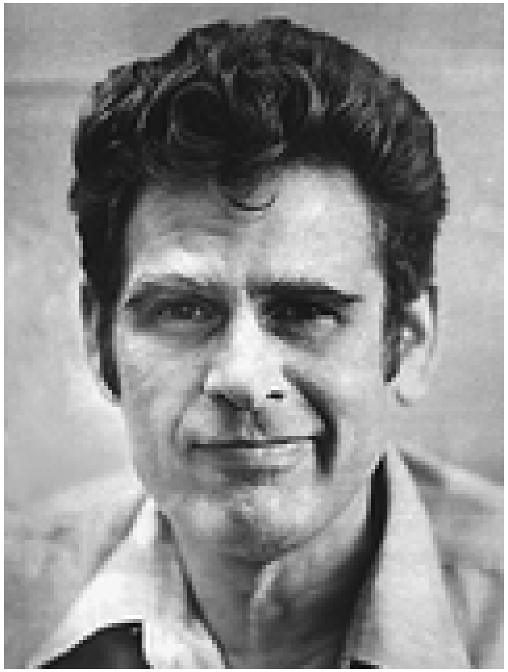
ECE 405/511
Error Control Coding

Binary Convolutional Codes

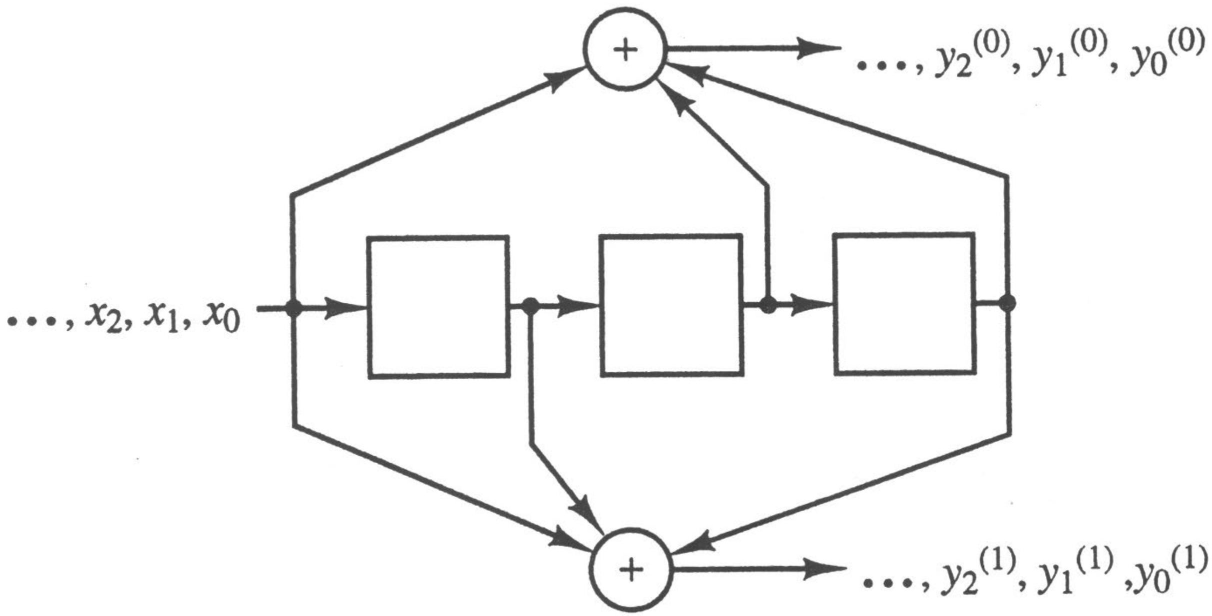
Peter Elias (1923-2001)

- Coding for Noisy Channels, 1955



- With block codes, the input data is divided into blocks of length *k* and the codewords have blocklength *n*.
- With convolutional codes, the input data is encoded in a continuous manner.
  - the output is a single codeword
  - encoding is done using only shift registers and XOR gates (binary arithmetic)
  - the encoder output is a function of the current input bits and the shift register contents

Rate 1/2 Linear Convolutional Encoder



Encoding

- Input data stream $\mathbf{x} = (x_0, x_1, x_2, \ldots)$

- Output data streams $\mathbf{y}^{(0)} = (y_0^{(0)}, y_1^{(0)}, y_2^{(0)}, \ldots)$

  $\mathbf{y}^{(1)} = (y_0^{(1)}, y_1^{(1)}, y_2^{(1)}, \ldots)$

- Multiplex into a single data stream

$$\mathbf{y} = (y_0^{(0)}, y_0^{(1)}, y_1^{(0)}, y_1^{(1)}, y_2^{(0)}, y_2^{(1)}, \ldots)$$

Convolutional Coding Applications

- Wi-Fi (802.11 standard) and cellular networks (3G, 4G, LTE standards)
- Deep space satellite communications
- Digital Video Broadcasting (Digital TV)
- Building block in more advanced codes such as Turbo Codes

ETSI TS 136 212 V8.4.0 (2008-11) Technical Specification

LTE; Evolved Universal Terrestrial Radio Access (E-UTRA);
Multiplexing and channel coding

Rate 1/3 Convolutional Encoder

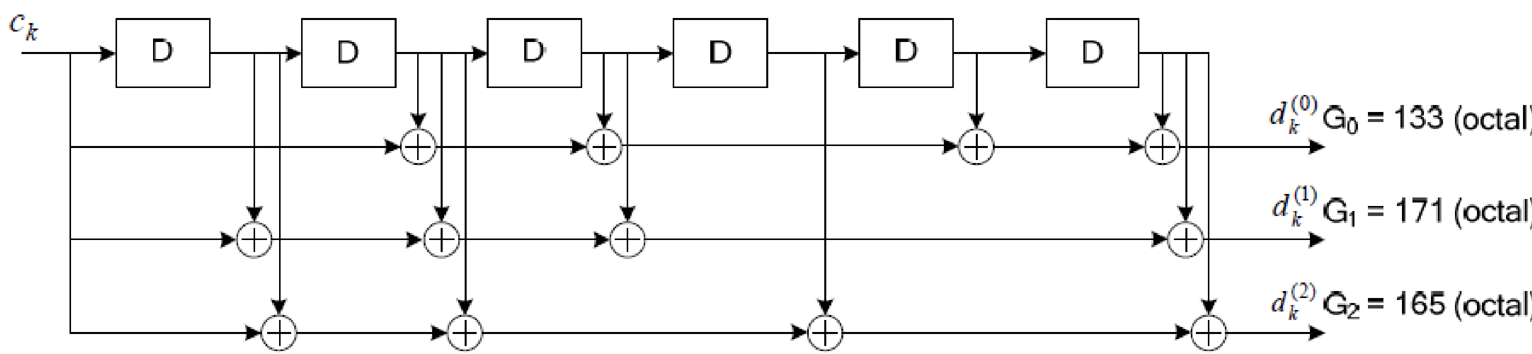
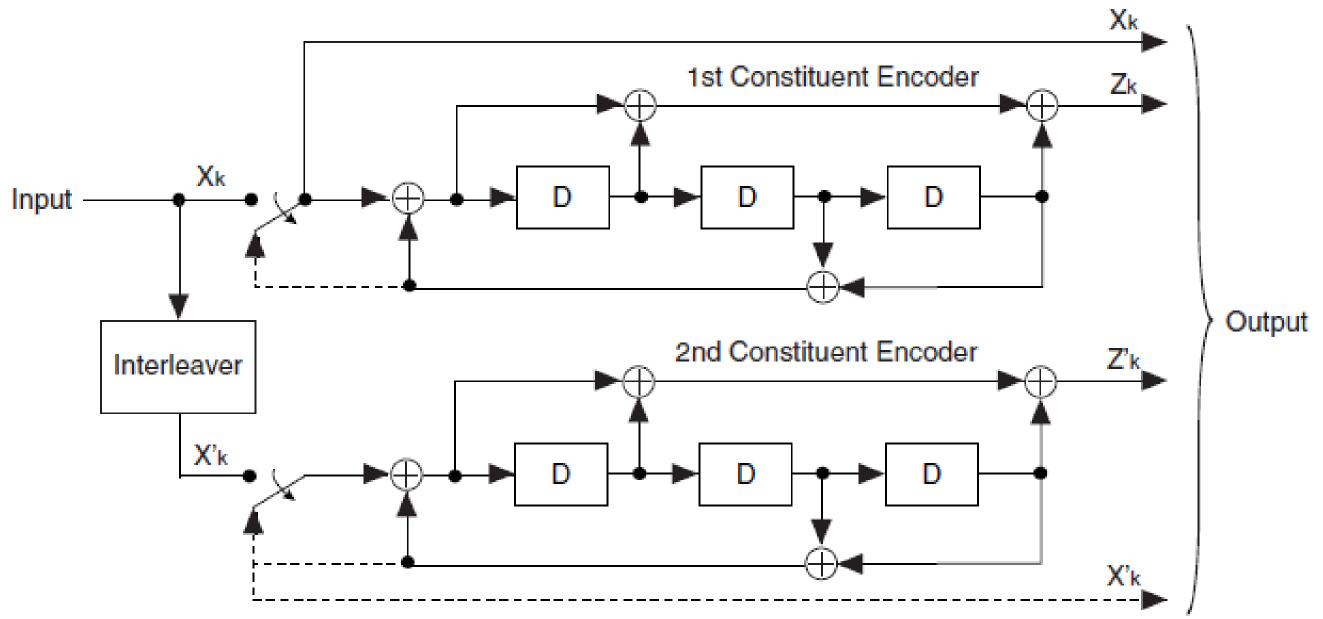
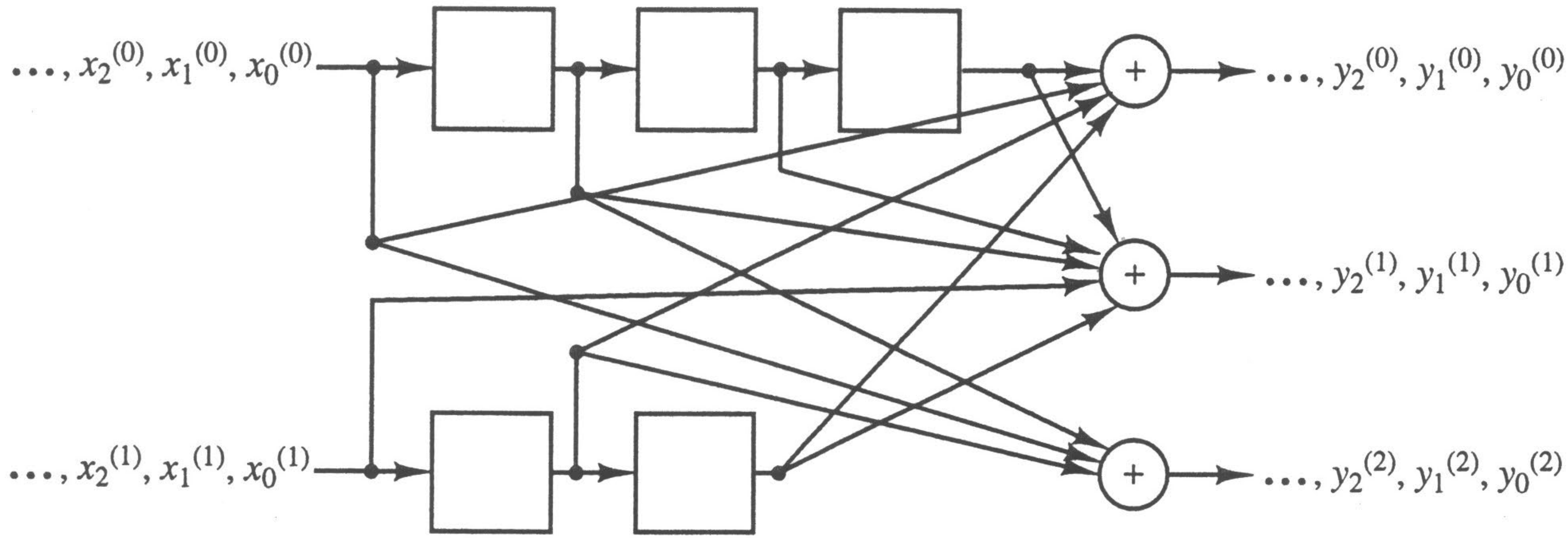


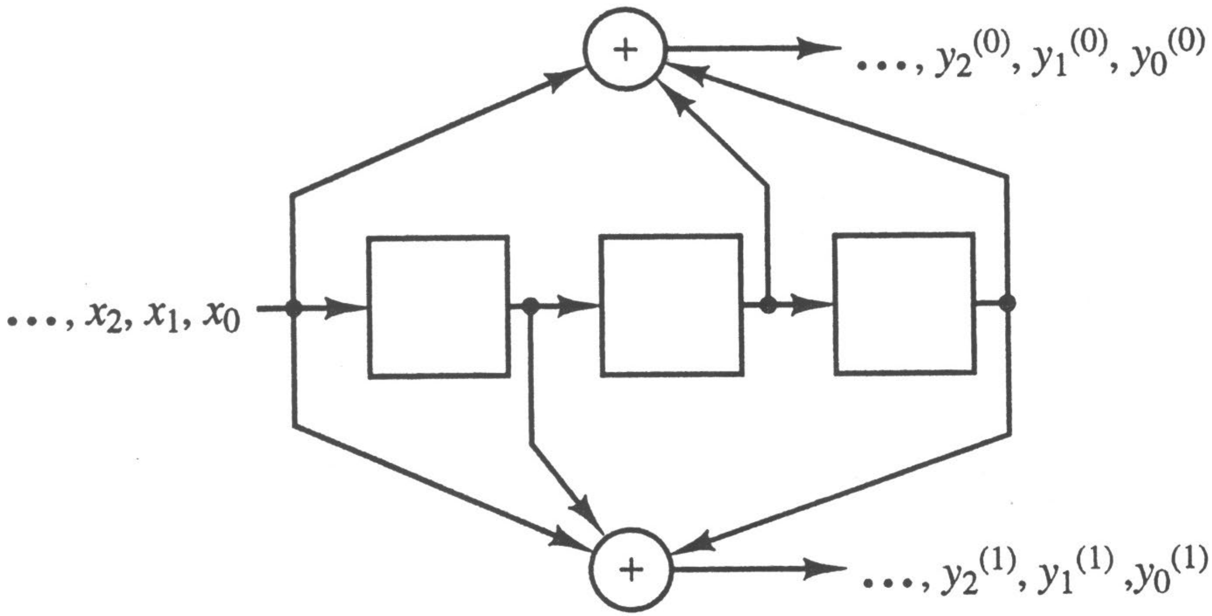
Figure 2. Structure of a Rate 1/3 Turbo Encoder



Rate 2/3 Linear Convolutional Encoder



Rate 1/2 Linear Convolutional Encoder



Example 11-1

- Input data stream $\mathbf{x} = (10110)$

  first input bit ↗

- Output data streams $\mathbf{y}^{(0)} = (10001010)$

  $\mathbf{y}^{(1)} = (11111110)$

- Multiplex into a single data stream

$$\mathbf{y} = (11,01,01,01,11,01,11,00)$$

Fractional Rate Loss

- Message length L
- Codeword length is $L(n/k) + nm$
- Effective code rate is

$$R_{effective} = \frac{L}{L(n/k) + nm}$$

- Fractional rate loss is

$$\gamma = \frac{R - R_{effective}}{R} = \frac{km}{L + km}$$

Example 11-2

- From Example 11-1, the message length is $L = 5$
- Codeword length is $L(n/k) + nm = 5(2/1)+2\cdot3 = 16$
- Effective code rate is

$$R_{effective} = \frac{L}{L(n/k)+nm} = \frac{5}{10+6} = \frac{5}{16}$$

- Fractional rate loss is

$$\gamma = \frac{R - R_{effective}}{R} = \frac{km}{L+km} = \frac{3}{8}$$

Fractional Rate Loss

- Typically k = 1 and L is large: L » m
- In this case

$$R_{effective} = \frac{1}{n}\frac{L}{\left(L+m\right)} \to \frac{1}{n} \text{ as } L \to \infty$$

- Fractional rate loss becomes

$$\gamma = \frac{m}{L+m} \to 0$$

- Memory order m: length of the longest shift register
  - $m = \max m_i$
- Constraint length: $K = 1 + \max m_i$
- Total memory (overall constraint length): sum of all memory elements $M = \Sigma m_i$
- Number of input streams: k
- Number of output streams: n
- Code rate: k/n
- State: contents of the shift registers
  - Initial state: 00…0
  - End state: 00…0 (input zeros at the end)

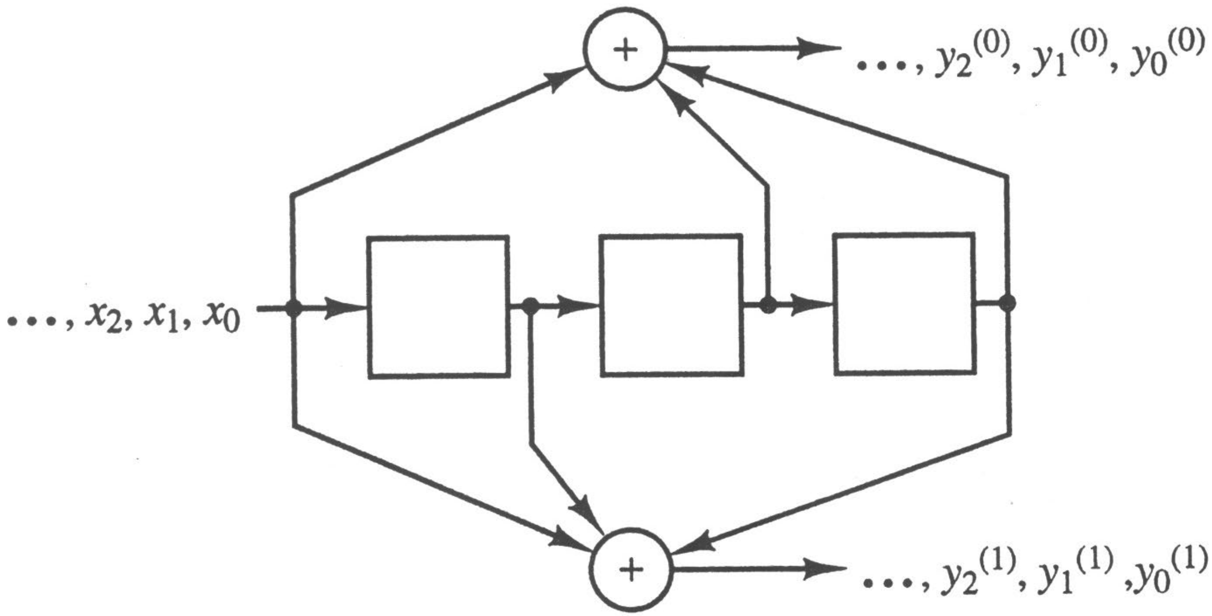
Punctured Convolutional Codes

- Complexity is a function of the total memory or overall constraint length
- Thus codes with $k > 1$ are rarely used
- Puncturing allows for higher code rates from a rate 1/2 code
- For rate 2/3, delete one of every 4 output bits

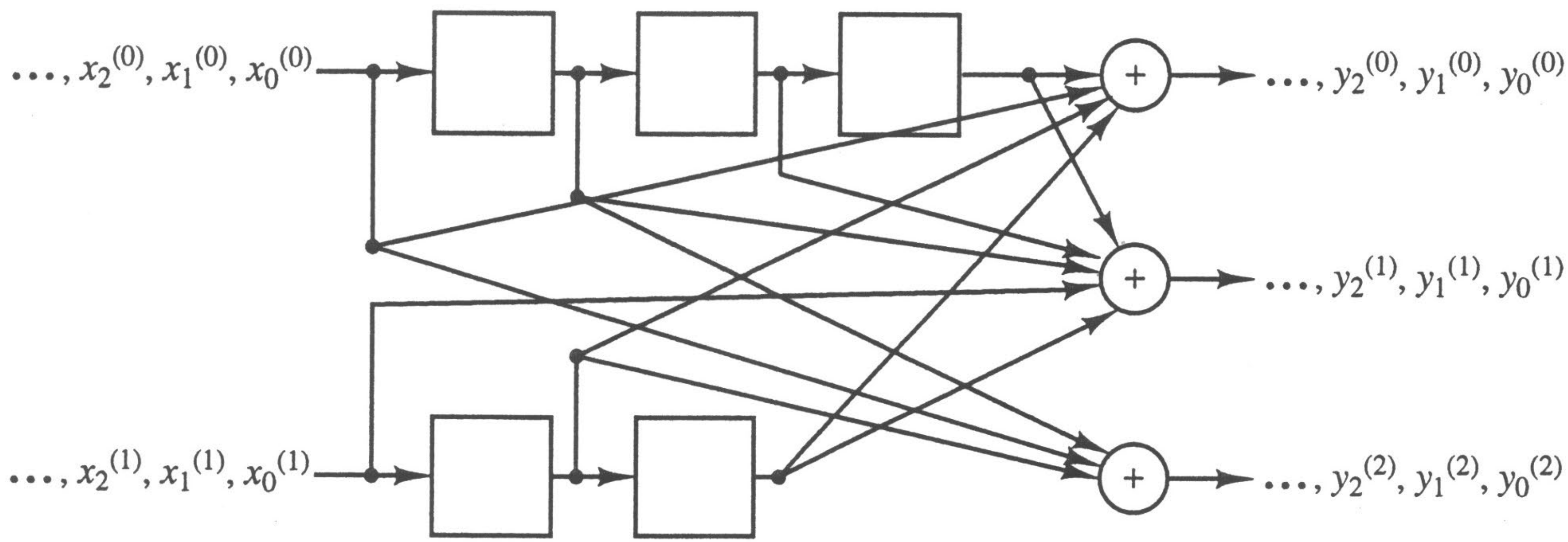
input x_1 x_2 output y_1 y_2 y_3 y_4

remove y_3 ↗

Rate 1/2 Linear Convolutional Encoder



Rate 2/3 Linear Convolutional Encoder



Generator Sequences

- $\mathbf{g}_j^{(i)}$ represents the *i*th output of an encoder when the *j*th input is a single 1 followed by a string of zeros

$$\mathbf{x}^{(j)} = \delta = (10000\ldots)$$

- Rate 1/2 example (K = 4)

$$\mathbf{g}^{(0)} = (1011)$$

$$\mathbf{g}^{(1)} = (1101)$$

- Rate 2/3 example

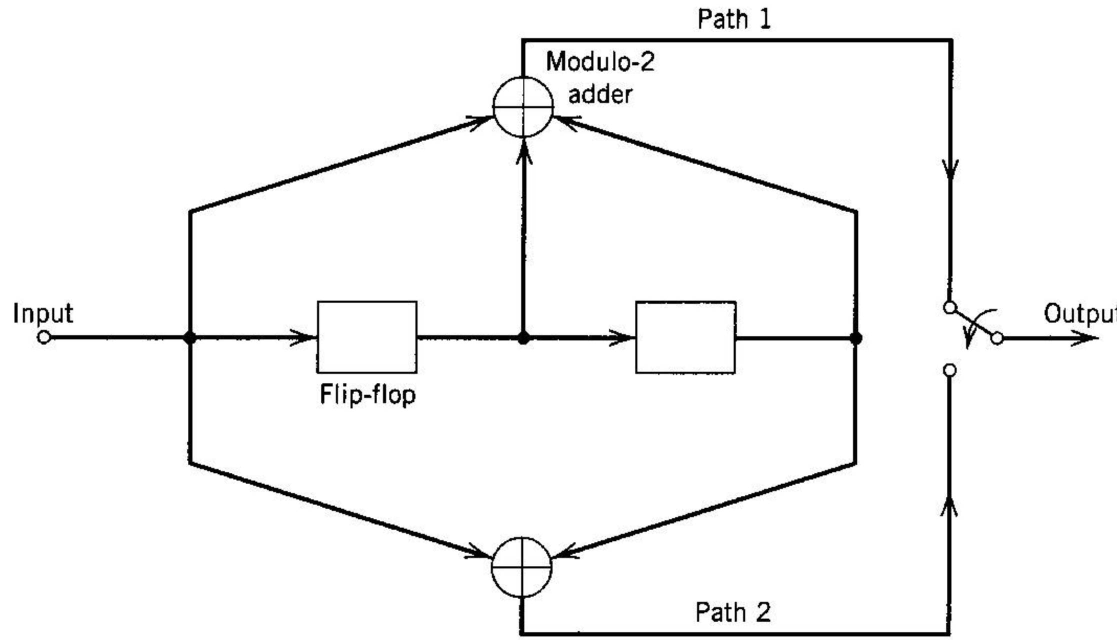
$$\mathbf{g}_0^{(0)} = (1001) \qquad \mathbf{g}_1^{(0)} = (0110)$$

$$\mathbf{g}_0^{(1)} = (0111) \qquad \mathbf{g}_1^{(1)} = (1010)$$

$$\mathbf{g}_0^{(2)} = (1100) \qquad \mathbf{g}_1^{(2)} = (0100)$$

Rate 1/2 Linear Convolutional Encoder

- (2,1,3) code
- $\mathbf{g}^{(0)} = 111$
- $\mathbf{g}^{(1)} = 101$



Why Convolutional Code?

- For the rate 1/2 (2,1,4) code with impulse response

  $\mathbf{g}^{(0)} = 1011 \qquad \mathbf{g}^{(1)} = 1101$

  The output of the encoder is

  $\mathbf{y}^{(0)} = \mathbf{x} * \mathbf{g}^{(0)} \qquad \mathbf{y}^{(1)} = \mathbf{x} * \mathbf{g}^{(1)}$

  If $\mathbf{x} = (10110)$, the output is

  $\mathbf{y}^{(0)} = (10110)*(1011)=10001010$

  $\mathbf{y}^{(1)} = (10110)*(1101)=11111110$

  $\mathbf{y}$ = 11 01 01 01 11 01 11 00

- A convolutional code is in fact a linear block code for finite input length L.
- For a rate 1/2 code, the generator matrix can be formed by interleaving $\mathbf{g}^{(0)}$ and $\mathbf{g}^{(1)}$

$$\mathbf{G} = \begin{bmatrix} g_0^{(0)} g_0^{(1)} & g_1^{(0)} g_1^{(1)} & \cdots & g_m^{(0)} g_m^{(1)} & & \mathbf{0} \\ & g_0^{(0)} g_0^{(1)} & g_1^{(0)} g_1^{(1)} & \cdots & g_m^{(0)} g_m^{(1)} & \\ & & \ddots & \ddots & & \ddots \\ \mathbf{0} & & & & & \end{bmatrix}$$

L x $2(m+L)$

- For the rate 1/2 (2,1,4) code

$$\mathbf{G} = \begin{bmatrix} 11 & 01 & 10 & 11 & & \mathbf{0} \\ & 11 & 01 & 10 & 11 & \\ \mathbf{0} & & \ddots & & \ddots & \end{bmatrix}$$

- For the rate 2/3 (3,2,4) code

$$\mathbf{G} = \begin{bmatrix} 101 & 011 & 010 & 110 & & \mathbf{0} \\ 010 & 101 & 110 & 000 & & \\ & 101 & 011 & 010 & 110 & \\ & 010 & 101 & 110 & 000 & \\ \mathbf{0} & & \ddots & & \ddots & \end{bmatrix}$$

Example 11-3

- (2,1,4) code
- input **x** = (1011), *L* = 4, *m* = 3
- **G** has dimensions 4×14

$$\mathbf{G} = \begin{bmatrix} 11 & 01 & 10 & 11 & 00 & 00 & 00 \\ 00 & 11 & 01 & 10 & 11 & 00 & 00 \\ 00 & 00 & 11 & 01 & 10 & 11 & 00 \\ 00 & 00 & 00 & 11 & 01 & 10 & 11 \end{bmatrix}$$

- **y=xG =** (11,01,01,01,11,01,11)

Polynomial Representation

- Introduce the delay operator D

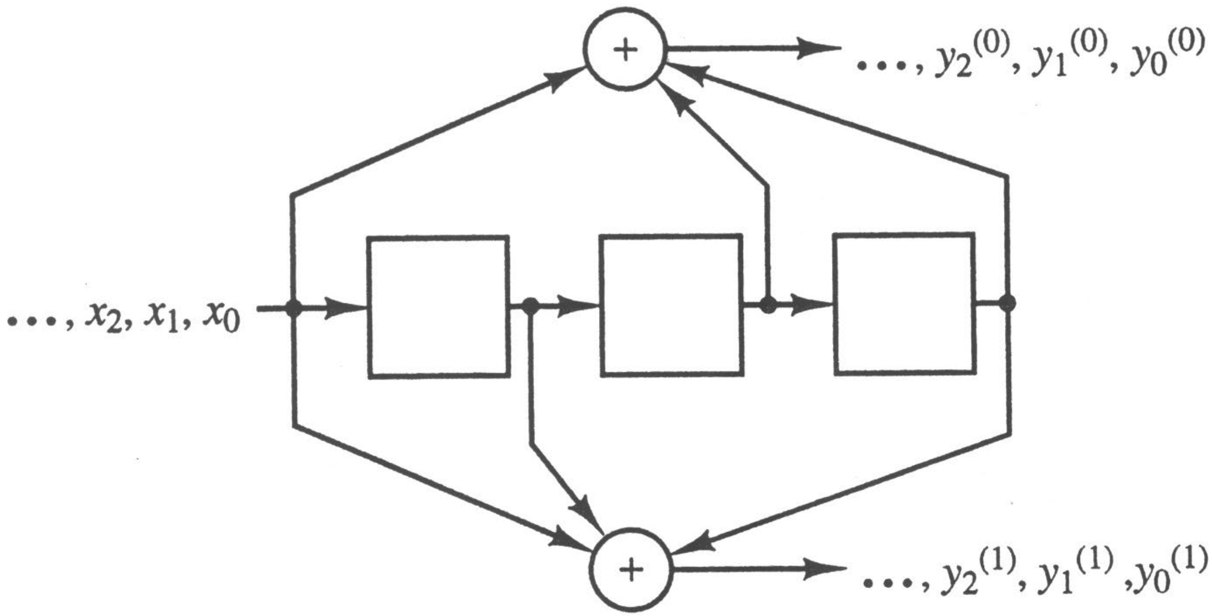
$$\mathbf{x} = (x_0, x_1, x_2, \cdots) \leftrightarrow \mathbf{X}(D) = x_0 + x_1 D + x_2 D^2 + \cdots$$

$$\mathbf{y}^{(0)} = (y_0^{(0)}, y_1^{(0)}, y_2^{(0)}, \ldots) \leftrightarrow \mathbf{Y}^{(0)}(D) = y_0^{(0)} + y_1^{(0)} D + y_2^{(0)} D^2 + \ldots$$

$$\mathbf{y}^{(1)} = (y_0^{(1)}, y_1^{(1)}, y_2^{(1)}, \ldots) \leftrightarrow \mathbf{Y}^{(1)}(D) = y_0^{(1)} + y_1^{(1)} D + y_2^{(1)} D^2 + \ldots$$

$$g^{(i)} \leftrightarrow \mathbf{G}^{(i)}(D)$$

(2,1,4) Convolutional Encoder



Polynomial Representation

- For the (2,1,4) code example

  $\mathbf{G}^{(0)}(D) = 1+D^2+D^3$ $\mathbf{X}(D) = 1+D^2+D^3$

  $\mathbf{G}^{(1)}(D) = 1+D+D^3$

- Use polynomial multiplication to obtain the output

  $\mathbf{Y}^{(0)}(D) = \mathbf{X}(D)\ \mathbf{G}^{(0)}(D) = 1+D^4+D^6$

  $\mathbf{Y}^{(1)}(D) = \mathbf{X}(D)\ \mathbf{G}^{(1)}(D) = 1+D+D^2+D^3+D^4+D^5+D^6$

  $\mathbf{Y}(D) = \mathbf{Y}^{(0)}(D^2) + D\mathbf{Y}^{(1)}(D^2)$

  $= 1+D+D^3+D^5+D^7+D^8+D^9+D^{11}+D^{12}+D^{13}$

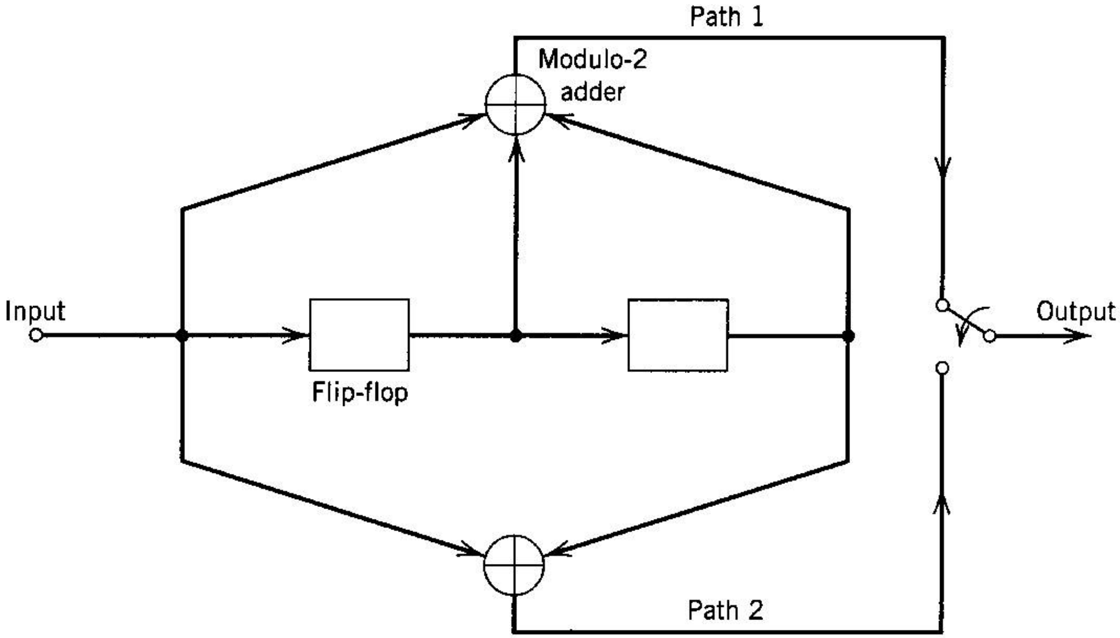
Rate 1/2 (2,1,3) Code

- $\mathbf{G}^{(0)}(D) = 1+D+D^2$
- $\mathbf{G}^{(1)}(D) = 1+D^2$
- $\mathbf{X}(D) = 1+D^3+D^4$
- $\mathbf{Y}^{(0)}(D) = \mathbf{X}(D)\ \mathbf{G}^{(0)}(D)$

  $= 1+D+D^2+D^3+D^6$
- $\mathbf{Y}^{(1)}(D) = \mathbf{X}(D)\ \mathbf{G}^{(1)}(D)$

  $= 1+D^2+D^3+D^4+D^5+D^6$
- $\mathbf{Y}(D) = \mathbf{Y}^{(0)}(D^2) + D\mathbf{Y}^{(1)}(D^2)$

  $= 1+D+D^2+D^4+D^5+D^6+D^7+D^9+D^{11}+D^{12}+D^{13}$



- For 3 outputs

$$\mathbf{Y}(D) = \mathbf{Y}^{(0)}(D^3) + D\mathbf{Y}^{(1)}(D^3) + D^2\mathbf{Y}^{(2)}(D^3)$$

- For a multiple input, multiple output system

$$\mathbf{Y}^{(i)}(D) = \sum_{j=0}^{k-1} \mathbf{X}^{(j)}(D)\mathbf{G}_j^{(i)}(D)$$

$$= [\mathbf{X}^{(0)}(D)\ \mathbf{X}^{(1)}(D)\cdots\mathbf{X}^{(k-1)}(D)] \begin{bmatrix} \mathbf{G}_0^{(i)}(D) \\ \mathbf{G}_1^{(i)}(D) \\ \vdots \\ \mathbf{G}_{k-1}^{(i)}(D) \end{bmatrix}$$

$$\mathbf{Y}(D)=\left(\mathbf{Y}^{(0)}(D),\mathbf{Y}^{(1)}(D),\cdots,\mathbf{Y}^{(n-1)}(D)\right)$$

$$=\mathbf{X}(D)\mathbf{G}(D)=[\mathbf{X}^{(0)}(D)\ \mathbf{X}^{(1)}(D)\cdots\mathbf{X}^{(k-1)}(D)]\begin{bmatrix}\mathbf{G}_0^{(0)}(D) & \mathbf{G}_0^{(1)}(D) & \cdots & \mathbf{G}_0^{(n-1)}(D)\\ \mathbf{G}_1^{(0)}(D) & \mathbf{G}_1^{(1)}(D) & \cdots & \mathbf{G}_1^{(n-1)}(D)\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{G}_{k-1}^{(0)}(D) & \mathbf{G}_{k-1}^{(1)}(D) & \cdots & \mathbf{G}_{k-1}^{(n-1)}(D)\end{bmatrix}$$

$$\mathbf{Y}(D)=\sum_{i=0}^{n-1}D^i\mathbf{Y}^{(i)}(D^n)$$

$\uparrow$ $\mathbf{G}(D)$

- $\mathbf{G}(D)$ is called the transfer-function matrix for the encoder

- For the (2,1,4) code

$\mathbf{G}(D) = [1+D^2+D^3 \;\; 1+D+D^3]$

- For the (3,2,4) code

$$\mathbf{G}_0^{(0)} = 1 + D^3 \qquad \mathbf{G}_1^{(0)} = D + D^2$$

$$\mathbf{G}_0^{(1)} = D + D^2 + D^3 \qquad \mathbf{G}_1^{(1)} = 1 + D^2$$

$$\mathbf{G}_0^{(2)} = 1 + D \qquad \mathbf{G}_1^{(2)} = D$$

$$\mathbf{G}(D) = \begin{bmatrix} 1 + D^3 & D + D^2 + D^3 & 1 + D \\ D + D^2 & 1 + D^2 & D \end{bmatrix}$$

Example 11-4 Rate 2/3 Code

- Let **x** = (11,10,11)
- $\mathbf{X}^{(0)}(D) = 1+D+D^2 \quad \mathbf{X}^{(1)}(D) = 1+D^2$
- $\mathbf{Y}(D) = \mathbf{X}(D)\mathbf{G}(D)$

$$= \begin{bmatrix} 1+D+D^2 & 1+D^2 \end{bmatrix} \begin{bmatrix} 1+D^3 & D+D^2+D^3 & 1+D \\ D+D^2 & 1+D^2 & D \end{bmatrix}$$

$$= \begin{bmatrix} 1+D^5 & 1+D+D^3+D^4+D^5 & 1+D \end{bmatrix}$$

- To convert from matrix form to the output string

$$\mathbf{Y}(D) = \sum_{i=0}^{2} D^i \mathbf{Y}^{(i)}(D^3)$$

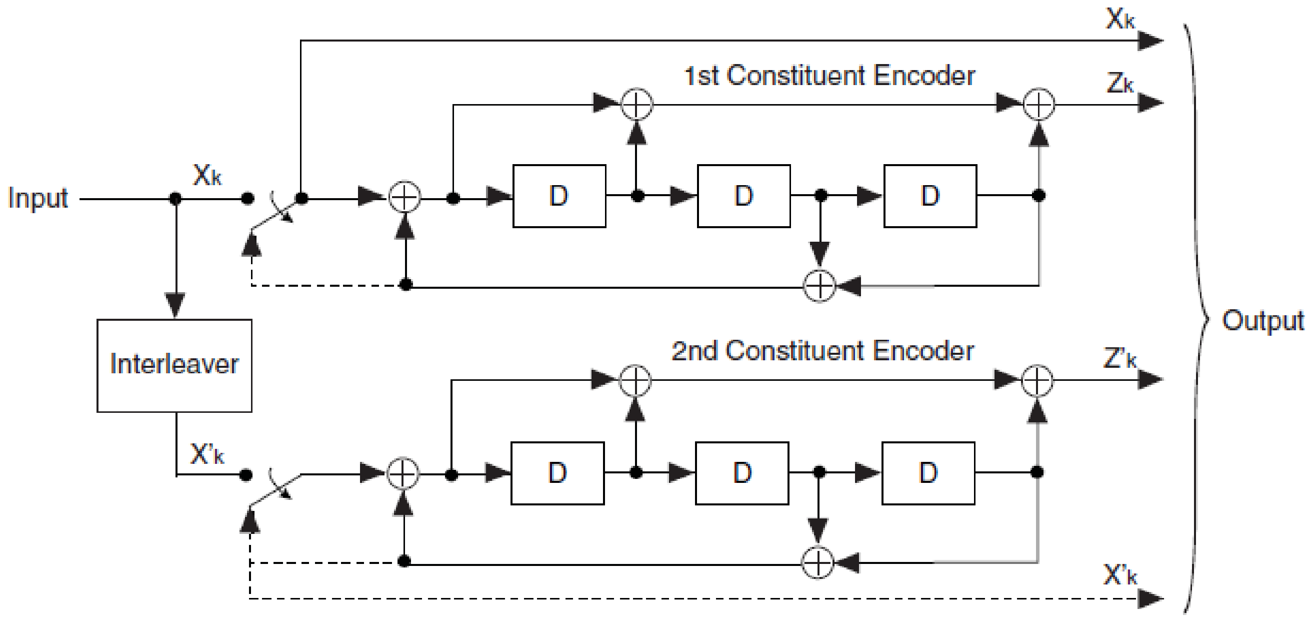
$$= 1 + D^{15} + D(1 + D^3 + D^9 + D^{12} + D^{15}) + D^2(1 + D^3)$$

$$= 1 + D + D^2 + D^4 + D^5 + D^{10} + D^{13} + D^{15} + D^{16}$$

- The output codeword is then

  y = (111,011,000,010,010,110)

LTE Turbo Code



$$\mathbf{G}(D) = \left[1, \quad \frac{\mathbf{G}_1(D)}{\mathbf{G}_0(D)} \right]$$

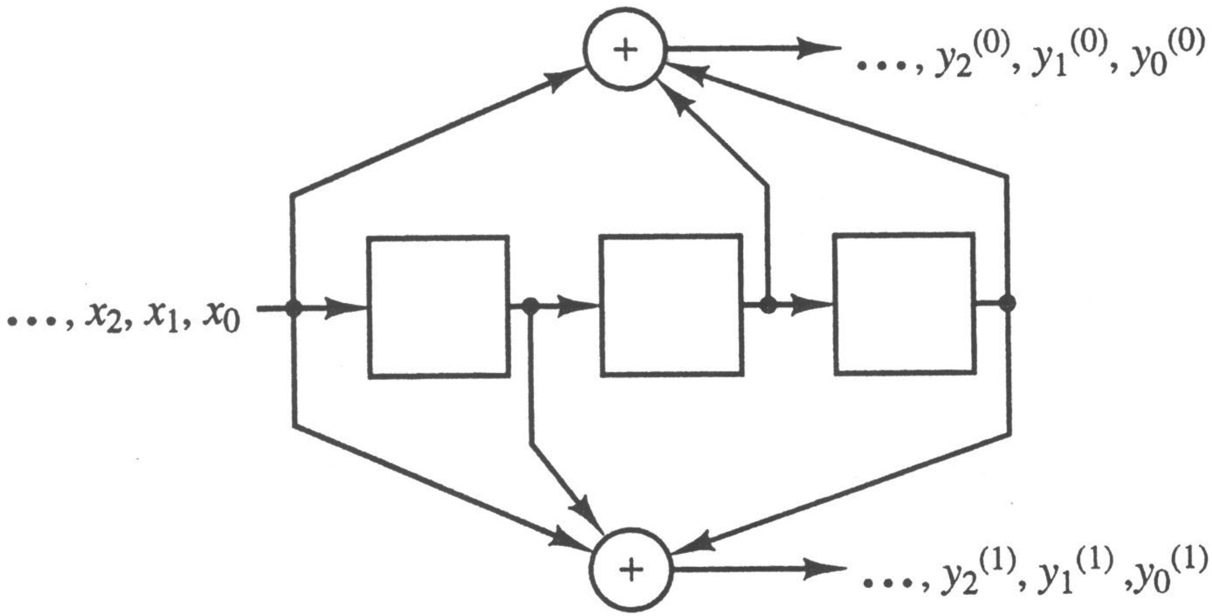
$$\mathbf{G}_0(D) = 1 + D^2 + D^3$$

$$\mathbf{G}_1(D) = 1 + D + D^3$$

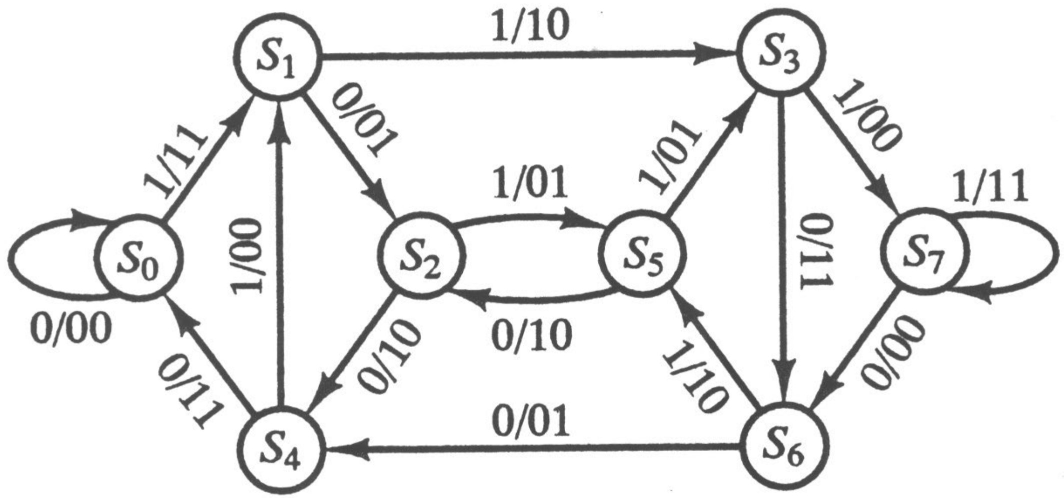
State Diagrams

- The state of an encoder is defined as the shift register contents
- For an $(n,1,K)$ code there are $2^{K\text{-}1} = 2^m$ states
- Each branch in the state diagram has a label of the form X/Y, where X is the input bit that causes the state transition and Y is the corresponding output bits.
- Path: a sequence of nodes connected by branches
- Circuit: a path that starts and stops at the same node
- Loop: a circuit that does not enter any node more than once.

Rate 1/2 Linear Convolutional Encoder



State Diagram for the (2,1,4) Encoder



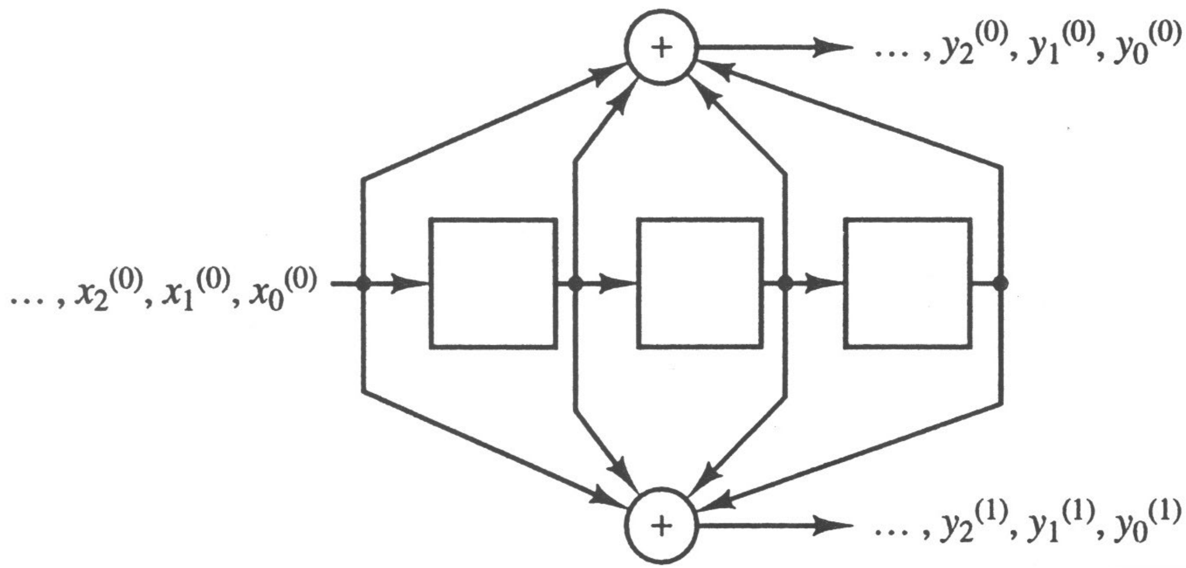
- Example 11-5: Consider the input **x** = (1011)
- Begin at state S_0

input	output	state
-	-	S_0
1	11	S_1
0	01	S_2
1	01	S_5
1	01	S_3
0	11	S_6
0	01	S_4
0	11	S_0

Catastrophic Codes

- Catastrophic code: A code for which an infinite weight input causes a finite weight output
- Result: A finite number of channel errors can cause an infinite number of decoding errors
- In terms of the state diagram: A loop corresponding to a nonzero input for which all the output bits are zero denotes a catastrophic code

Catastrophic Encoder



- $g^{(0)} = (1111)$ $g^{(1)} = (1111)$
- **x** = (111111…)
- **y** = (11,00,11,00,00,00,…)
- Now suppose **r** = (10,00,00,00,00,00,…)
- A maximum likelihood decoder will choose

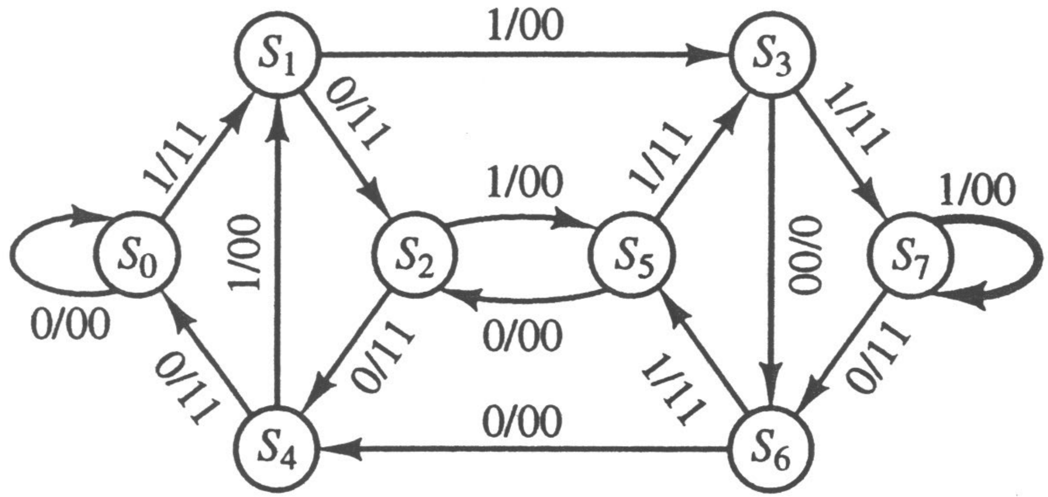
  c' = (00,00,00,00,00,00,…)

  which corresponds to

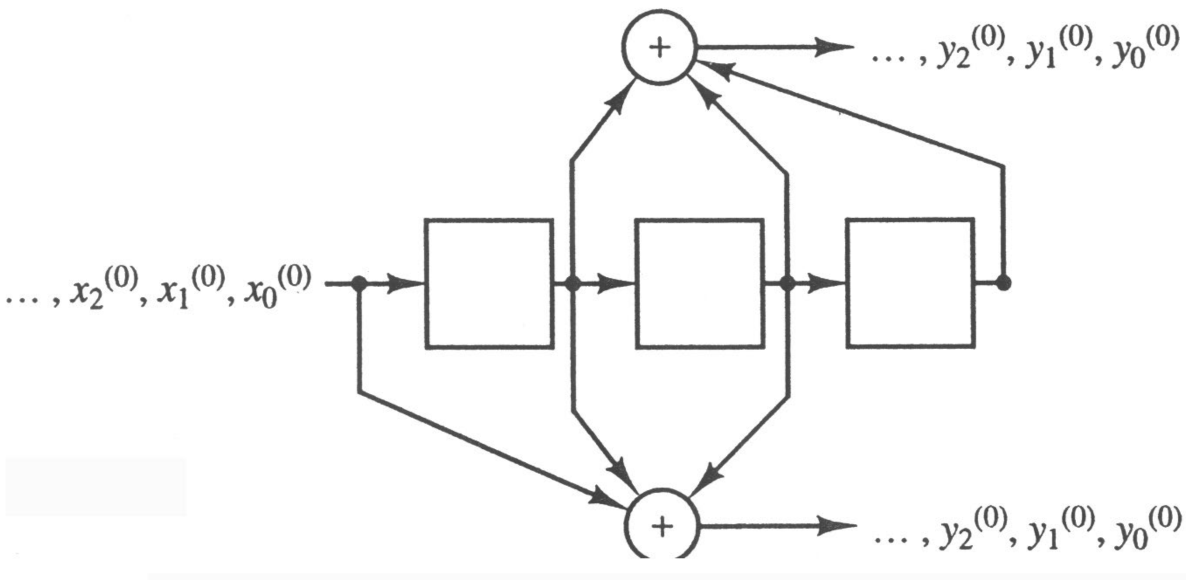
  x' = (000000…)

  and an infinite number of errors

State Diagram for a Catastrophic Code



Example 11-6
Another Catastrophic Encoder

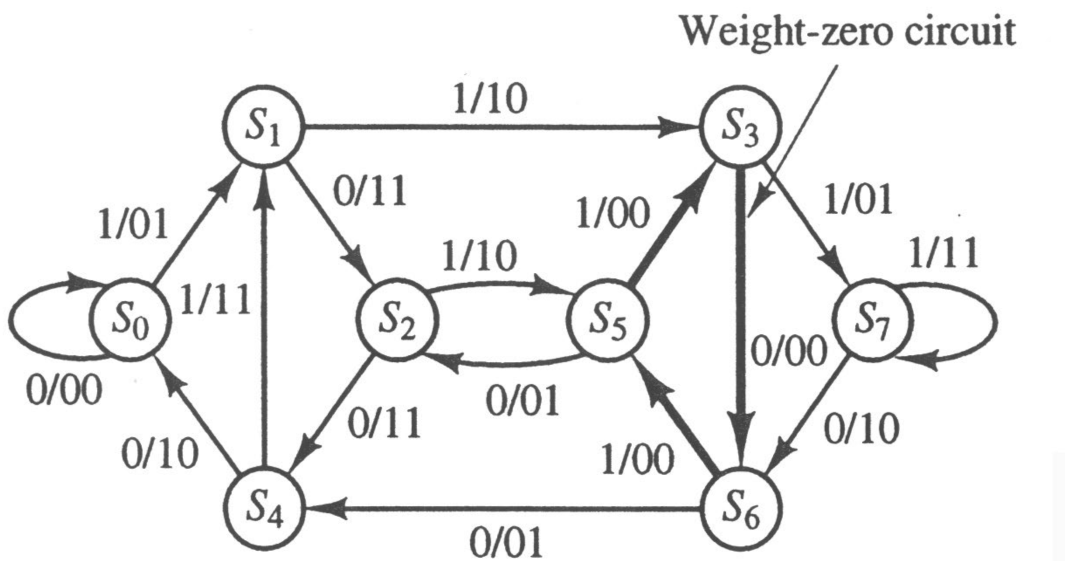


$\mathbf{g}^{(0)} = (0111)$ $\mathbf{g}^{(1)} = (1110)$

$\mathbf{x} = (11011011\ldots)$

$\mathbf{y} = (01,10,00,00,00,00,\ldots)$

The Corresponding State Diagram



Theorem 11-1 Catastrophic Codes

Let C be a rate $1/n$ convolutional code with transfer function matrix $\mathbf{G}(D)$ whose generator sequences have the transforms $\{\mathbf{G}^{(0)}(D), \mathbf{G}^{(1)}(D), \ldots, \mathbf{G}^{(n-1)}(D)\}$.

C is not catastrophic if and only if

$$\mathrm{GCD}(\mathbf{G}^{(0)}(D), \mathbf{G}^{(1)}(D), \ldots, \mathbf{G}^{(n-1)}(D)) = D^{l}$$

for $l \geq 0$.

Catastrophic Code Examples

- Example 11-6 (previous) code
  - $\mathbf{g}^{(0)} = (0111)$ $\mathbf{g}^{(1)} = (1110)$
  - $\mathbf{G}^{(0)}(D) = D+D^2+D^3$ $\mathbf{G}^{(1)}(D) = 1+D+D^2$
  - $\mathrm{GCD}(\mathbf{G}^{(0)}(D),\mathbf{G}^{(1)}(D)) = 1+D+D^2 \neq D^l$

- For the other code
  - $\mathbf{g}^{(0)} = (1111)$ $\mathbf{g}^{(1)} = (1111)$
  - $\mathrm{GCD}(\mathbf{G}^{(0)}(D),\mathbf{G}^{(1)}(D)) = 1+D+D^2+D^3 \neq D^l$

Weight Enumerator

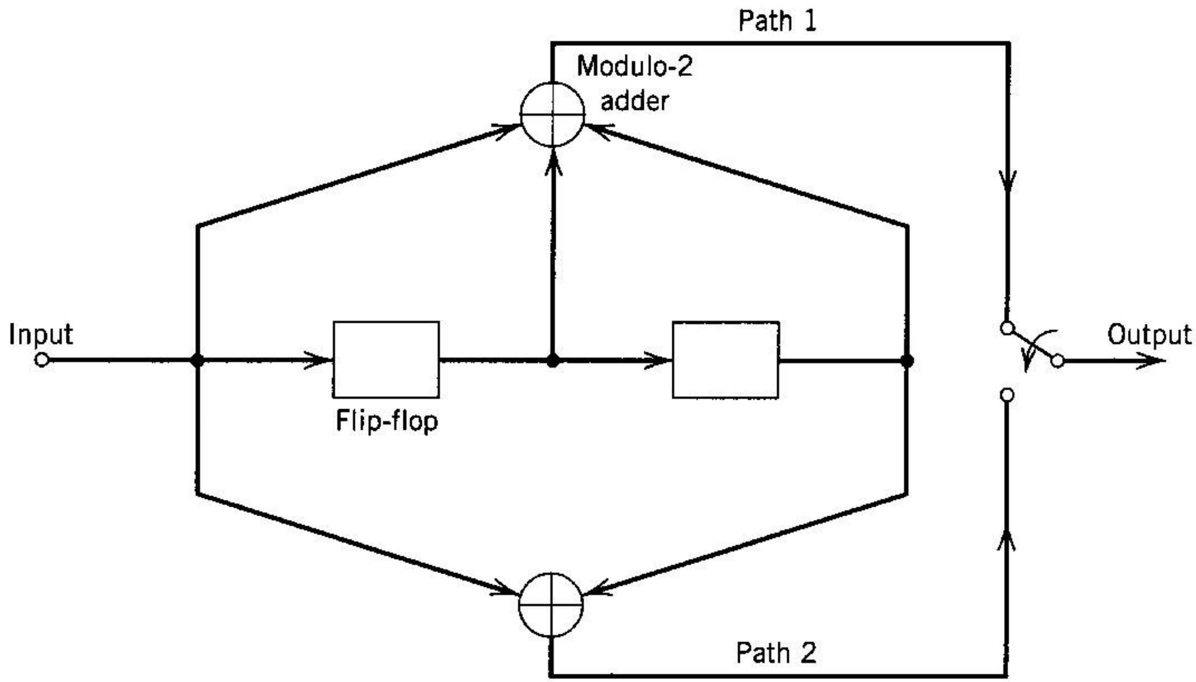
- Label each branch in the state diagram with X^iY^j, where i is the weight of the output sequence and j is the weight of the input sequence.
- The generating function is

$$T(X,Y) = \sum_i \sum_j a_{i,j} X^i Y^j$$

where $a_{i,j}$ is the number of codewords of weight i corresponding to input sequences of weight j.

- $T(X,Y)$ can be obtained using Mason's formula.

(2,1,3) Convolutional Encoder



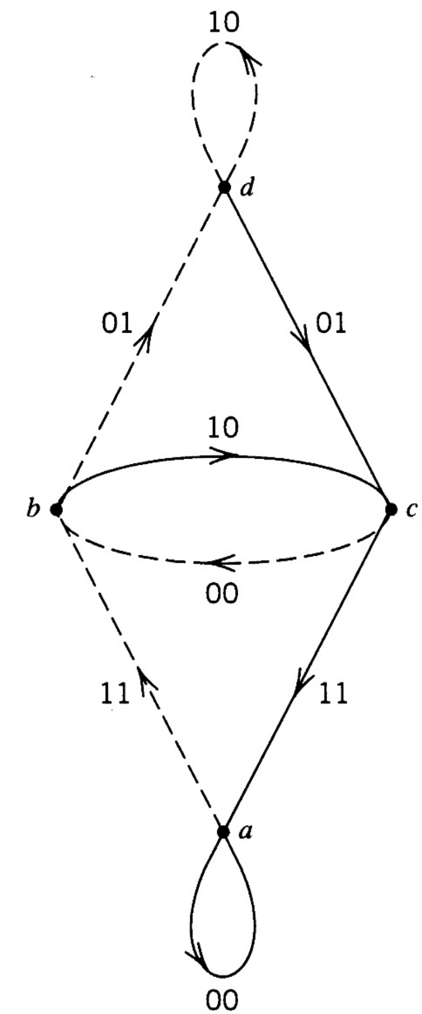
(a)

$$\mathbf{G}(D) = [1+D+D^2 \;\; 1+D^2]$$

State Diagram for the (2,1,3) Encoder

- Solid lines represent input 0
- Dashed lines represent input 1

state	SR contents
a	00
b	10
c	01
d	11



Example

- Consider the input sequence **x** = (10011)
- Begin at state **a**

input	1	0	0	1	1
output	11	10	11	11	01
new state	**b**	**c**	**a**	**b**	**d**

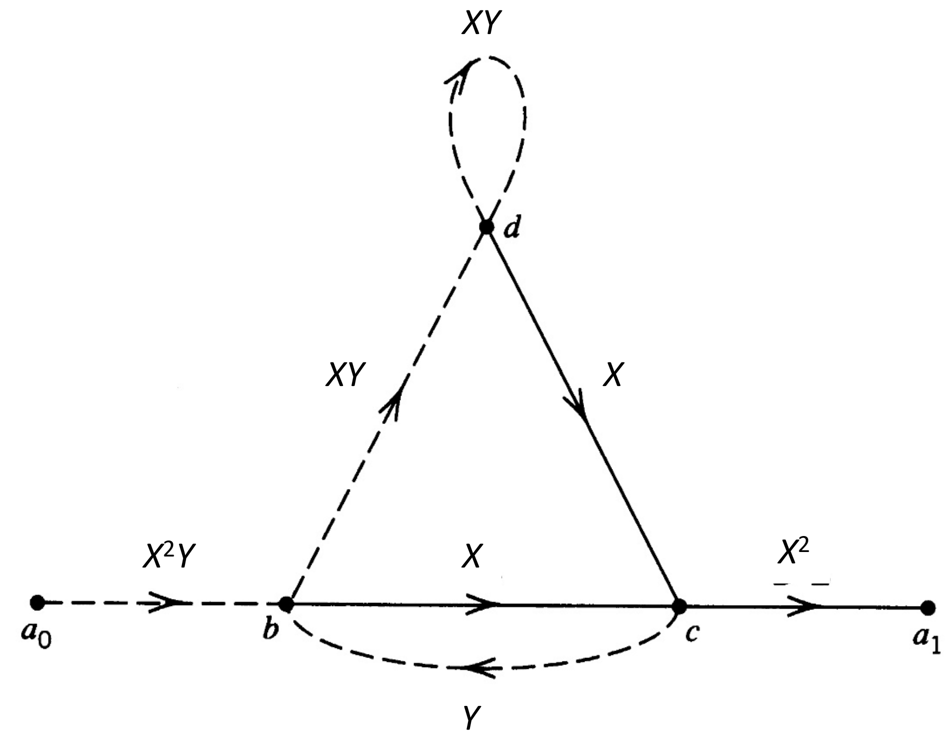
- To return to the all zero state

input	0	0
output	01	11
new state	**c**	**a**

Modified State Diagram for the (2,1,3) Code

- X = output weight
- Y = input weight

- $T(X,Y) = \frac{a_1}{a_0}$



Mason's Formula

$$T(X,Y) = \frac{\sum_i F_i \Delta_i}{\Delta}$$

- Forward paths F_i
- Graph determinant

$$\Delta = 1 - \sum_{L_l} C_l + \sum_{(L_l, L_m)} C_l C_m - \sum_{(L_l, L_m, L_n)} C_l C_m C_n + \cdots$$

- Cofactor of path K_i

$$\Delta_i = 1 - \sum_{(K_i, L_l)} C_l + \sum_{(K_i, L_l, L_m)} C_l C_m - \sum_{(K_i, L_l, L_m, L_n)} C_l C_m C_n + \cdots$$

(2,1,3) Code

- Two forward paths
  - a_0bca_1 $\quad F_1= X^2Y \bullet X \bullet X^2 = X^5Y$
  - a_0bdca_1 $\quad F_2= X^2Y \bullet XY \bullet X \bullet X^2 = X^6Y^2$
- Three loops
  - dd $\quad C_1= XY$
  - $bdcb$ $\quad C_2= XY \bullet X \bullet Y = X^2Y^2$
  - bcb $\quad C_3= X \bullet Y = XY$
- $\Delta=1-(XY+X^2Y^2+XY)+XY \bullet XY=1-2XY$
- $\Delta_1=1-XY \quad \Delta_2=1$

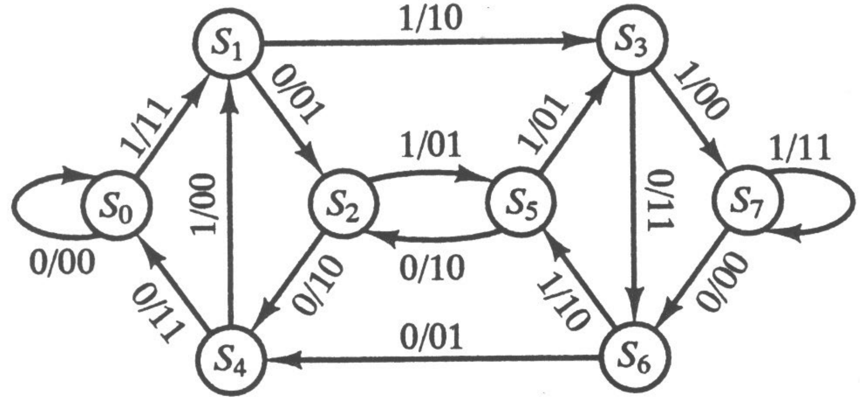
(2,1,3) Code

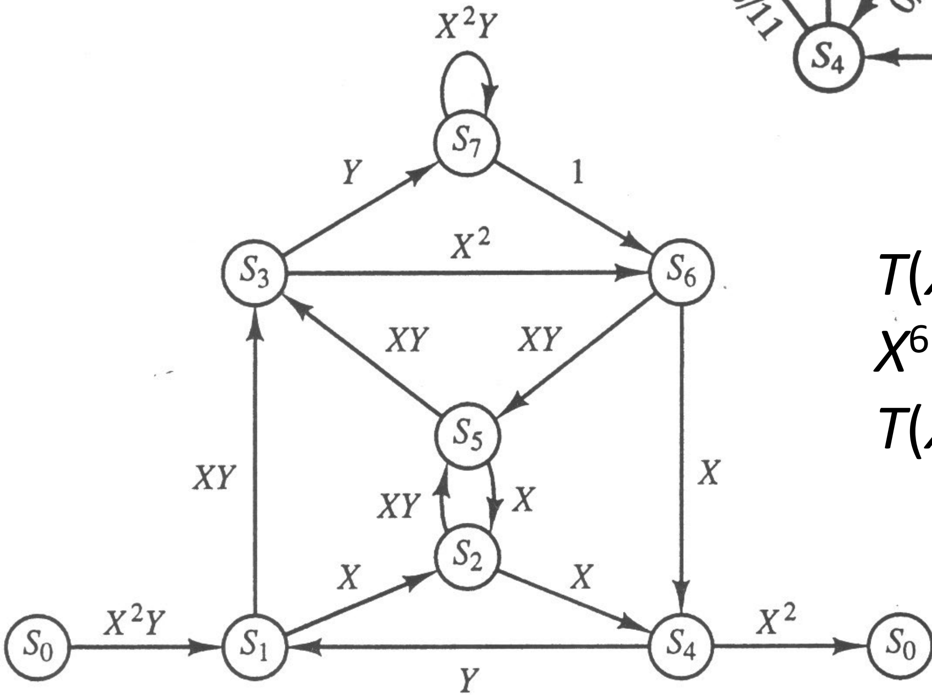
$$T(X,Y)=\frac{\sum_i F_i \Delta_i}{\Delta}=\frac{X^5 Y(1-XY)+X^6 Y^2}{1-2XY}$$

$$=\frac{X^5 Y}{1-2XY}$$

$$T(X)=\frac{X^5}{1-2X}=X^5+2X^6+4X^7+\cdots$$

(2,1,4) Code





$T(X,Y) =$
$X^6(Y+Y^3)+X^8(3Y^2+5Y^4+2Y^6)+\ldots$
$T(X) = 2X^6+10X^8+49X^{10}+\ldots$

Minimum Free Distance

- (2,1,4) code $T(X) = 2X^6+10X^8+49X^{10}+\ldots\ldots$
- The minimum free distance d_{free} is the minimum Hamming distance between all pairs of convolutional codewords. It is the minimum weight of an output sequence starting from the all-zero state and ending in the all-zero state

$$d_{free} = \min\{d(\mathbf{y}',\mathbf{y}'') \mid \mathbf{y}' \neq \mathbf{y}''\} = \min\{w(\mathbf{y}) \mid \mathbf{y} \neq 0\}$$

- For the (2,1,4) code, there are two codewords of weight 6, thus $d_{\text{free}} = 6$.

Maximum Free Distance Codes

- Nonsystematic convolutional codes typically provide a higher minimum free distance than systematic codes with the same constraint length and rate.

TABLE 11-1 Maximum d_{free} for rate-1/3 convolutional codes

Constraint length K	Systematic maximum d_{free}	Nonsystematic maximum d_{free}
2	5	5
3	6	8
4	8	10
5	9	12
6	10	13
7	12	15

TABLE 11-2 Maximum d_{free} for rate-1/2 convolutional codes

Constraint length K	Systematic maximum d_{free}	Nonsystematic maximum d_{free}
2	3	3
3	4	5
4	4	6
5	5	7
6	6	8
7	6	10

Best Known Convolutional Codes

TABLE 11-3 Rate-1/4 convolutional codes with maximal minimum free distance [Lin1][2]

K	$\mathbf{g}^{(0)}$	$\mathbf{g}^{(1)}$	$\mathbf{g}^{(2)}$	$\mathbf{g}^{(3)}$	d_{free}
3	5	7	7	7	10
4	54	64	64	74	13
5	52	56	66	76	16
6	53	67	71	75	18
7	564	564	634	714	20
8	472	572	626	736	22
9	463	535	733	745	24
10	4474	5724	7154	7254	27
11	4656	4726	5562	6372	29
12	4767	5723	6265	7455	32
13	44624	52374	66754	73534	33
14	42226	46372	73256	73276	36

TABLE 11-4 Rate-1/3 convolutional codes with maximal minimum free distance [Lin1]

K	$\mathbf{g}^{(0)}$	$\mathbf{g}^{(1)}$	$\mathbf{g}^{(2)}$	d_{free}
3	5	7	7	8
4	54	64	74	10
5	52	66	76	12
6	47	53	75	13
7	554	624	764	15
8	452	662	756	16
9	557	663	711	18
10	4474	5724	7154	20
11	4726	5562	6372	22
12	4767	5723	6265	24
13	42554	43364	77304	24
14	43512	73542	76266	26

ECE 405/511 Final Exam

- Saturday, April 22, 2023 9:00 AM ECS 130
  - 30% of the final grade
- Covers all course material up to and including convolutional codes (not weight enumerators)
  - Wicker Chapters 5, 8, 9, 11 (available on Brightspace)
  - Moreira and Farrell Chapters 2, 3, 4, 5, 6, Appendix B
- Aids allowed: 2 sheets of paper 8.5×11 in^2
  Calculator

Peter Elias (1923-2001)

- Coding for Noisy Channels