Viterbi Decoder

EE382V SoC Design, Fall 2009

Oct 15, 2009

Discussion Session

Hyungman Park (TA)

Contents

- Purpose
- Introduction
- Simple Example (Viterbi decoding)

Purpose

- To understand the Viterbi Algorithm
- For lab3, You will need to
  - Isolate the Viterbi Decoder from DRM
  - Modify your code to make it fit into Catapult’s coding requirements
- HW3 will be an exercise on convolutional decoding

Viterbi Decoder (Intro.)

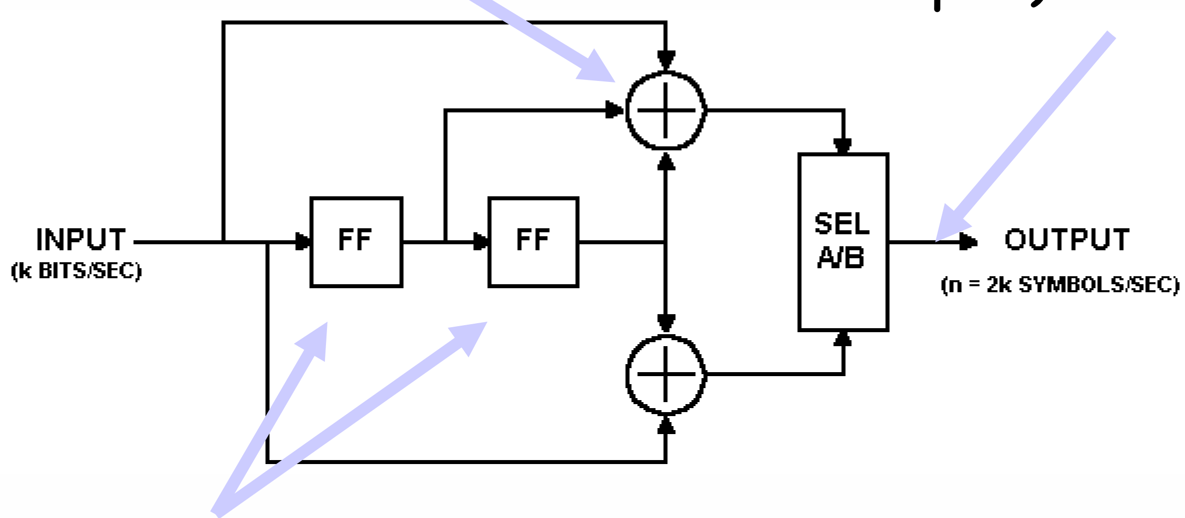
- Convolutional Coder
  - Finite memory system
  - Introduces redundancy to correct random errors occurring in the transmission of a data sequence
  - Added redundant bits are generated by modulo-two convolutions (Modulo-2 Adders, FFs, MUXs)

- Viterbi algorithm
  - Applicable to decode convolutional codes
  - Maximum-likelihood detector: finds the path through the trellis diagram with minimum path metric (accumulated error metric)
  - Widely used in digital communication

Convolution Encoder

Modulo-2 Addition

merge into one bit stream (upper input followed by lower input)

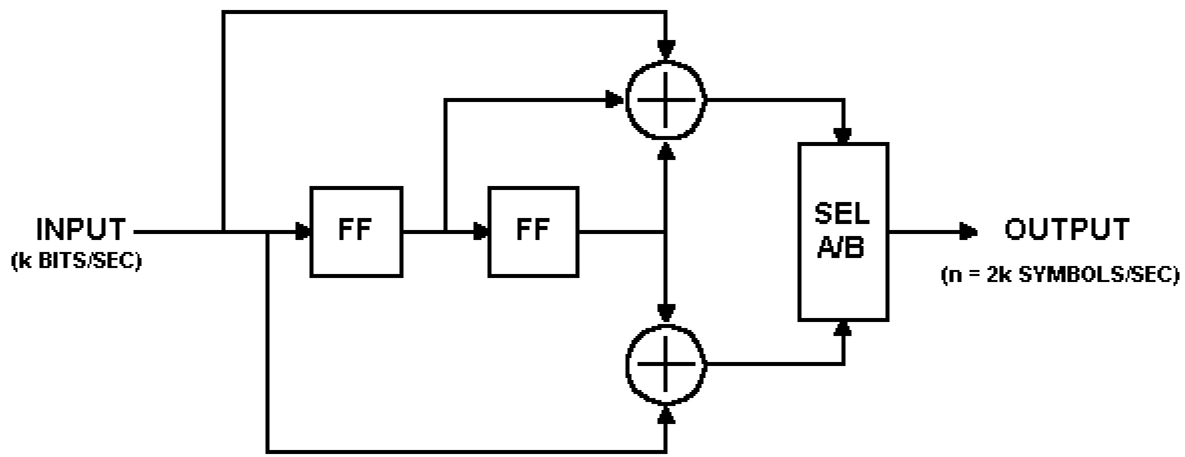


flip flop
(stores one bit)

Convolutional Encoder

- Terminology
  - k number of message symbols
  - n number of codeword symbols
  - 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)

Encoding Example: r=½, K=3, (7, 5)



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

(7, 5): code generator polynomial

Both flip-flops set to 0 initially.

Input: 01011100101000100

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

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

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

State transition table

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

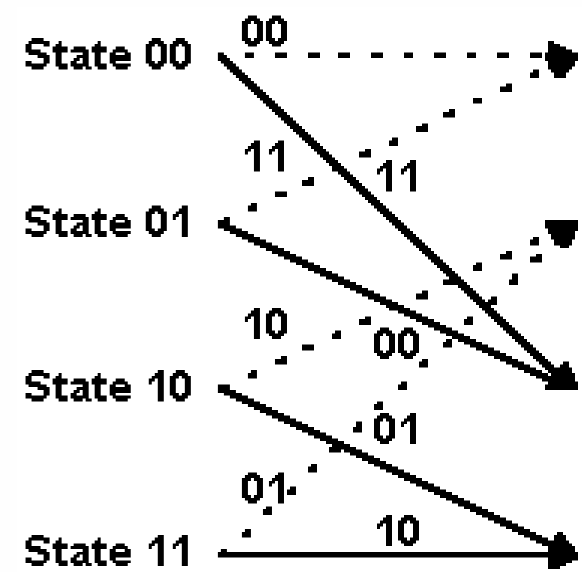
Output table

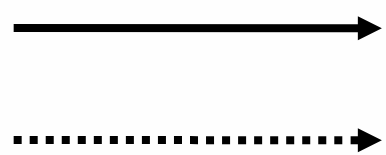
2^{K-1} rows, 2^k columns

State Transitions

	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



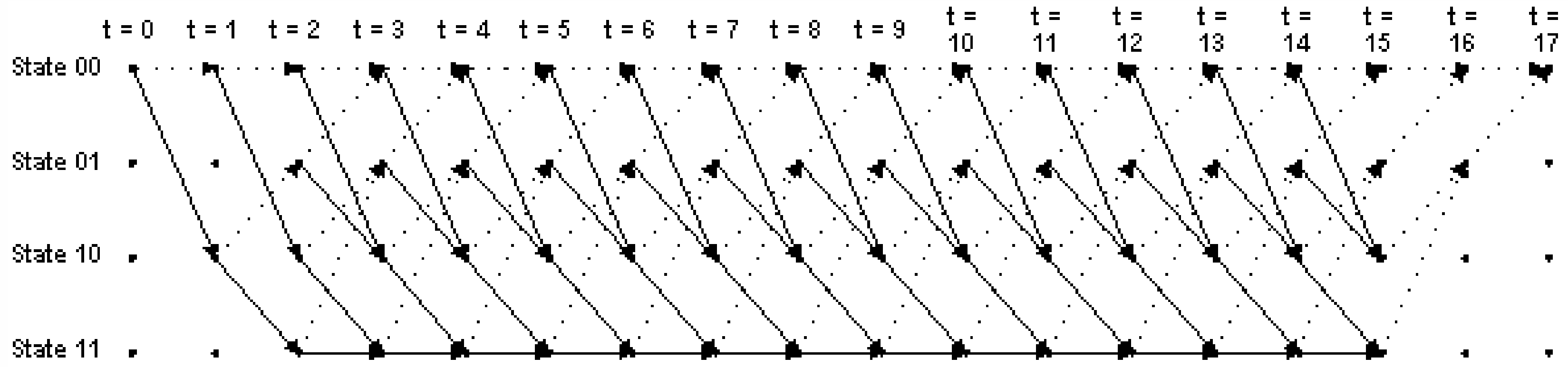


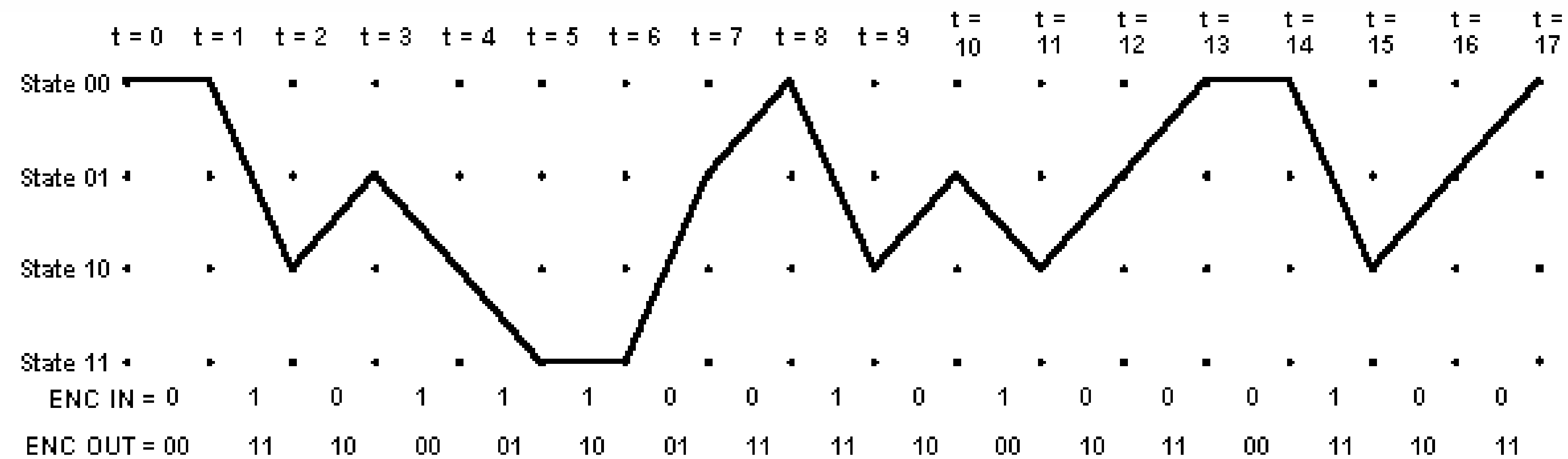
input symbol is 1

input symbol is 0

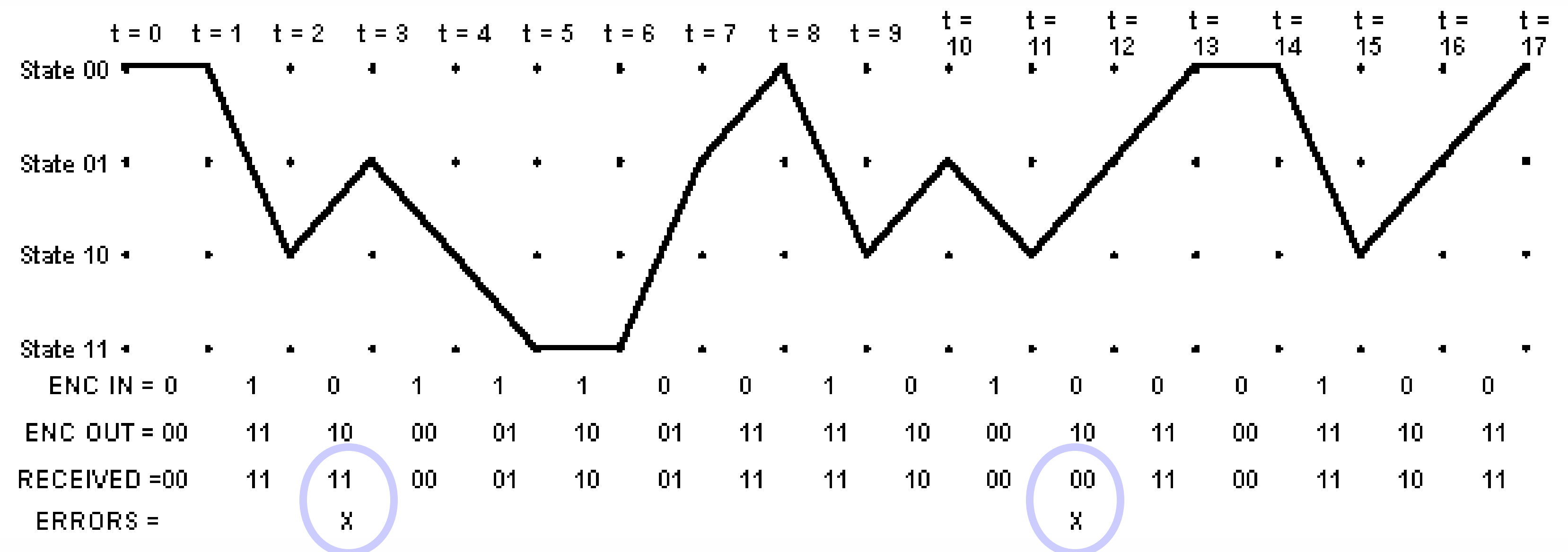
arcs labeled with output symbols

Trellis

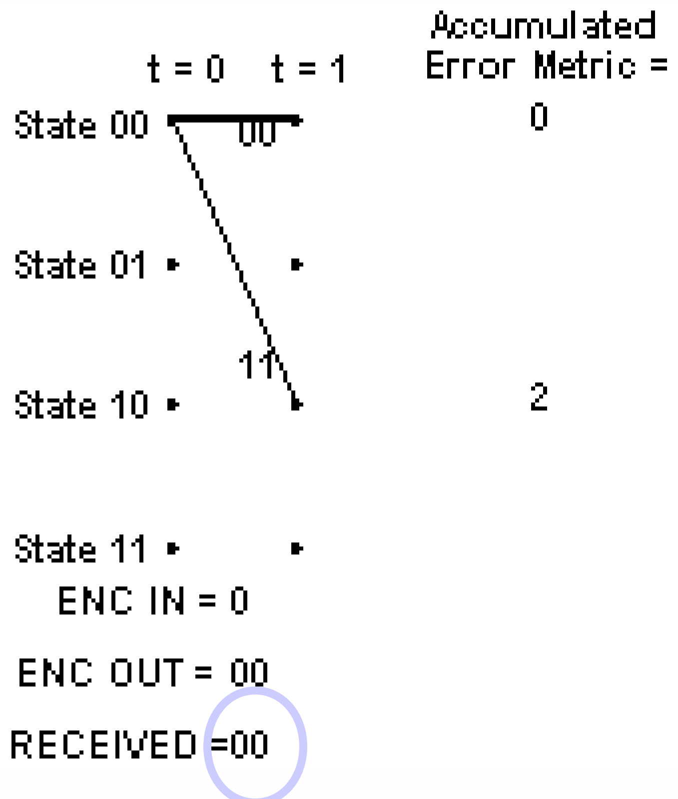

t = 0
t = 1
t = 2
t = 3
t = 4
t = 5
t = 6
t = 7
t = 8
t = 9
t = 10
t = 11
t = 12
t = 13
t = 14
t = 15
t = 16
t = 17
State 00
State 01
State 10
State 11


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

Errors in Received Bits



Accumlated Error Metric

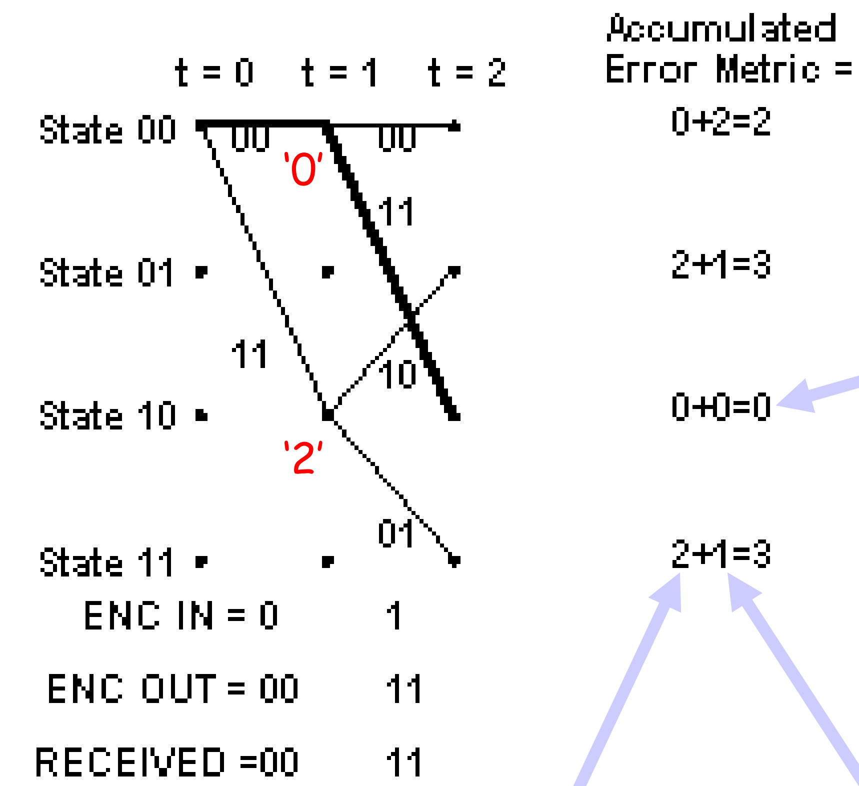


- Hamming distance: # of positions at which the corresponding symbols (the received symbol pair and the possible symbol pair) are different.

- Branch Metric: Use Hamming distance in our example

- Current AEM = Previous AEM + Current BM

Decoder Trellis (t=2)



Select one state having the smallest AEM and save

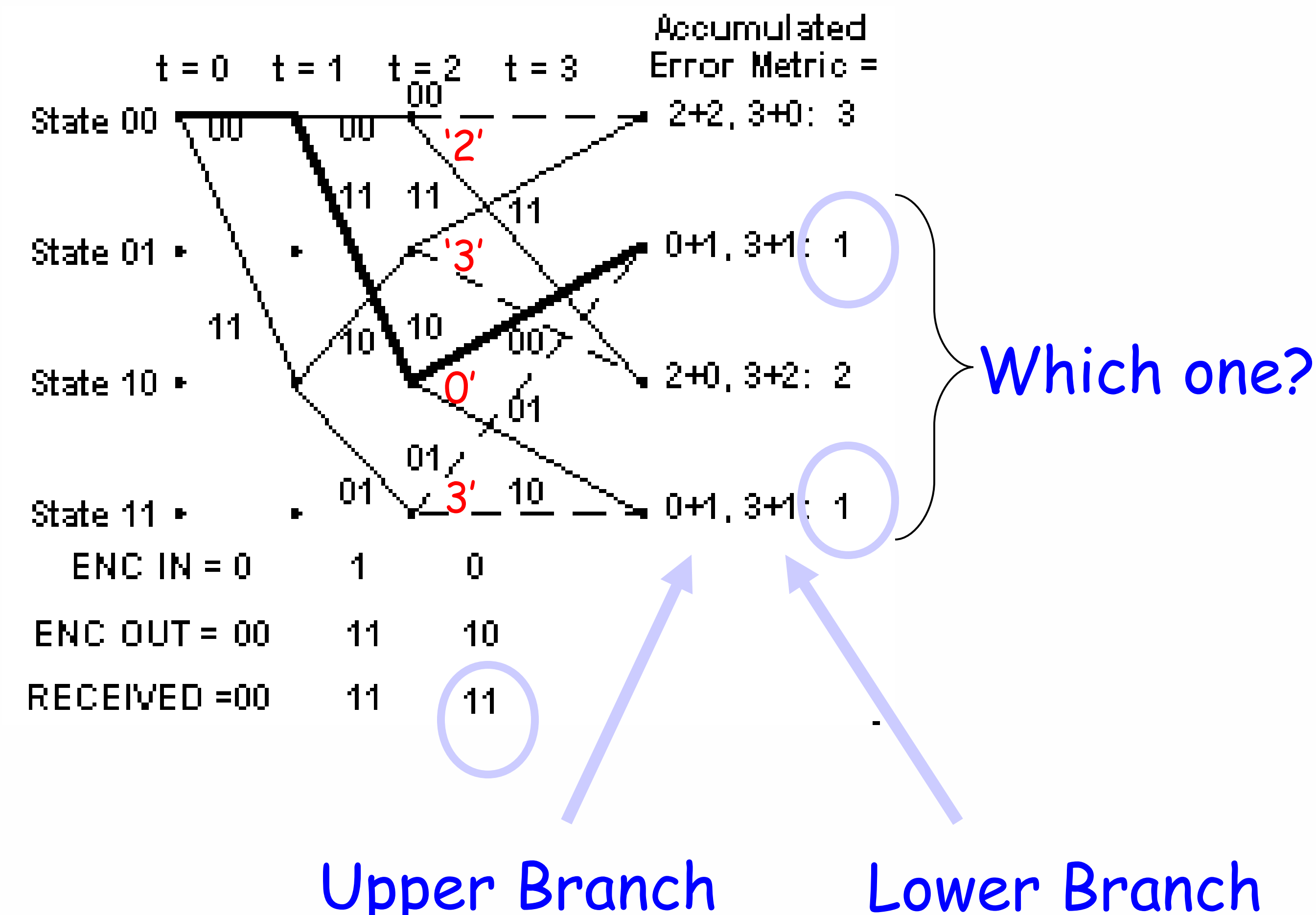
- State History
- AEMs for each state

Previous AEM

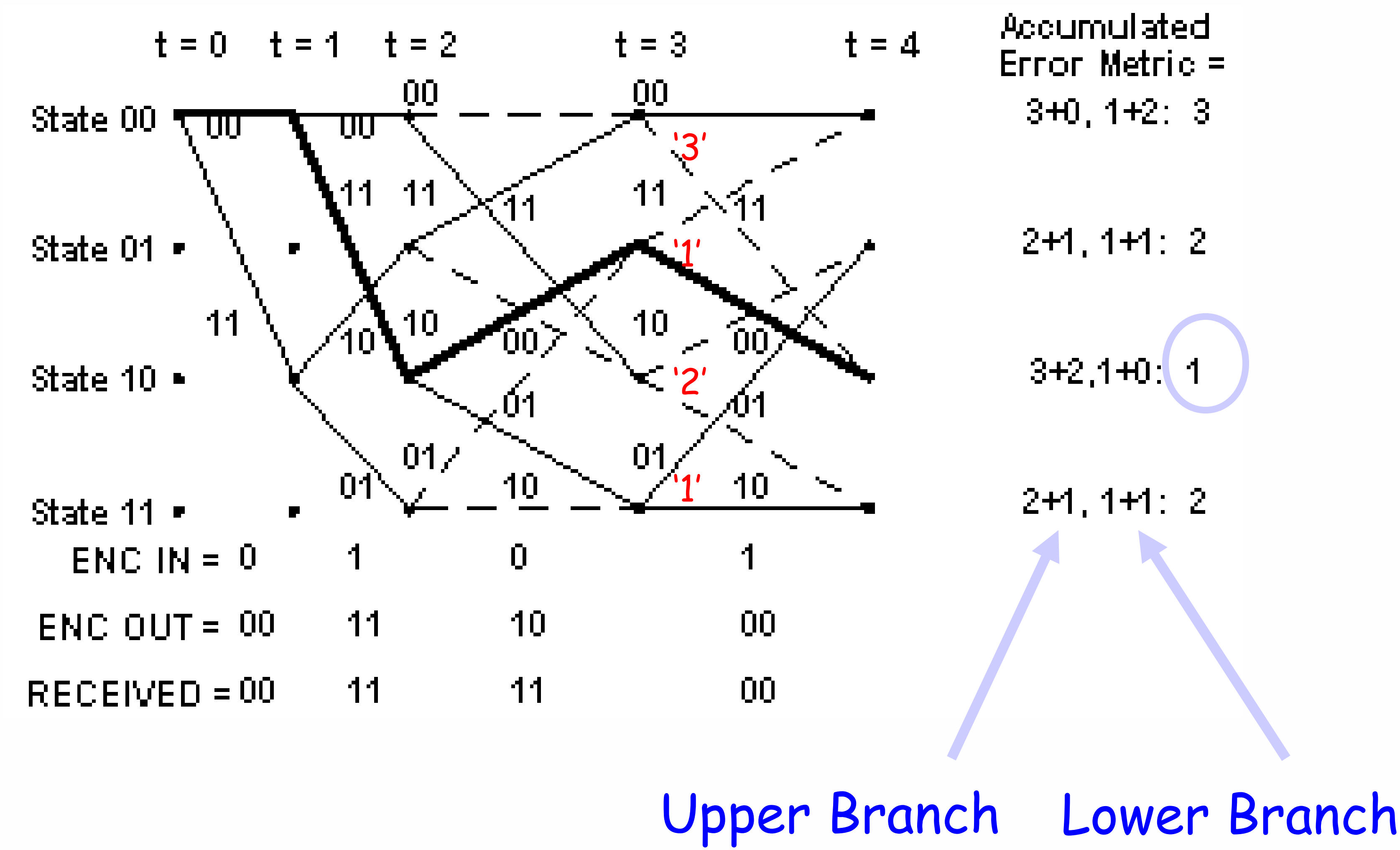
Current BM

Decoder Trellis (t=3)

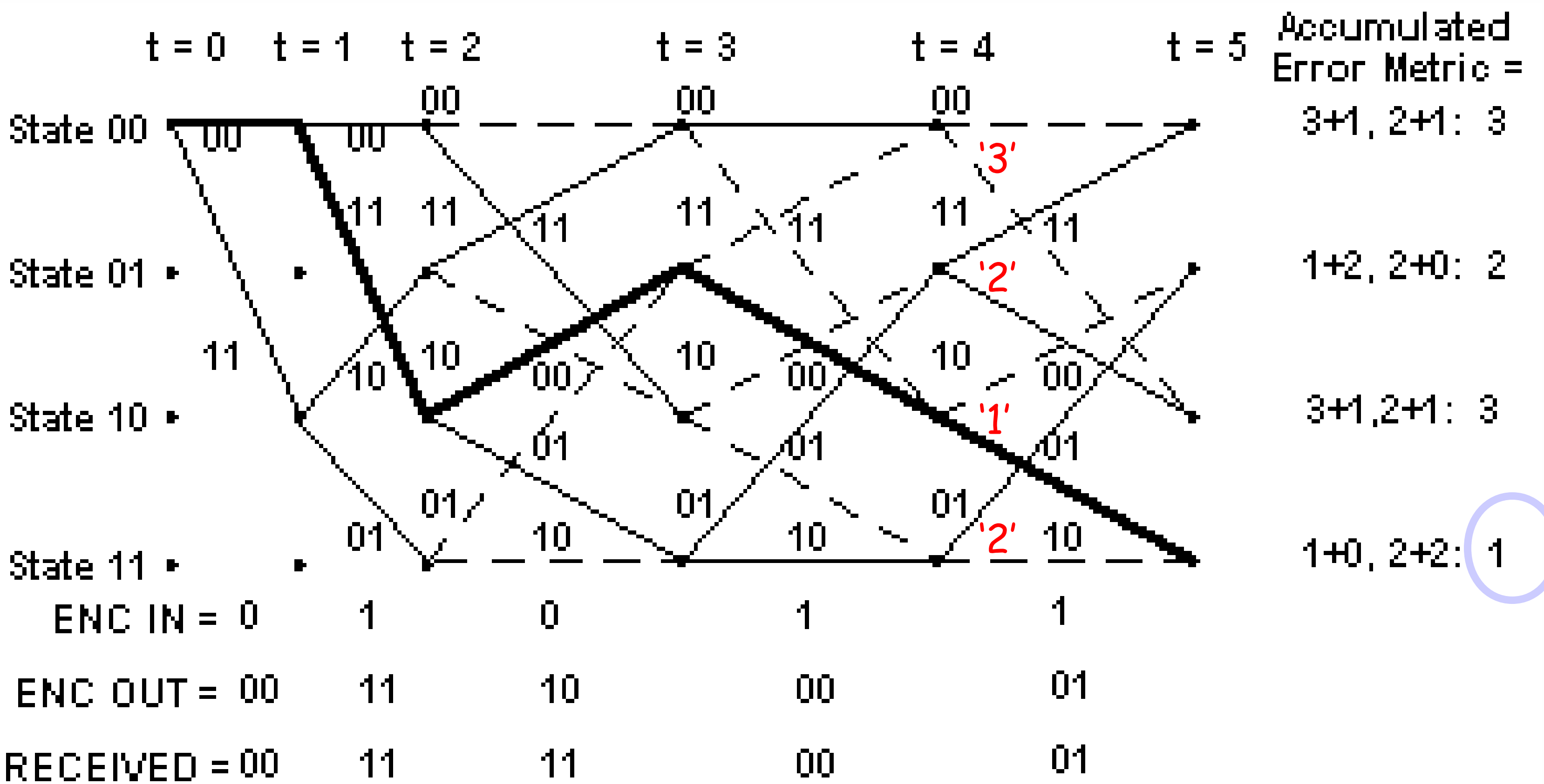
From t=3, we now have two different ways of state transition associated with each state



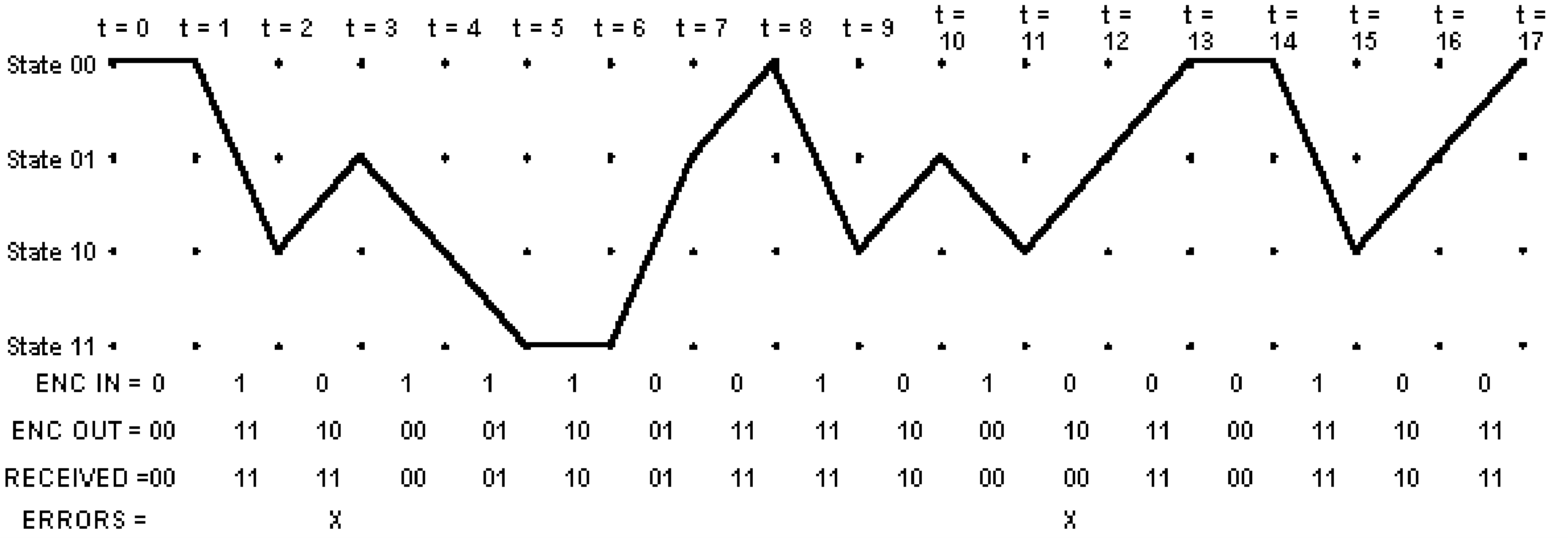
Decoder Trellis (t=4)



Decoder Trellis (t=5)



Final Decoder Trellis



Accumulated Error Metric Table

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		

Trace Back

- State History Table (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	0	0	0	1	0	1	1	0	1	0	0	1	0	1	0	0	0	1
State 01_2	0	0	2	2	3	3	2	3	3	2	2	3	2	3	2	2	2	0
State 10_2	0	0	0	0	1	1	1	0	1	0	0	1	1	0	1	0	0	0
State 11_2	0	0	2	2	3	2	3	2	3	2	2	3	2	3	2	2	0	0

Recreating the Original Message

- Selected States

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

- Input Table

	Input was, Given =			
	$00_2 = 0$	$01_2 = 1$	$10_2 = 2$	$11_2 = 3$
$00_2 = 0$	0	x	1	x
$01_2 = 1$	0	x	1	x
$10_2 = 2$	x	0	x	1
$11_2 = 3$	x	0	x	1

- Final Decoded Bit Stream

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