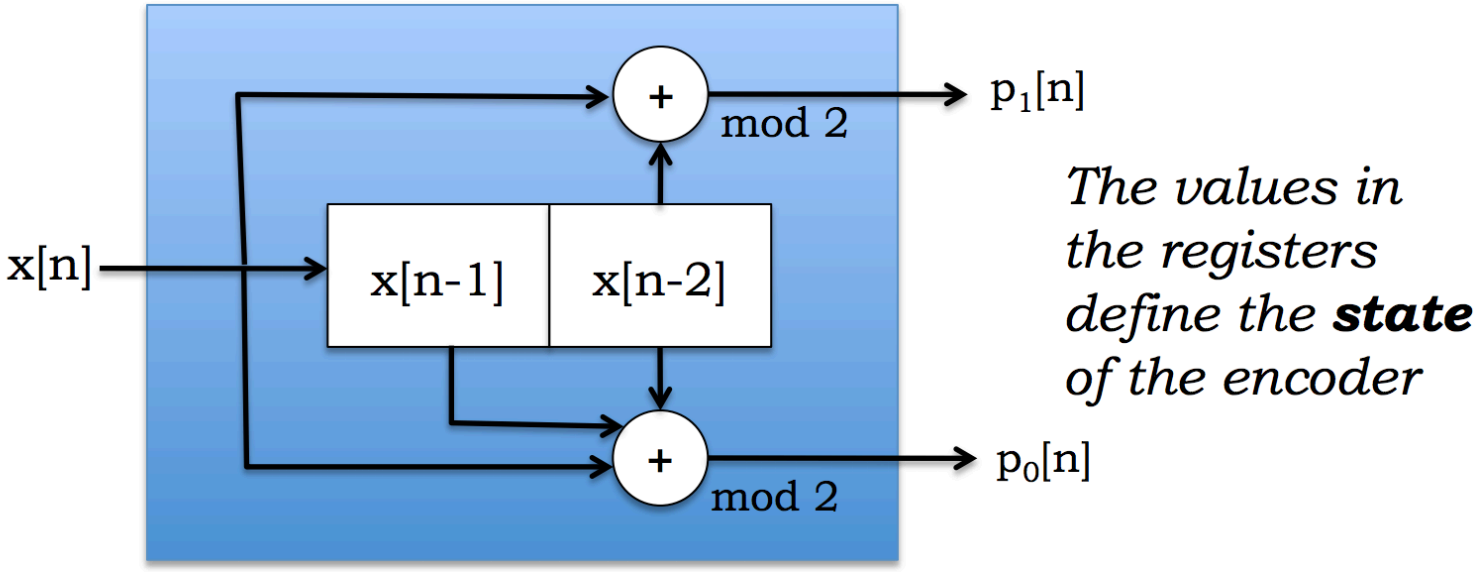
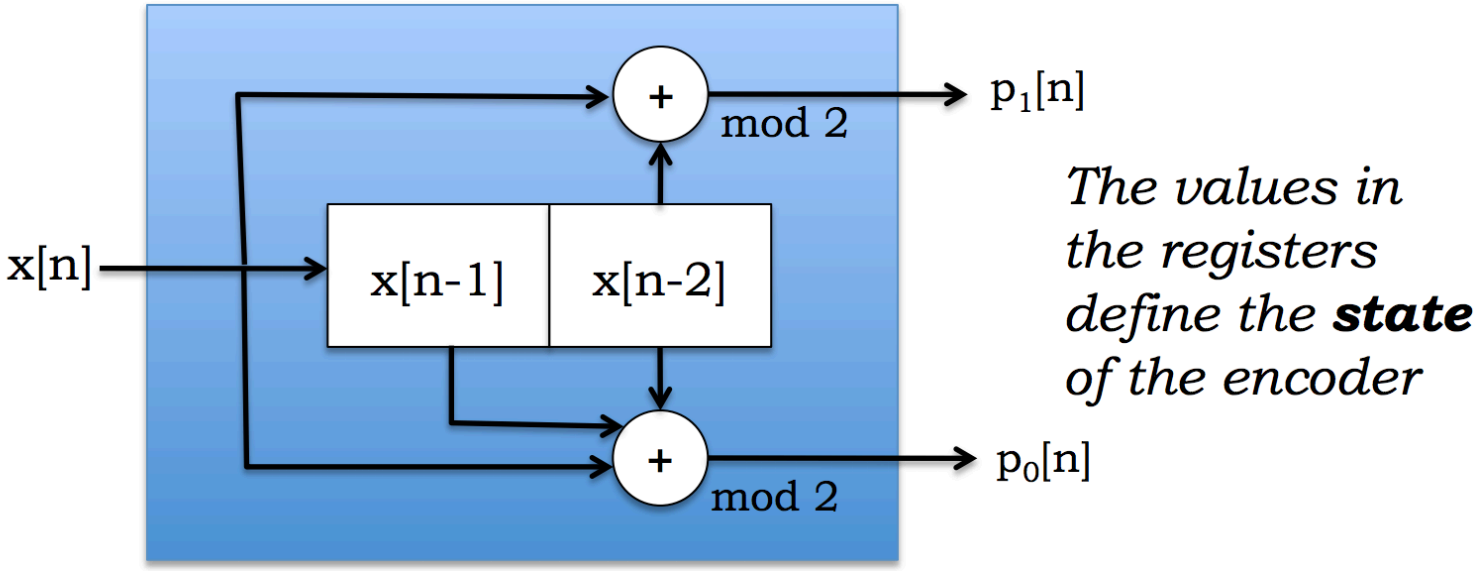
Shift Register View



- **One message bit x[n] in, two parity bits out**
  - **Each timestep:** message bits **shifted right** by one, the **incoming bit** moves into the **left-most register**

Equation View

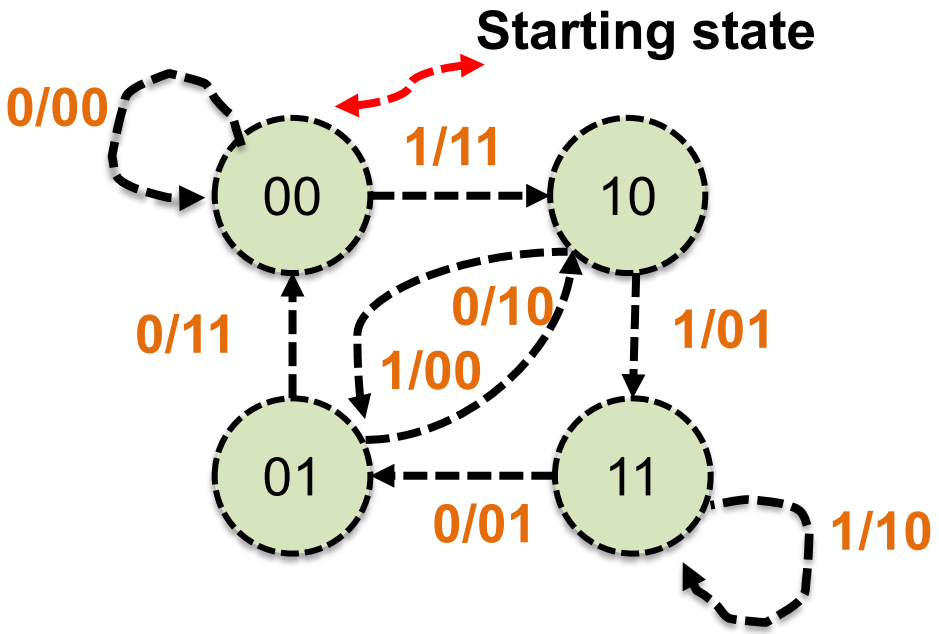


0th stream: $p_0[n] = x[n] + x[n-1] + x[n-2] \pmod 2$

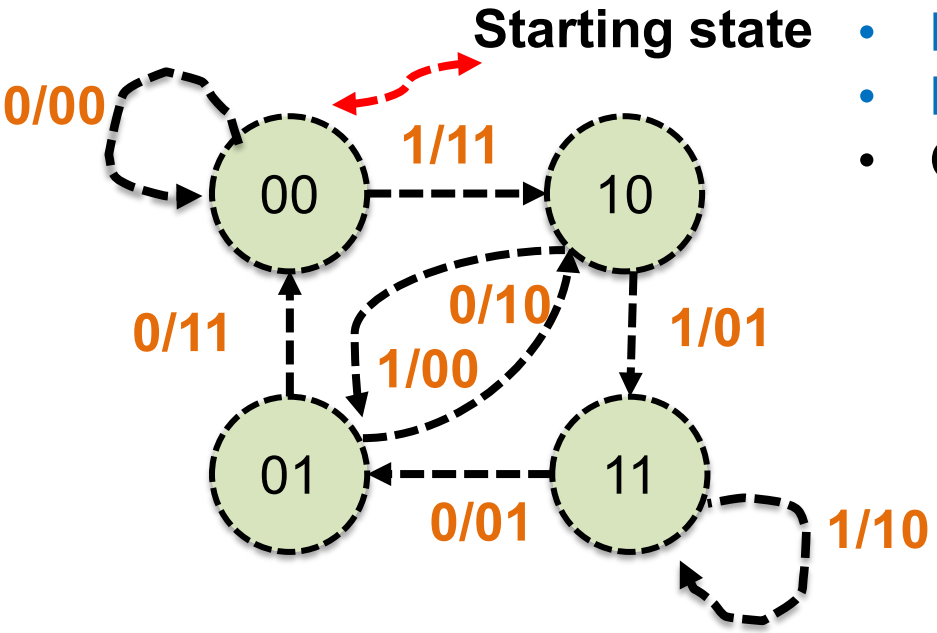
1st stream: $p_1[n] = x[n] + x[n-2] \pmod 2$

State Machine View

- Example: **K = 3, code rate = ½, convolutional code**
  - There are 2^{K-1} states
  - **States** labeled with **(x[n-1], x[n-2])**
  - **Arcs** labeled with **$x[n]/p_0[n]p_1[n]$**
  - **Generator: $g_0 = 111$, $g_1 = 101$**
  - **msg** = 101100



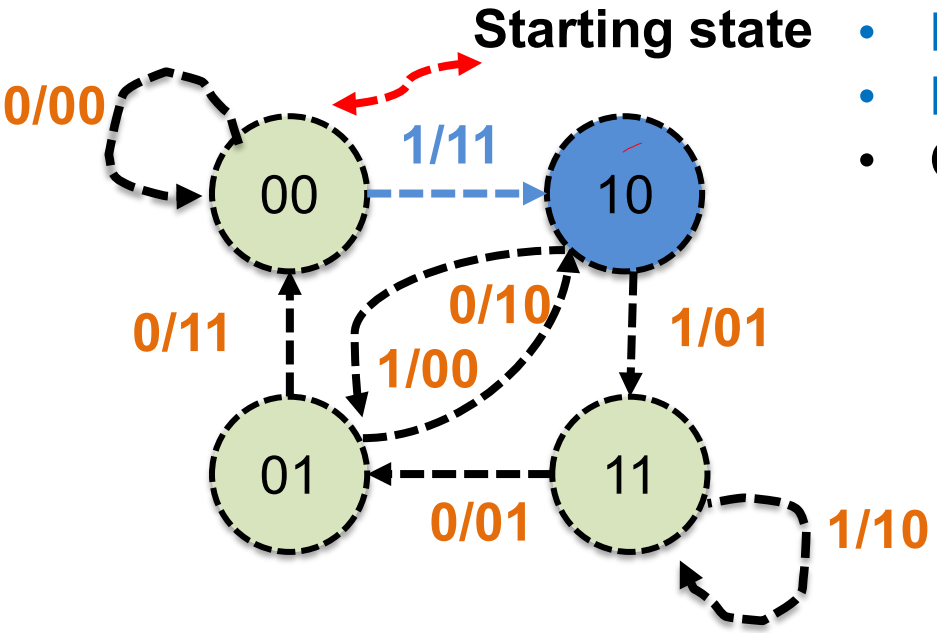
State Machine View



- $P_0[n] = (1*x[n] + 1*x[n-1] + 1*x[n-2]) \bmod 2$
- $P_1[n] = (1*x[n] + 0*x[n-1] + 1*x[n-2]) \bmod 2$
- **Generators**: $g_0 = 111$, $g_1 = 101$

- **msg** = 101100
- **Transmit**:

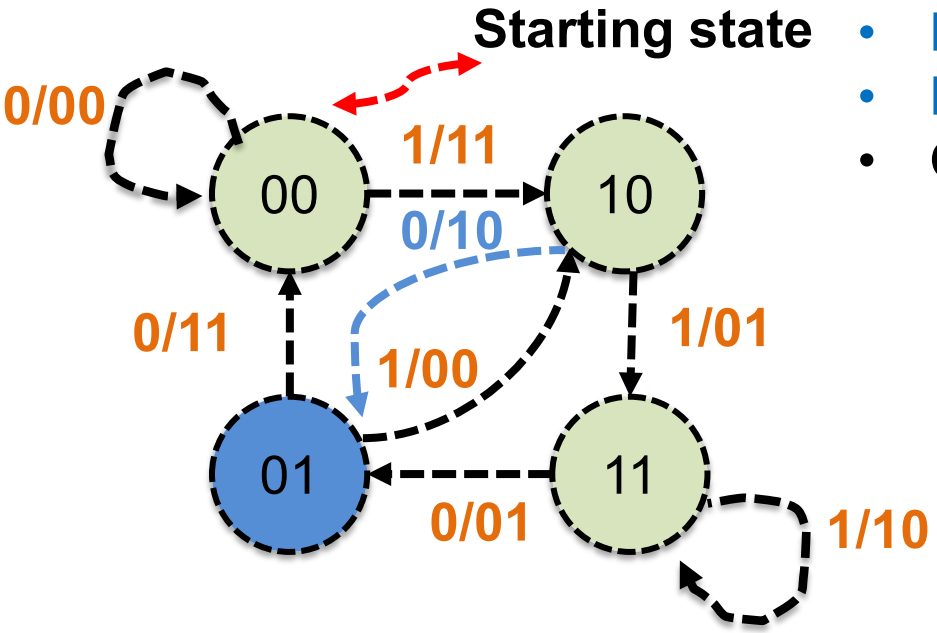
State Machine View



- $P_0[n] = 1*1 + 1*0 + 1*0 \bmod 2$
- $P_1[n] = 1*1 + 0*0 + 1*0 \bmod 2$
- **Generators**: $g_0 = 111$, $g_1 = 101$

- **msg** = 101100
- **Transmit**: **11**

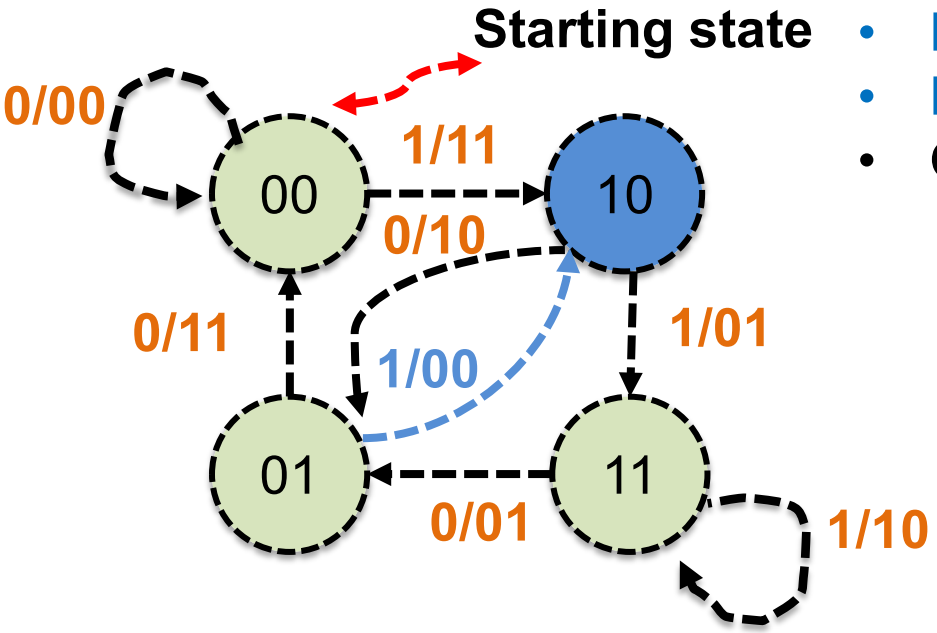
State Machine View



- $P_0[n] = 1*0 + 1*1 + 1*0 \bmod 2$
- $P_1[n] = 1*0 + 0*1 + 1*0 \bmod 2$
- **Generators**: $g_0 = 111$, $g_1 = 101$

- **msg** = 1**0**1100
- **Transmit**: 11 **10**

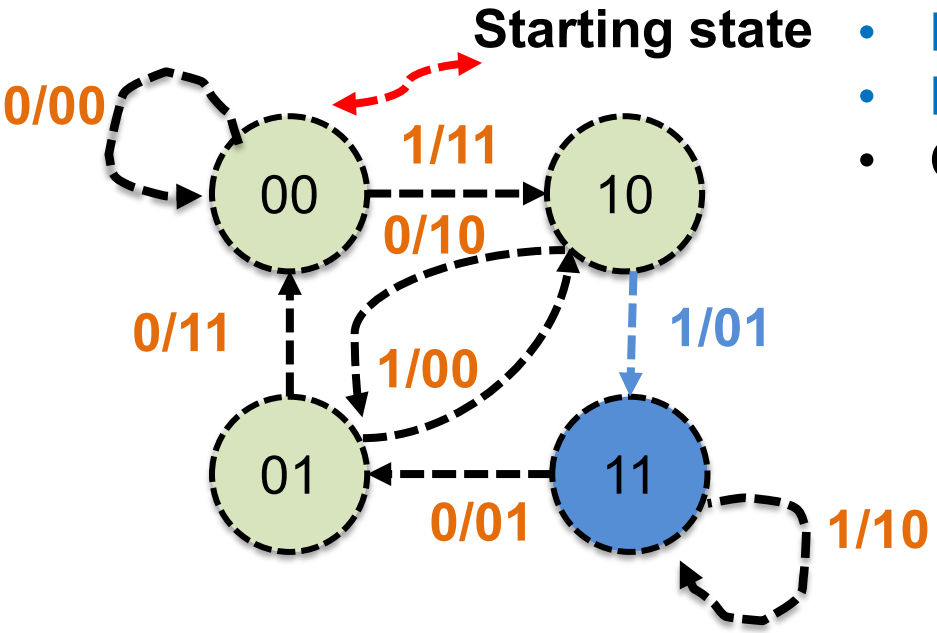
State Machine View



- $P_0[n] = 1*1 + 1*0 + 1*1 \bmod 2$
- $P_1[n] = 1*1 + 0*0 + 1*1 \bmod 2$
- **Generators**: $g_0 = 111$, $g_1 = 101$

- **msg** = 10**1**100
- **Transmit**: 11 10 **00**

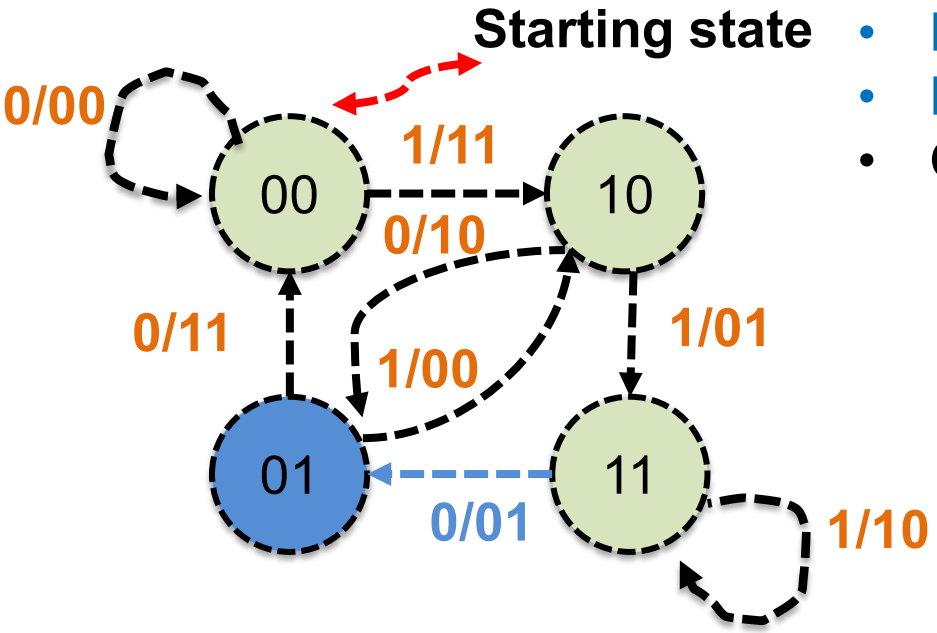
State Machine View



- $P_0[n] = 1*1 + 1*1 + 1*0$
- $P_1[n] = 1*1 + 0*1 + 1*0$
- **Generators**: $g_0 = 111$, $g_1 = 101$

- **msg** = 101**1**00
- **Transmit**: 11 10 00 **01**

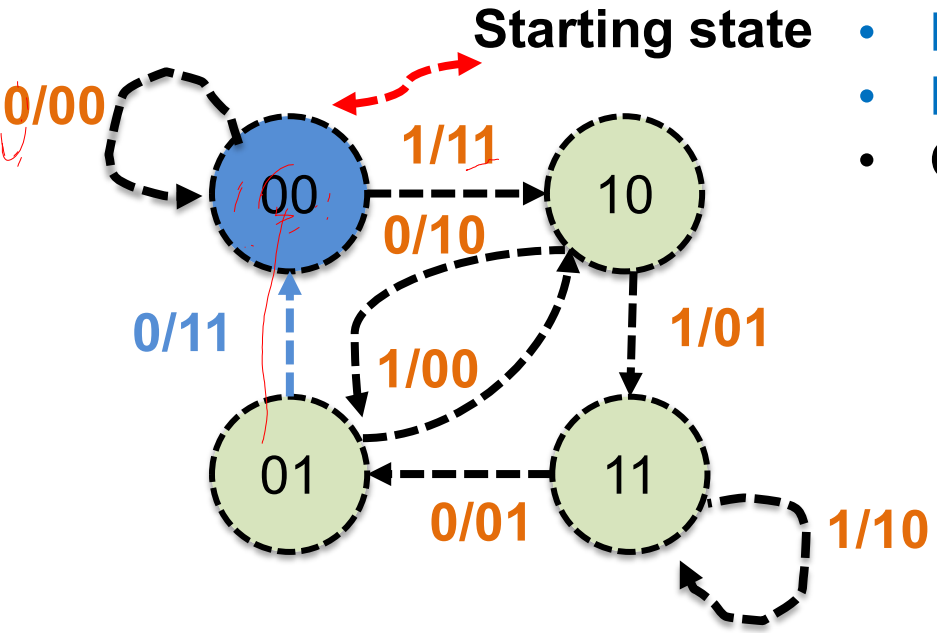
State Machine View



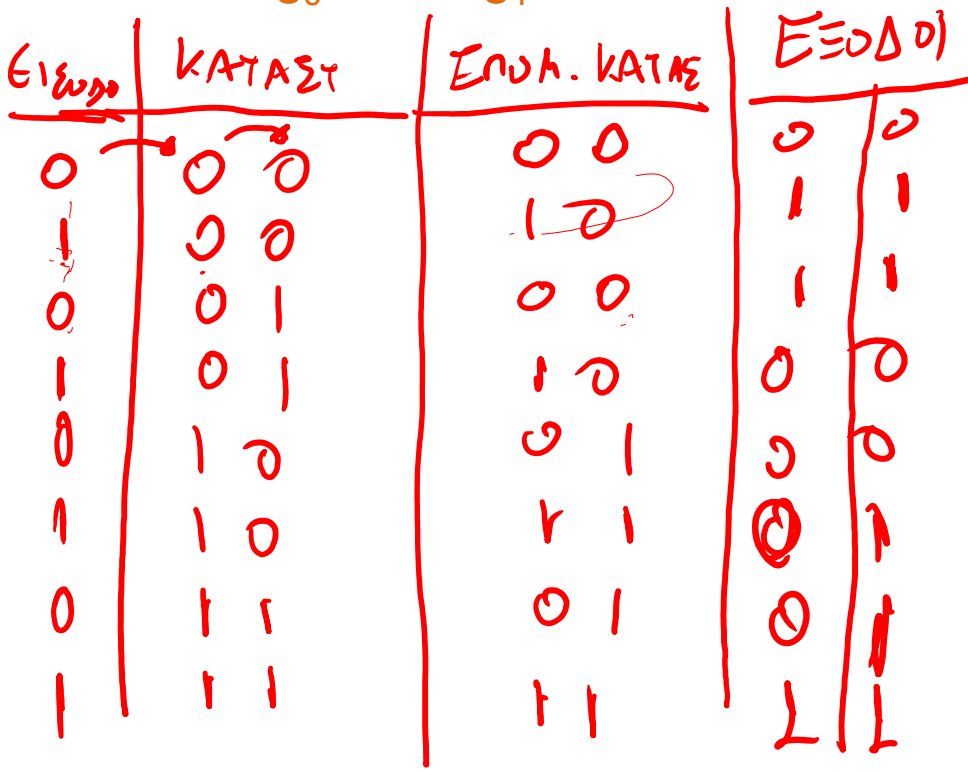
- $P_0[n] = 1*0 + 1*1 + 1*1$
- $P_1[n] = 1*0 + 0*1 + 1*1$
- **Generators**: $g_0 = 111$, $g_1 = 101$

- **msg** = 101100
- **Transmit**: 11 10 00 01 **01**

State Machine View

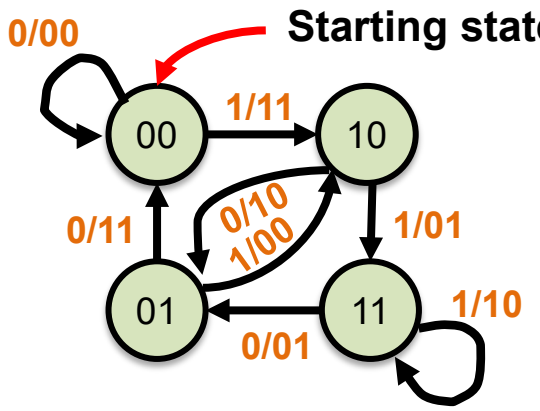


- $P_0[n] = 1*0 + 1*0 + 1*1$
- $P_1[n] = 1*0 + 0*0 + 1*1$
- **Generators**: $g_0 = 111$, $g_1 = 101$



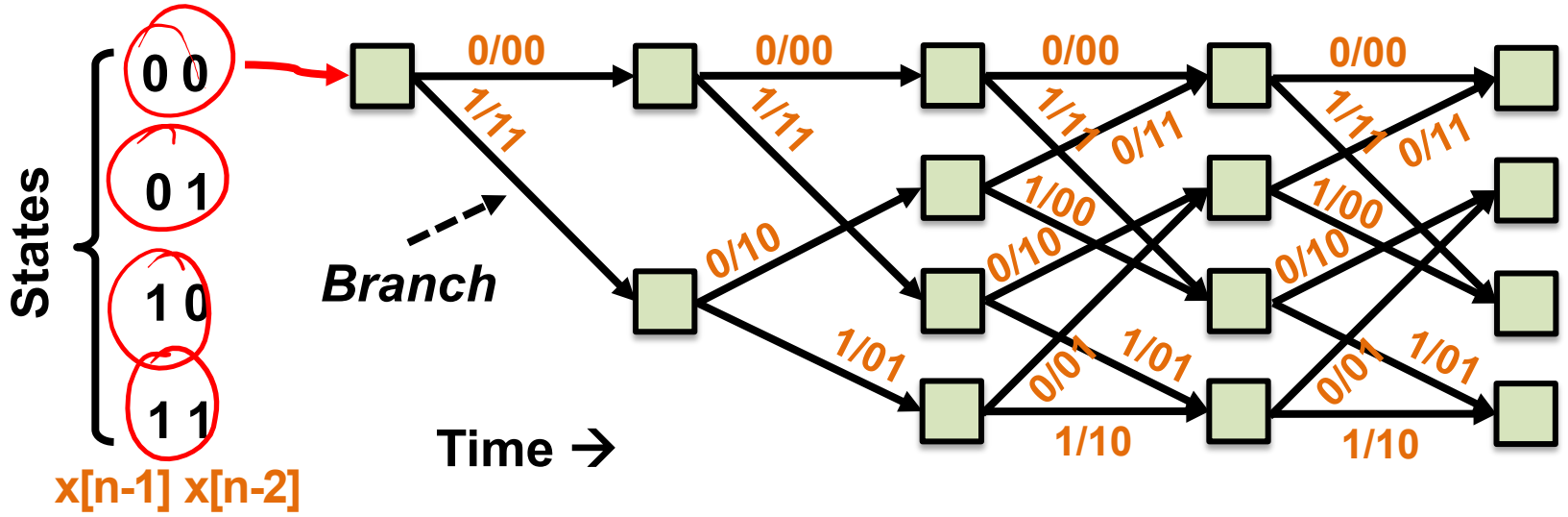
- **msg** = 101100
- **Transmit**: 11 10 00 01 01 **11**

The Trellis



- **Vertically,** lists encoder **states**
- **Horizontally,** tracks **time steps**
- ***Branches*** connect states in successive time steps

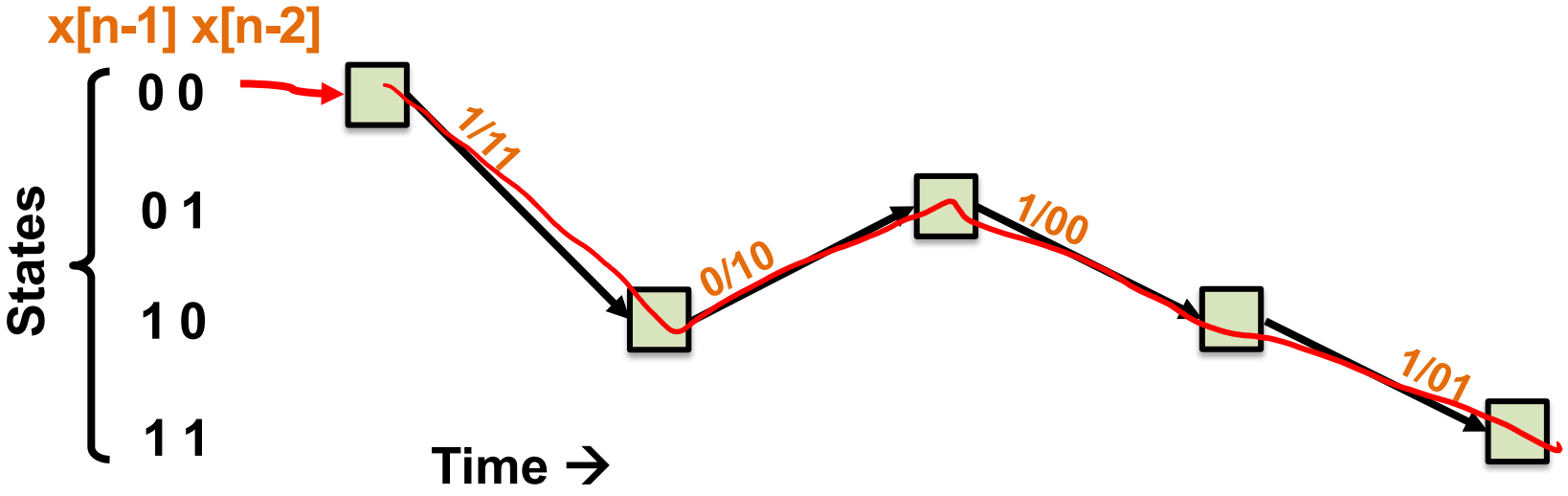
Trellis:



The Trellis: Sender's View

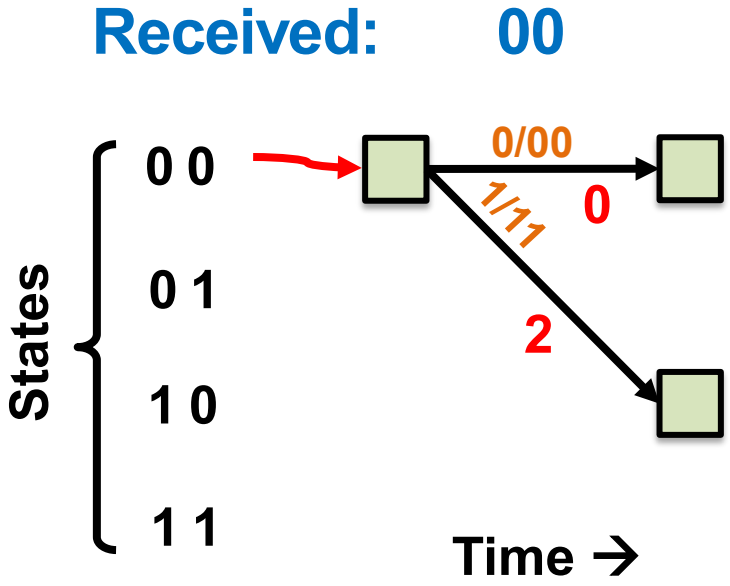
- **At the sender, transmitted bits** trace a unique, single ***path*** **of branches through the trellis**
  - *e.g.* transmitted data bits **1 0 1 1**

- **Recover transmitted bits ⇔ Recover path**



Hard-input branch metric

- Suppose we know encoder is in **state 00,** **receive bits: 00**

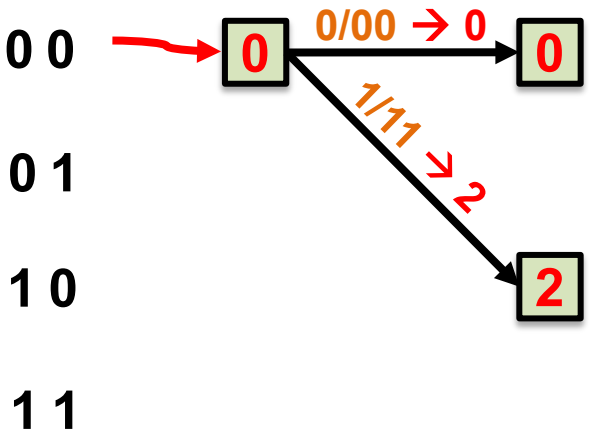


Hard-input path metric

- ***Hard-input path metric:*** **Sum** Hamming distance between **sent** and **received bits** **along path**

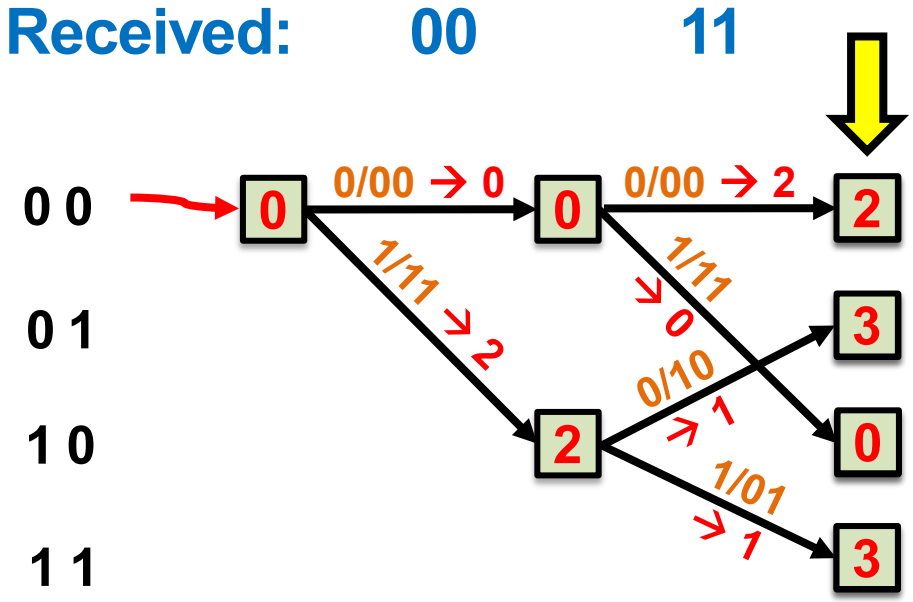
- Encoder is initially in **state 00, receive bits: 00**

Received: 00



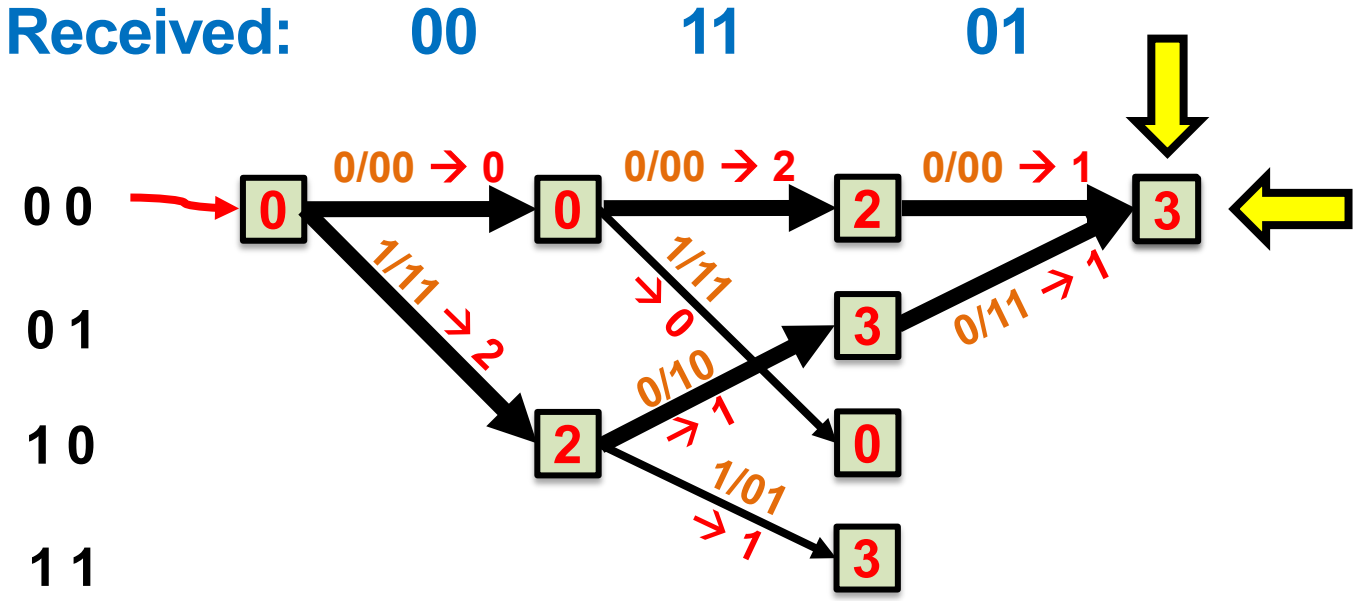
Hard-input path metric

- Right now, each state has a **unique** **predecessor** state
- Path metric: Total bit errors **along path ending at state**
  - Path metric of **predecessor** + **branch metric**



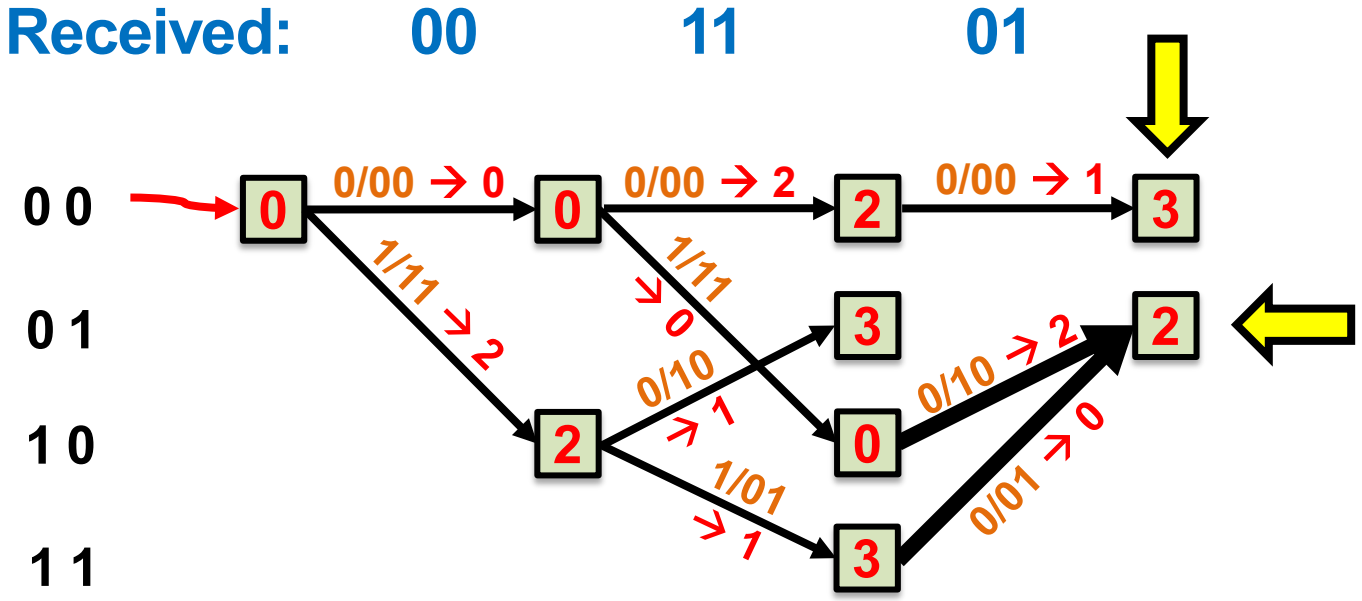
Hard-input path metric

- Each state has **two predecessor states, two *predecessor paths*** (which to use?)
- **Winning** branch has **lower** path metric (**fewer** bit errors): ***Prune losing*** branch



Hard-input path metric

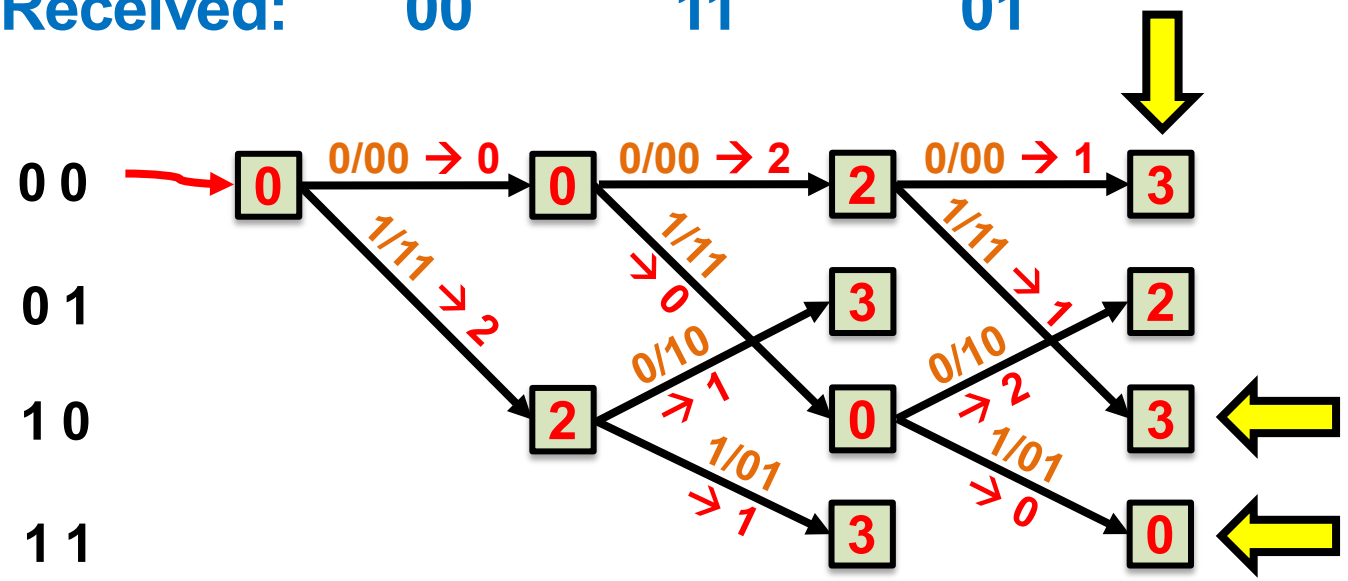
- Prune losing branch **for each state** in trellis



Pruning non-surviving branches

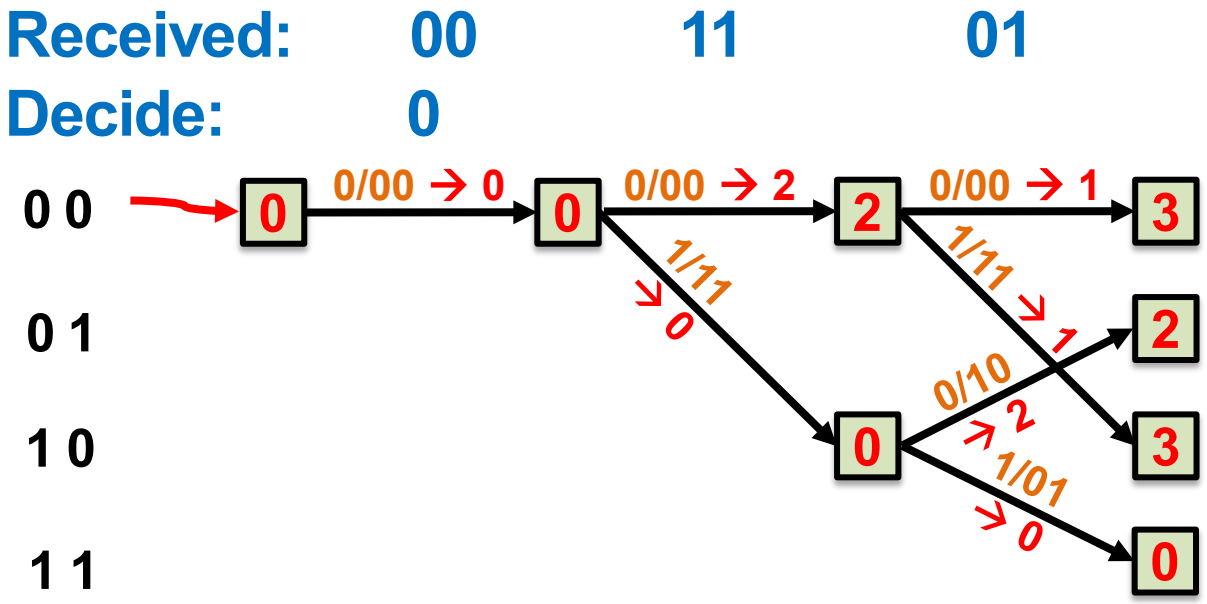
- ***Survivor path*** begins at each state, traces **unique path** back to **beginning** of trellis
  - **Correct path** is one of **four** survivor paths

- Some branches are not part of any survivor: **prune them**



Making bit decisions

- When **only one branch remains** at a stage, the Viterbi algorithm **decides** that branch's **input bits:**



End of received data

- **Trace back** the survivor with **minimal path metric**
- Later stages **don't get benefit** of future error correction, had data not ended

Received:	00	11	01	10
Decide:	0	1	1	1