

# Error Correction Code (2)

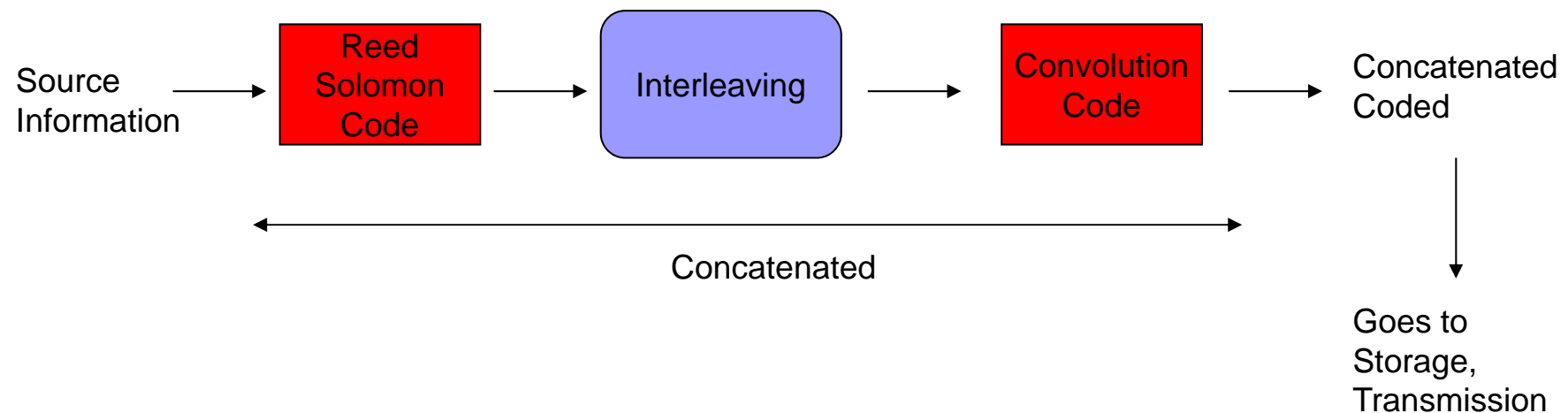
Fire Tom Wada

Professor, Information

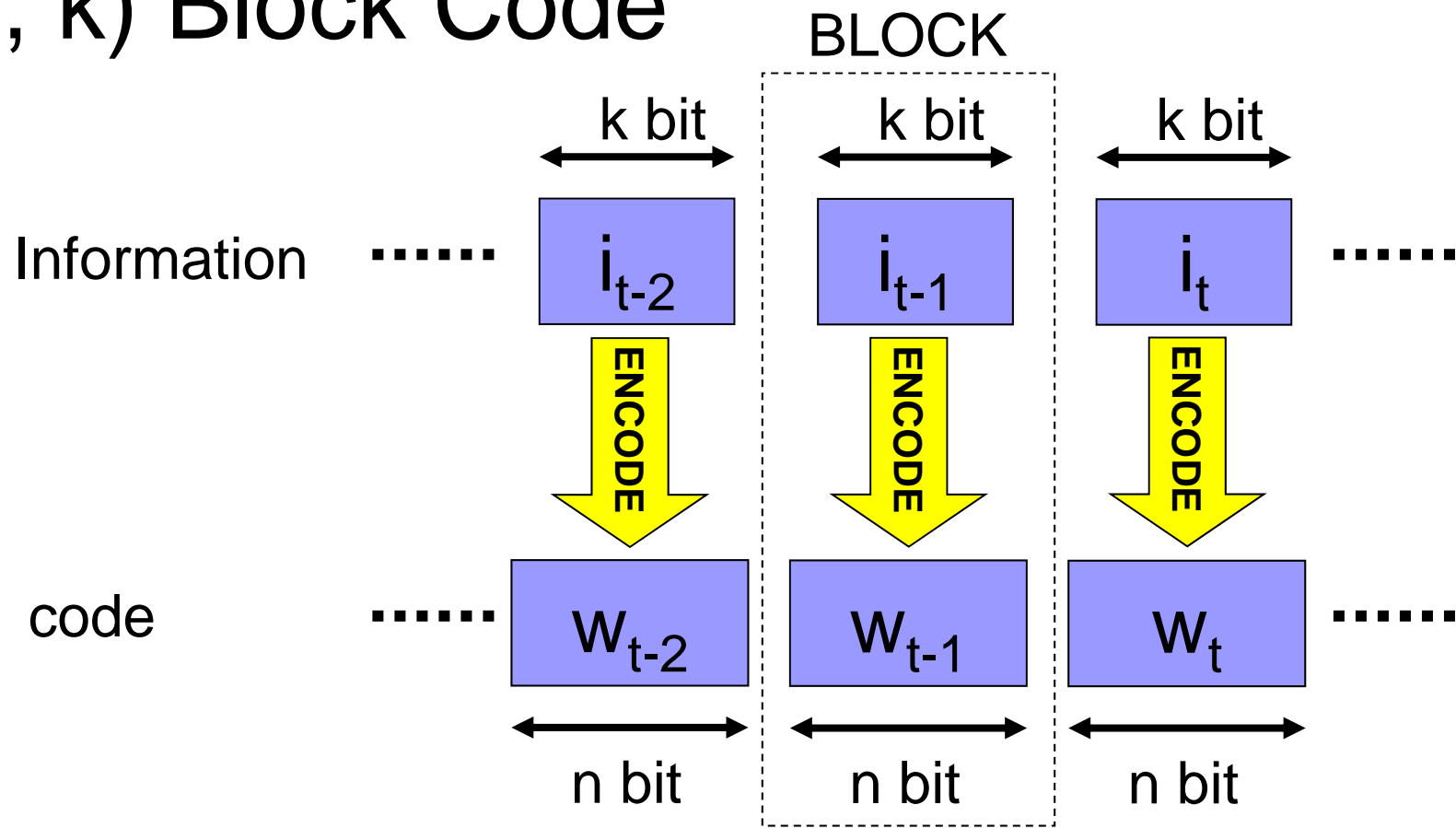
Engineering, Univ. of the Ryukyus

# Two major FEC technologies

1. Reed Solomon code (Block Code)
2. Convolutional code
3. Serially concatenated code

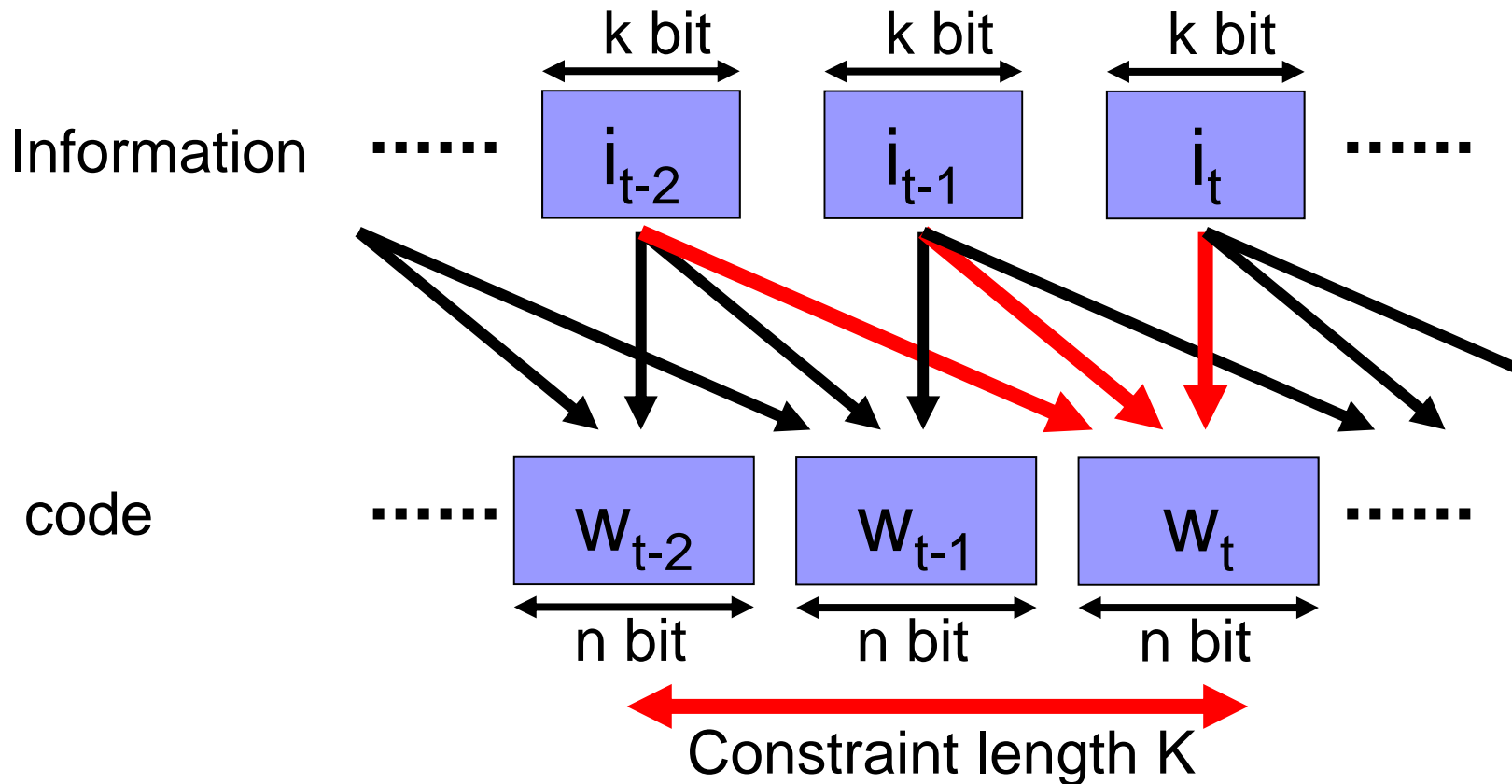


# (n, k) Block Code



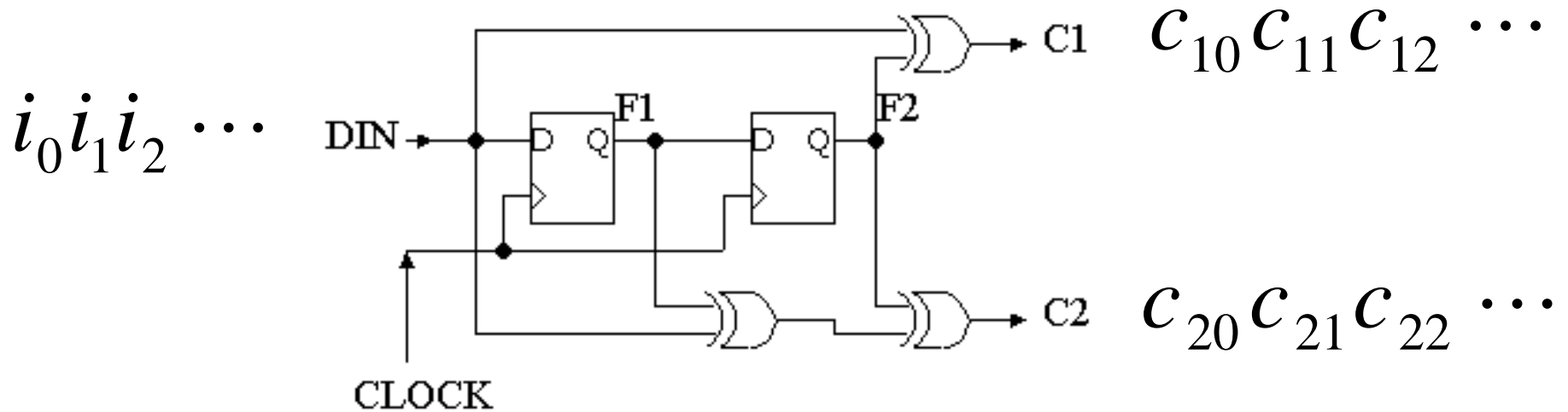
- Time  $t$  information block  $i_t$  is coded to time  $t$  code  $w_t$ .
- $k$  : information length
- $n$  : code length
- Code Rate :  $R=k/n$

# Convolutional Code



- Time  $t$  code  $w_t$  is determined by past  $K$  information.
- $K$  : Constraint length
- Code Rate :  $R=k/n$

# Simple Convolutional Coder



$$c_{1t} = i_t \oplus i_{t-2} = (1 + D^2)i_t$$

$$c_{2t} = i_t \oplus i_{t-1} \oplus i_{t-2} = (1 + D + D^2)i_t$$

- D is delay operator
- $c_{1t}, c_{2t}$  depends on not only current it bu also past  $i_{t-1}, i_{t-2}$

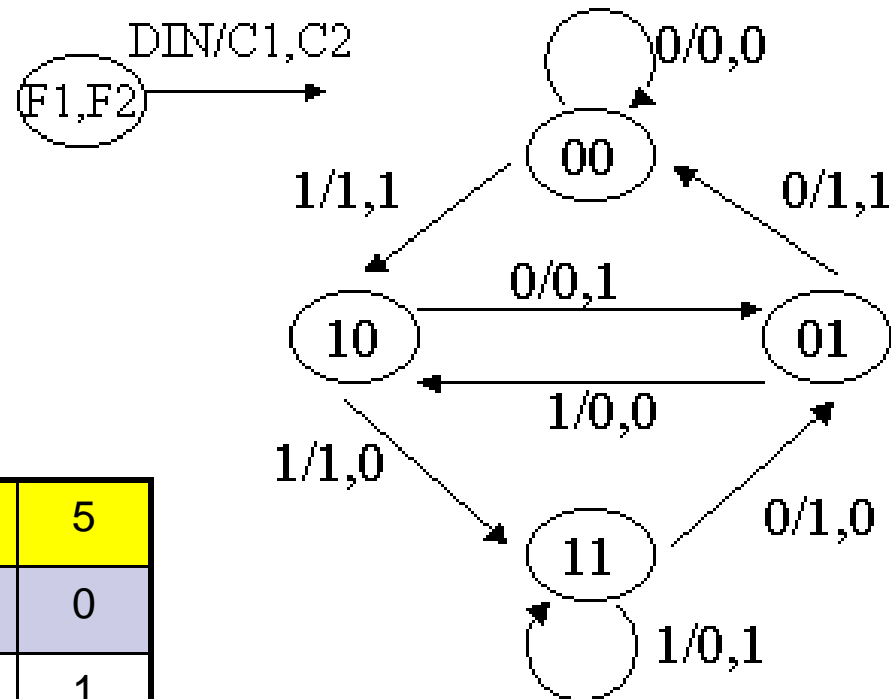
# Simple Convolutional Coder(2)

Time t	0	1	2	3	4	5
$i_t$	1	1	0	0	1	0
$F1=i_{t-1}$	0	1	1	0	0	1
$F2=i_{t-2}$	0	0	1	1	0	0
$C_{1t}$	1	1	1	1	1	0
$C_{2t}$	1	0	0	1	1	1

- If input information is “110010” then
- Output code is “11 10 10 11 11 01”.
- Constraint length  $K=3$ ,  $R=1/2$

# State Transition Diagram

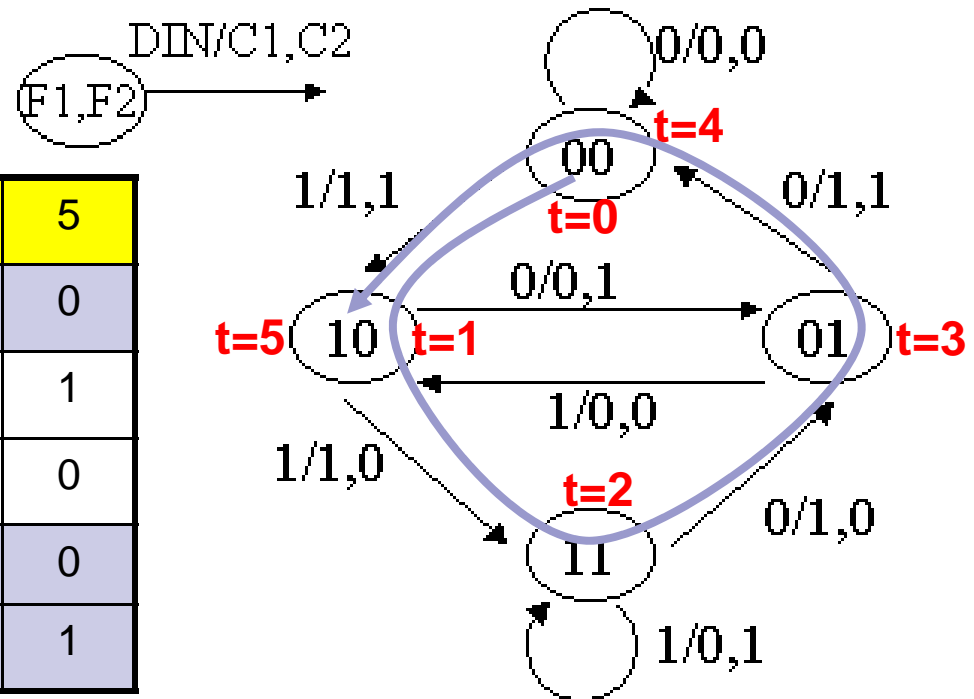
- Number of State: 4



Time t	0	1	2	3	4	5
$i_t$	1	1	0	0	1	0
$F1=i_{t-1}$	0	1	1	0	0	1
$F2=i_{t-2}$	0	0	1	1	0	0
$C_{1t}$	1	1	1	1	1	0
$C_{2t}$	1	0	0	1	1	1

# State Transition Diagram(2)

Time t	0	1	2	3	4	5
$i_t$	1	1	0	0	1	0
$F1=i_{t-1}$	0	1	1	0	0	1
$F2=i_{t-2}$	0	0	1	1	0	0
$c_{1t}$	1	1	1	1	1	0
$c_{2t}$	1	0	0	1	1	1

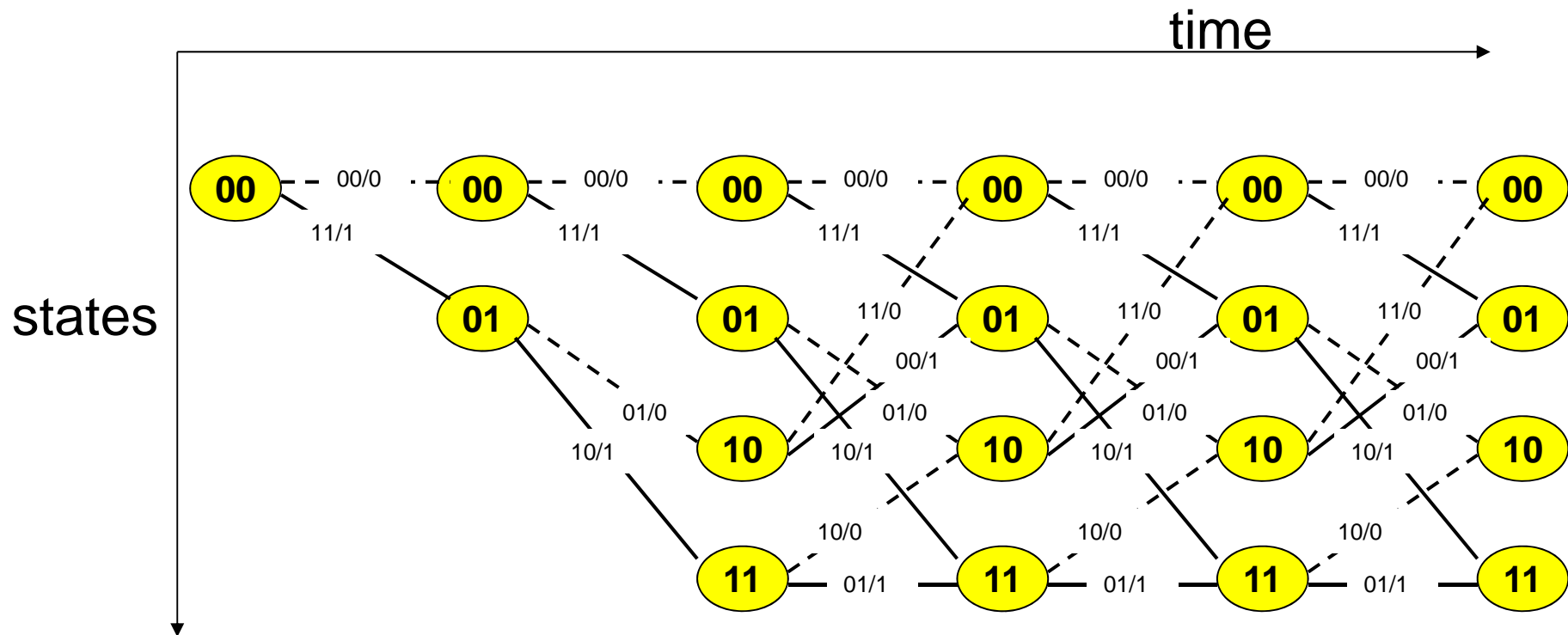


- In each time  $t$ , a traveler stays in one state.
- And move to different state cycle by cycle.



# Trellis Diagram

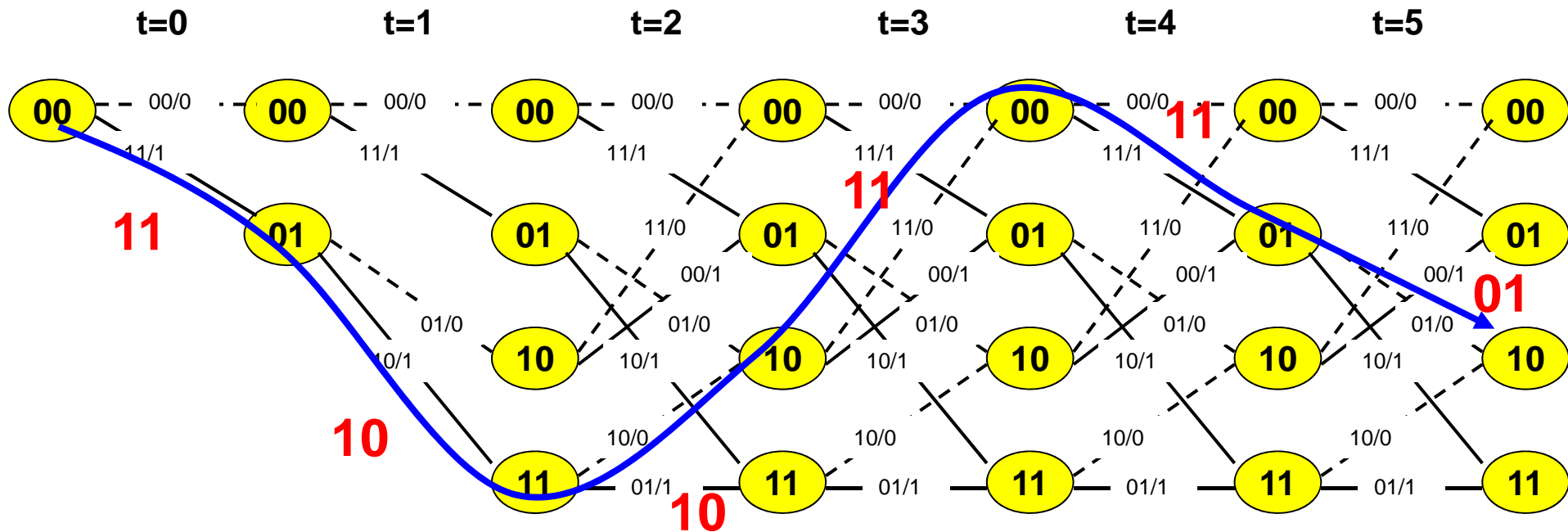
- Convert the State transition diagram to 2D diagram
  - Vertical axis : states
  - Horizontal axis : time



# Encoding in Trellis

- Input information determines a path in Trellis
- Each Branch outputs 2bit code.

Time t	0	1	2	3	4	5
$i_t$	1	1	0	0	1	0
$F1=i_{t-1}$	0	1	1	0	0	1
$F2=i_{t-2}$	0	0	1	1	0	0
$C_{1t}$	1	1	1	1	1	0
$C_{2t}$	1	0	0	1	1	1



# Punctured Convolutional Code

- The example Encoder has Code Rate  $R=1/2$
- Punctured Convolutional Code means that one or some code output is removed.
- Then Code Rate can be modified

$$R_{original} = \frac{k}{n} = \frac{ck}{cn}$$

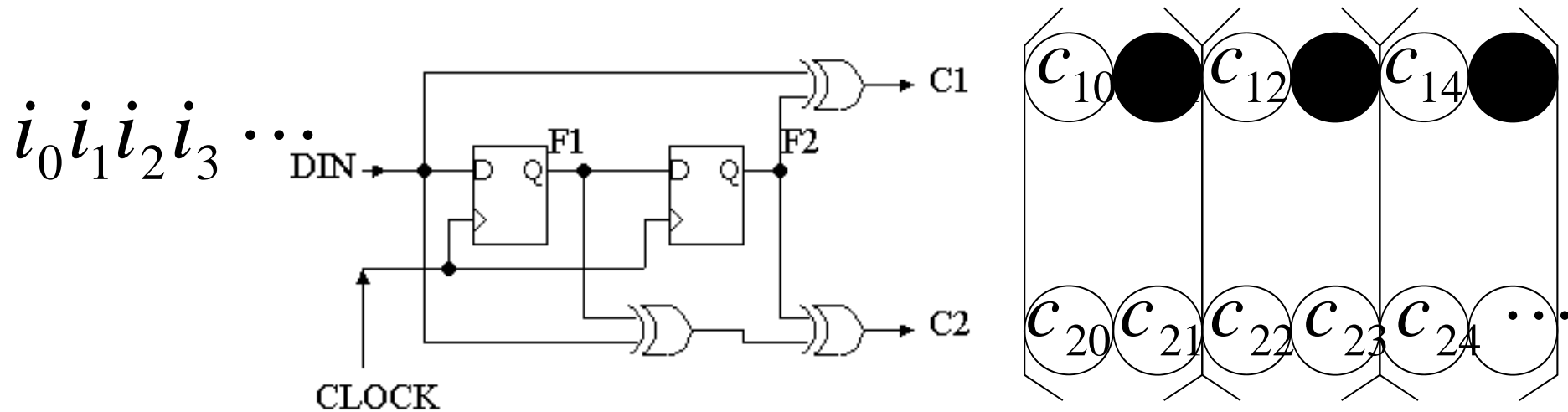
$$R_{punctured} = \frac{ck}{cn - l}$$



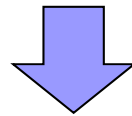
# Merit of Punctured Code

- Larger code rate is better.
- Using Punctured technology,
  1. When communication channel condition is good, weak error correction but high code rate can be chosen.
  2. When communication channel condition is bad, strong error correction but low code rate can be chosen.
  3. This capability can be supported by small circuit change.  
One coder or decoder can be used for several code rate adaptively

# R=2/3 example



$i_0 i_1 i_2 i_3$



*Non – punctured* :  $C_{10} C_{20} C_{11} C_{21} C_{12} C_{22} C_{13} C_{23}$

*Punctured* :  $C_{10} C_{20} C_{21} C_{12} C_{22} C_{23}$

# Viterbi Decode

- Viterbi Decode is one method of Maximum likelihood decoding for Convolutional code.
- Maximum likelihood decoding
  - Likelihood function is  $P(x_i|r)$ :
  - Probability of sending  $x_i$  under the condition of receiving  $r$



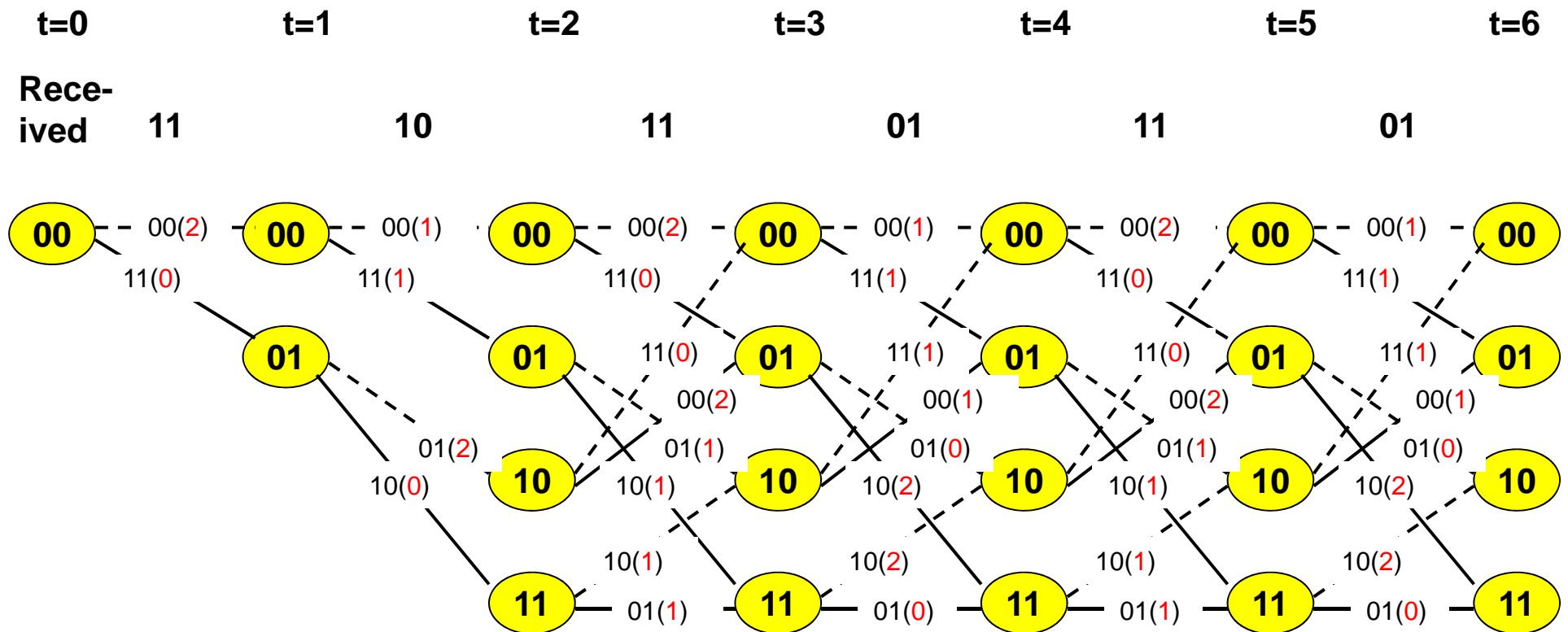


## Viterbi Decode(2)

- **Branch metric** is the Likelihood function of each branch.
  - $-\text{Ln}\{p(x_k|r_i)\}$
  - High possibility  $\rightarrow$  small value
  - Example : Hamming distance
- **Path metric** is the sum of branch metrics along the possible TRELLIS PATH.

# Viterbi Decoding Example

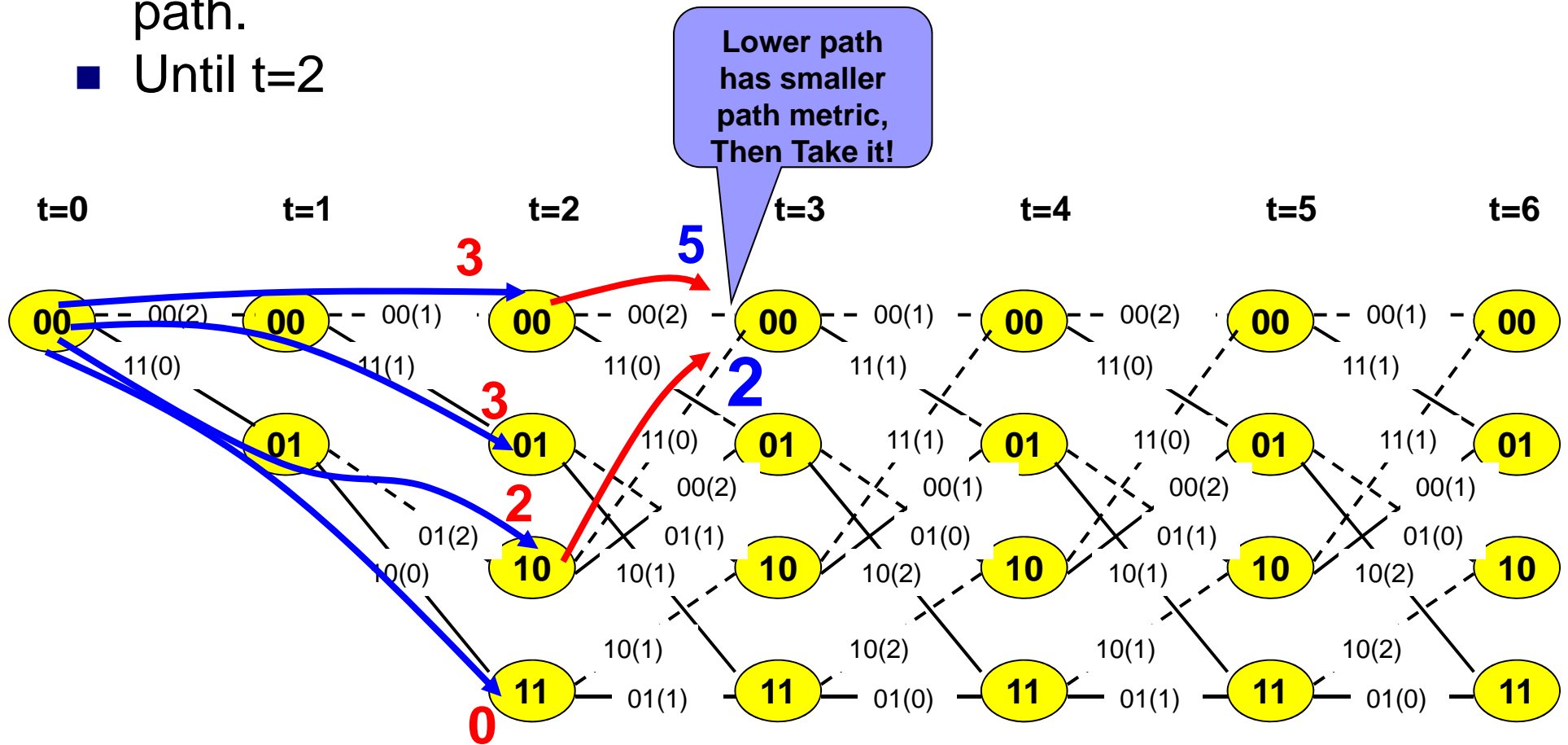
- R=1/2, Receive 11 10 11 01 11 01
- Calculate Each Branch metric (This time Hamming distance)





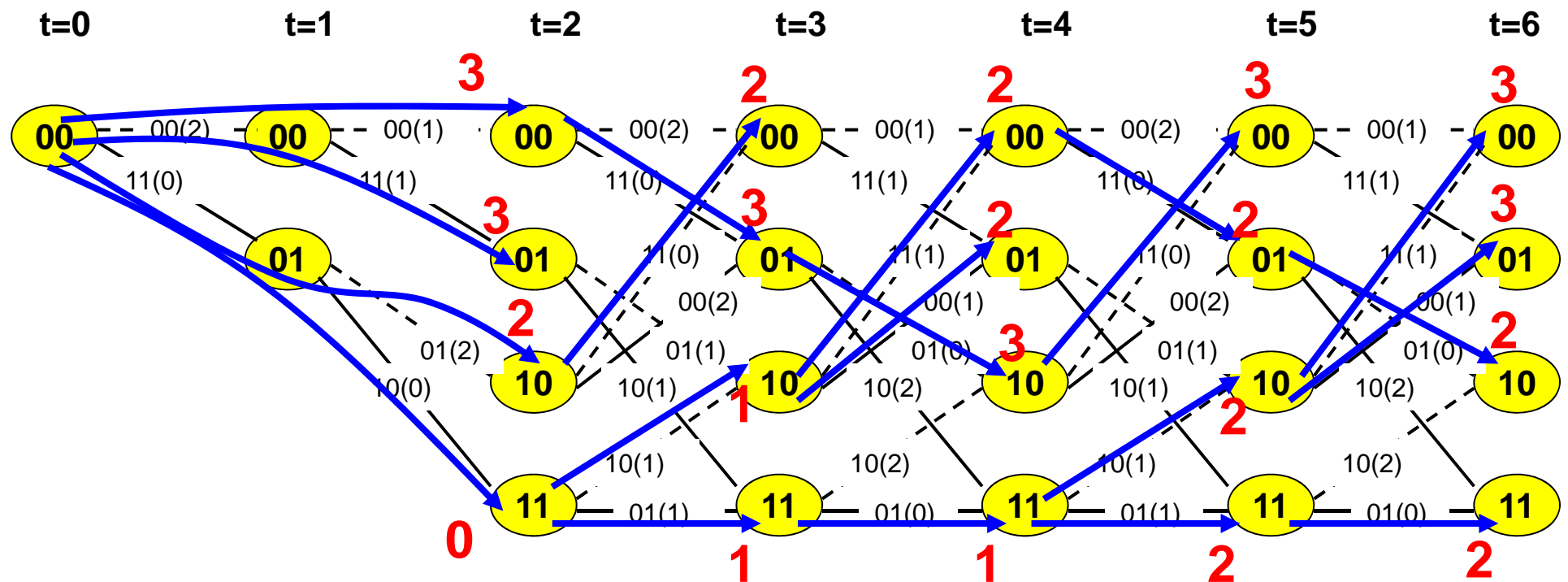
# Viterbi Decoding Example(2)

- Calculate Path metric in order to find minimum path metric path.
- Until  $t=2$



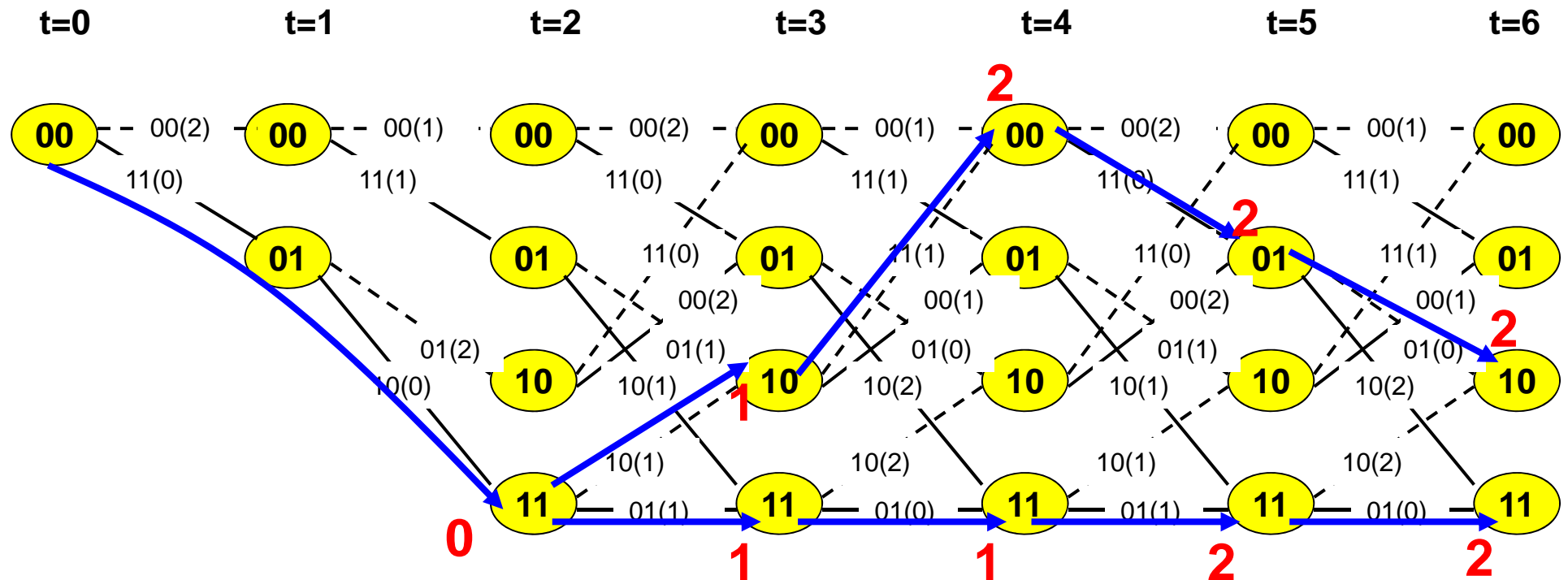
# Viterbi Decoding Example(3)

- Calculate Path metric in order to find minimum path metric path.
- Until  $t=6$



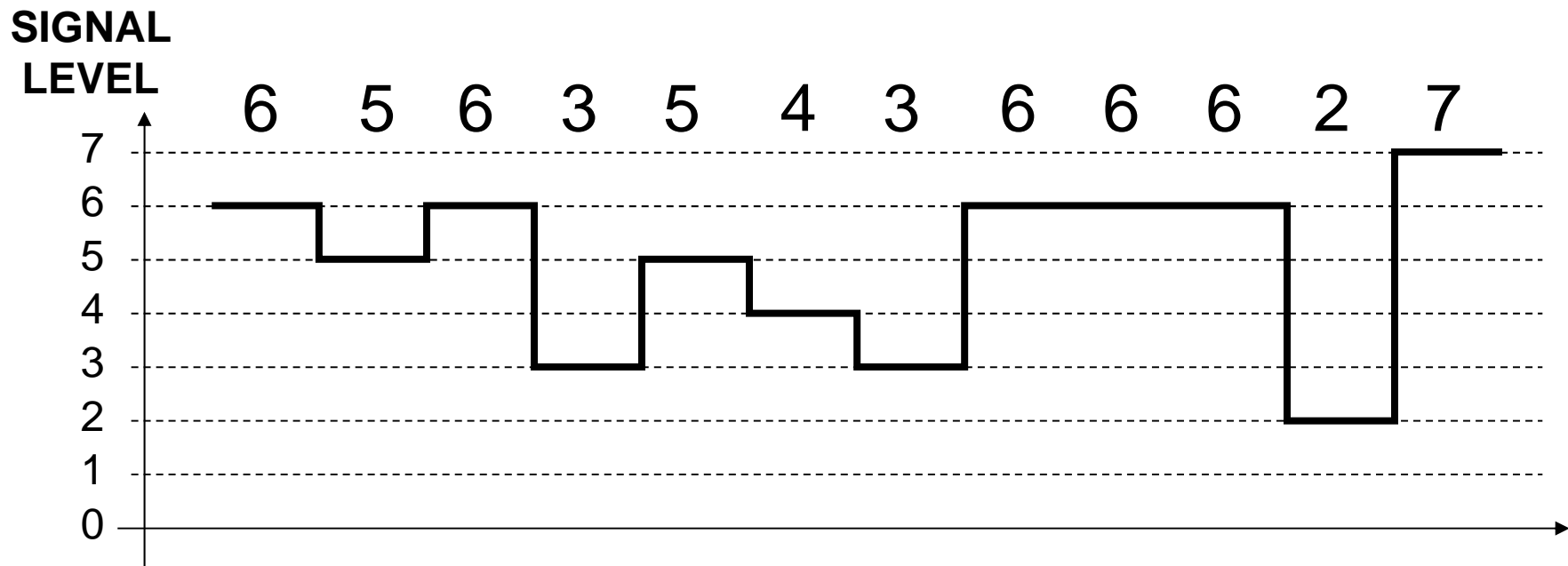
# Viterbi Decoding Example(4)

- Select Minimum Path Metric and get original information
- In this example, two minimum path
  - Upper path : 1 1 0 0 1 0
  - Lower path : 1 1 1 1 1 1
- If we increase the time, we might find ONLY ONE MINIMUM PATH.

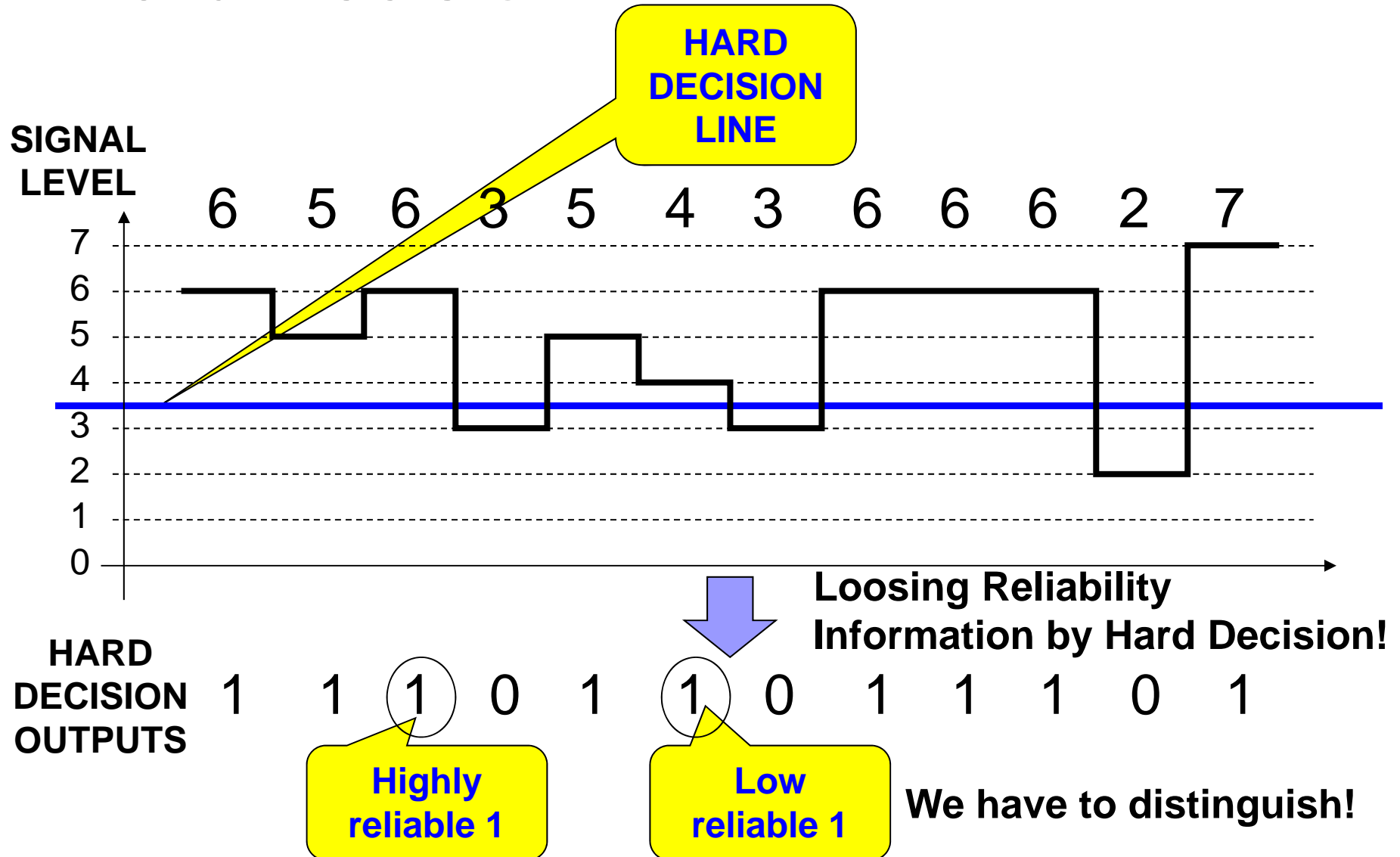


# Received signal has many level

- In the previous example, we have assumed the received sequence is
  - 11 10 11 01 11 01
- Usually, received signal is analog (Many Levels) such as



# Hard Decision



# Soft Decision

- Use soft decision metric

- One Example

LEVEL	0	1	2	3	4	5	6	7
Branch Metric for '0'	0	0	0	0	0	1	2	3
Branch Metric for '1'	3	2	1	0	0	0	0	0
Difference	-3	-2	-1	0	0	1	2	3

Reliable '0'



Reliable '1'



**HOW TO COMPUTE BRANCH METRIC**  
**LEVEL= 65**

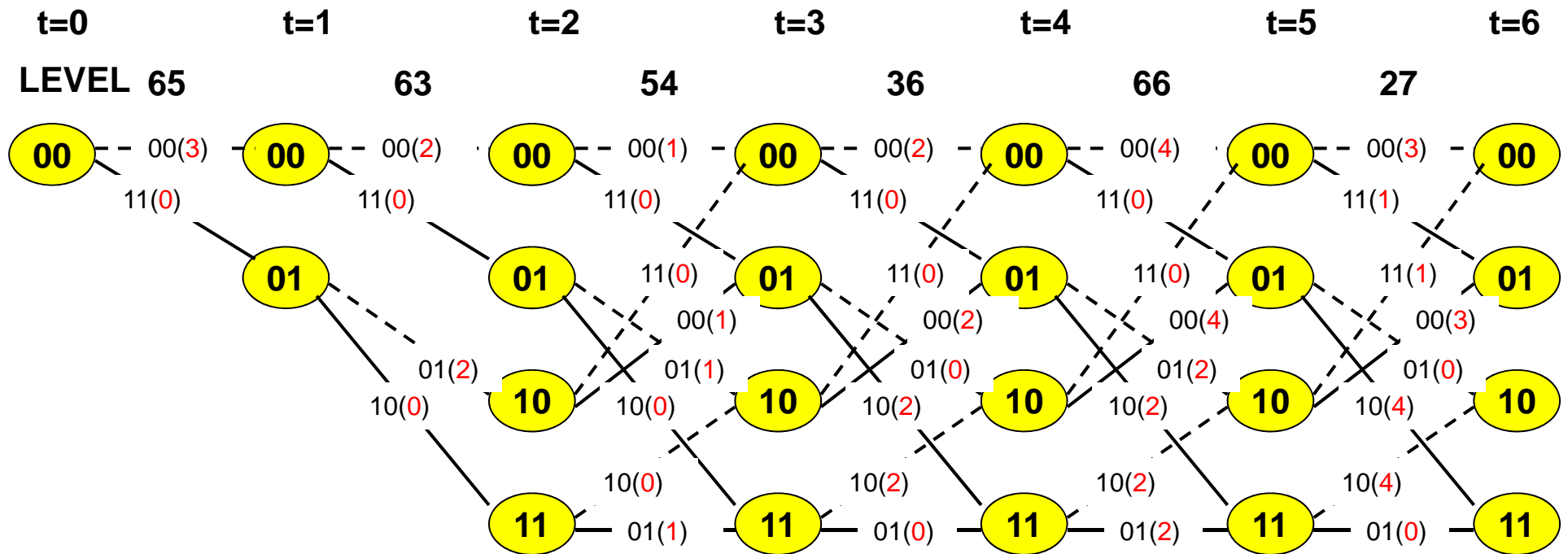
**00** - - 00 (branch metric = 1 + 2 = 3)

No effect on Viterbi decoding!

Looks like Erased or Punctured!

# Soft Decision Viterbi

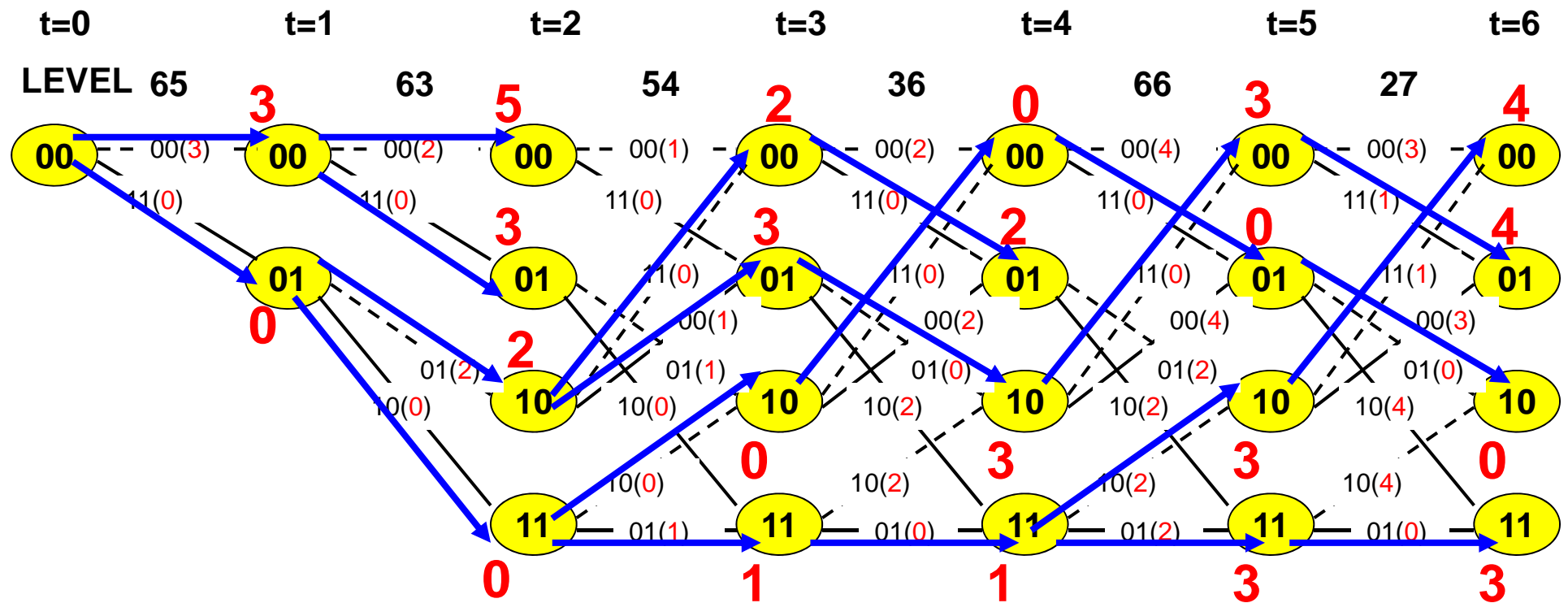
- Calculate Branch Metric based on Soft-Table



LEVEL	0	1	2	3	4	5	6	7
Branch Metric for '0'	0	0	0	0	0	1	2	3
Branch Metric for '1'	3	2	1	0	0	0	0	0

# Soft Decision Viterbi(2)

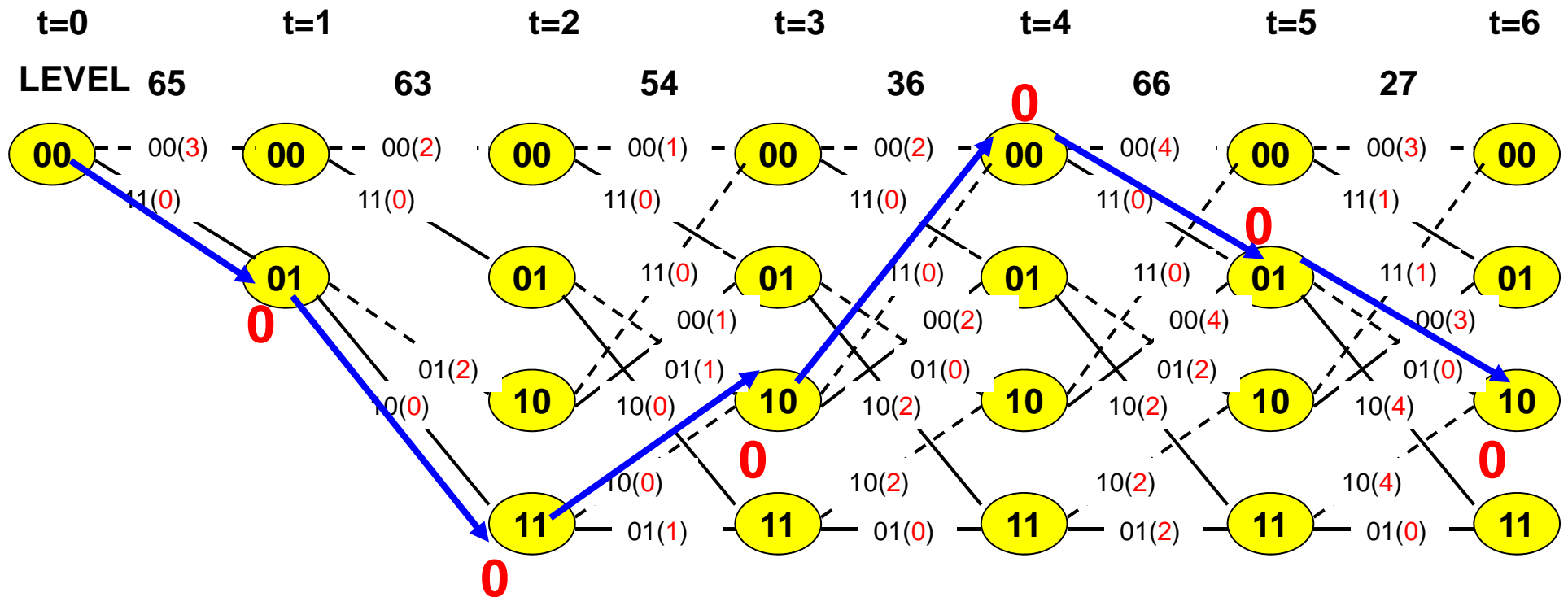
- Calculate Path metric in order to find minimum path metric path.
- Until  $t=6$





# Soft Decision Viterbi(3)

- Select Minimum Path Metric and get original information
- In this example : 1 1 0 0 1 0





# Summary

- 2 types of FEC
  - Block code such as RS
  - Convolutional Code
- Convolutional Code
  - Code Rate
  - Punctured
- Viterbi Decoder : Maximum likelihood decoding
  - Trellis
  - Hard Decision vs. Soft Decision
  - Branch Metric, Path Metric