column by column.

The resulting code $\mathcal{C}^{\lambda}$ is the interleaved code.

λ is the interleaving depth.

If the minimum distance of $\mathcal{C}$ is d_{min} the minimum distance of the interleaved code is also d_{min}.

An interleaved code can be obtained also using λ codewords from λ different codes having minimum distance d_{min}.



Block interleaving techniques are effective for correcting errors that cluster to form bursts.

Hadamart Codes

A code such that any row differs from any other row in exactly $\frac{n}{2}$ positions is called Hadamart code.

In a Hadamart code one row of the code contains all zeros. The other rows contains $\frac{n}{2}$ zeros and $\frac{n}{2}$ ones.

Construction of Hadamart Codes

A Hadamart code is obtained by selecting as codewords the rows of a Hadamart matrix.

$$\boldsymbol{H}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \qquad \boldsymbol{H}_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \qquad \boldsymbol{H}_{2n} = \begin{bmatrix} \boldsymbol{H}_n & \boldsymbol{H}_n \\ \boldsymbol{H}_n & \overline{\boldsymbol{H}}_n \end{bmatrix} = \boldsymbol{H}_2 \otimes \boldsymbol{H}_n$$

Cyclic Codes

Definition: An (n, k) linear block code is a cyclic code if and only if any cyclic shift of a codeword produces another codeword.

Example: The (7,4) Hamming code with generator matrix

$$\boldsymbol{G} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

is a cyclic code. For example, the sequences

$$\{(0001011), (1000101), (1100010), (0110001), (1011000), (0101100), (0010110)\}$$

are codewords!

Polynomial Representation of a Codeword: A codeword $\boldsymbol{v} = (v_0, v_1, \ldots, v_{n-1})$ is represented by the code polynomial

$$\boldsymbol{v}(X) = v_0 + v_1 X + v_2 X^2 + \ldots + v_{n-1} X^{n-1}$$

Properties of Cyclic Codes (1)

Theorem: **Let v be the codeword of a cyclic code with code polynomial $\boldsymbol{v}(X)$ and let $\boldsymbol{v}^{(i)}(X)$ be the code polynomial of a codeword obtained by right cyclic shift of i positions. Then, $\boldsymbol{v}^{(i)}(X)$ is the remainder resulting from the division of $X^i\boldsymbol{v}(X)$ by (X^n+1), i.e.**

$$X^i\boldsymbol{v}(X) = q(X)(X^n+1) + \boldsymbol{v}^{(i)}(X)$$

where $q(X)$ is the quotient polynomial of degree not greater than $i-1$.

Proof:

1. **Let us write explicitly $X^i\boldsymbol{v}(X)$**

$$X^i\boldsymbol{v}(X) = v_{n-1}X^{n+i-1} + v_{n-2}X^{n+i-2} + \ldots + v_1X^{i+1} + v_0X^i$$

2. **Add to $X^i\boldsymbol{v}(X)$ twice the i terms $v_{n-1}X^{i-1}, v_{n-2}X^{i-2}, \ldots v_{n-i}$. Then,**

$$X^i\boldsymbol{v}(X) = v_{n-1}X^{i-1}(X^n+1) + \ldots + v_{n-i}(X^n+1) + v_{n-i-1}X^{n-1} \ldots + v_1X^{i+1} + v_0X^i + v_{n-1}X^{i-1} + \ldots v_{n-1}$$

3. **We can rearrange**

$$\begin{aligned} X^i\boldsymbol{v}(X) &= (v_{n-1}X^{i-1} + v_{n-2}X^{i-2} + \ldots + v_{n-i})(X^n+1) + v_{n-i-1}X^{n-1} \ldots + v_1X^{i+1} + v_0X^i + v_{n-1}X^{i-1} + \ldots v_{n-1} \\ &= q(X)(X^n+1) + v^{(i)}(X) \end{aligned}$$

Properties of Cyclic Codes (2)

The nonzero code polynomial of minimum degree in a cyclic code $\mathcal{C}$ is unique.

Proof:

Suppose $g(X)$ is a code polynomial of minimum degree r. If $g(X)$ is not unique there exists $g'(X)$ that is a code polynomial of degree r. Since the code is linear $g(X) + g'(X) = (g_0 + g'_0) + (g_1 + g'_1)X + \ldots + (g_{r-1} + g'_{r-1})X^{r-1}$ is still a code polynomial in $\mathcal{C}$ of degree $r - 1$. This is impossible if g is a code polynomial of minimum degree.

Let $g(X) = g_0 + g_1X + \ldots + g_{r-1}X^{r-1} + X^r$ be the non zero code polynomial of minimum degree in an (n, k) cyclic code $\mathcal{C}$. Then, the constant term g_0 must be equal to 1.

Proof:

If $g_0 = 0$ then $g(X) = X(g_1 + g_2X \ldots + g_{r-1}X^{r-2} + X^{r-1}) = Xg^{(1)}(X)$ with $g^{(1)}(X)$ code polynomial of degree $r - 1$. This is in contrast with the assumption on $g(X)$.

Properties of Cyclic Codes (3)

Let $g(X) = g_0 + g_1X + \ldots + g_{r-1}X^{r-1} + X^r$ be the non zero polynomial of minimum degree in an (n, k) cyclic code $\mathcal{C}$. A binary polynomial v of degree $n-1$ or less is a code polynomial if and only if it is a multiple of $g(X)$, i.e. $v(X) = u(X)g(X)$, with $u(X)$ polynomial of degree at most $n-r-1$.

1. **Suppose $v(X)$ is a binary polynomial of degree at most $n-1$ multiple of $g(X)$. Then,**

$$\begin{aligned} \boldsymbol{v}(X) &= (u_0 + u_1X + \ldots + u_{n-r-1}X^{n-r-1})\boldsymbol{g}(X) \\ &= u_0\boldsymbol{g}(X) + u_1X\boldsymbol{g}(X) + u_2X^2\boldsymbol{g}(X) + \ldots + u_{n-r-1}X^{n-r-1}\boldsymbol{g}(X) \end{aligned}$$

   is a linear combination of code polynomials in $\mathcal{C}$.

2. **Let $v(X)$ be a code polynomial in $\mathcal{C}$ and let us divide it by $g(X)$, $v(X) = u(X)g(X) + b(X)$ with $b(X)$ polynomial of degree at most $r-1$ (remainder of division). If $b \neq 0$, b is a linear combination of codewords in $\mathcal{C} \Rightarrow$ It is a codeword of degree at most $r-1$. This is impossible because $g(X)$ is the minimum degree code polynomial.**

If $g(X)$ is a code polynomial of $\mathcal{C}$ of minimum degree r, there are 2^{n-r} code polynomials in $\mathcal{C}$ and $k = n - r$.

Generator Polynomial and Generator Matrix

In an (n, k) cyclic code, there exists one and only one code polynomial of degree $n - k$,

$$\boldsymbol{g}(X) = g_0 + g_1 X + \ldots + g_{n-k} X^{n-k}$$

A polynomial $\boldsymbol{v}(X)$ of degree $n - 1$ or less is a code polynomial if and only if

$$\boldsymbol{v}(X) = \boldsymbol{u}(X)\boldsymbol{g}(X)$$

with $\boldsymbol{u}(X)$ polynomial of degree $k - 1$.

$\boldsymbol{g}(X)$ is called the generator polynomial.

Generator Matrix

$$\boldsymbol{G} = \begin{pmatrix} g_0 & g_1 & g_2 & \ldots & g_{n-k} & 0 & 0 & \ldots & 0 \\ 0 & g_0 & g_1 & \ldots & g_{n-k-1} & g_{n-k} & 0 & \ldots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ldots & \ldots \\ 0 & \ldots & \ldots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \ldots & 0 & g_0 & g_1 & \ldots & g_{n-k-1} & g_{n-k} \end{pmatrix}$$

Code Polynomials in Systematic Form

We want to encode $\boldsymbol{u} = (u_0, u_1, \ldots, u_{k-1})$ so that the code polynomial is

$$\begin{aligned}\boldsymbol{u} &= \boldsymbol{b} + X^{n-k}\boldsymbol{u} \\ &= b_0 + b_1 X + \ldots + b_{n-k-1}X^{n-k-1} + u_0 X^{n-k} + u_1 X^{n-k+1} + \ldots u_{k-1}X^{n-1}\end{aligned}$$

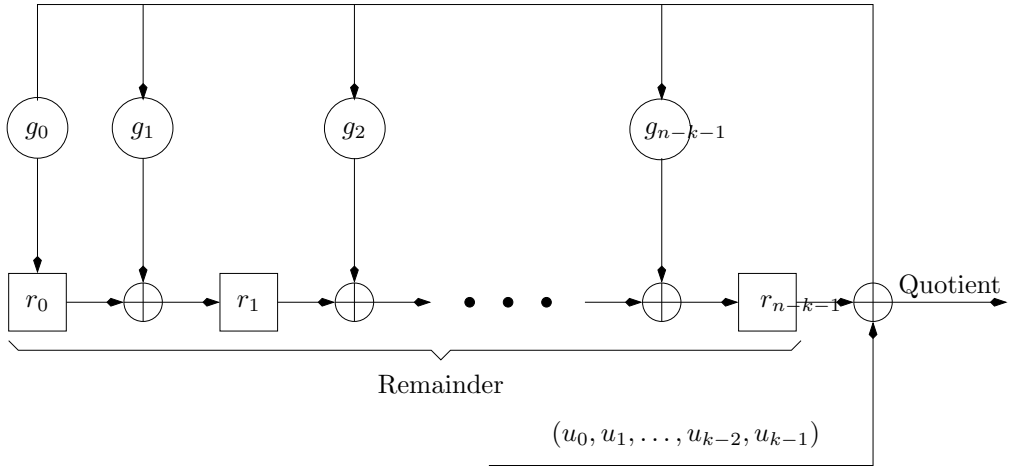
Step 1 **Premultiply the message $\boldsymbol{u}(X)$ by X^{n-k}.**

Step 2 **Obtain the remainder $\boldsymbol{b}(X)$ by dividing $X^{n-k}\boldsymbol{u}(X)$ by the generator polynomial $\boldsymbol{g}(X)$.**

Step 3 **Combine $\boldsymbol{b}(X)$ and $X^{n-k}\boldsymbol{u}(X)$ to obtain the code polynomial $\boldsymbol{b} + X^{n-k}\boldsymbol{u}(X)$.**

Division Circuit

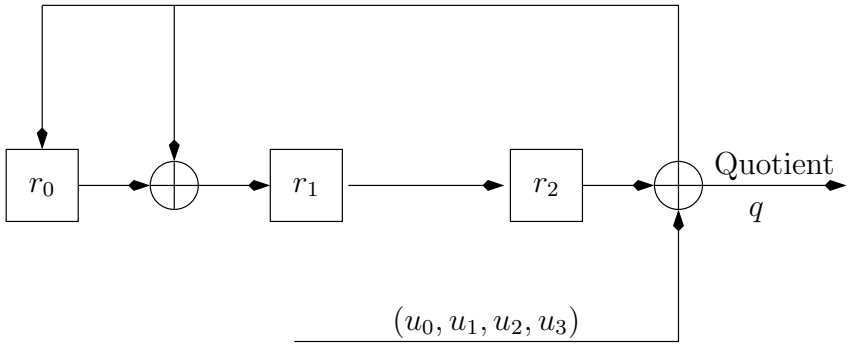
Circuit performing the division of the polynomial $X^{n-k}(u_{k-1}X^{k-1} + u_{k-2}X^{k-2} + \dots u_0)$ by $\boldsymbol{g}(X) = X^{n-k} + g_{n-k-1}X^{n-k-1} + \dots + g_1X + g_0$



- **Initial status of the register:** $(0, 0, \dots 0)$.
- **Final status: When all bits** $(u_0, u_1, \dots u_{k-1})$ **have been processed the register contains the reminder of the division.**

Example

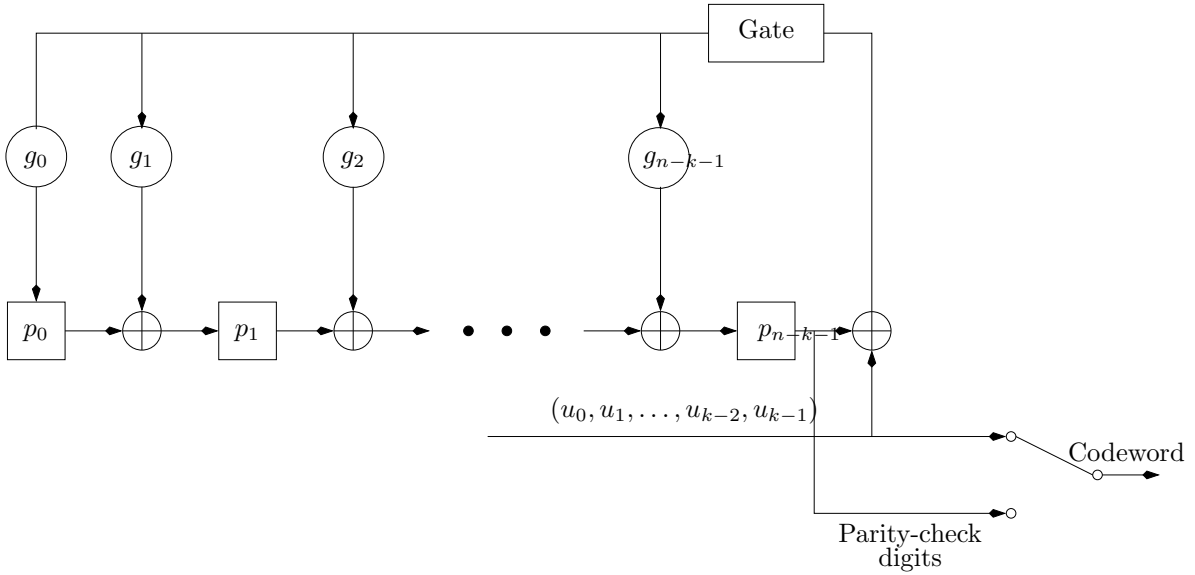
Assume $\boldsymbol{g}(X) = X^3 + X + 1$ and $\boldsymbol{u}(X) = X^3 + X^2 + 1$.



u_i	r_0	r_1	r_2	q
–	0	0	0	–
u_3=1	1	1	0	1
u_2=1	1	0	1	1
u_1=0	1	0	0	1
u_0=1	1	0	0	1

			X^3	$+X^2$	$+X$	$+1$			
X^3	X	1	X^6	$+X^5$		$+X^3$			
			X^6		$+X^4$	$+X^3$			
				X^5	$+X^4$				
				X^5		$+X^3$	$+X^2$		
					X^4	$+X^3$	$+X^2$		
					X^4		$+X^2$	$+X$	
						X^3		$+X$	
						X^3		$+X$	$+1$
									1

Systematic Encoder



- **Set the shift register to $(0, 0, \ldots 0)$. Open the gate and shift the message bits $(u_0, u_1, \ldots u_{k-1})$ both into the circuit and into the channel. When all bits $(u_0, u_1, \ldots u_{k-1})$ have been processed the register contains the parity check bits.**
- **When all bits $(u_0, u_1, \ldots u_{k-1})$ have been processed break the feedback connection by turning off the gate.**
- **Shift the parity-check digits out and send them into the channel.**

Properties of Cyclic Codes (4)

The generator polynomial $g(X)$ of an (n, k) cyclic code is a factor of $X^n + 1$.

Proof:

Consider $X^k \boldsymbol{g}(X)$ and divide it by $X^n + 1$

$$X^k \boldsymbol{g}(X) = X^n + 1 + \boldsymbol{g}^{(k)}(X)$$

$\boldsymbol{g}^{(k)}(X)$ is the code polynomial obtained by shifting $g(X)$ by k positions on the right $\Rightarrow$ $\boldsymbol{g}^{(k)}(X) = \boldsymbol{a}(x) * \boldsymbol{g}(X)$. It follows

$$X^n + 1 = (X^k + \boldsymbol{a}(X))\boldsymbol{g}(X).$$

For any n and k there exists a cyclic code?

If $g(X)$ is a polynomial of degree $n - k$ and is a factor of $X^n + 1$, then $g(X)$ generates an (n, k) cyclic code.

Parity Check Matrices for Cyclic Codes (1)

$$X^n + 1 = \boldsymbol{g}(X)\boldsymbol{h}(X)$$

being $\boldsymbol{h}(X)$ a polynomial of degree k.

The reciprocal of $\boldsymbol{h}(X)$

$$X^k \boldsymbol{h}(X^{-1}) \stackrel{\triangle}{=} h_k + h_{k-1}X + h_{k-2}X^2 + \ldots + h_0 X^k$$

is a factor of $X^n + 1$. $\Rrightarrow$ We can consider the $(n, n-k)$ cyclic code with generator matrix

$$\boldsymbol{H} = \begin{pmatrix} h_k & h_{k-1} & h_{k-2} & \ldots & h_0 & 0 & 0 & \ldots & 0 \\ 0 & h_k & h_{k-1} & \ldots & h_1 & h_0 & 0 & \ldots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ldots & \ldots \\ 0 & \ldots & \ldots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \ldots & 0 & h_k & h_{k-1} & \ldots & h_1 & h_0 \end{pmatrix}$$

Parity Check Matrices for Cyclic Codes (2)

Let $\mathcal{C}$ be an (n, k) cyclic code with generator polynomial $g(X)$. The dual code of $\mathcal{C}$ is also cyclic and it is generated by the polynomial $X^k h(X^{-1})$, where $h(X) = \frac{X^n+1}{g(X)}$.

Proof:

For any codeword $v(X) = (v_0, v_1, \ldots, v_{n-1}) = a(X)g(X) \in \mathcal{C}$

$$v(X)h(X) = a(X)g(X)h(X) = a(X)(X^n + 1) = a(X)X^n + a(X)$$

Since $a(X)$ has degree at most $k-1$, the monomials X^k, X^{k+1}, X^{n-1} do not appear in the r.h.s. It follows

$$\sum_{i=0}^{k} h_i v_{n-i-j} = 0 \quad \text{for} \quad i \leq j \leq n-k.$$

$$\Downarrow$$

Any codeword $v \in \mathcal{C}$ is orthogonal to the matrix H.

H is the parity check matrix of $\mathcal{C}$.

$h(X)$ is called the parity polynomial of $\mathcal{C}$.

Syndrome Computation

Let $\boldsymbol{r}(X) = r_0 + r_1 X + \ldots + r_{n-1} X^{n-1}$ be the received vector.

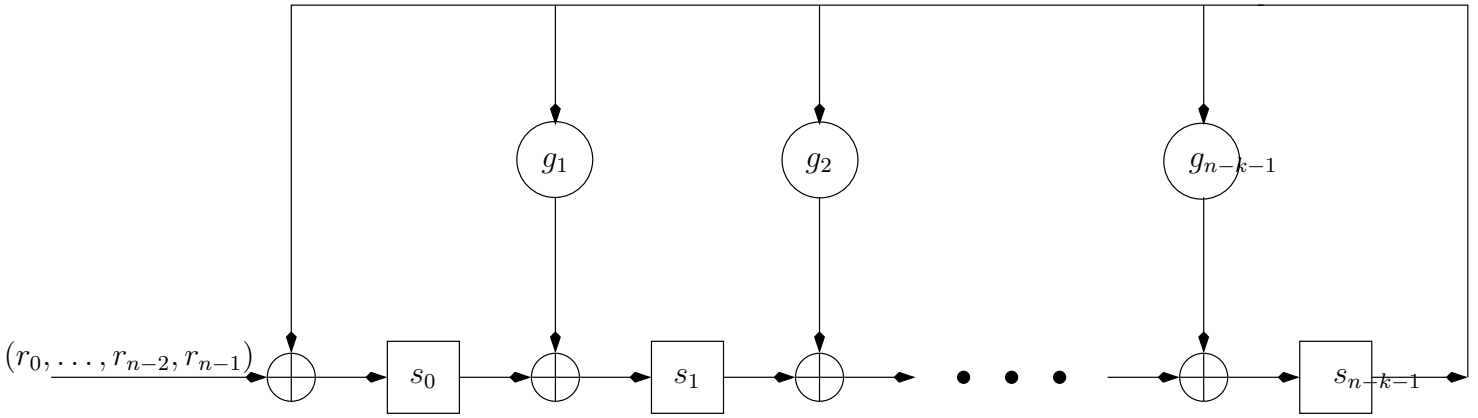
Let us divide $\boldsymbol{r}(X)$ by the generator polynomial $\boldsymbol{g}(X)$.

$$\boldsymbol{r}(X) = \boldsymbol{a}(X)\boldsymbol{g}(X) + \boldsymbol{s}(X)$$

$\boldsymbol{s}(X) = 0$ if and only if $\boldsymbol{r}(X)$ is a code polynomial.

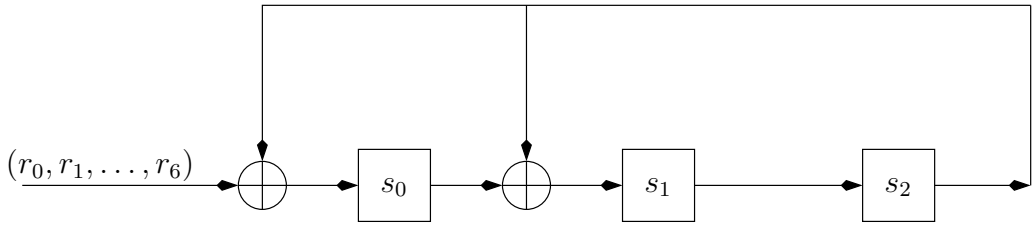
$\boldsymbol{s}(X)$ is a polynomial of degree $n-k-1$ and its $n-k$ coefficients form the syndrome $\boldsymbol{s}$.

Syndrome Circuit



Syndrome: Properties and Examples

Let $\boldsymbol{s}(X)$ be the syndrome of the received polynomial $\boldsymbol{r}(X) = r_0 + r_1 X + \ldots + r_{n-1} X^{n-1}$. Then, the remainder $\boldsymbol{s}^{(1)}(X)$ resulting from dividing $X\boldsymbol{s}(X)$ by the generator polynomial $\boldsymbol{g}(X)$ is the syndrome of $\boldsymbol{r}^{(1)}(X)$, which is a cyclic shift of $\boldsymbol{r}(X)$.



Assume
$\boldsymbol{g}(X) = X^3 + X + 1$
and
$\boldsymbol{r}(X) = X^5 + X^4 + X^2 + 1.$
The gate is disabled to compute $\boldsymbol{s}^{(1)}(X)$ and $\boldsymbol{s}^{(2)}(X)$.

r_i	s_0	s_1	s_2	Comments
r_6=0	0	0	0	Initial state
r_5=1	1	0	0	
r_4=1	1	1	0	
r_3=0	0	1	1	
r_2=1	0	1	1	
r_1=0	1	1	1	
r_0=1	1	0	1	Syndrome $\boldsymbol{s}$
–	1	0	0	Syndrome $\boldsymbol{s}^{(1)}$
–	0	1	0	Syndrome $\boldsymbol{s}^{(2)}$

Error Detection Capabilities of Cyclic Codes (1)

End-around burst

$$\begin{array}{cccccccccccccccccccc} (1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0) \\ \leftarrow & \leftarrow & \hookleftarrow & & & & & & & & & & \hookrightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow \end{array}$$

An (n, k) cyclic code is capable of detecting any error burst of length $n - k$ or less, including the end-around errors.

Proof:

Let $b(X)$ a polynomial of degree $n - k - 1$ or less. Any burst error of length n-k **can be expressed as**

$$\boldsymbol{e}(X) = X^j \boldsymbol{b}(X) \qquad 0 \leq j \leq n - 1.$$

$b(X)$ is not divisible for $g(X)$ because $g(X)$ has degree $n - k$. X is not a factor of $g(X)$ because $g(X)$ is a factor of $X^n + 1$. $\Rightarrow$ $e(X) = X^j b(X)$ is not divisible by $g(X)$ and the syndrome caused by $e(X)$ is nonzero. Any cyclic rotation of e including end-around burst can be detected.

Error Detection Capabilities of Cyclic Codes (2)

The fraction of undetectable burst of length $n-k+1$ is $2^{-(n-k-1)}$

Proof:

Consider the error pattern e with error confined to the digits $e_i, e_{i+1}, \ldots e_{i+n-k}$ with $e_i = e_{i+n-k} = 1$. There are 2^{n-k-1} of such error bursts. The only one that cannot be detected is $\boldsymbol{e}(X) = X^i \boldsymbol{g}(X)$. This holds for any i. Then, the fraction of undetectable errors is $2^{-(n-k-1)}$.

The fraction of undetectable burst of length $\ell > n-k+1$ is $2^{-(n-k)}$

Proof:

For $\ell > n-k+1$ there are $2^{\ell-2}$ burst errors with errors confined to the digits $e_i, e_{i+1}, \ldots e_{i+\ell-1}$ with $e_i = e_{i+\ell-1} = 1$. Among those the undetectable ones have form

$$\boldsymbol{e}(X) = X^i \boldsymbol{a}(X) \boldsymbol{g}(X)$$

where $\boldsymbol{a}(X)$ is a polynomial of degree $\ell-(n-k)-1$ and $a_0 = a_{\ell-(n-k)-1} = 1$. There are $2^{\ell-(n-k)-2}$ of such polynomials. This holds for any i including end-around burst. Then, the fraction of undetectable errors is $2^{-(n-k)}$.

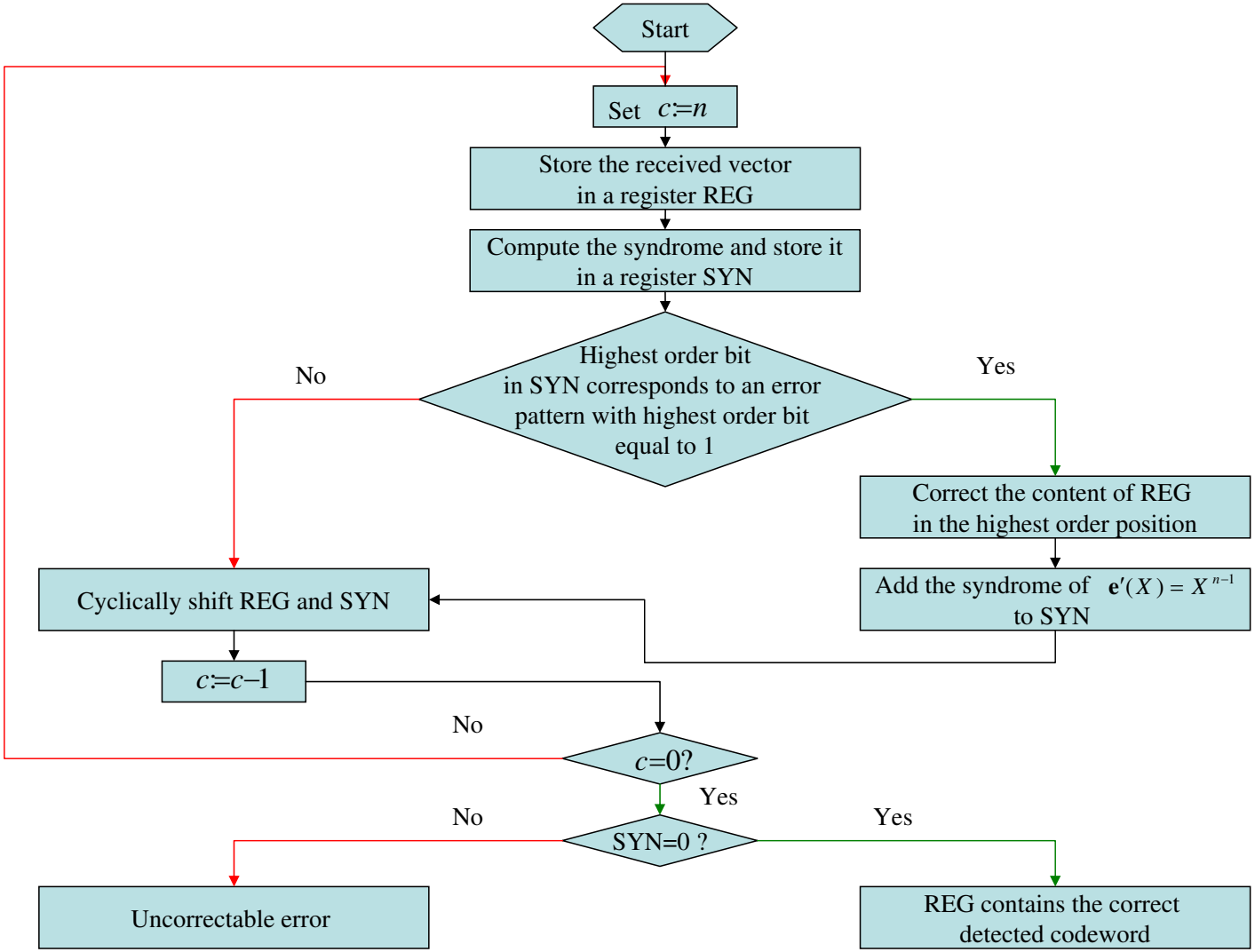
Error Correction for Cyclic Codes with Standard Array

Step 1: Syndrome computation	**Division by the generator polynomial.** **Hardware: Division circuit.** **Complexity: Linearly proportional to $n-k$.**
Step 2: Error pattern detection	**Search into the look-up table.** **Hardware: Combinatorial circuit that stores the look-up table.** **Complexity: It increases exponentially with the code length n and the number of errors to be detected.**
Step 3: Addition modulo 2 of the received vector to the error pattern	**Hardware: EXOR circuit. Possible serial implementation.**

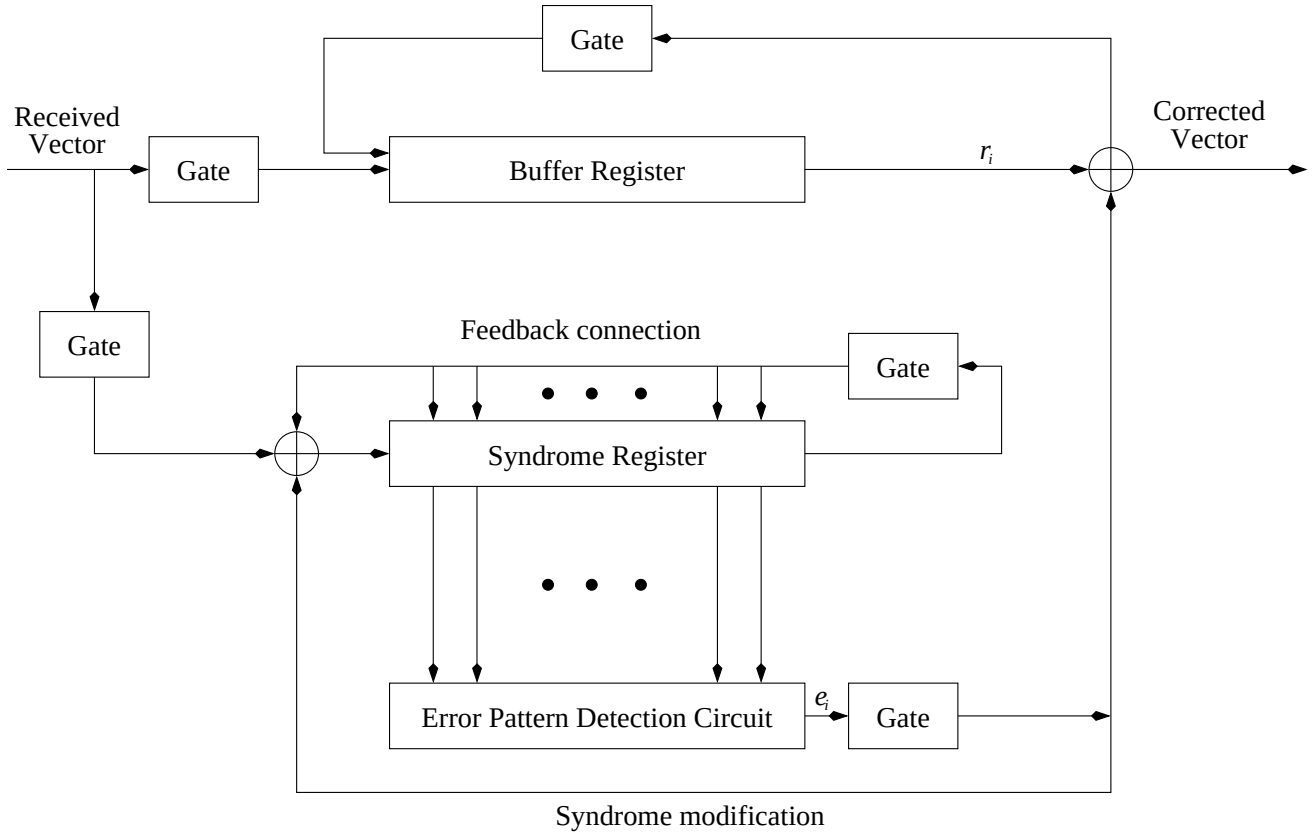
Step 2 is still too costly...

How to use the properties of cyclic codes to reduce complexity?

Meggitt Decoder: Flow Chart



Meggitt Decoder: Circuit



Remarks

Initially we have in the syndrome register a syndrome corresponding to the error $\boldsymbol{e}(X) = e_0 + e_1 X + \ldots + e_{n-1} X^{n-1}$.

The error pattern detection circuit is built to detect an error $\boldsymbol{e}'(X) = X^{n-1}$. If an error $\boldsymbol{e}'(X)$ is detect then $e_{n-1} = 1$ in $\boldsymbol{e}(X)$ is corrected by replacing $\boldsymbol{r}(X)$ with $\boldsymbol{r}_1(X) = r_0 + r_1 X + \ldots + (r_{n-1} \oplus e_{n-1}) X^{n-1}$

Then the vector $\boldsymbol{r}_1(X)$ is cyclic shifted by one position and the vector $\boldsymbol{r}_1^{(1)}(X)$ is stored. Simultaneously the syndrome $\boldsymbol{s}_1^{(1)}(X)$ of $\boldsymbol{r}_1^{(1)}(X)$ is computed. It is easy to verify that it is given by

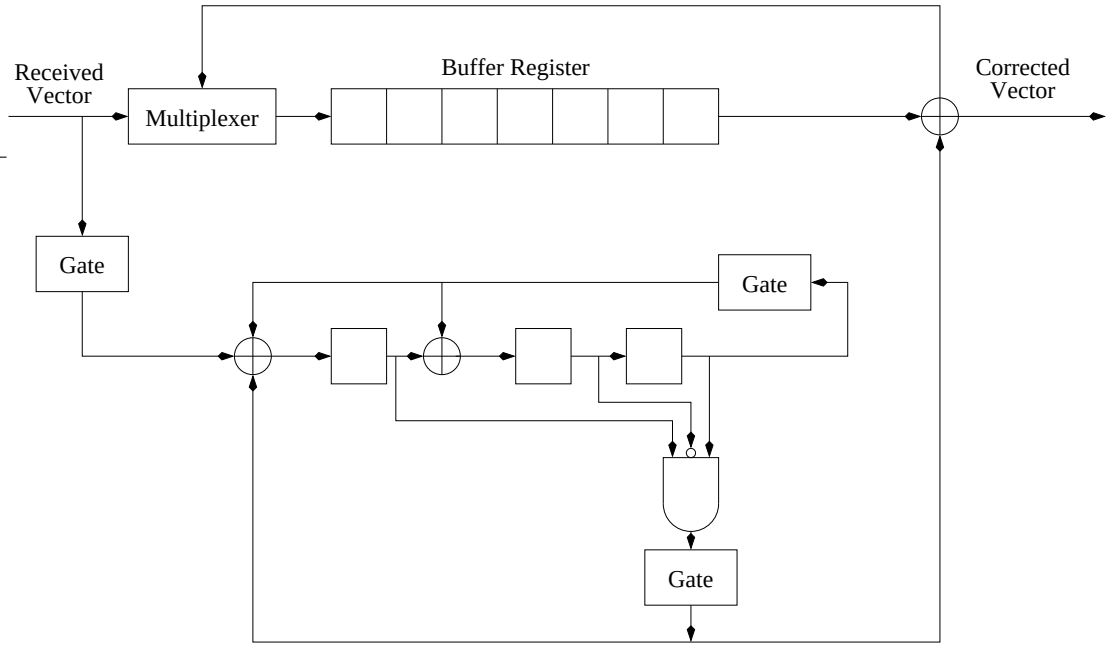
$$\boldsymbol{s}_1^{(1)}(X) = \boldsymbol{s}^{(1)}(X) + 1$$

The processing continues by detecting and correcting the error X^{n-2}.

Meggitt Decoder: Example

Decoding circuit for the $(7,4)$ cyclic code generated by $g(X) = 1 + X + X^3$.

Error Pattern $e(X)$	Syndrome $s(X)$	Syn. Vector (s_0, s_1, s_2)
$e(X) = X^6$	$s(X) = 1 + X^2$	(101)
$e(X) = X^5$	$s(X) = 1 + X + X^2$	(111)
$e(X) = X^4$	$s(X) = X + X^2$	(011)
$e(X) = X^3$	$s(X) = 1 + X$	(110)
$e(X) = X^2$	$s(X) = X^2$	(001)
$e(X) = X$	$s(X) = X$	(010)
$e(X) = 1$	$s(X) = 1$	(100)



Special Cyclic Codes: Cyclic Hamming Codes

A polynomial of degree ℓ is an irreducible polynomial if it is not divisible for any polynomial of degree lower than ℓ and greater than zero.

An irreducible polynomial $g(X)$ of degree ℓ is called primitive when the smallest n such that $(X^n + 1)$ is a multiple of $g(X)$ is $n = 2^\ell - 1$.

A primitive polynomial $g(X)$ generates a $(2^\ell - 1, 2^\ell - \ell - 1)$ cyclic code that is an Hamming code.

Special Cyclic Codes: BCH Codes

The Bose, Chaudhuri, and Hocquenghem (BCH) codes form a large class of powerful error-correcting cyclic codes.

For any pair of positive integers m and t there exists a binary BCH code with the following parameters

$$n = 2^m - 1, \qquad n - k \leq mt, \qquad d_{min} \geq 2t + 1$$

- **BCH codes are obtained from factors of $(X^{2^m-1} + 1)$;**
- **A BCH code can correct all combinations of t errors;**
- **Very flexible in the choice of parameters;**
- **At block-lengths of few hundreds or less, many of these codes are among the best known codes of the same length and rate.**

References

- S. Lin and D. Costello, **Error Control Coding,** *Prentice Hall,* 1983. Chapters 2-5.
- S. Benedetto and E. Biglieri, **Principles of Digital Transmission with Wireless Applications,** , Chapter 10.