Algorithms in the Real World

Convolutional Coding & Viterbi Decoding

And now a word from my father…

"First, computer software and hardware are the most complex and rapidly developing intellectual creations of modem man."

-- p. iii, Internet and Computer Law, P. B. Maggs, J. T. Soma, and J. A. Sprowl, 2001

Today's lecture is based on

A Tutorial on Convolutional Coding with Viterbi Decoding

Chip Fleming

Spectrum Applications

http://home.netcom.com/~chip.f/viterbi/tutorial.html

Origin of Viterbi Decoding

Andrew J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Transactions on Information Theory*, Volume IT-13, pp. 260-269, April 1967.

Viterbi is a founder of Qualcomm.

Terminology

k number of message symbols (as before)

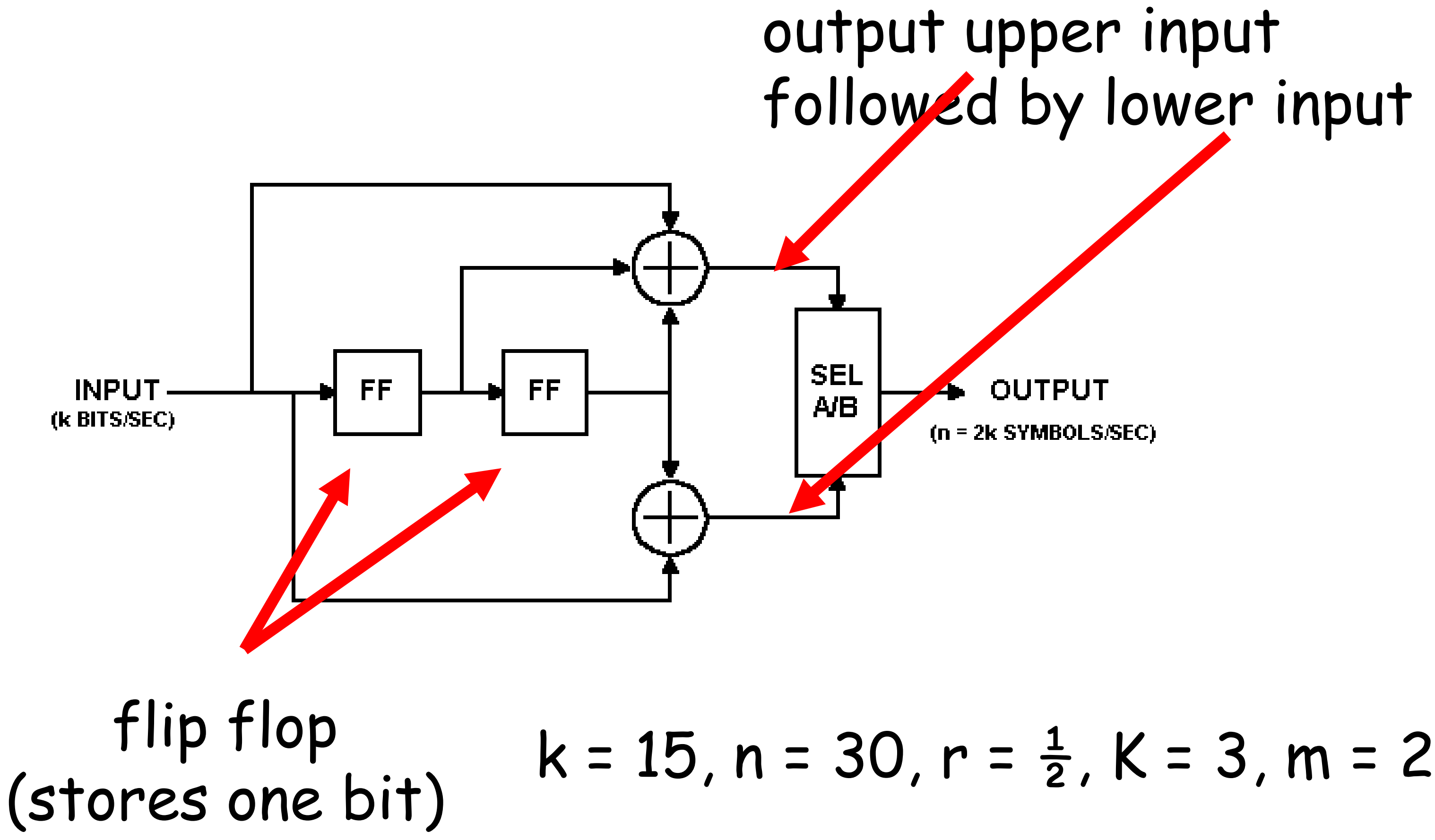
n number of codeword symbols (as before)

r rate = k/n

m number of encoding cycles an input symbol is stored

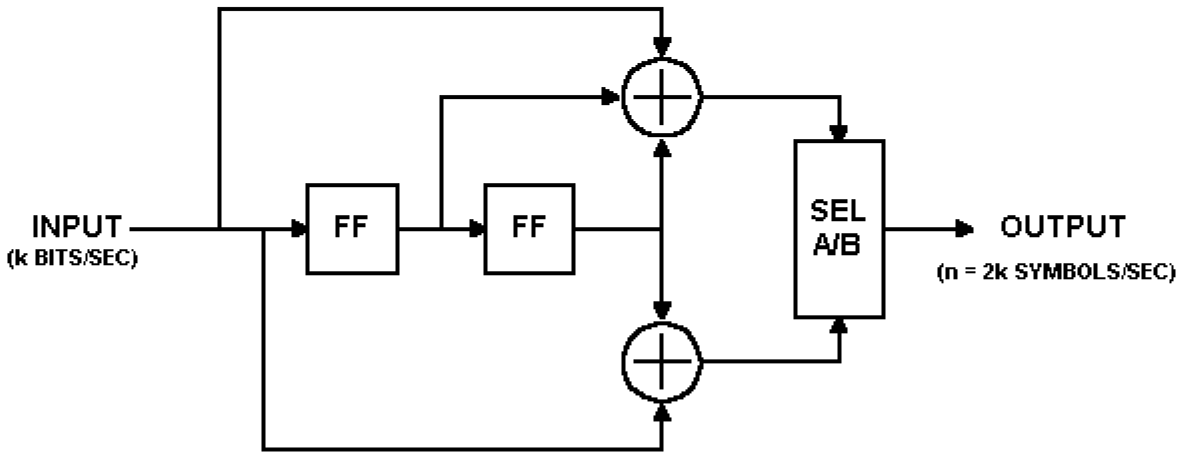
K number of input symbols used by encoder to compute each output symbol (decoding time exponentially dependent on K)

Convolution Encoder



$k = 15, n = 30, r = \frac{1}{2}, K = 3, m = 2$

Encoding Example



Both flip flops set to 0 initially.

Input: 010111001010001

Output: 00 11 10 00 01 10 01 11 11 10 00 10 11 00 11

Flush encoder by clocking m = 2 times with 0 inputs.

Viterbi Decoding Applications

- decoding trellis-coded modulation in modems
- most common FEC technique used in space communications ($r = \frac{1}{2}$, $K = 7$)
- usually implemented as serial concatenated block and convolutional coding – first Reed-Solomon, then convolutional
- Turbo codes are a new parallel-concatenated convolutional coding technique

State Transition and Output Tables

	Next State, if	
Current State	**Input = 0:**	**Input = 1:**
00	00	10
01	00	10
10	01	11
11	01	11

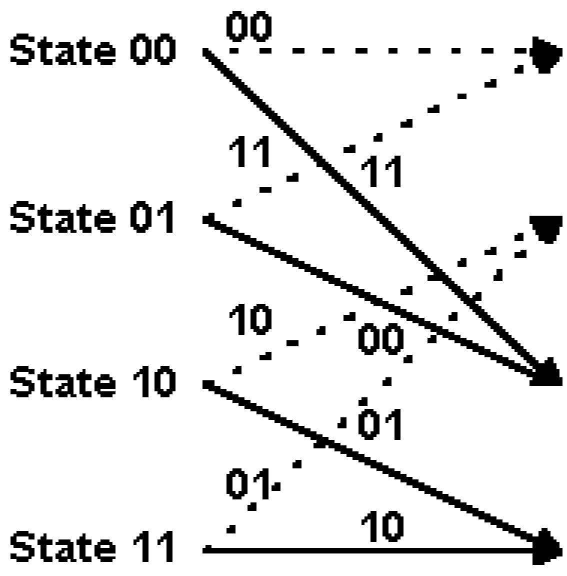
	Output Symbols, if	
Current State	**Input = 0:**	**Input = 1:**
00	00	11
01	11	00
10	10	01
11	01	10

State transition table

Output table

2^m rows

State Transitions

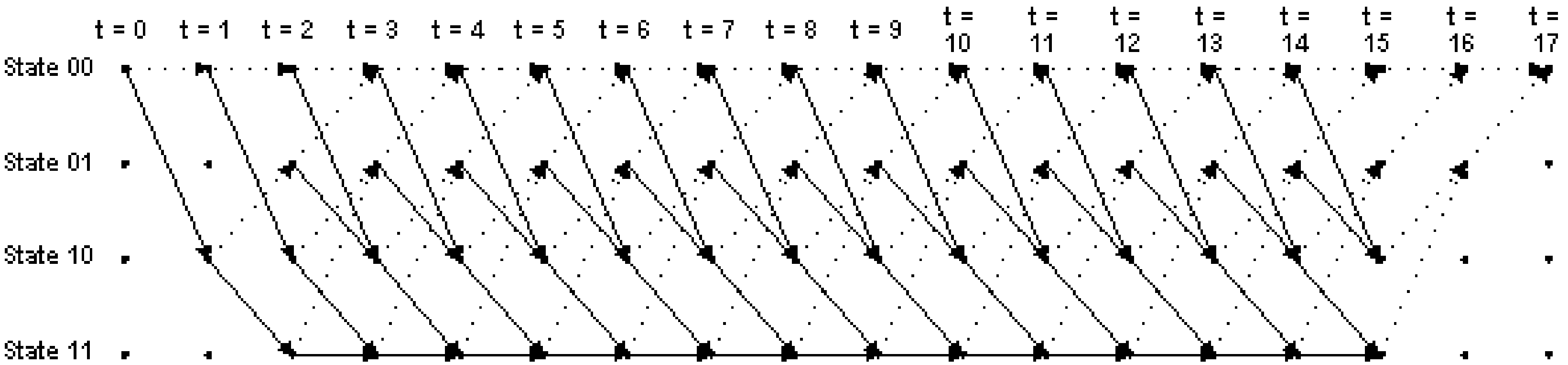


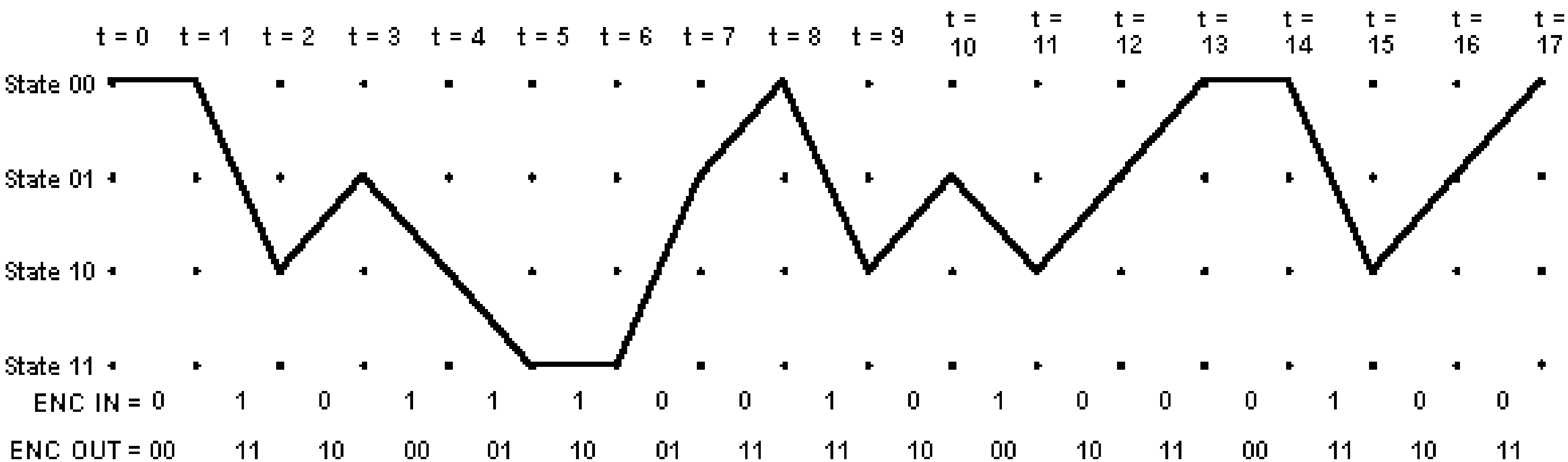
input symbol is 1

input symbol is 0

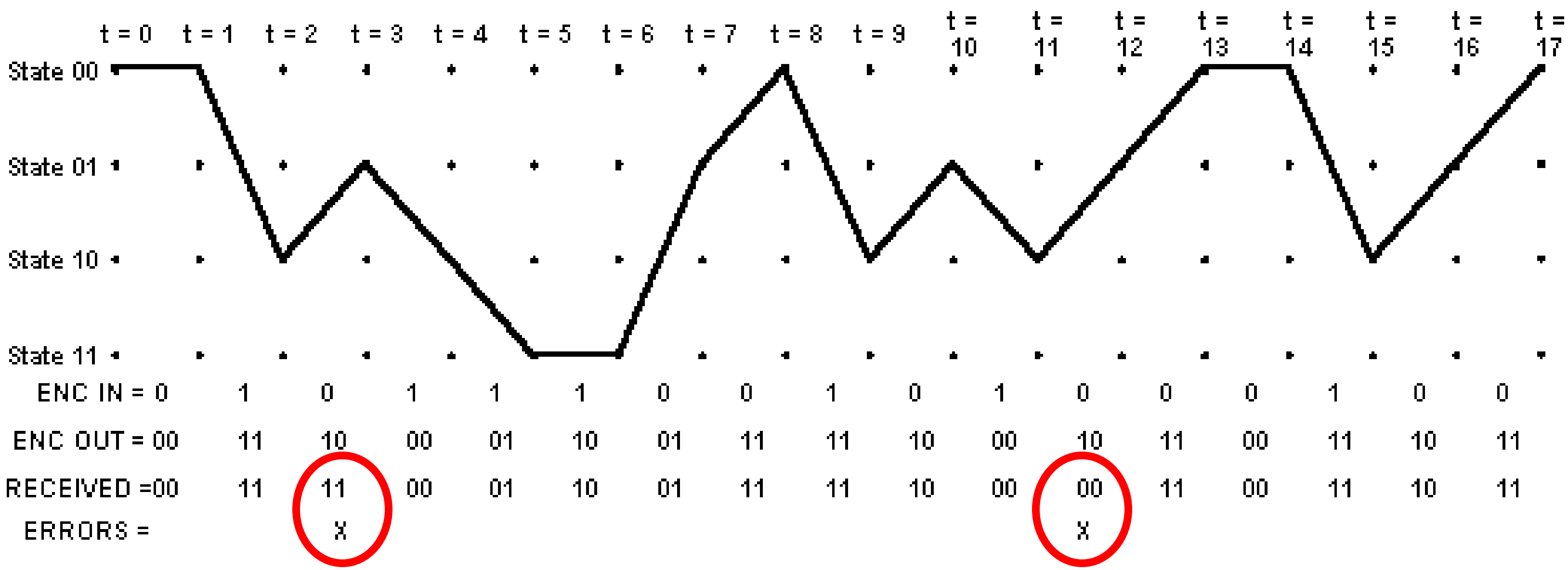
arcs labeled with output symbols

Trellis

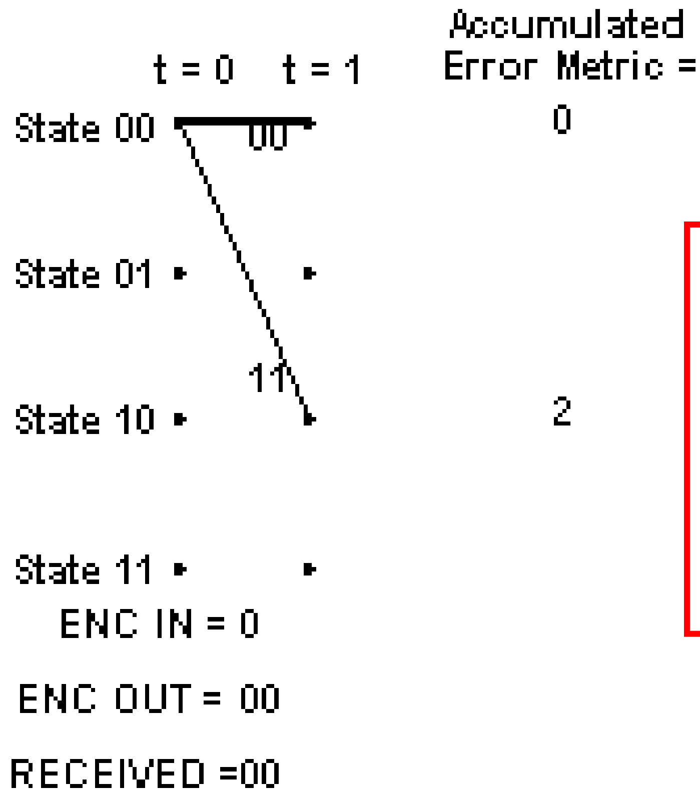

t = 0 t = 1 t = 2 t = 3 t = 4 t = 5 t = 6 t = 7 t = 8 t = 9 t = 10 t = 11 t = 12 t = 13 t = 14 t = 15 t = 16 t = 17
State 00
State 01
State 10
State 11


t = 0 t = 1 t = 2 t = 3 t = 4 t = 5 t = 6 t = 7 t = 8 t = 9 t = 10 t = 11 t = 12 t = 13 t = 14 t = 15 t = 16 t = 17
State 00
State 01
State 10
State 11
ENC IN = 0 1 0 1 1 1 0 0 1 0 1 0 0 0 1 0 0
ENC OUT = 00 11 10 00 01 10 01 11 11 10 00 10 11 00 11 10 11

Oh no! Errors in received bits!



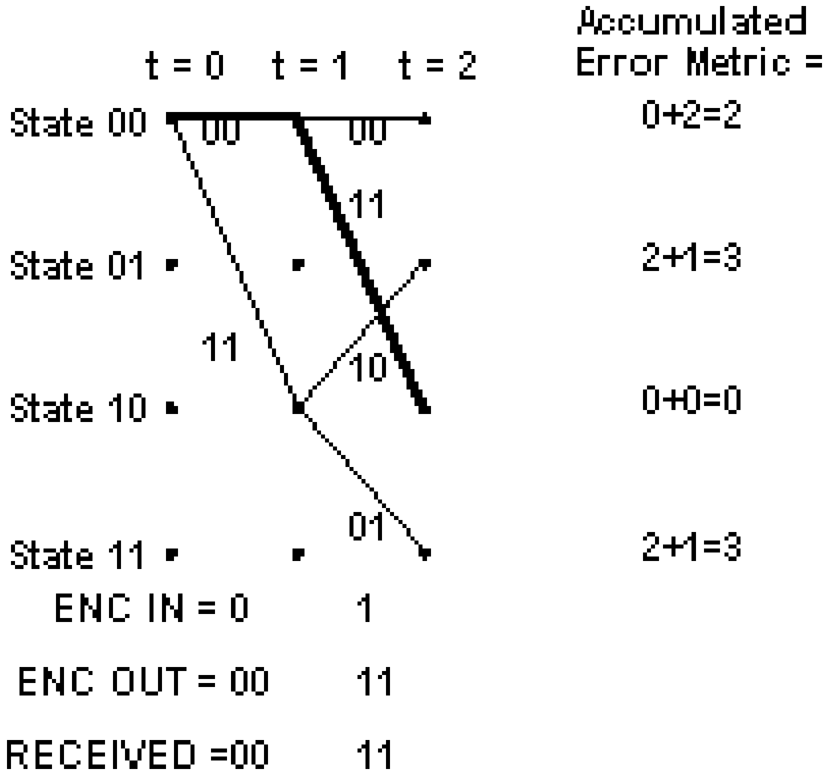
Viterbi Decoding - Accumulated Error Metric



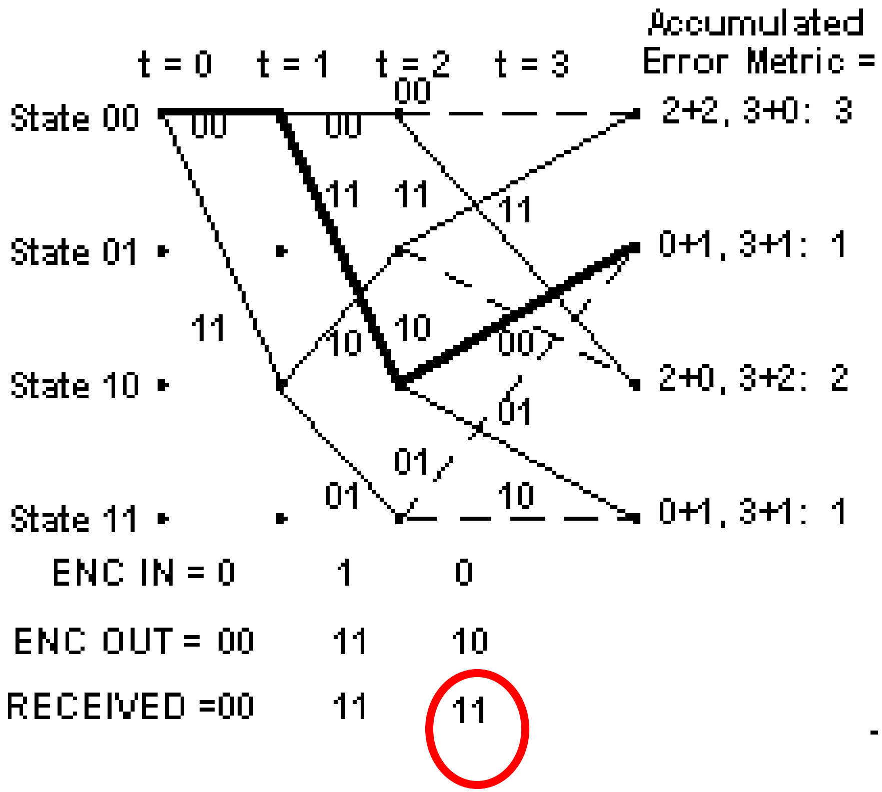
Trying to find the input sequence who corresponding output matches the received output as closely as possible.

(use Hamming distance in our example)

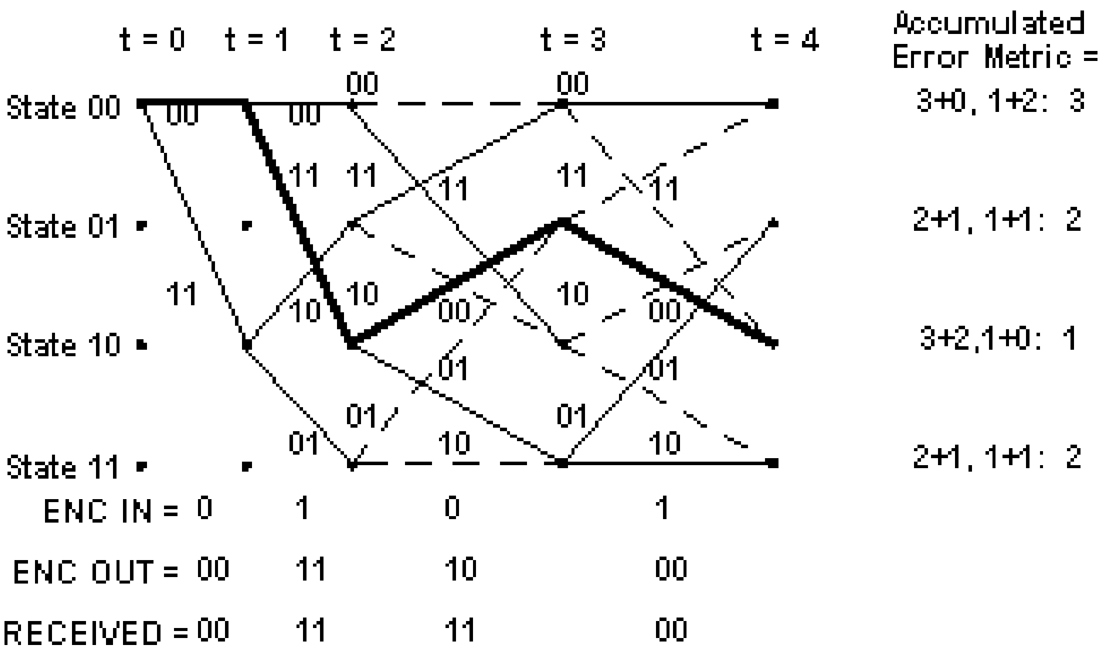
Accumulated Error Metric



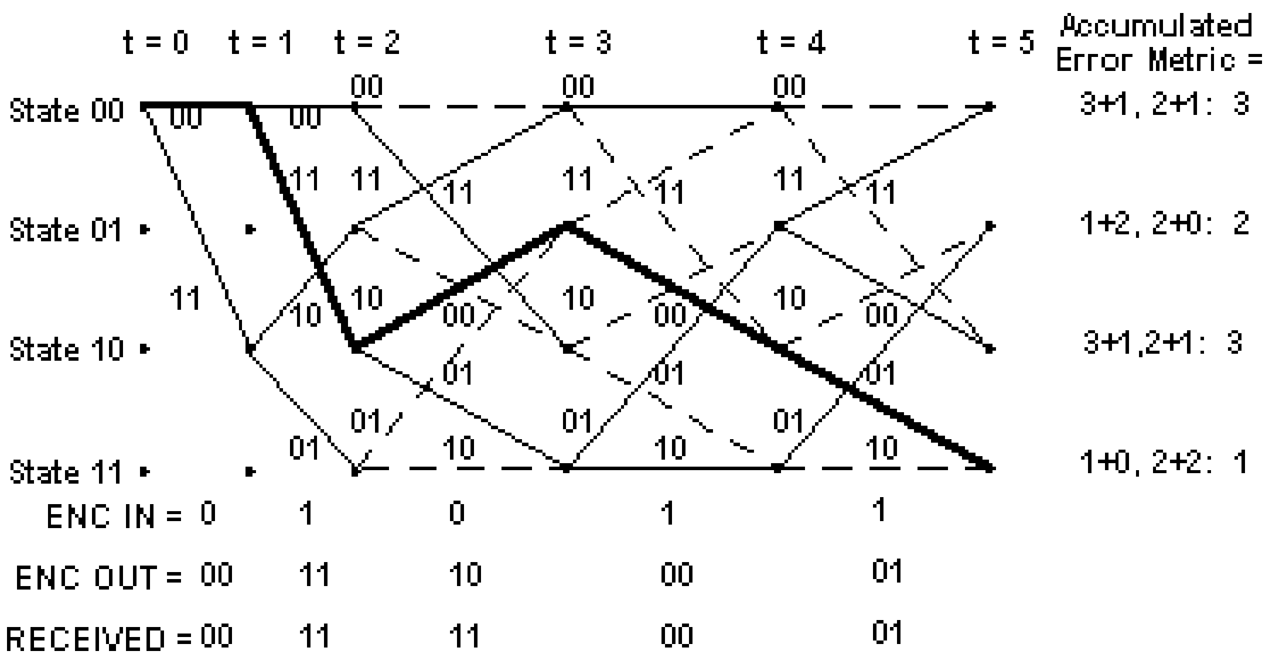
Decoder Trellis



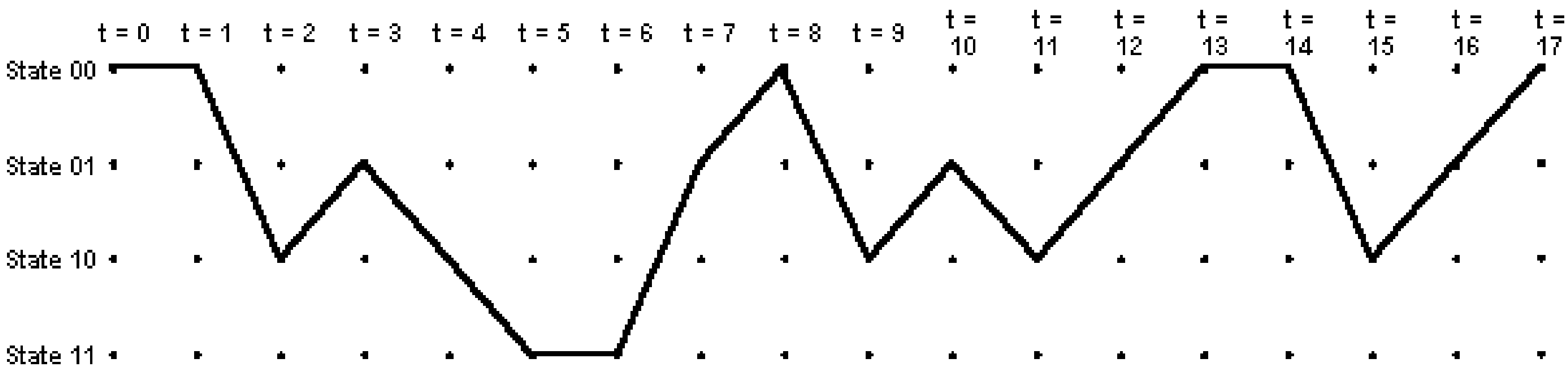
Decoder Trellis



Decoder Trellis



Final Decoder Trellis



ENC IN = 0	1	0	1	1	1	0	0	1	0	1	0	0	0	1	0	0
ENC OUT = 00	11	10	00	01	10	01	11	11	10	00	10	11	00	11	10	11
RECEIVED =00	11	11	00	01	10	01	11	11	10	00	00	11	00	11	10	11
ERRORS =		X									X					

Accumulated Error Metric over Time

t =	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
State 00_2		0	2	3	3	3	3	4	1	3	4	3	3	2	2	4	5	2
State 01_2			3	1	2	2	3	1	4	4	1	4	2	3	4	4	2	
State 10_2		2	0	2	1	3	3	4	3	1	4	1	4	3	3	2		
State 11_2			3	1	2	1	1	3	4	4	3	4	2	3	4	4		

Last two inputs known to be zero.

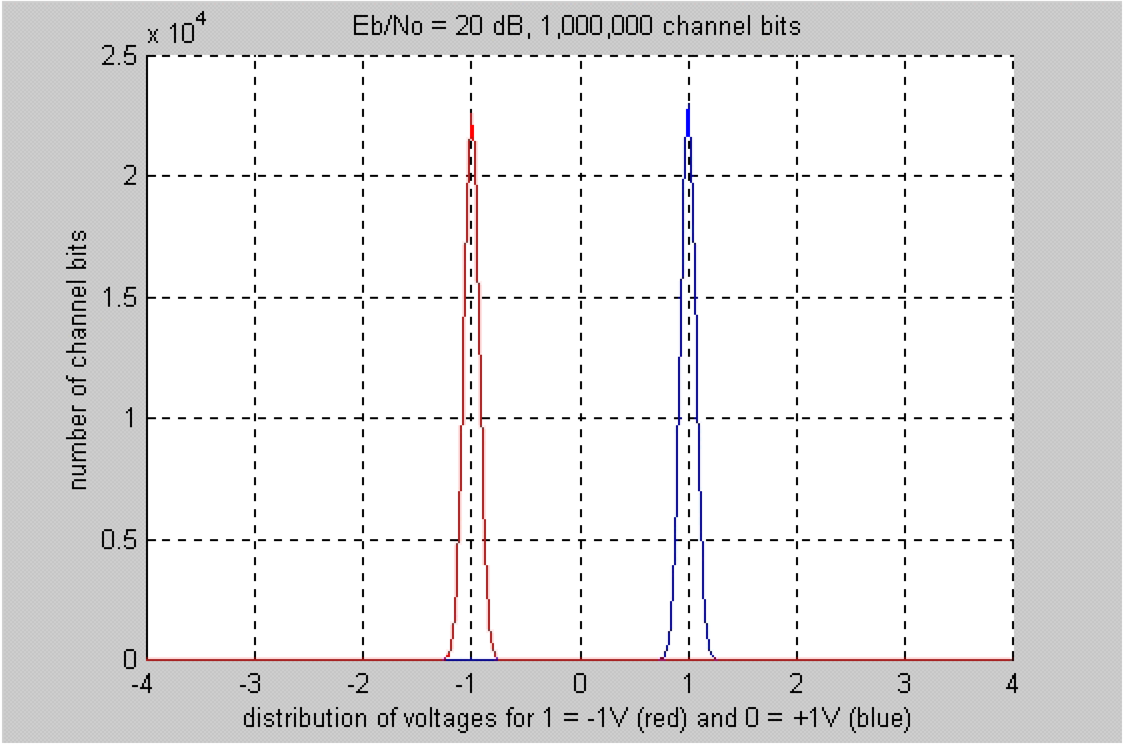
Surviving Predecessor States

t =	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
State 00_2	00	00	00	01	00	01	01	00	01	00	00	01	00	01	00	00	00	01
State 01_2	00	00	10	10	11	11	10	11	11	10	10	11	10	11	10	10	10	00
State 10_2	00	00	00	00	01	01	01	00	01	00	00	01	01	00	01	00	00	00
State 11_2	00	00	10	10	11	10	11	10	11	10	10	11	10	11	10	10	00	00

States Selected when Tracing Back

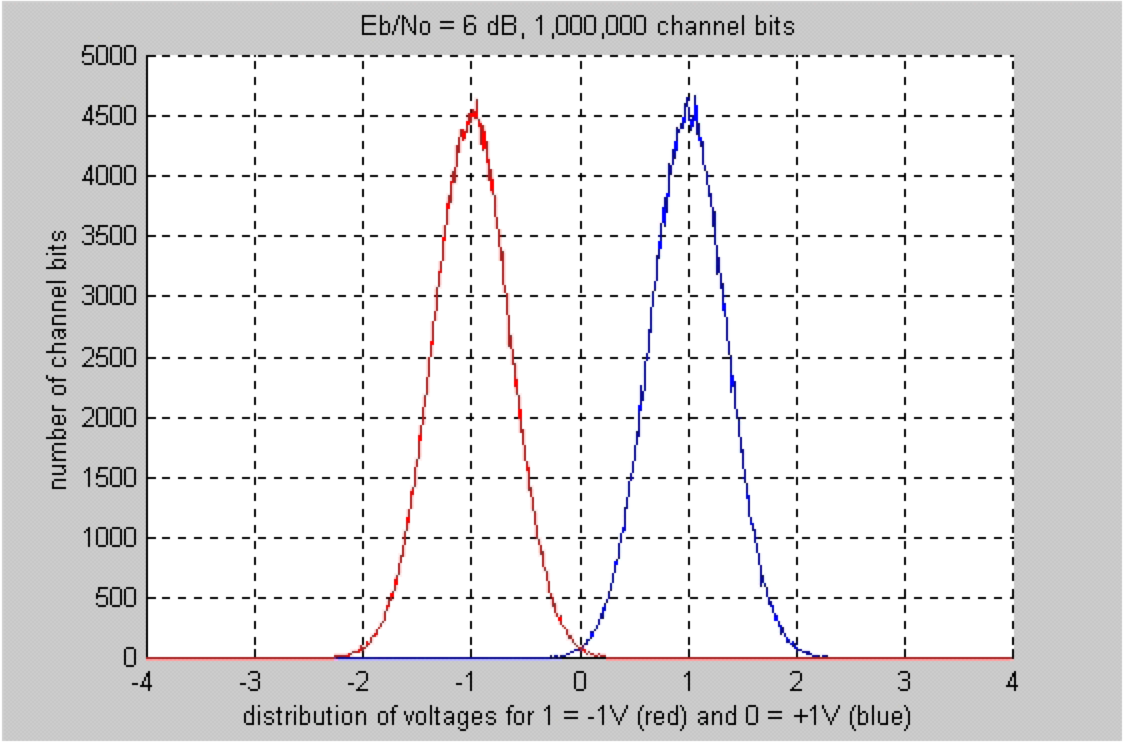
t =	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
	00	00	10	01	10	11	11	01	00	10	01	10	01	00	00	10	01	00

Coding Gain



Transmission voltages (signal to noise ratio SNR 20 dB).
No errors.

Coding Gain



Transmission voltages with Gaussian noise (SNR 6dB)
bit error rate (BER) of about 0.235%

Coding Gain

convolutional coding with Viterbi decoding can achieve a BER of less than 1×10^{-7} at the same SNR, 6 dB

$r = \frac{1}{2}$, $K = 3$

Use 5db less power to achieve 1×10^{-7} BER than without coding

Coding uses twice as much (3dB) bandwidth

Coding gain: 5dB-3dB = 2dB less energy

References (from Fleming)

Some Books about Forward Error Correction

S. Lin and D. J. Costello, *Error Control Coding*. Englewood Cliffs, NJ: Prentice Hall, 1982.

A. M. Michelson and A. H. Levesque, *Error Control Techniques for Digital Communication*. New York: John Wiley & Sons, 1985.

W. W. Peterson and E. J. Weldon, Jr., *Error Correcting Codes,* 2 nd ed. Cambridge, MA: The MIT Press, 1972.

V. Pless, *Introduction to the Theory of Error-Correcting Codes*, 3rd ed. New York: John Wiley & Sons, 1998.

C. Schlegel and L. Perez, *Trellis Coding*. Piscataway, NJ: IEEE Press, 1997

S. B. Wicker, *Error Control Systems for Digital Communication and Storage* . Englewood Cliffs, NJ: Prentice Hall, 1995.