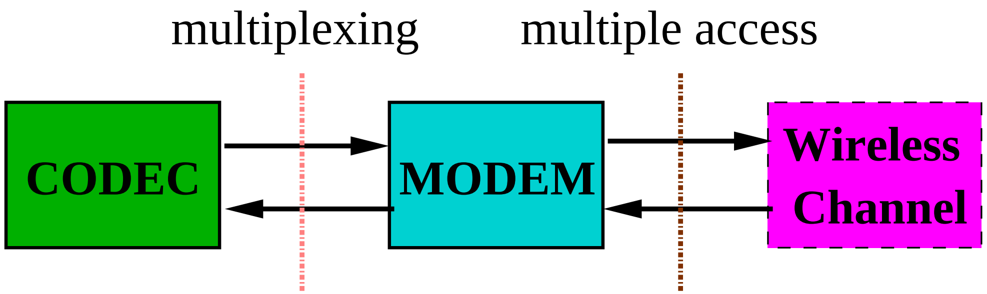
Revision of Lecture Twelve

- Previous lecture concentrates on the remaining issue of Modem, namely, differential detection for non-coherent systems, which does not require CSI

- Thus we have completed Modem, under ideal AWGN or flat fading channel condition



If channel is dispersive, equalisation is required ⇒ we will return to this issue as well as issue of multiple access later

- We now turn to CODEC part ⇒ we will concentrate on channel coding and decoding, but not source coding and decoding

Source coder and decoder are well covered in Digital Coding and Transmission

Channel Coding Introduction

- Mobile channels are very hostile environments, and yet real systems work satisfactorily. One of the contributors to this success is channel coding
- **Channel coding** is used to detect and often correct symbols that are received in error
- **Error detection** can be used by receiver to generate ARQ to transmitter for a re-transmission of the frame in error, as in computer networks (stop & wait, go-back-n, selective repeat protocols)
- When re-transmission is not an option: **forward error correction** coding, which introduces extra information (redundancy) into transmitted data for receiver to detect and correct errors

<table>
<tr><td colspan="6">FEC coding</td></tr>
<tr><td colspan="5">Block codes</td><td>Convolutional codes</td></tr>
<tr><td>Others</td><td colspan="4">linear</td><td></td></tr>
<tr><td></td><td>non-cyclic</td><td colspan="3">Polynomial (cyclic)</td><td></td></tr>
<tr><td></td><td></td><td>Golay</td><td colspan="2">Bose-Chaudhuri-Hocquenhem</td><td></td></tr>
<tr><td></td><td></td><td></td><td>Reed-Solomon</td><td>Binary BCH</td><td></td></tr>
</table>

Some examples:

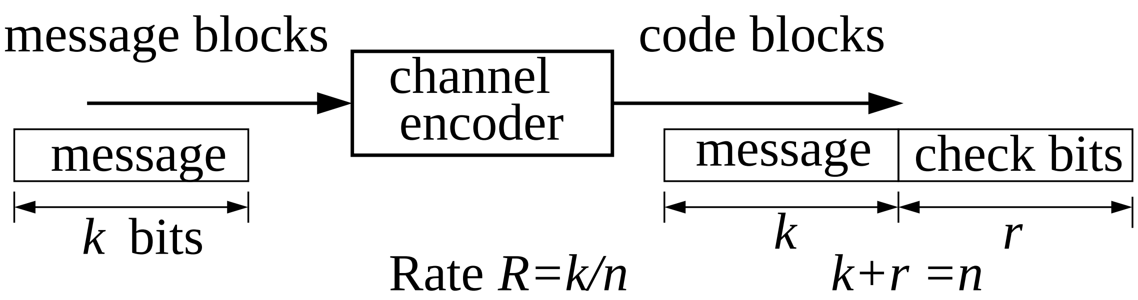
- Binary BCH and Convolutional codes widely used in various practical communications systems
- Reed-Solomon codes used in music CD
- Golay codes used in Mars explorer

- Advanced topics: Soft-decision decoding, turbo codes, iterative or turbo detection and decoding

Block Code Introduction

- There are systematic and non-systematic codes. For block codes, systematic ones are more powerful
  - Rate $R = k/n$ block code: k information bits plus $r = n - k$ check bits forms a **codeword**
  - All valid codewords form a **codebook**

- (n, k) systematic block code



  - Systematic: k information bits must be explicitly transmitted (more strict definition also requires they are transmitted together as a block)

- Systematic **linear** block code: first k bits of a codeword are message bits, and last $n - k$ check bits are linear combinations of the k message bits

Linear Block Code: Encoding

- Let $\mathbf{c}$ be n-bit codeword and $\mathbf{d}$ be k-bit message, written in row-vector form
- An (n, k) linear block code is defined by its $k \times n$ **generating matrix** G

$$G = [I_k \mid P]$$

  – I_k is identity matrix of order k, $k \times (n - k)$ matrix P specifies the given (n, k) linear block code, and all elements in P are binary valued

- Encoding process can then be written as

$$\mathbf{c} = \mathbf{d}G$$

  – Binary (**modulo-2**) arithmetic operations are carried out

- Where error detection and correction capability comes from:
  – A binary sequence of n bits should have 2^n patterns, denoting as $\bar{\mathbf{c}}_i$, $1 \le i \le 2^n$
  – but $\mathbf{c}$ only contains 2^k codewords, i.e. it can only take some of $\{\bar{\mathbf{c}}_i\}$, called **legal** sequences $\rightarrow$ only these legal sequences can be transmitted
  – If receiver encounters an illegal sequence $\bar{\mathbf{c}}_i$ (not a codeword), what it says?

Example

- $(6,3)$ linear block code with generating matrix and codebook

$$G = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{array} \right]$$

massages	codewords
000	000 000
001	001 110
010	010 101
011	011 011
100	100 011
101	101 101
110	110 110
111	111 000

- For example, for message **d**=110, parity check bits are

$$\begin{aligned} c_4 &= 1\cdot 0 + 1\cdot 1 + 0\cdot 1 = 0 + 1 + 0 = 1 \\ c_5 &= 1\cdot 1 + 1\cdot 0 + 0\cdot 1 = 1 + 0 + 0 = 1 \\ c_6 &= 1\cdot 1 + 1\cdot 1 + 0\cdot 0 = 1 + 1 + 0 = 0 \end{aligned}$$

  Note the binary modulo-2 arithmetic operations involved

- $2^6 = 64$ but only $2^3 = 8$ legal codewords e.g. 111111 is not a legal codeword

- If receiver encounters 111111 it must be due to error, as 111111 will never be sent

Linear Block Code: Decoding

- Each $k \times n$ generating matrix $G = [I_k \mid P]$ is associated with a $(n-k) \times n$ **parity check matrix**

$$H = [P^T \mid I_{n-k}]$$

  – Basic **property of codeword**: $\mathbf{c}$ is a codeword in the (n,k) block code generated by G, if and only if $\mathbf{c}H^T = 0$

- Received row vector $\mathbf{r}$ can be written as

$$\mathbf{r} = \mathbf{c} + \mathbf{e}$$

  – All the elements are binary valued, e.g. if the transmitted $c_i = 1$ and is received in error: $r_i = 0$, then $e_i = 1$

- $(n-k)$ (row vector) **error syndrome**

$$\mathbf{s} = \mathbf{r}H^T = (\mathbf{c}+\mathbf{e})H^T = \mathbf{c}H^T + \mathbf{e}H^T = \mathbf{e}H^T$$

  – $\mathbf{s}$ is related to the error vector $\mathbf{e}$, and can be used to detect and correct errors

Error Detection and Correction Capabilities

- **Weight** of a codeword $\mathbf{c}$ is the number of nonzero elements in $\mathbf{c}$
- **Hamming distance** between two codewords $\mathbf{c}_1$ and $\mathbf{c}_2$ is the number of elements in which they differ
- **Minimum distance** of a codebook, $d_{\min}$, is the smallest Hamming distance between any pair of codewords in the codebook
- The minimum distance $d_{\min}$ of a linear block code is equal to the minimum weight of any nonzero codeword in the code
- Code with $d_{\min}$ can detect up to $d_{\min} - 1$ errors and correct up to $(d_{\min} - 1)/2$ errors in each codeword

A note: Here we are considering binary codes, where Hamming distance is defined

For error correction capability, we refer to hard-input hard-output decoding, i.e. decoder input is in hard bits and it outputs hard bits, later we will see soft-input decoding has better capability

Cyclic Codes

- **Cyclic** or polynomial generated codes are subset of linear block codes with some nice properties
  - Definition of cyclic: if $(c_0, c_1, \cdots c_{n-2}, c_{n-1})$ is a codeword then $(c_{n-1}, c_0, \cdots c_{n-3}, c_{n-2})$ is also a codeword in the same code
- A k-bit message $\mathbf{d} = (d_0, d_1, \cdots, d_{k-1})$ can be described by a message polynomial $d(x)$:

$$d(x) = d_0 + d_1 x^1 + \cdots + d_{k-1} x^{k-1}$$

- The code is defined by its **generating polynomial**

$$g(x) = g_0 + g_1 x^1 + \cdots + g_r x^r \quad \text{with} \quad g_0 = 1 \quad \text{and} \quad g_r = 1$$

- The n-bit codeword $\mathbf{c} = (c_0, c_1, \cdots, c_{n-1})$ for $\mathbf{d}$ is described by a polynomial

$$c(x) = \mathrm{Rem}\left(\frac{x^r \cdot d(x)}{g(x)}\right) + x^r \cdot d(x)$$

  - The remainder of $x^r \cdot d(x)/g(x)$, $\mathrm{Rem}(x^r \cdot d(x)/g(x))$, is a polynomial up to order x^{r-1} (i.e. r check bits), called **parity check** polynomial for $d(x)$
  - All calculations use modulo-2 arithmetic, remainder is found by "long division"

Cyclic Codes (continue)

- Example of (7,4) cyclic code with $g(x) = 1 + x^2 + x^3$
  - For message $\mathbf{d} = 0101$, $d(x) = x^1 + x^3$, $x^3 \cdot d(x) = x^4 + x^6$, $\mathrm{Rem}(x^3 \cdot d(x)/g(x)) = 1$, $c(x) = 1 + x^4 + x^6$, and thus

	check	message
c=	1 0 0	0 1 0 1

- In decoding, the received $r(x) = c(x) + e(x)$ with nonzero terms in $e(x)$ indicating errors, and the **syndrome** polynomial is calculated:

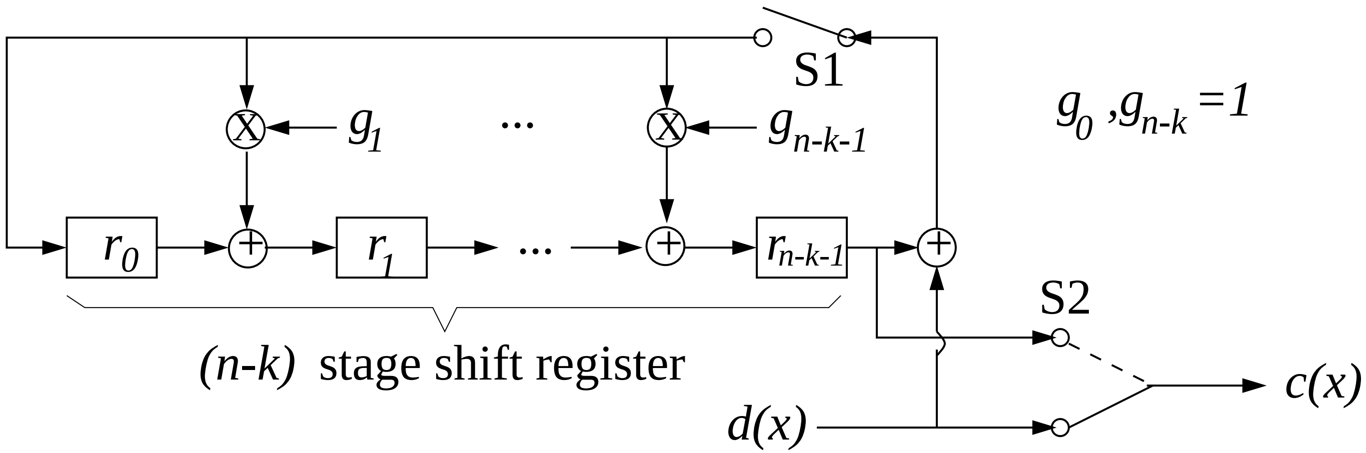
$$\mathrm{Rem}\left(\frac{c(x) + e(x)}{g(x)}\right) = \mathrm{Rem}\left(\frac{e(x)}{g(x)}\right) = s(x)$$

  - If a zero syndrome: no error or undetectable errors ($e(x)$ contains factor $g(x)$)
  - If a nonzero syndrome: errors detected and it is used for error correction

- Encoding and syndrome calculation can easily be implemented using shift register feedback circuits

Cyclic Code Encoder

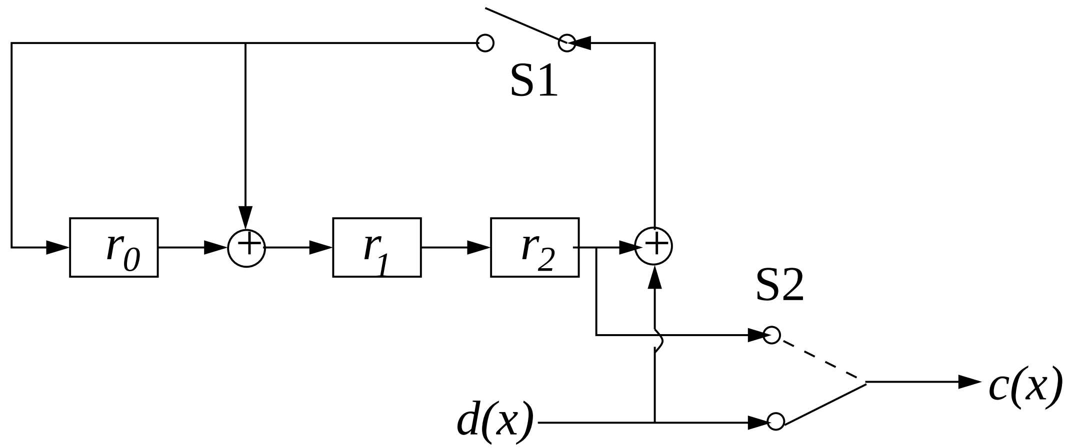
- (n,k) cyclic code encoder: an $(n-k)$ stage shift register with a feedback circuit



- The circuit operates under a clock and an encoding cycle consists of n shifts
  - Shift register always starts at zero state, i.e. all $r_i = 0$, and ends at zero state
  - During the first k shifts, S1 is closed $\rightarrow$ shift $d(x)$ into the shift register; and S2 is down $\rightarrow$ copy $d(x)$ directly to $c(x)$
  - After the k-th shift, the contents of the $(n-k)$ stage shift register are the $n-k$ parity check bits for $d(x)$
  - During the remaining $n-k$ shifts, S1 is open and S2 is up $\rightarrow$ clear the shift register contents out to $c(x)$

Example

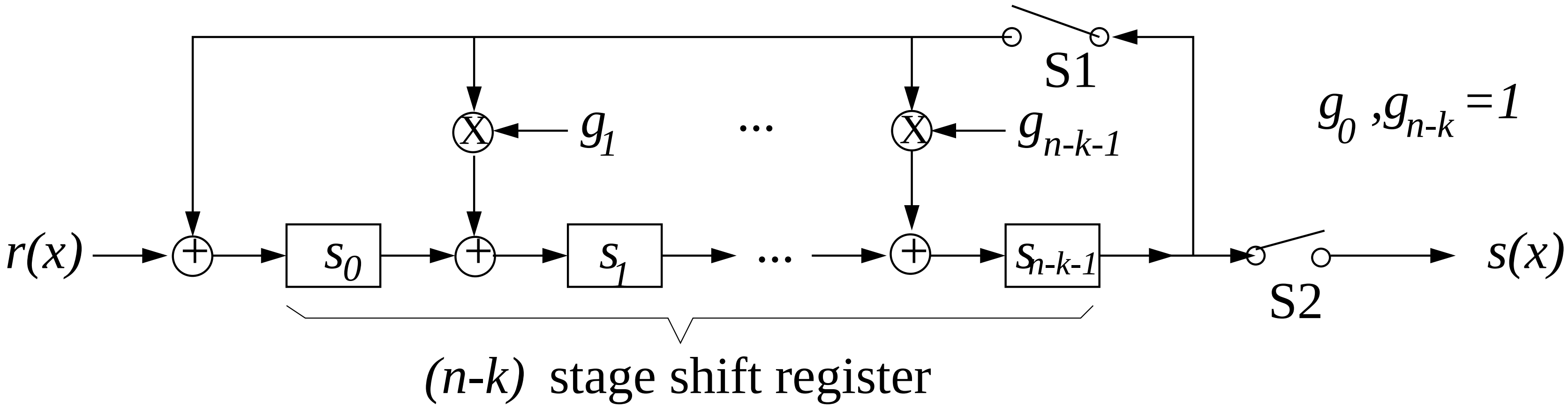
(7,4) cyclic code with
$g(x) = 1 + x + x^3$
Given message
$d(x) = 1 + x^2 + x^3$:



input				shift	register						codeword			
				index	r_0	r_1	r_2	c_0	c_1	c_2	c_3	c_4	c_5	c_6
1	0	1	1	0	0	0	0	-	-	-	-	-	-	-
	1	0	1	1	1	1	0	-	-	-	-	-	-	1
		1	0	2	1	0	1	-	-	-	-	-	1	1
			1	3	1	0	0	-	-	-	-	0	1	1
			-	4	1	0	0	-	-	-	1	0	1	1
			-	5	0	1	0	-	-	0	1	0	1	1
			-	6	0	0	1	-	0	0	1	0	1	1
			-	7	0	0	0	1	0	0	1	0	1	1

Cyclic Code Syndrome Calculation

- (n, k) cyclic code syndrome calculation circuit:



- The register is initialised to the zero state

1. S1 is closed and S2 is opened $\rightarrow$ the received $r(x)$ is shifted into register
2. After this, contents of register are $s(x)$
3. S1 is opened and S2 is closed $\rightarrow$ $s(x)$ is shifted out and the register is cleared, ready for the next cycle
4. Error syndrome $s(x)$ is then used in error detection and correction

Other FEC Codes

- BCH: subset of cyclic codes with largest $d_{\min}$ for given (n, k), denoted by $(n, k, d_{\min})$
  - This is a class of powerful and widely used FEC codes
  - Non-binary (i.e. can take values not just 0 and 1) version is called Reed-Solomon code and is used e.g. in music CD
- Golay codes: e.g. Mars explorer uses Golay code
- Convolutional codes:
  - In block codes, a n-bit codeword at a time unit t, $\mathbf{c}(t)$, depends only on the k-bit data, $\mathbf{d}(t)$, at the time t
  - For convolutional codes, $\mathbf{c}(t)$ also depends on N $(N > 0)$ previous blocks of data $\mathbf{d}(t-i)$, $1 \leq i \leq N$
  - $CC(n, k, N)$: rate $R = k/n$, constraint length N (or memory $N+1$), usually n, k and N are small
- Parallel-concatenated turbo codes, serial-concatenated turbo codes

Summary

- Channel coding introduction: FEC coding and classification
- Systematic block codes $\supset$ linear block codes $\supset$ cyclic (polynomial generated) codes $\supset$ binary BCH codes

  Error detection and correction capabilities
- Systematic linear block codes: generating matrix and encoding; parity check matrix and syndrome
- Cyclic codes: how every things can be described by polynomials, encoder and syndrome calculation (shift register feedback circuits)

  BCH: subset of cyclic codes with the largest $d_{\min}$ for given (n, k)
- Convolutional codes: differences with linear block codes