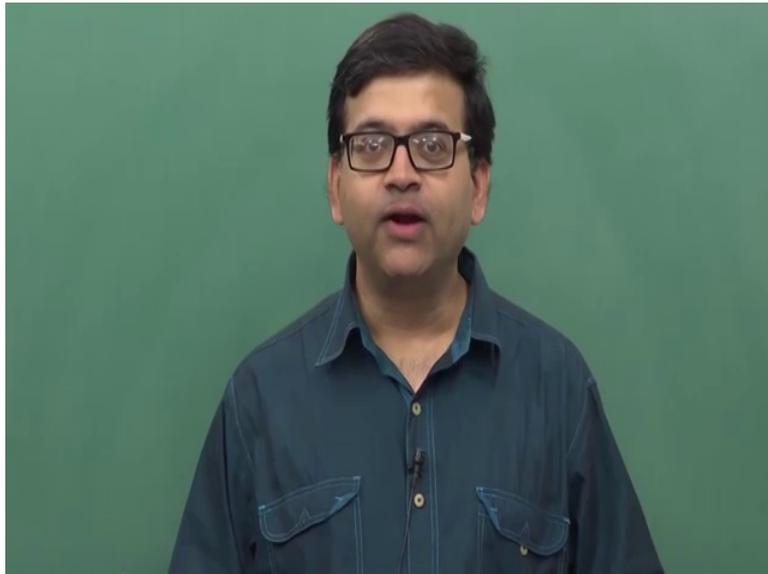


**An Introduction to Coding Theory**  
**Professor Adrish Banerji**  
**Department of Electrical Engineering**  
**Indian Institute of Technology, Kanpur**  
**Module 04**  
**Lecture Number 16**

**Introduction to convolutional codes-II: state diagram, trellis diagram**

(Refer Slide Time 00:13)



Today we are going to talk about how to draw the state diagram and Trellis diagram of convolutional code. So today's lecture

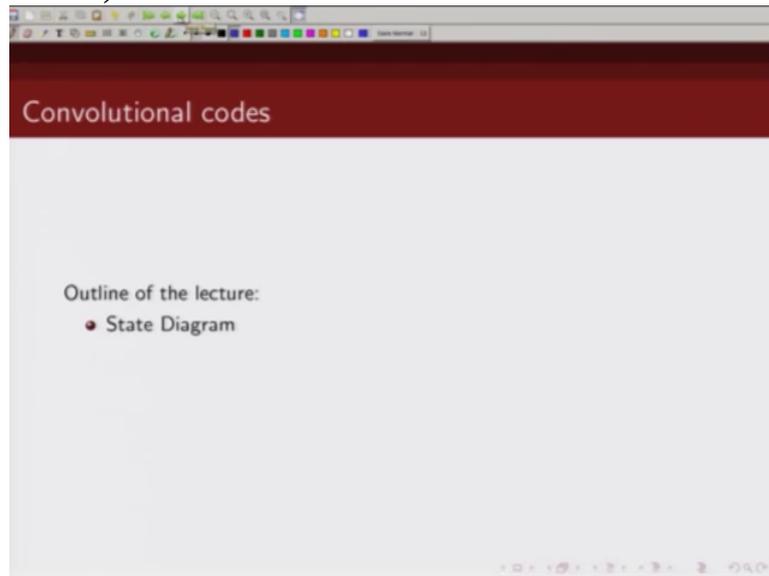
(Refer Slide Time 00:22)



**Lecture #8B: Introduction to convolutional codes-II: state diagram, trellis diagram**

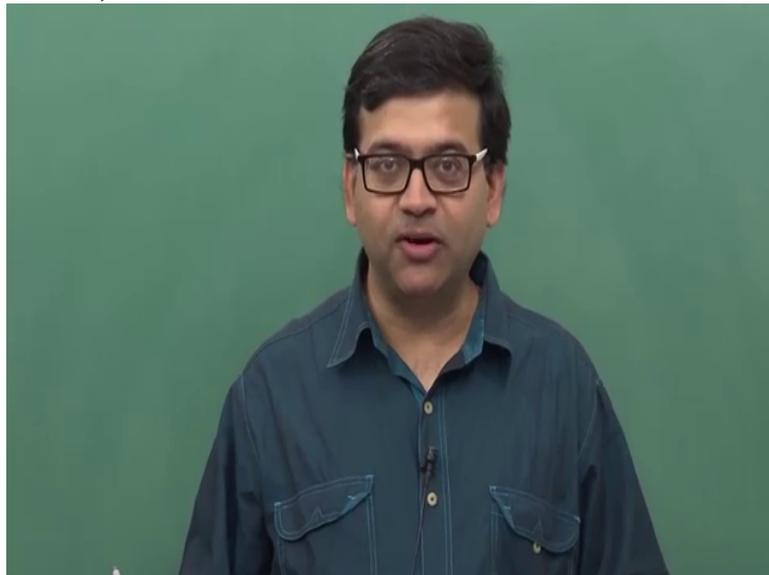
deals with state diagram and Trellis diagram of a convolutional code.

(Refer Slide Time 00:27)



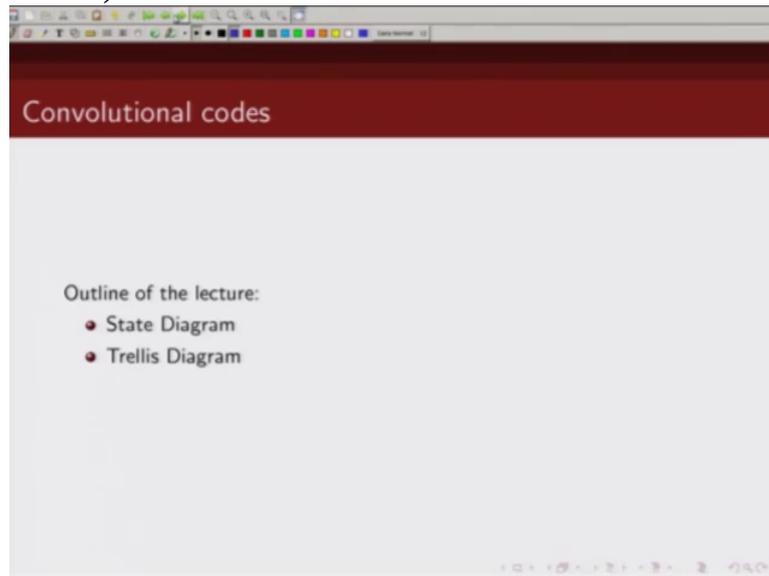
So we will start of this lecture with description of

(Refer Slide Time 00:31)



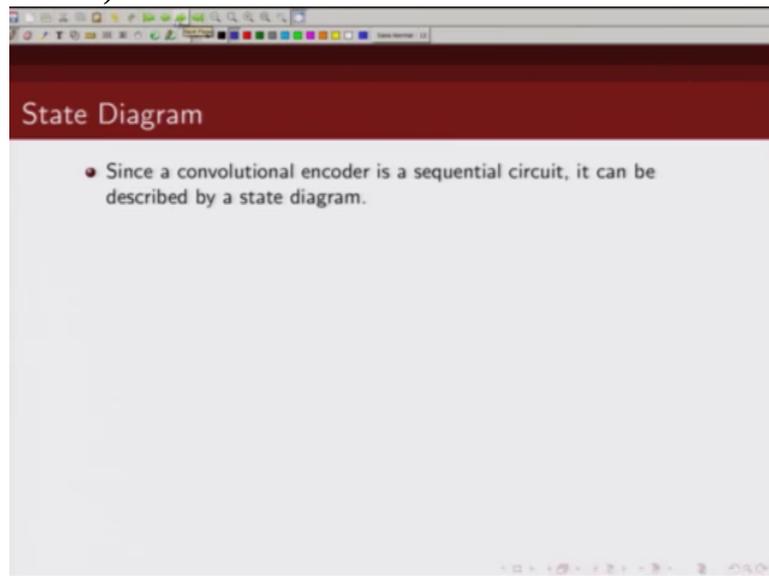
how to construct the state diagram of convolutional code and then we will talk about the Trellis diagram.

(Refer Slide Time 00:39)



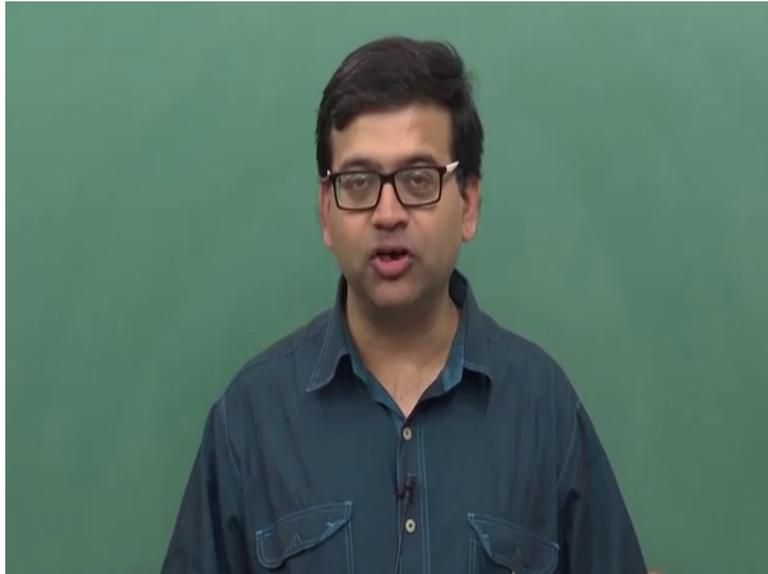
So before we talk about

(Refer Slide Time 00:44)



state diagram and Trellis diagram, just few words about convolutional code and

(Refer Slide Time 00:49)



its equivalent state diagram. Since it is a sequential circuit, the convolutional encoder we can represent it using a state diagram and we are going to make use of state diagram and Trellis diagram when we will be decoding the convolutional code. So state diagram is described

(Refer Slide Time 01:11)

A screenshot of a presentation slide. The slide has a dark red header with the title "State Diagram" in white. Below the header, there is a white background with black text. The text includes two bullet points, a mathematical equation, and a sentence. The slide is shown within a window frame with a standard operating system taskbar at the top.

**State Diagram**

- Since a convolutional encoder is a sequential circuit, it can be described by a state diagram.
- The state of the encoder is defined as its shift register contents. For a  $(n, 1, m)$  code at time instant  $l$ , the state is defined by the  $m$ -tuple

$$S_l = (x_{l-1}, x_{l-2}, \dots, x_{l-m})$$

where  $(x_{l-1}, x_{l-2}, \dots, x_{l-m})$  are the  $m$  bits stored in the shift register.

follows. The state of an encoder is basically defined by the contents of its shift register. So if you have a rate 1 by  $n$  code whose memory order is  $m$  then we define the state at time  $l$  by  $S_{sub l}$  where  $x_{l-1}$ ,  $x_{l-2}$ ,  $x_{l-m}$  these are essentially the contents of the shift registers. So the contents of the shift register will define what is the state of the convolutional encoder.

(Refer Slide Time 01:54)

**State Diagram**

- Since a convolutional encoder is a sequential circuit, it can be described by a state diagram.
- The state of the encoder is defined as its shift register contents. For a  $(n, 1, m)$  code at time instant  $l$ , the state is defined by the  $m$ -tuple
$$S_l = (x_{l-1}, x_{l-2}, \dots, x_{l-m})$$
where  $(x_{l-1}, x_{l-2}, \dots, x_{l-m})$  are the  $m$  bits stored in the shift register.
- There are  $2^m$  number of possible states for a  $(n, 1, m)$  convolutional code.

So if we have the convolutional code whose memory is  $m$  rate 1 by  $n$  code then essentially we will have total  $2$  raised to power  $m$  states because we will have one set of shift register to represent this convolutional code because  $k$  is equal to 1 and since memory is  $m$  so total there are  $2^m$  different possibilities for the state depending on whether the bit is 0 or 1 which is stored in the shift register.

(Refer Slide Time 02:29)

**State Diagram**

- Since a convolutional encoder is a sequential circuit, it can be described by a state diagram.
- The state of the encoder is defined as its shift register contents. For a  $(n, 1, m)$  code at time instant  $l$ , the state is defined by the  $m$ -tuple
$$S_l = (x_{l-1}, x_{l-2}, \dots, x_{l-m})$$
where  $(x_{l-1}, x_{l-2}, \dots, x_{l-m})$  are the  $m$  bits stored in the shift register.
- There are  $2^m$  number of possible states for a  $(n, 1, m)$  convolutional code.
- The output of a convolutional code at each time instant  $l$ , depends on the input and the current state.
$$v_l = f(u_l, S_l)$$

And then output of a convolutional code as we know is function of what is the input and what is the current state. And what

(Refer Slide Time 02:40)

State Diagram

- The convolutional encoder undergoes a state transition whenever a new information bit is input to the encoder.

Time unit	Message bit	State
$l$	$u_l$	$S_l = (x_{l-1}, x_{l-2}, \dots, x_{l-m})$
$l+1$	$u_{l+1}$	$S_{l+1} = (x_l, x_{l-1}, \dots, x_{l-m+1})$

happens when we receive a new bit? When we receive a new bit the convolutional encoder is in a particular state,

(Refer Slide Time 02:47)



let us say at time  $t$ , when it receives a new bit, that new bit will transform it, will take it some other state and some output will be emitted. So we undergo what we call a state

(Refer Slide Time 03:02)

State Diagram

- The convolutional encoder undergoes a state transition whenever a new information bit is input to the encoder.

Time unit	Message bit	State
$l$	$u_l$	$S_l = (x_{l-1}, x_{l-2}, \dots, x_{l-m})$
$l+1$	$u_{l+1}$	$S_{l+1} = (x_l, x_{l-1}, \dots, x_{l-m+1})$

transition whenever a new information bit arrives. So what happens, when the new information bit will arrive, it will transform it from a given state, let's call it  $S_l$  to another

(Refer Slide Time 03:14)

State Diagram

- The convolutional encoder undergoes a state transition whenever a new information bit is input to the encoder.

Time unit	Message bit	State
$l$	$u_l$	$S_l = (x_{l-1}, x_{l-2}, \dots, x_{l-m})$
$l+1$	$u_{l+1}$	$S_{l+1} = (x_l, x_{l-1}, \dots, x_{l-m+1})$

state  $S_{l+1}$  and this

(Refer Slide Time 03:18)

State Diagram

- The convolutional encoder undergoes a state transition whenever a new information bit is input to the encoder.

Time unit	Message bit	State
$l$	$u_l$	$S_l = (x_{l-1}, x_{l-2}, \dots, x_{l-m})$
$l+1$	$u_{l+1}$	$S_{l+1} = (x_l, x_{l-1}, \dots, x_{l-m+1})$

transition happens when you receive a new information message bit.

(Refer Slide Time 03:25)

State Diagram

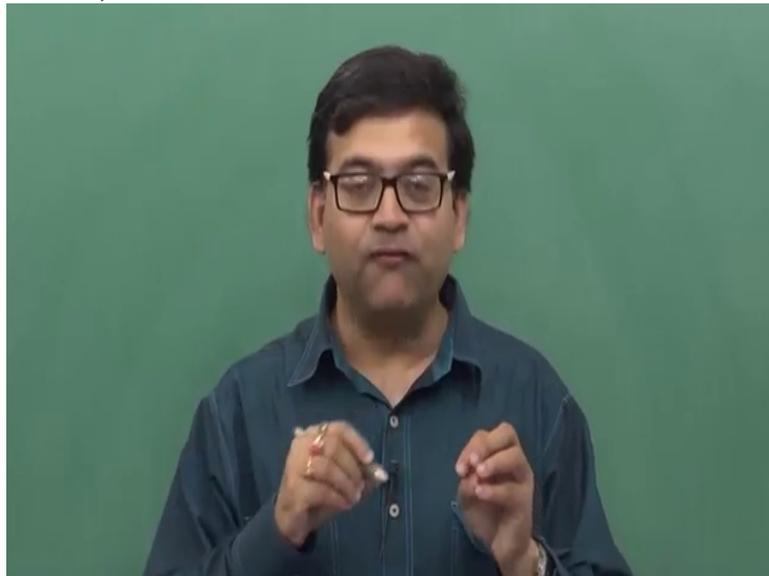
- The convolutional encoder undergoes a state transition whenever a new information bit is input to the encoder.

Time unit	Message bit	State
$l$	$u_l$	$S_l = (x_{l-1}, x_{l-2}, \dots, x_{l-m})$
$l+1$	$u_{l+1}$	$S_{l+1} = (x_l, x_{l-1}, \dots, x_{l-m+1})$

- A state transition is represented by a directed edge connecting two states,  $S_l$  and  $S_{l+1}$ .

And these state transitions are represented in a state diagram by a directed edge. So if there is a directed edge from state

(Refer Slide Time 03:36)



$S_l$  to state  $S_{l+1}$ , that means from state  $S_l$  you will undergo state transition to state  $S_{l+1}$  plus 1 when a particular new bit has arrived. And

(Refer Slide Time 03:50)

**State Diagram**

- The convolutional encoder undergoes a state transition whenever a new information bit is input to the encoder.

Time unit	Message bit	State
$l$	$u_l$	$S_l = (x_{l-1}, x_{l-2}, \dots, x_{l-m})$
$l+1$	$u_{l+1}$	$S_{l+1} = (x_l, x_{l-1}, \dots, x_{l-m+1})$

- A state transition is represented by a directed edge connecting two states,  $S_l$  and  $S_{l+1}$ .

normally what we do is we

(Refer Slide Time 03:53)

**State Diagram**

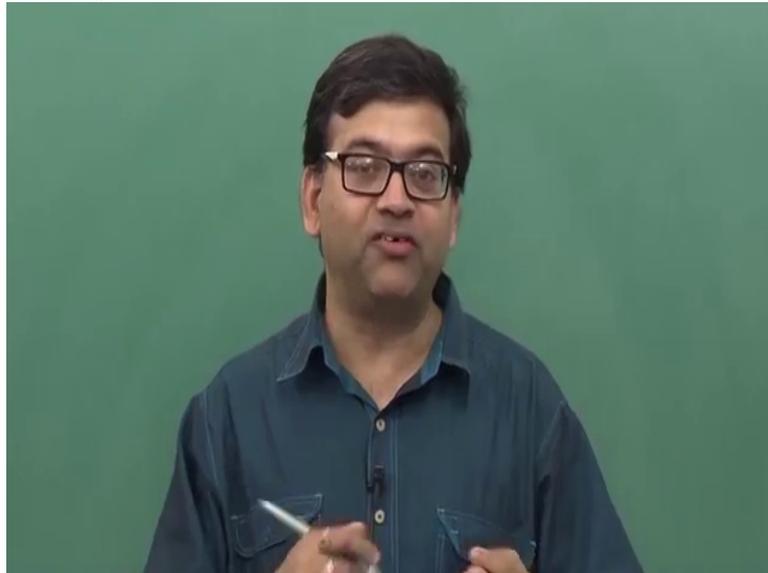
- The convolutional encoder undergoes a state transition whenever a new information bit is input to the encoder.

Time unit	Message bit	State
$l$	$u_l$	$S_l = (x_{l-1}, x_{l-2}, \dots, x_{l-m})$
$l+1$	$u_{l+1}$	$S_{l+1} = (x_{l+1}, x_{l-1}, \dots, x_{l-m+1})$

- A state transition is represented by a directed edge connecting two states,  $S_l$  and  $S_{l+1}$ .
- The state transitions are labeled with the information and coded bits corresponding to that transition.

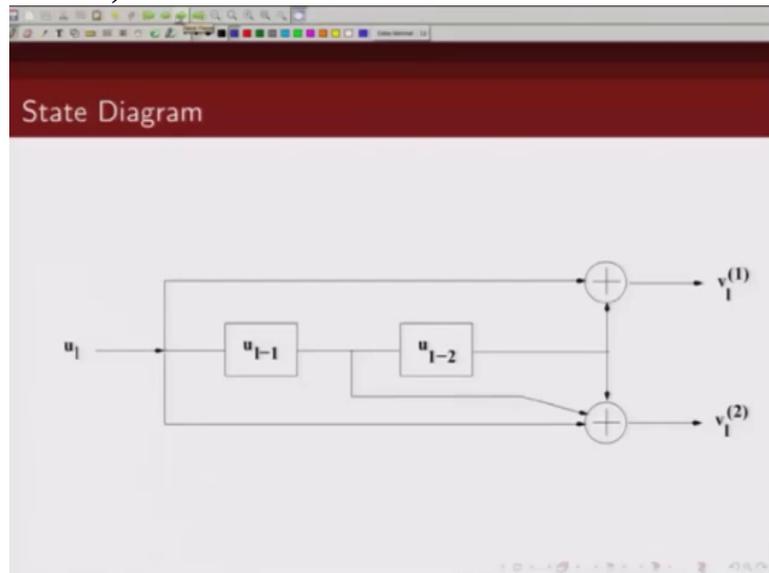
label these state transitions with what is that information bit that caused that state transition and what are the coded bits corresponding to the state transition. So let's say there is information bit 0 which causes

(Refer Slide Time 04:09)



you to move from state 0 0 to state 0 1 then in this we will draw directed graph, directed edge from state 0 0 to state 0 1 and we will label it by input which is 0 and we will also write what is the corresponding output. So that's how we will basically label all these state transitions. And in a state diagram we are essentially enumerating all possible state diagram corresponding to all possible states.

(Refer Slide Time 04:42)



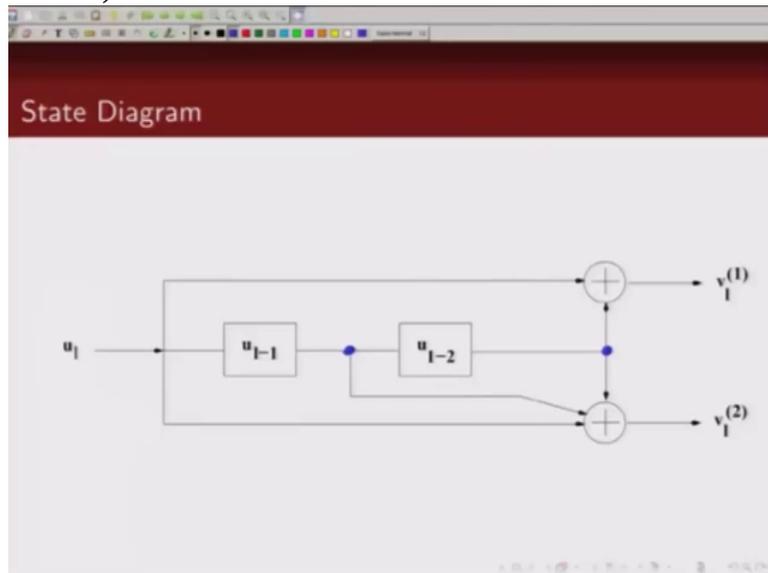
So let us take the same example that we have been dealing so far. This is a rate 1 by 2 convolutional code with memory 2. And what are the states of this? State of this convolutional encoder is given by what is the content of this shift register which I am denoting by blue dots. So what is the content here and what is the content here, that is essentially the state of the convolutional encoder. As you can see basically, when you get a new

(Refer Slide Time 05:19)



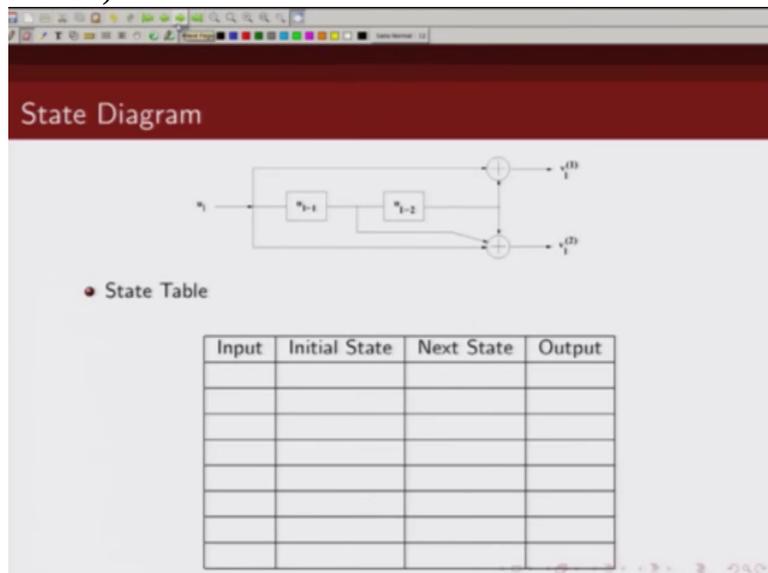
information bit, depending upon what the state is, you will get

(Refer Slide Time 05:25)



corresponding output and what happens in this new information bit will cause, in the next cycle this information bit will move here and whatever was here will move here, so there will be a state transition because this  $u_l$  will move to this location and what was stored here will move to this location, Ok. So let us try to draw the state diagram for this convolutional encoder.

(Refer Slide Time 05:59)



So as we said basically these are the states of the convolutional encoder. We have to, we first enumerate all possible state all possible input. So since the memory is 2, all possible states are 0 0, 0 1, let's say 0 0, 0 1, 1 0 and 1 1. And corresponding

(Refer Slide Time 06:29)

State Diagram

State Table

Input	Initial State	Next State	Output
	0 0		
	0 0		
	0 1		
	0 1		
	1 0		
	1 0		
	1 1		
	1 1		

to each state we can have input 0 or 1, input 0 or 1, input 0 or 1, input 0 or 1.

(Refer Slide Time 06:40)

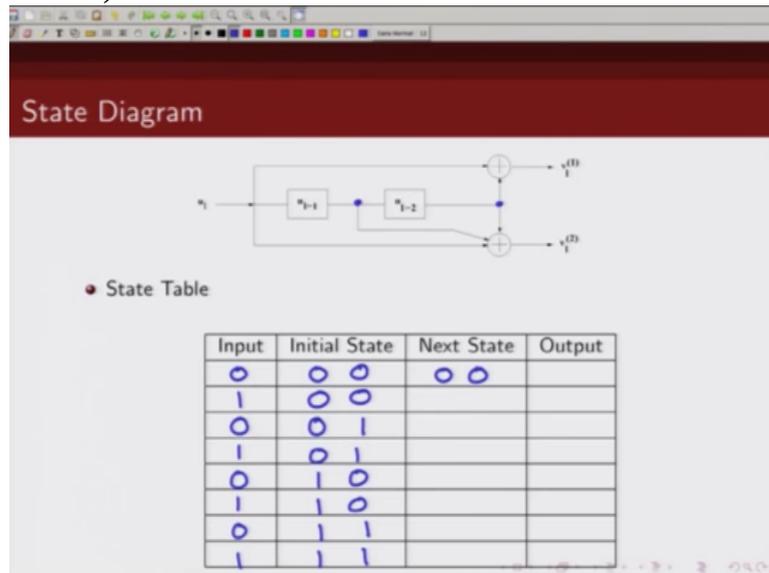
State Diagram

State Table

Input	Initial State	Next State	Output
0	0 0		
1	0 0		
0	0 1		
1	0 1		
0	1 0		
1	1 0		
0	1 1		
1	1 1		

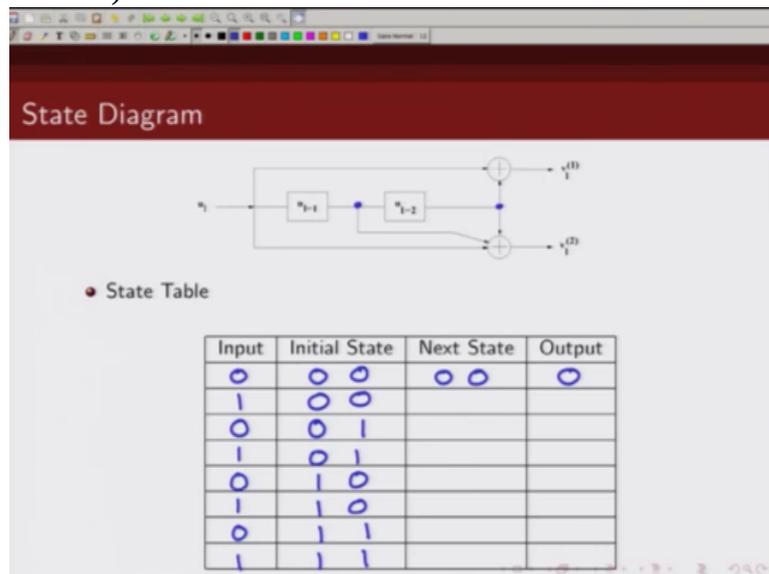
Ok. So now let us look at each case separately. So we have input 0 and initial state 0. Now what's going to happen in the next state? So when there is a transition, this 0 will move in here and this is 0 0 so this 0 will move here. So when a zero comes and you are in initial state 0 0, this will move state 0 0 so it will remain

(Refer Slide Time 07:08)



in state 0 0 if the input is 0. And what is the corresponding output? Since the input is 0 and this state is 0 what you will get here is 0 plus 0 is 0 and 0 plus 0 plus 0 is 0, so output will be 0.

(Refer Slide Time 07:29)



Now let's look at this. If the input is 1, and the initial state is 0 0, then this 1, the 1 which is here will now move to this location, right. So the next state would be 1 0

(Refer Slide Time 07:51)

State Diagram

State Table

Input	Initial State	Next State	Output
0	0 0	0 0	0
1	0 0	1 0	
0	0 1		
1	0 1		
0	1 0		
1	1 0		
0	1 1		
1	1 1		

because 1 from this input will move here and 0 from here will move. So the next state would be 1 0. And what would be the output, since input is 0, initial states are 0, so 1 plus 0 will be 1 and

(Refer Slide Time 08:10)

State Diagram

State Table

Input	Initial State	Next State	Output
0	0 0	0 0	0
1	0 0	1 0	1
0	0 1		
1	0 1		
0	1 0		
1	1 0		
0	1 1		
1	1 1		

1 plus 0 plus 0 will be 1 so this output will be 1, and this output is 0 0, we have already mentioned this.

(Refer Slide Time 08:17)

State Diagram

State Table

Input	Initial State	Next State	Output
0	0 0	0 0	0 0
1	0 0	1 0	1 1
0	0 1		
1	0 1		
0	1 0		
1	1 0		
0	1 1		
1	1 1		

Let's look at next case. Initial state 0 1 so initially this is 0 and this is 1, if the input is 0, if the input is 0 what is going to happen, this 0 will move here and this 0 will move here so when you get a 0 and the initial state is 0 0, next state will be 0 0

(Refer Slide Time 08:47)

State Diagram

State Table

Input	Initial State	Next State	Output
0	0 0	0 0	0 0
1	0 0	1 0	1 1
0	0 1	0 0	
1	0 1		
0	1 0		
1	1 0		
0	1 1		
1	1 1		

because this 0 which is here will move to here and this 0 will move to here. So next state will be 0 0 and what is the output? This will be 1 and this will be 1 so the output is 1 1.

(Refer Slide Time 09:06)

State Diagram

State Table

Input	Initial State	Next State	Output
0	0 0	0 0	0 0
1	0 0	1 0	1 1
0	0 1	0 0	1 1
1	0 1		
0	1 0		
1	1 0		
0	1 1		
1	1 1		

Next, if the initial state is 0 1 but the input is 1, in that case the next state will be this, 0 will move here and this 1 will move here, so next state would be

(Refer Slide Time 09:26)

State Diagram

State Table

Input	Initial State	Next State	Output
0	0 0	0 0	0 0
1	0 0	1 0	1 1
0	0 1	0 0	1 1
1	0 1		
0	1 0		
1	1 0		
0	1 1		
1	1 1		

(Refer Slide Time 09:28)

State Diagram

State Table

Input	Initial State	Next State	Output
0	0 0	0 0	0 0
1	0 0	1 0	1 1
0	0 1	0 0	1 1
1	0 1	1 0	
0	1 0		
1	1 0		
0	1 1		
1	1 1		

and what's the corresponding output? This would be 1 plus 1 that's 0 and this will be 1 plus 0 plus 1, that's again 0,

(Refer Slide Time 09:39)

State Diagram

State Table

Input	Initial State	Next State	Output
0	0 0	0 0	0 0
1	0 0	1 0	1 1
0	0 1	0 0	1 1
1	0 1	1 0	0 0
0	1 0		
1	1 0		
0	1 1		
1	1 1		

Ok.

Now let's look at another case. So next you have initial state 1 0. So initial state 1 0, if the input is 0, then the next state will be 0 1

(Refer Slide Time 10:02)

State Diagram

State Table

Input	Initial State	Next State	Output
0	0 0	0 0	0 0
1	0 0	1 0	1 1
0	0 1	0 0	1 1
1	0 1	1 0	0 0
0	1 0	0 1	
1	1 0		
0	1 1		
1	1 1		

and what's the output, when input is 0, the output will be 0 plus 0 is 0, and 0 plus 1 plus 0 is 1. So output is

(Refer Slide Time 10:13)

State Diagram

State Table

Input	Initial State	Next State	Output
0	0 0	0 0	0 0
1	0 0	1 0	1 1
0	0 1	0 0	1 1
1	0 1	1 0	0 0
0	1 0	0 1	0 1
1	1 0		
0	1 1		
1	1 1		

0 1. And if the input is 1, then the next state would be 1 1

(Refer Slide Time 10:23)

State Diagram

State Table

Input	Initial State	Next State	Output
0	0 0	0 0	0 0
1	0 0	1 0	1 1
0	0 1	0 0	1 1
1	0 1	1 0	0 0
0	1 0	0 1	0 1
1	1 0	1 1	1 1
0	1 1		
1	1 1		

and the output will be 1 plus 0 is 1, and 1 plus 1 plus 0 is 0.

And finally

(Refer Slide Time 10:33)

State Diagram

State Table

Input	Initial State	Next State	Output
0	0 0	0 0	0 0
1	0 0	1 0	1 1
0	0 1	0 0	1 1
1	0 1	1 0	0 0
0	1 0	0 1	0 1
1	1 0	1 1	1 1
0	1 1		
1	1 1		

if the next state is 1 1, let's say if the input is 1 then the next state is 1 1 and the output will be

1

(Refer Slide Time 10:45)

State Diagram

State Table

Input	Initial State	Next State	Output
0	0 0	0 0	0 0
1	0 0	1 0	1 1
0	0 1	0 0	1 1
1	0 1	1 0	0 0
0	1 0	0 1	0 1
1	1 0	1 1	1 0
0	1 1		
1	1 1	1 1	

plus 1 is 0 and 1 plus 1 plus 1 is 1 and if

(Refer Slide Time 10:51)

State Diagram

State Table

Input	Initial State	Next State	Output
0	0 0	0 0	0 0
1	0 0	1 0	1 1
0	0 1	0 0	1 1
1	0 1	1 0	0 0
0	1 0	0 1	0 1
1	1 0	1 1	1 0
0	1 1		
1	1 1	1 1	0 1

this is, instead of 1 if this is 0 then next state would be 0 1 and output will be

(Refer Slide Time 11:00)

State Diagram

State Table

Input	Initial State	Next State	Output
0	0 0	0 0	0 0
1	0 0	1 0	1 1
0	0 1	0 0	1 1
1	0 1	1 0	0 0
0	1 0	0 1	0 1
1	1 0	1 1	1 0
0	1 1	0 1	
1	1 1	1 1	0 1

0 plus 1, which is 1 and this is 0 plus 1 plus 1, this is 0.

(Refer Slide Time 11:10)

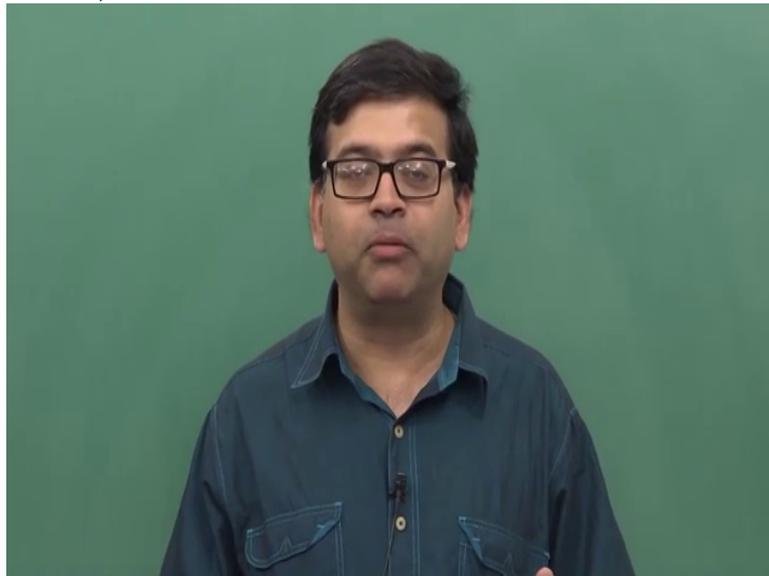
State Diagram

State Table

Input	Initial State	Next State	Output
0	0 0	0 0	0 0
1	0 0	1 0	1 1
0	0 1	0 0	1 1
1	0 1	1 0	0 0
0	1 0	0 1	0 1
1	1 0	1 1	1 0
0	1 1	0 1	1 0
1	1 1	1 1	0 1

So note that we have drawn all possible state transitions corresponding to all possible inputs and all possible initial states. Now what the state diagram does is it represents the state table

(Refer Slide Time 11:27)



graphically and how do we show it graphically, so

(Refer Slide Time 11:32)

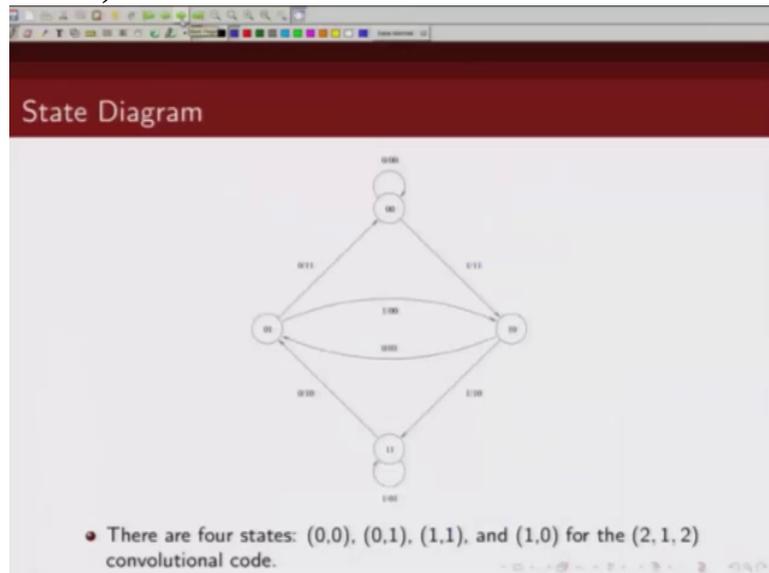
The slide is titled "State Diagram" and shows a block diagram of a system. The input is  $x_1$ . The system consists of two delay elements,  $z^{-1}$ , and two adders. The first adder takes the input  $x_1$  and the output of the first delay element,  $y_1^{(1)}$ , and produces the output  $y_1^{(2)}$ . The second adder takes the output of the first adder,  $y_1^{(2)}$ , and the output of the second delay element,  $y_2^{(1)}$ , and produces the output  $y_2^{(2)}$ .

• State Table

Input	Initial State	Next State	Output
0	0 0	0 0	0 0
1	0 0	1 0	1 1
0	0 1	0 0	1 1
1	0 1	1 0	0 0
0	1 0	0 1	0 1
1	1 0	1 1	1 0
0	1 1	0 1	1 0
1	1 1	1 1	0 1

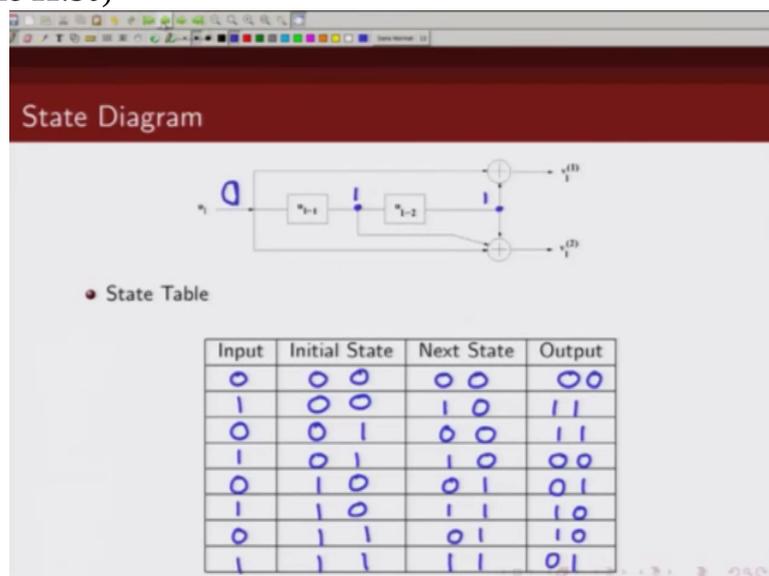
we will go back to

(Refer Slide Time 11:34)



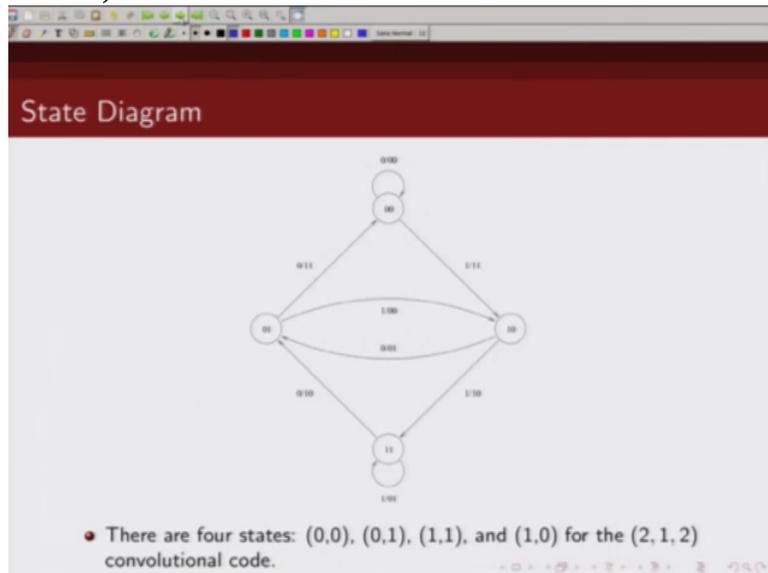
this, so note here we have 4 states, 0 0, 0 1, 1 0, 1 1 and we are showing each transition graphically. So if you go back and see here,

(Refer Slide Time 11:50)



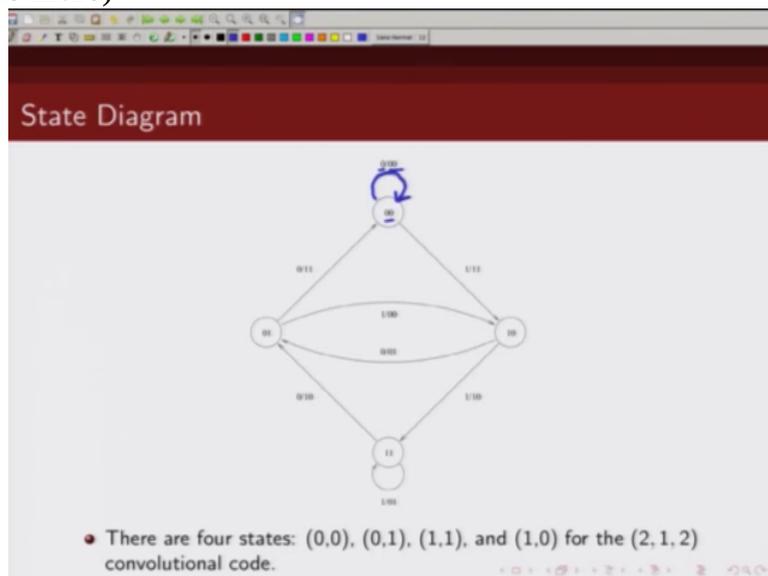
if the initial state is 0 and input is 0, it remains in 0 state and output is 0 0. How do we denote this? Initial state

(Refer Slide Time 12:00)



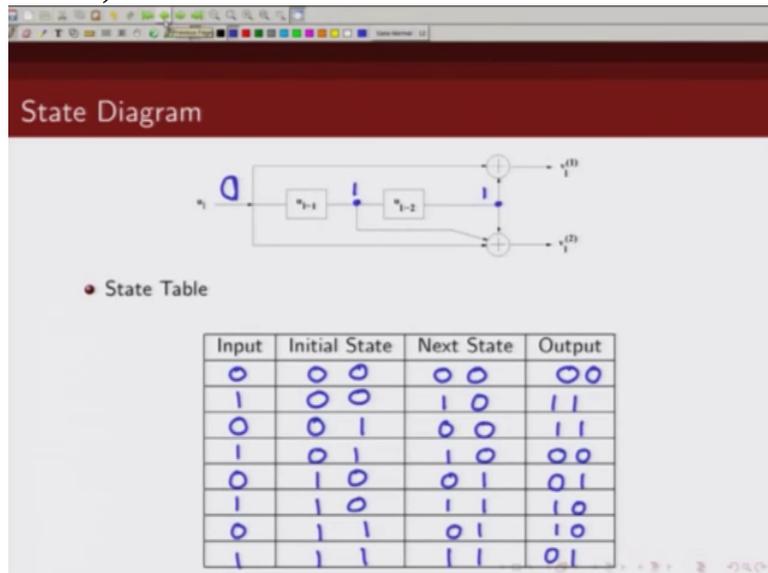
is 0 0, input is 0, corresponding output is 0 0 and it remains in all 0 state. So that we are

(Refer Slide Time 12:10)



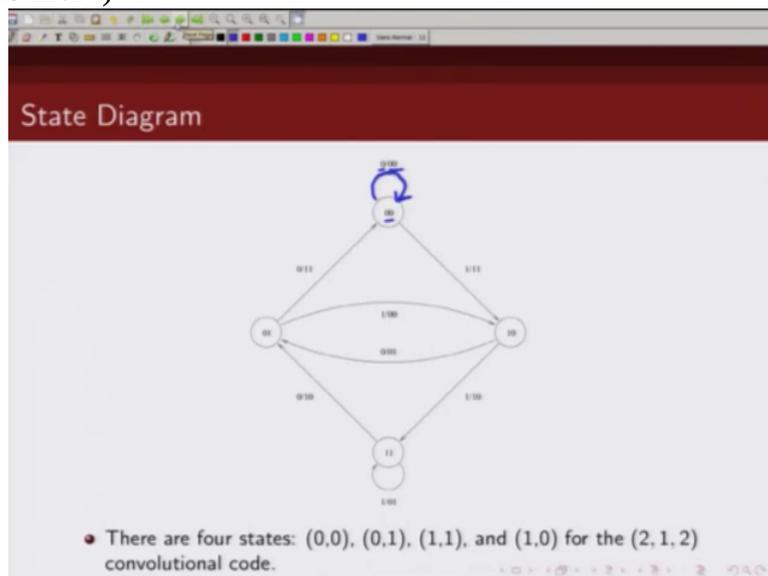
denoting like this. Now what happens when you are in 0 0 state and you receive a 1, if you are in

(Refer Slide Time 12:19)



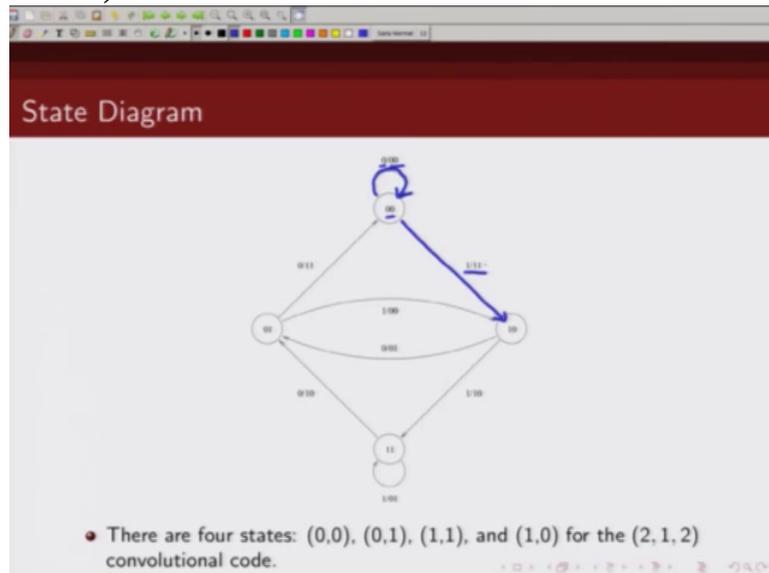
0 0 state and you receive a 1, you move to next state 1 0 and the output is 1 1,

(Refer Slide Time 12:27)



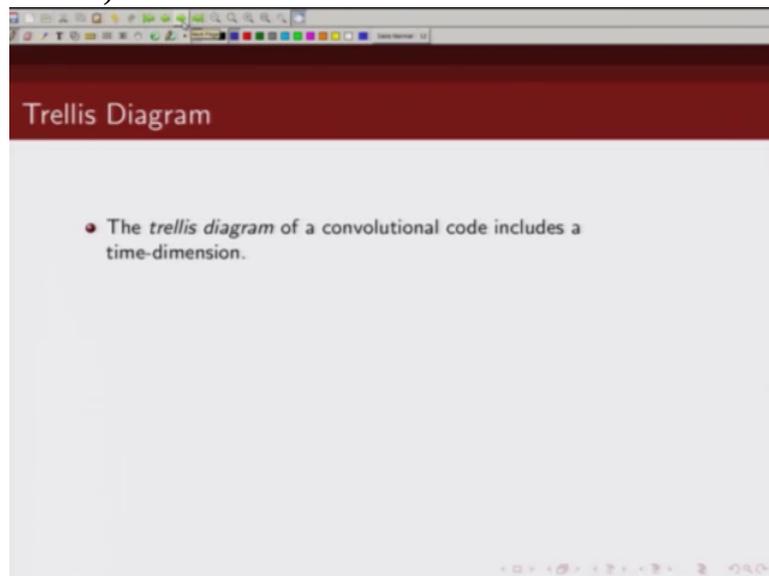
so when you get a 1, you move to next state which is 1 0 and the output is 1 1. So note how I

(Refer Slide Time 12:36)



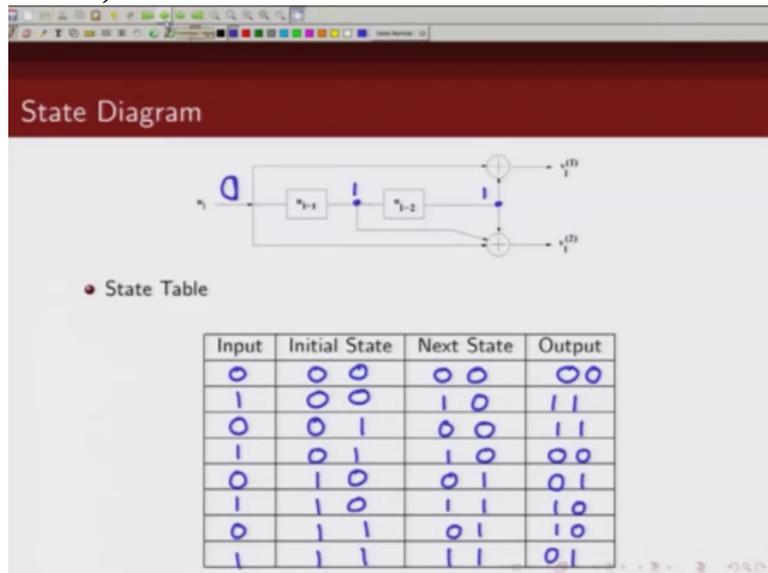
am writing this state transition graphically. So I show by directed graph, transition from state 0 0 to state 1 0 and I am labeling this state transition by the corresponding input and output. So 1 is the input and after slash I have the output. Similarly you do this for all

(Refer Slide Time 13:03)



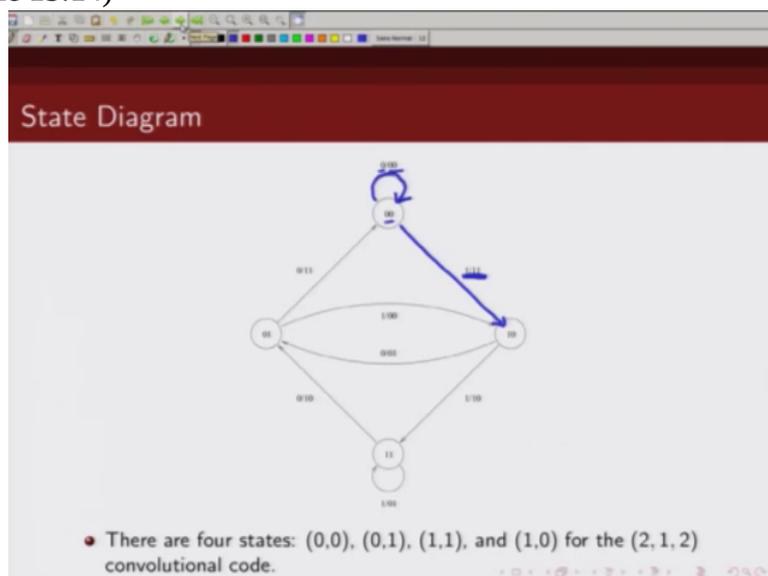
the states. So

(Refer Slide Time 13:05)



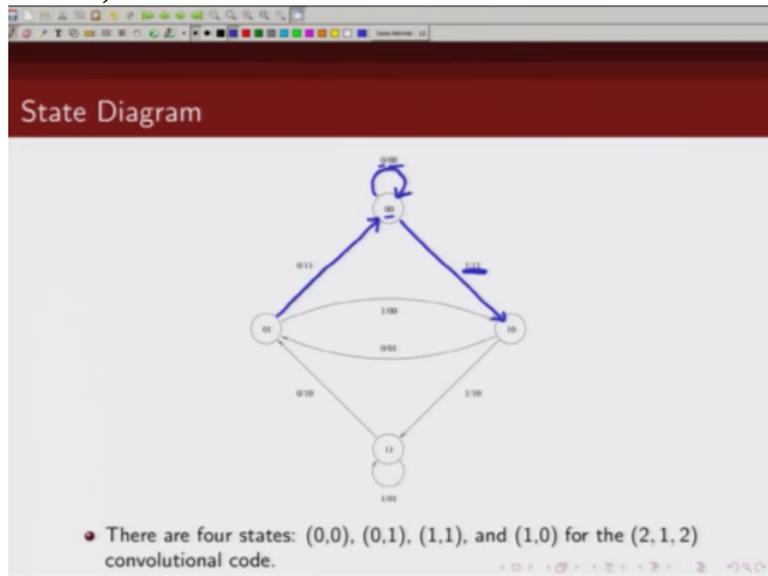
if the initial state is 0 1 and you get a 0, you move to 0 0 state and output is 1 1. So if you are in

(Refer Slide Time 13:14)



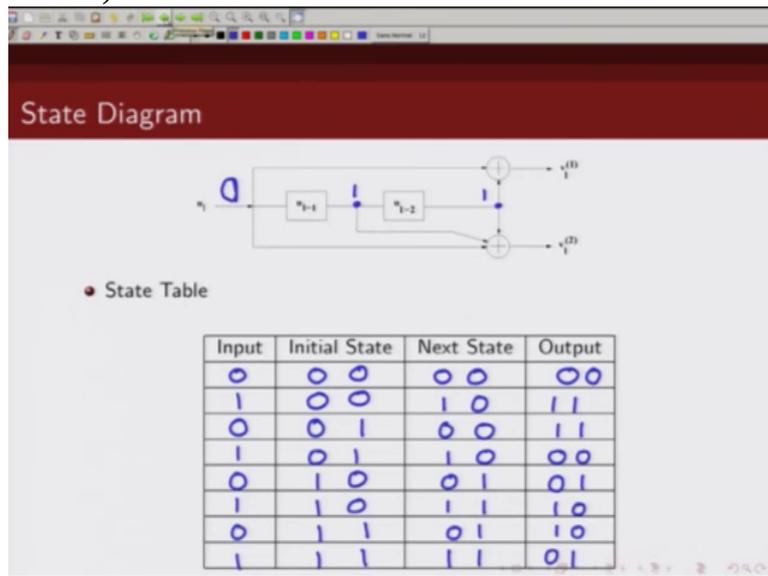
initial state 0 1, you receive a 0, you go back to all zero state and output is 1 1.

(Refer Slide Time 13:20)



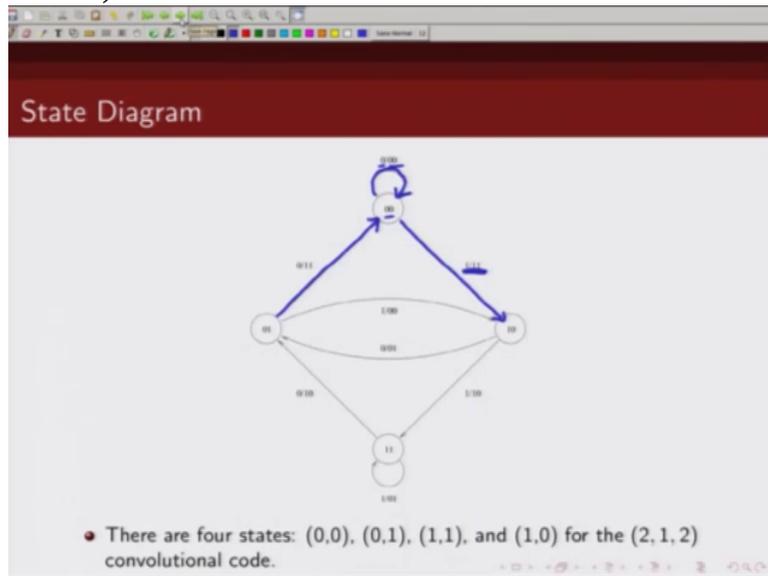
So this taken care of,

(Refer Slide Time 13:22)



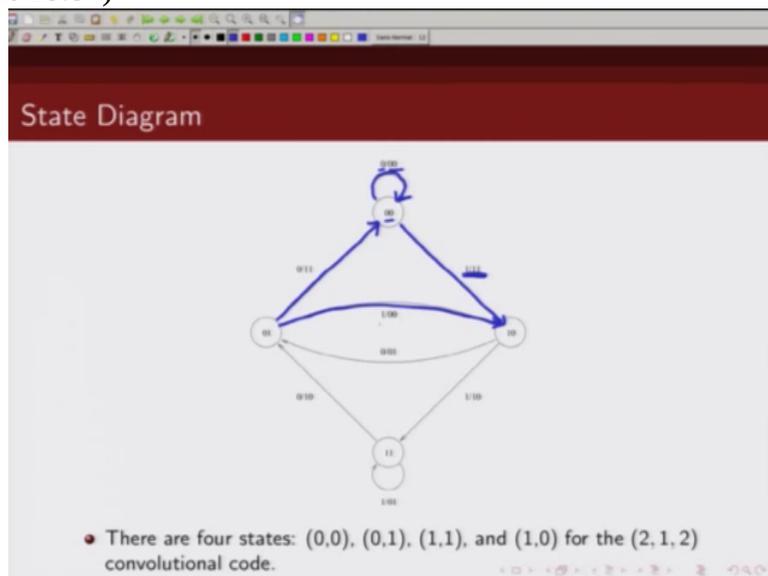
this is taken care of. If you are in initial state 0 1, you receive a 1, you move to 1 0 state and output is 0 0.

(Refer Slide Time 13:32)



You are in 0 1, you receive a 1, you move to state 1 0, input is

(Refer Slide Time 13:37)



1, output is 0 0. This

(Refer Slide Time 13:42)

State Diagram

• State Table

Input	Initial State	Next State	Output
0	0 0	0 0	0 0
1	0 0	1 0	1 1
0	0 1	0 0	1 1
1	0 1	1 0	0 0
0	1 0	0 1	0 1
1	1 0	1 1	1 0
0	1 1	0 1	1 0
1	1 1	1 1	0 1

is done, state 1 0 initial input is 0. You move to state 0 1 and output is 0 1. So you are in state

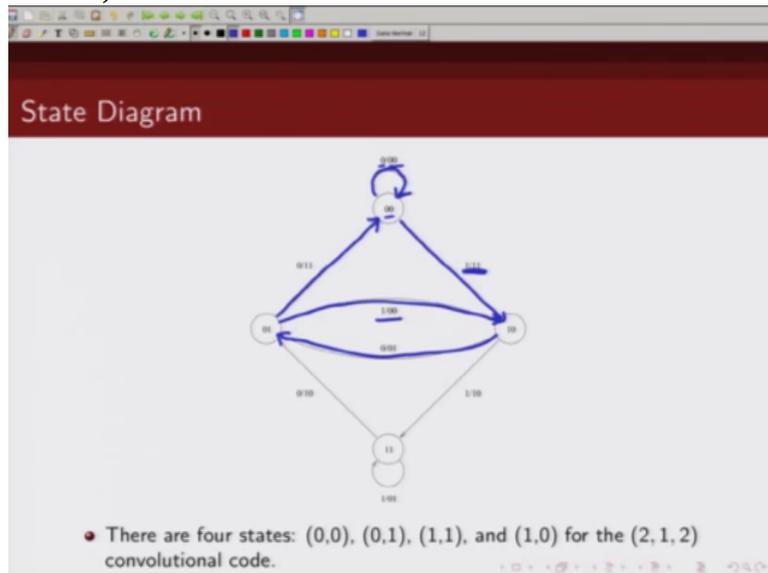
(Refer Slide Time 13:56)

State Diagram

• There are four states: (0,0), (0,1), (1,1), and (1,0) for the (2,1,2) convolutional code.

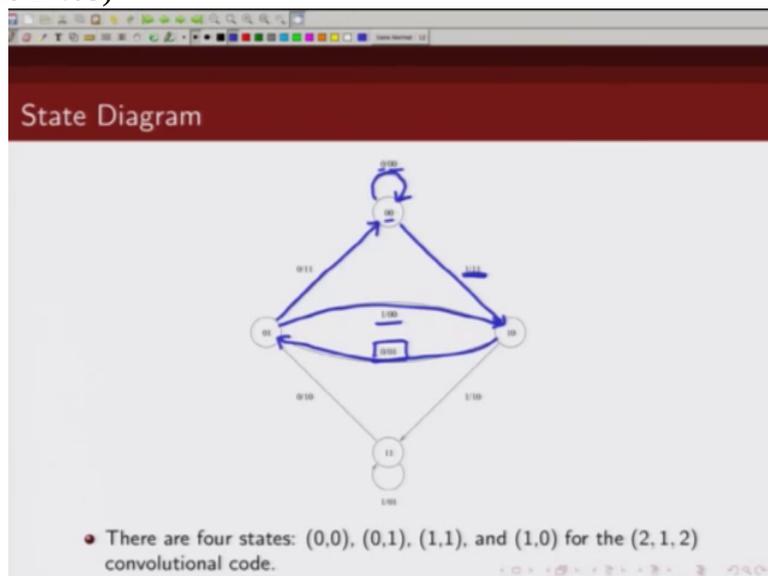
1 0, you receive a 0, you move to state 0 1 and output is

(Refer Slide Time 14:01)



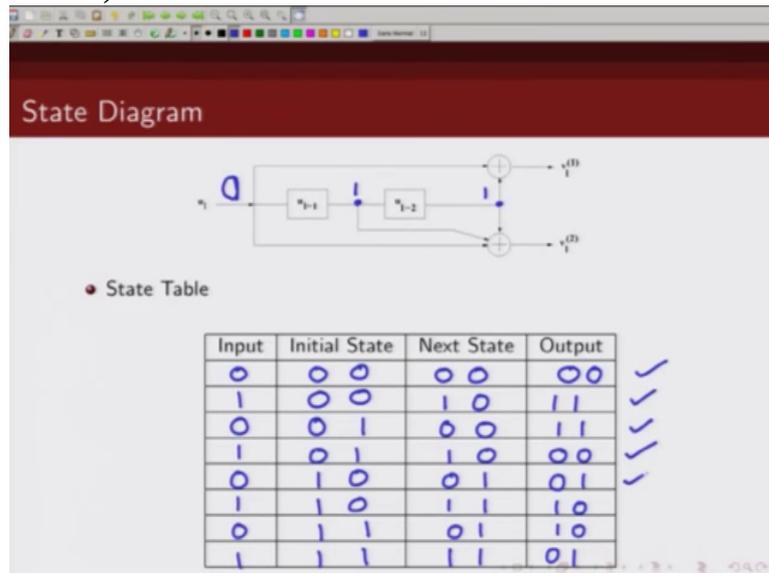
0 1. So that's this, Ok

(Refer Slide Time 14:05)



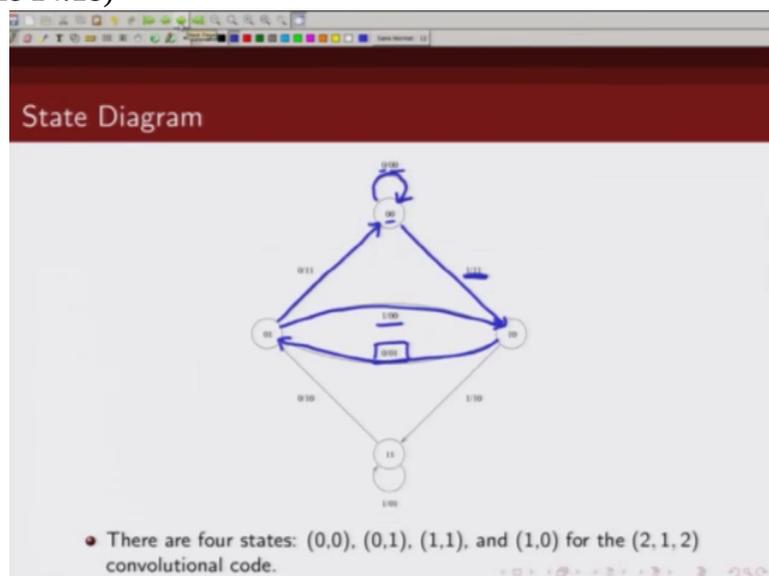
and so this is taken care of

(Refer Slide Time 14:08)



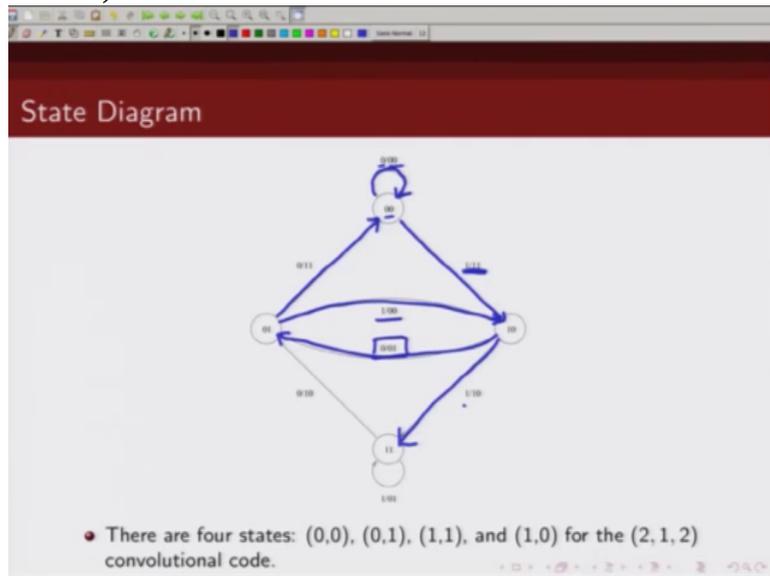
and what happens when you receive a 1 and you are in state 1 0, you move to next state 1 1 and output is 1 0. So you

(Refer Slide Time 14:18)



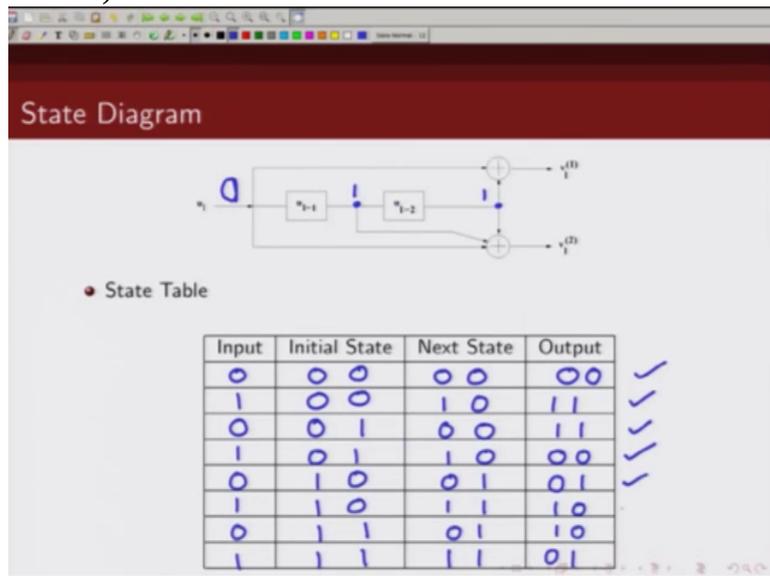
receive a 1, when you are in initial state 1 0, you move to state 1 1 which is this, this is the input

(Refer Slide Time 14:26)



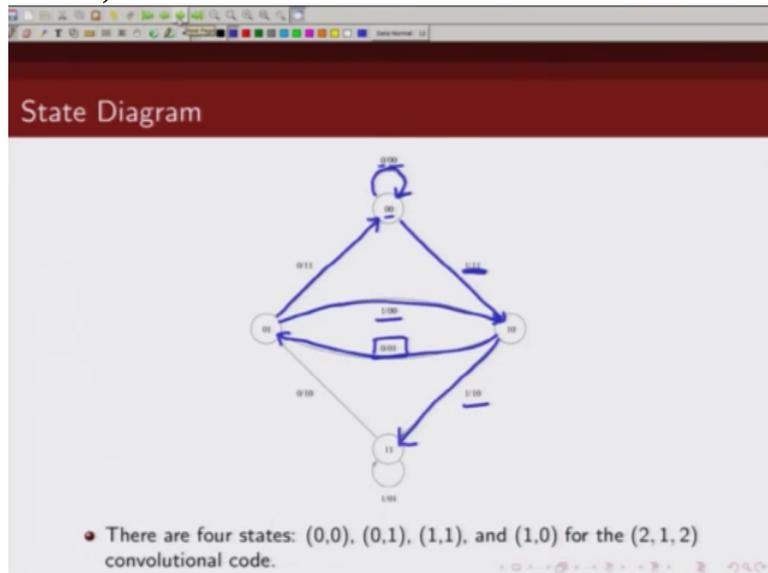
and corresponding output is 1 0. And

(Refer Slide Time 14:31)



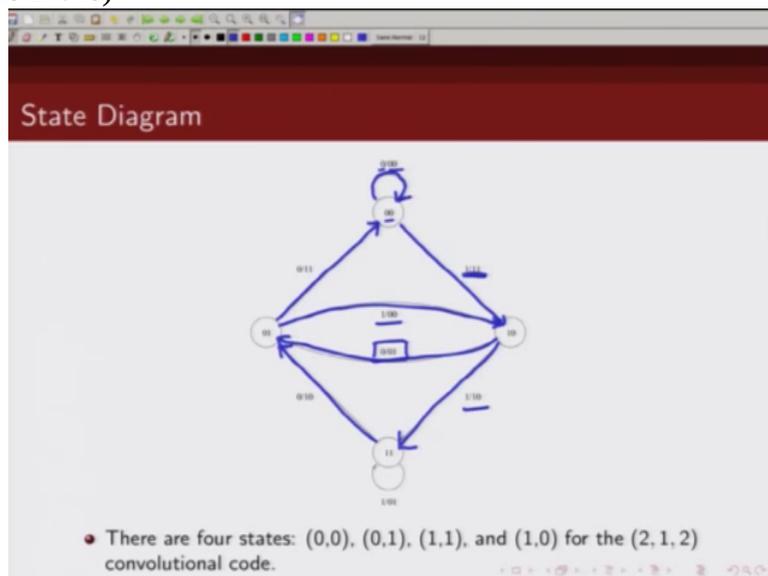
finally when you are in state 1 1, if you receive a 0, you move to state 0 1 and output is 1 0. So you are in state 1 1,

(Refer Slide Time 14:41)



you receive a 0, you move to state 0 1, input is

(Refer Slide Time 14:46)



0, output is 1 0.

(Refer Slide Time 14:51)

State Diagram

State Table

Input	Initial State	Next State	Output
0	00	00	00
1	00	10	11
0	01	00	11
1	01	10	00
0	10	01	01
1	10	11	10
0	11	01	10
1	11	11	01

And finally state 1 1, input 1, you remain in state 1 1 and output is 0 1. You receive

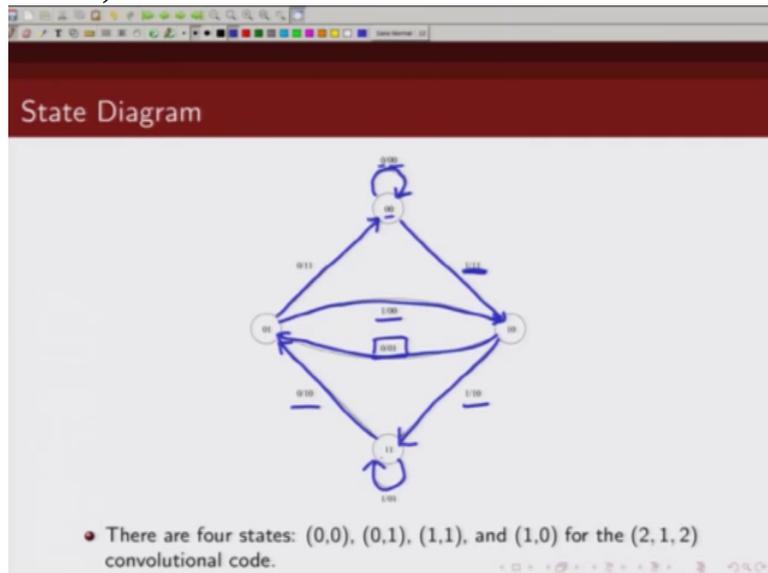
(Refer Slide Time 15:01)

State Diagram

There are four states: (0,0), (0,1), (1,1), and (1,0) for the (2,1,2) convolutional code.

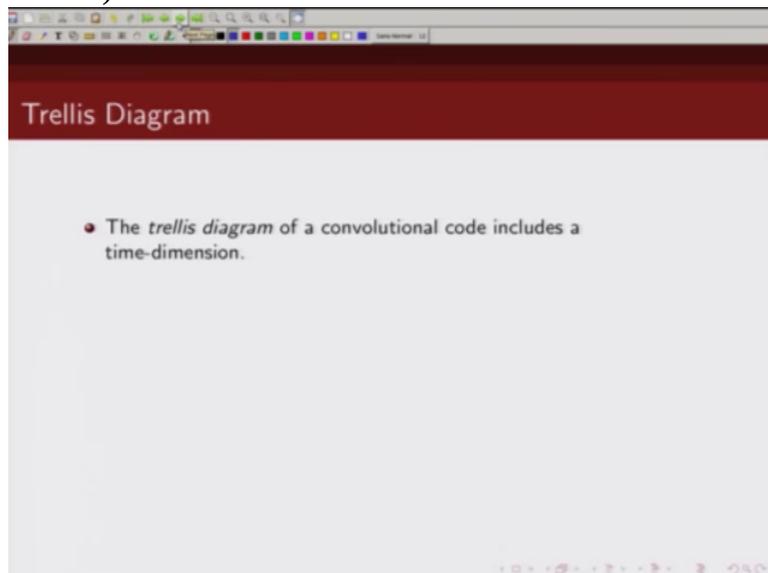
a 1, you remain in state 1 1,

(Refer Slide Time 15:05)



input is 1 and output is 0 1. So note that here I have graphically represented all possible state transitions corresponding to the convolutional encoder that we have considered. And this is my state diagram, Ok.

(Refer Slide Time 15:26)



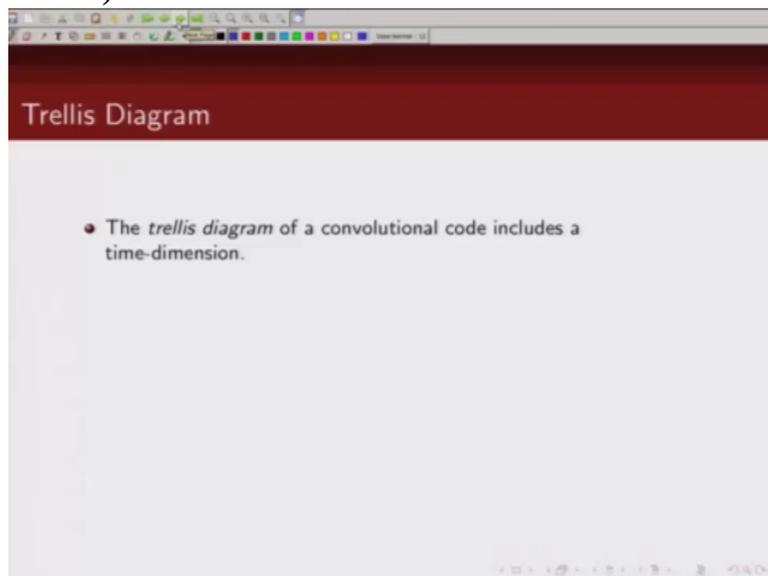
So next, let's talk about Trellis

(Refer Slide Time 15:28)



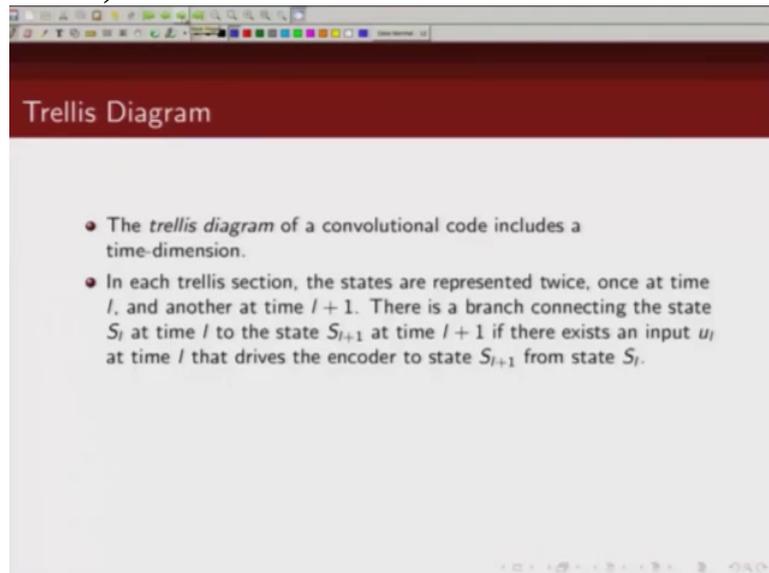
diagram. So in Trellis diagram we include another dimension which is the time dimension and how do we show

(Refer Slide Time 15:37)



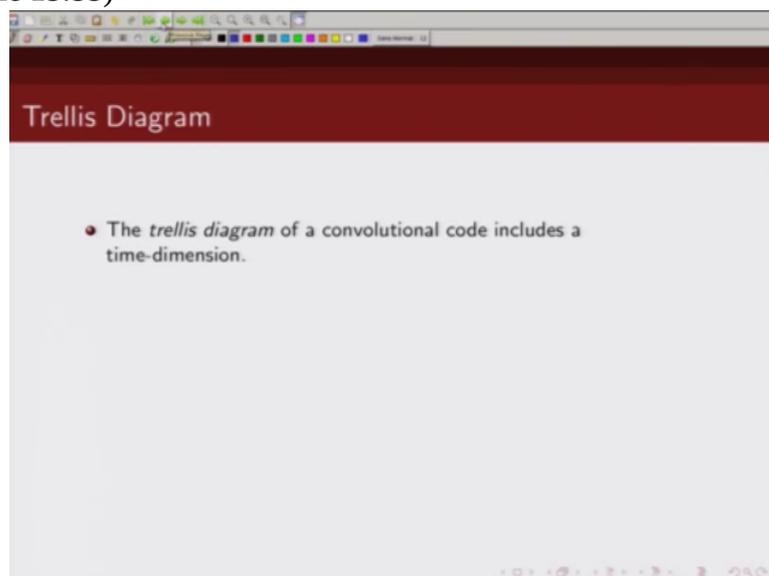
that

(Refer Slide Time 15:38)



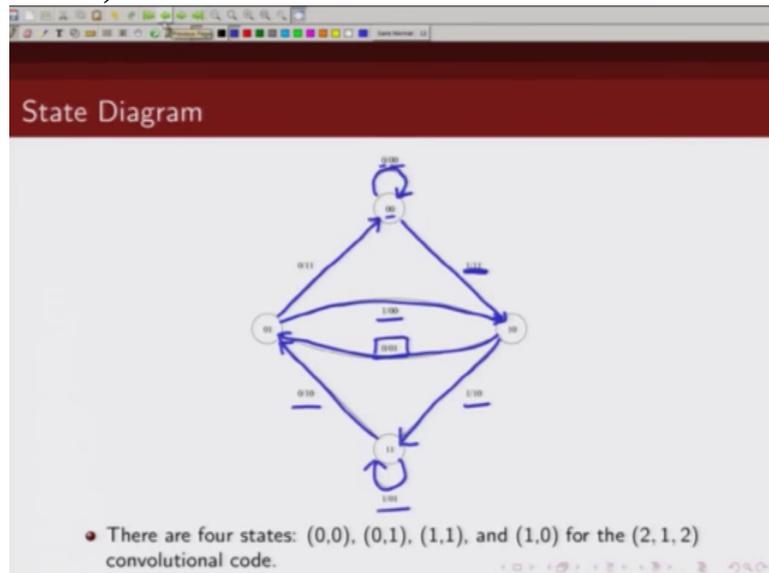
time dimension? So let us first talk about what is a Trellis section? So in a Trellis section we represent each state twice. How do we represent each state? Let us take this example.

(Refer Slide Time 15:55)



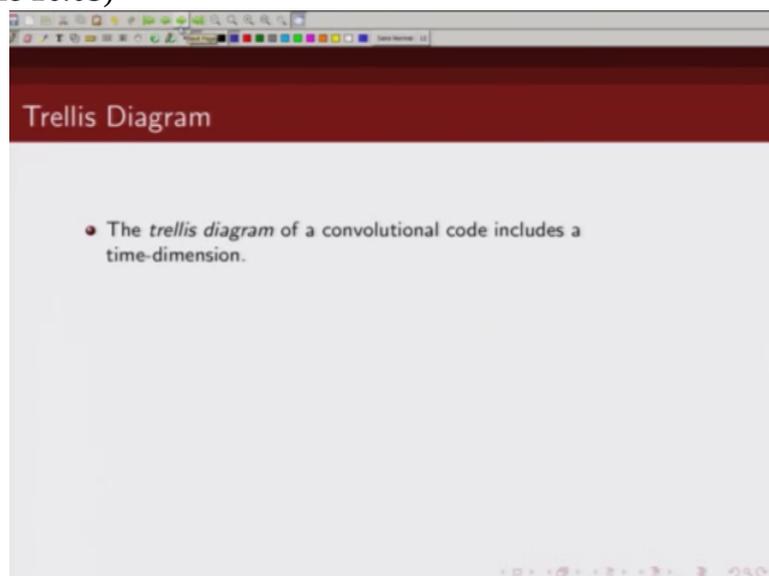
So here we have 4 states, right?

(Refer Slide Time 15:57)



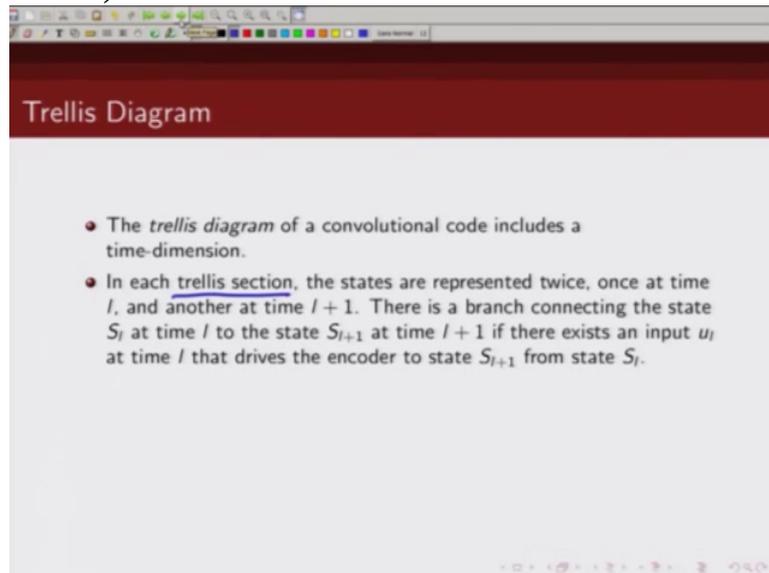
So if I have to draw a Trellis section corresponding to this encoder how will I do it? What I am going to do is,

(Refer Slide Time 16:05)



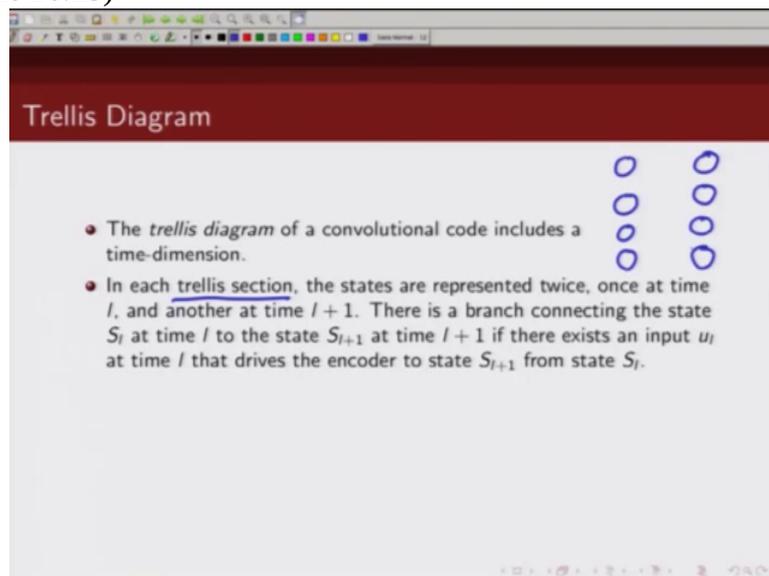
I am going to write

(Refer Slide Time 16:06)



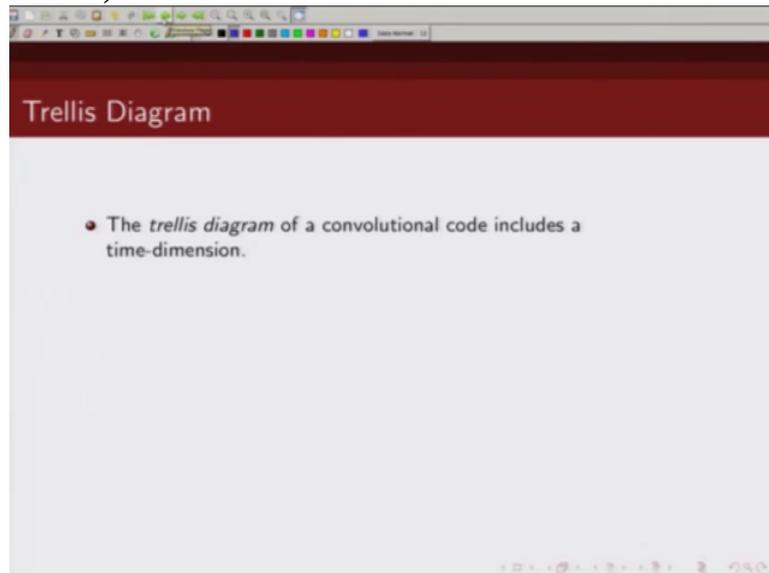
all these 4 states here and I will write all those 4 states here. Next

(Refer Slide Time 16:15)



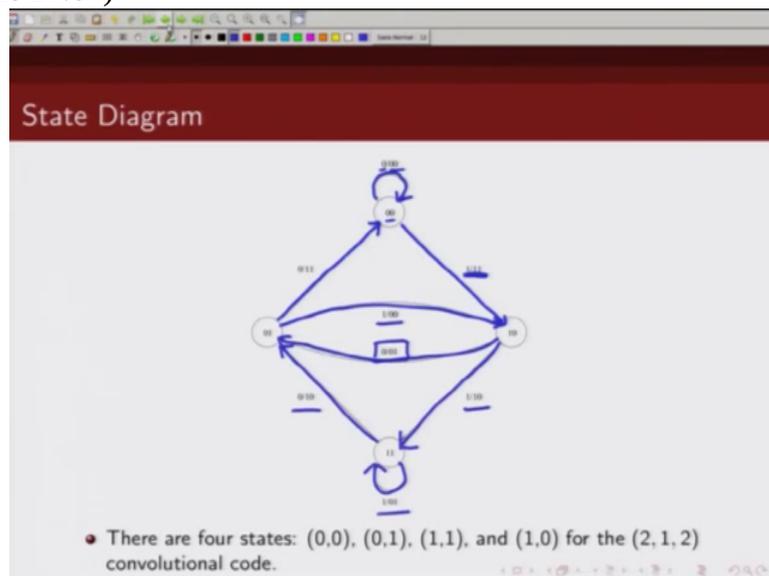
what I am going to do, I am going to add a branch connecting a state at this time instance, at time  $l$  to a state at time  $l$  plus 1, if there exists an input that will drive the encoder to state  $S_{l+1}$  from state  $S_l$ . So what we do is we, at time let's say  $l$ , these are the states corresponding to the time  $l$  and these are the states corresponding to time  $l$  plus 1. So there will be a valid transition from a state at time  $l$  to a state at time  $l$  plus 1 only if there exists an input that will drive from this state to some another state in the next time instance. For example, let's go back

(Refer Slide Time 17:06)



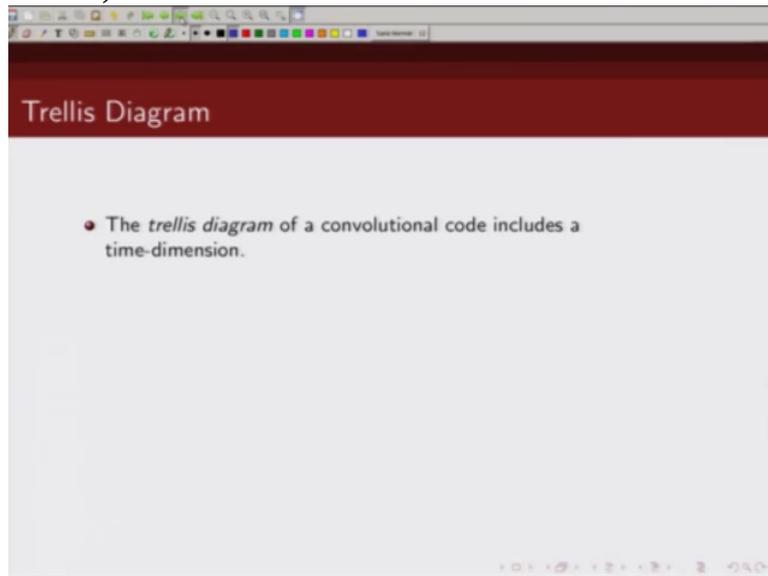
to this.

(Refer Slide Time 17:07)



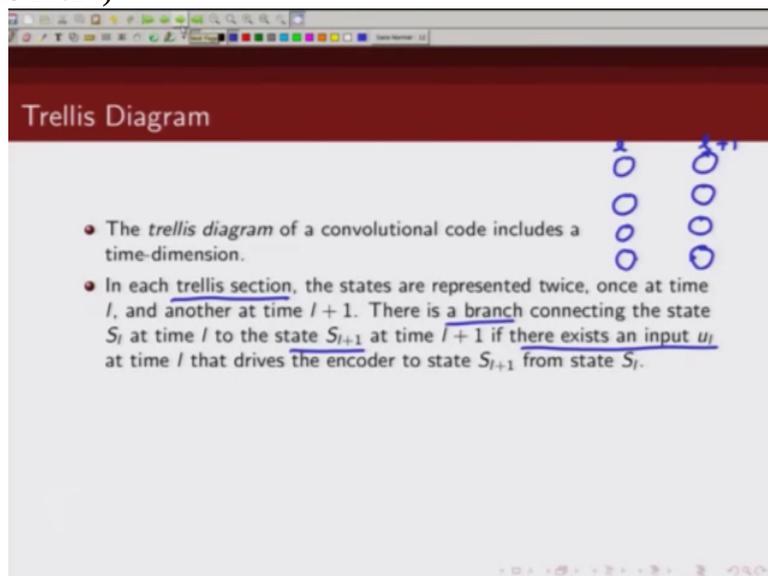
If you are in state 0 and you get a 0, you remain in all zero state. So this can be represented here.

(Refer Slide Time 17:16)



. Let's say

(Refer Slide Time 17:17)



this is my all zero state. This is 0 0 state and this is also my 0 0 state so there is a transition from here to here and what is that

(Refer Slide Time 17:30)

**Trellis Diagram**

- The *trellis diagram* of a convolutional code includes a time-dimension.
- In each trellis section, the states are represented twice, once at time  $l$ , and another at time  $l + 1$ . There is a branch connecting the state  $S_j$  at time  $l$  to the state  $S_{j+1}$  at time  $l + 1$  if there exists an input  $u_l$  at time  $l$  that drives the encoder to state  $S_{j+1}$  from state  $S_j$ .

input that causes this transition? That input is 0 and what is the output corresponding to that? It is 0 0. So I

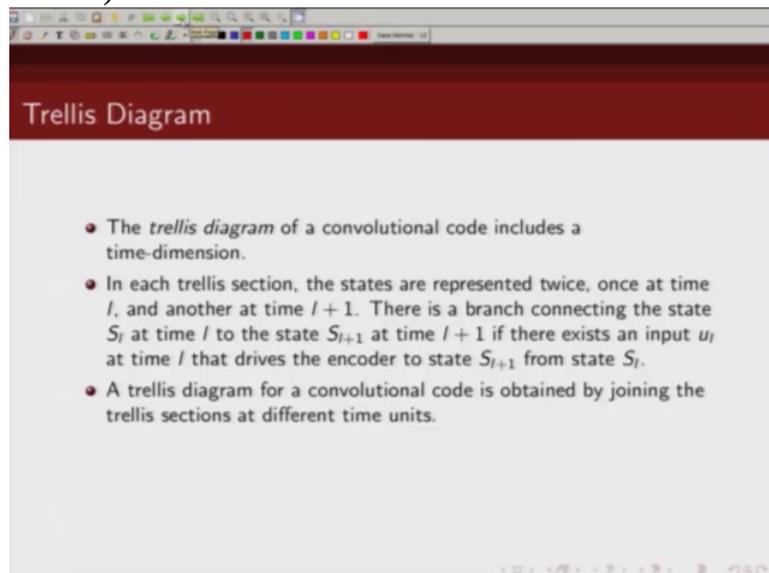
(Refer Slide Time 17:40)

**Trellis Diagram**

- The *trellis diagram* of a convolutional code includes a time-dimension.
- In each trellis section, the states are represented twice, once at time  $l$ , and another at time  $l + 1$ . There is a branch connecting the state  $S_j$  at time  $l$  to the state  $S_{j+1}$  at time  $l + 1$  if there exists an input  $u_l$  at time  $l$  that drives the encoder to state  $S_{j+1}$  from state  $S_j$ .

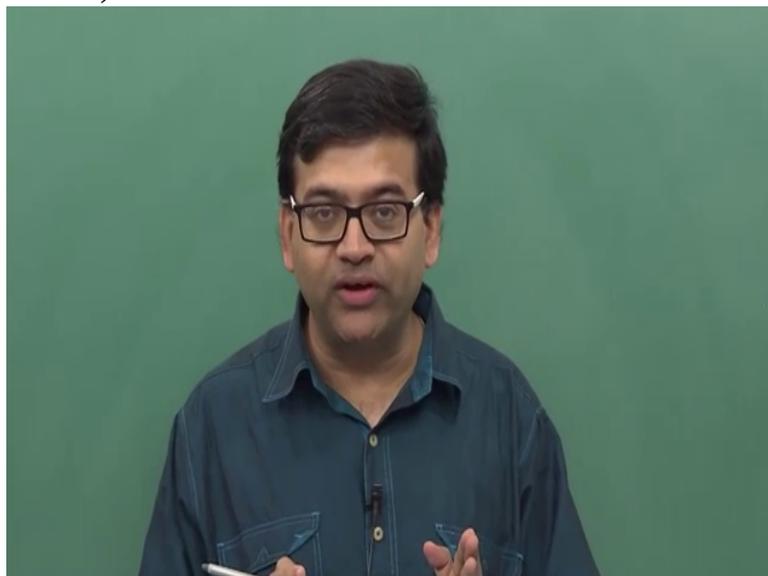
will label all those transitions which are happening from state at time  $S_l$ , I mean state at time  $l$  to any state at time  $S_l + 1$ , provided there is a valid transition, Ok. And

(Refer Slide Time 18:00)



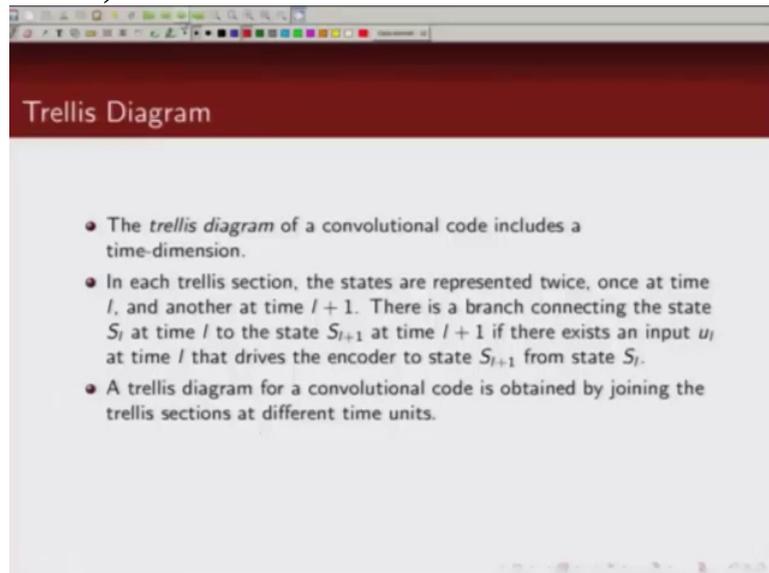
a Trellis diagram is obtained by joining Trellis section at different time instances. So what I am going to do is,

(Refer Slide Time 18:07)



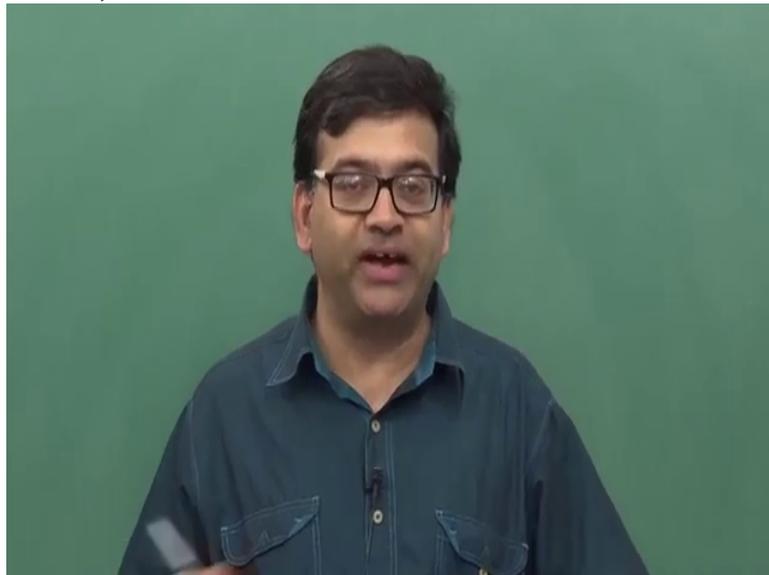
to draw the complete state diagram I am going to join Trellis section at time  $t$ . For various times I will join them together and that's how my Trellis section will be, Trellis diagram will be drawn.

(Refer Slide Time 18:21)



The interesting thing is in a Trellis then we can see all possible codewords. Any valid

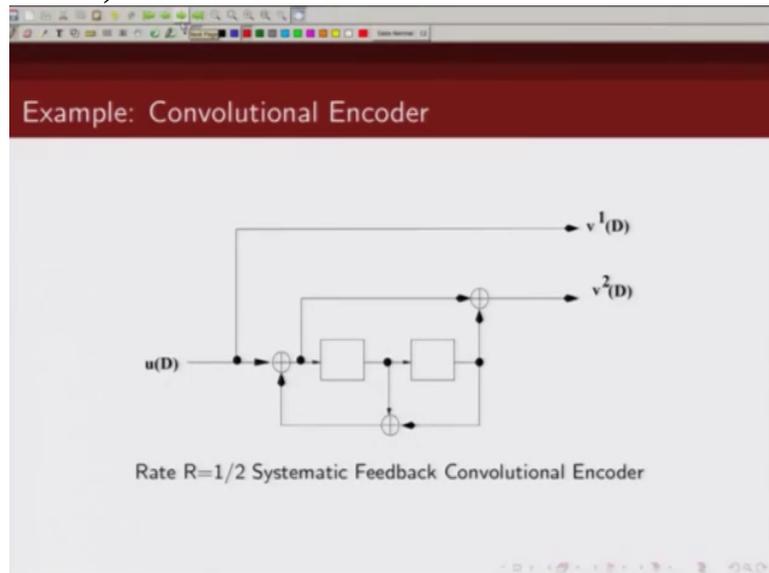
(Refer Slide Time 18:29)



transition basically is a valid codeword so let's say initially encoder is in all zero state. And it goes back to all zero state. Then all possible paths from all zero state to the final all zero state will enumerate all possible codewords that we have.

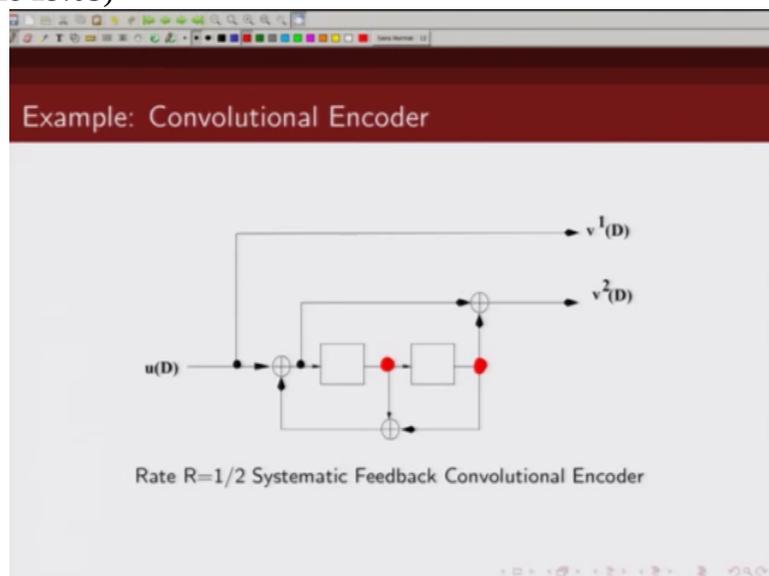
So let us take another example.

(Refer Slide Time 18:49)



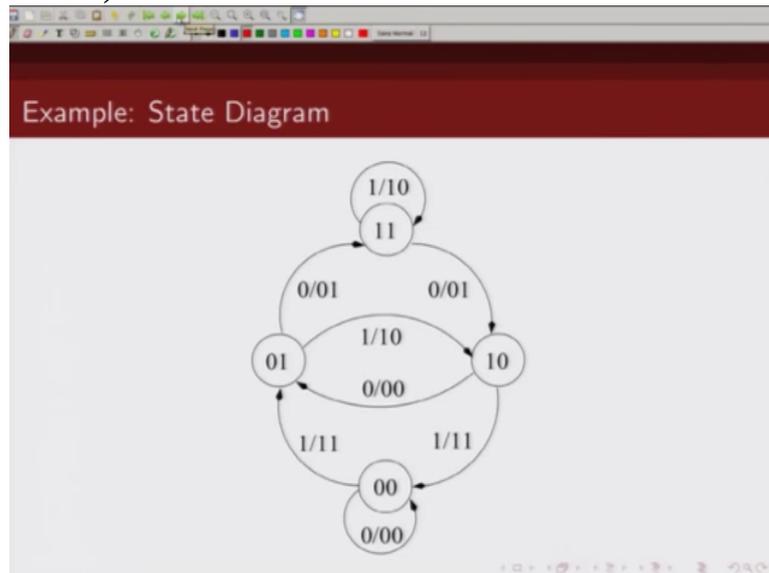
This is an encoder. Now following the same procedure as we explained earlier, we can draw the state diagram for this. So where are the states? This is one state, this is another state. And similarly we can find out for all

(Refer Slide Time 19:05)



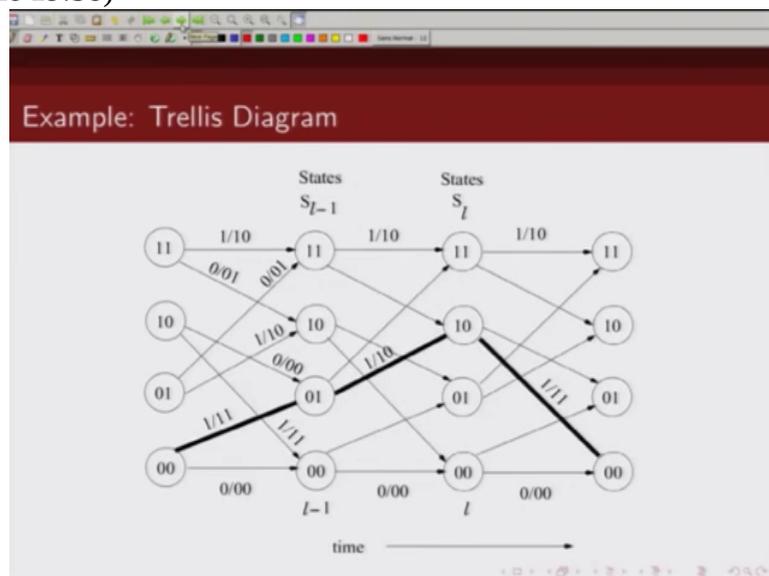
possible initial state and all possible inputs what are the next state and the corresponding output. So this, in this case

(Refer Slide Time 19:17)



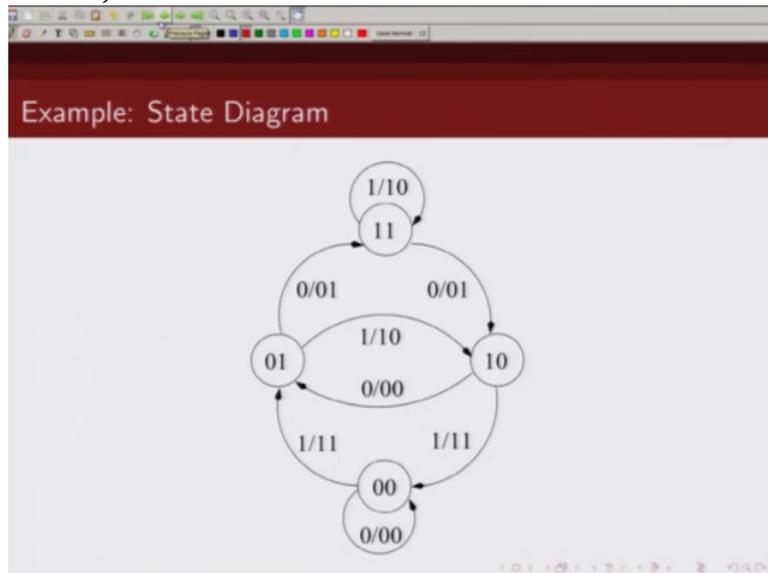
turns out to be this. So when you are all zero state if you get a zero, the output is 0 0 and you remain in all zero state but if you get a 1, your output is 1 1 and you move to state 0 1. So this is the state diagram for the convolutional encoder I had just shown. Now how

(Refer Slide Time 19:38)



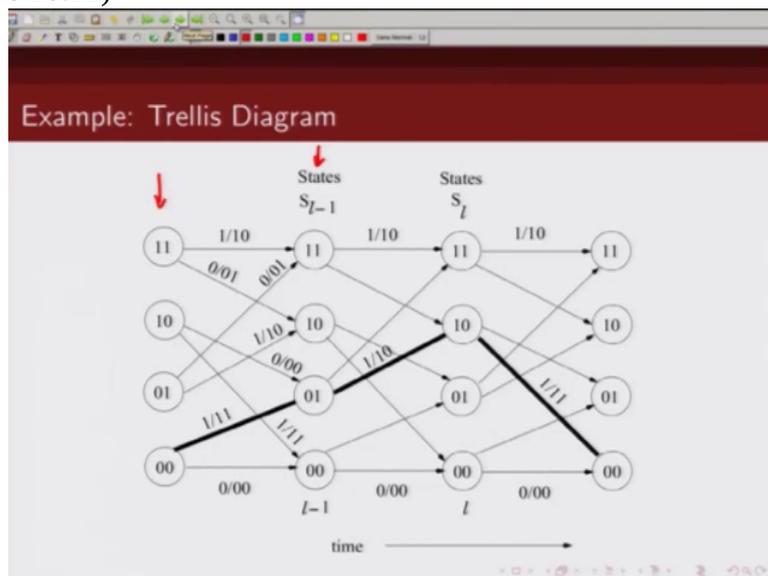
do we draw the Trellis diagram for this convolutional encoder? So as we said, at each time instance, let us consider this time instance, this and this, so this is one particular time instance. So what we are going to do is we are first going to enumerate all possible encoders and we write them twice. So in this example we have 4

(Refer Slide Time 20:03)



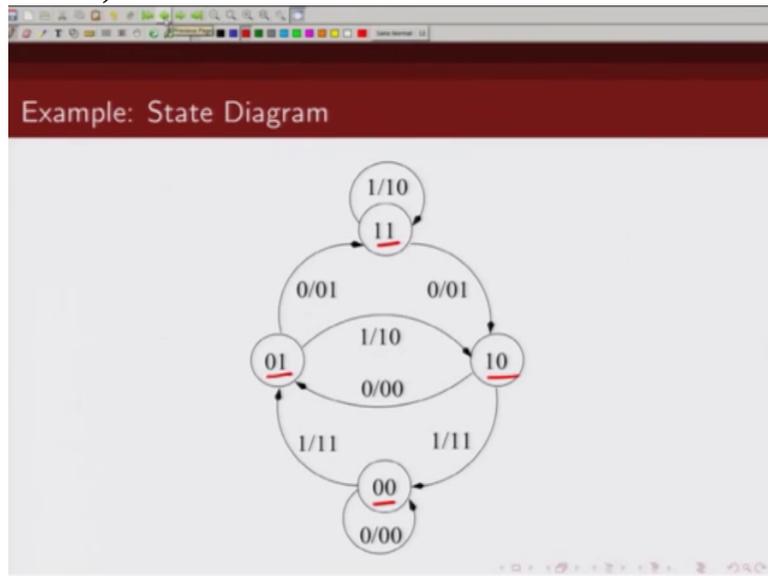
states, 0 0, 0 1, 1 0, 1 1. So what we do is we write these states

(Refer Slide Time 20:11)



twice; 0 0, 0 1, 1 0, 1 1. Next we draw state transition from each state to next state provided there is a valid transition. So what we are going to do is we are going to map all these

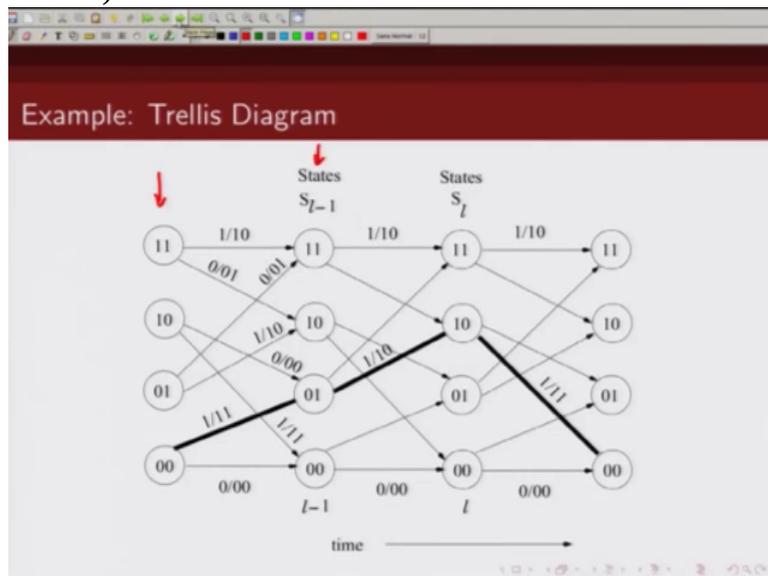
(Refer Slide Time 20:31)



state transitions into the diagram that we just showed.

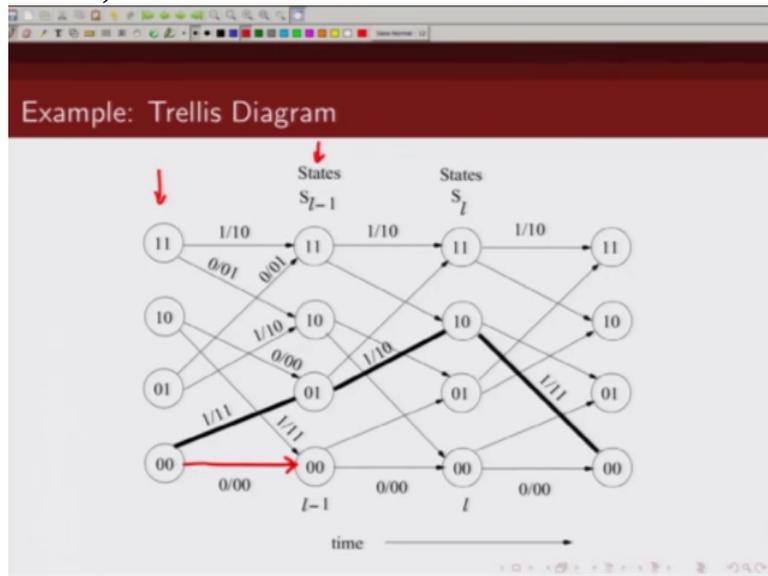
So let's take this example. So we are in 0 0 state and we get a 0. The output is 0 0 and we remain in state 0 0. This is shown by

(Refer Slide Time 20:48)



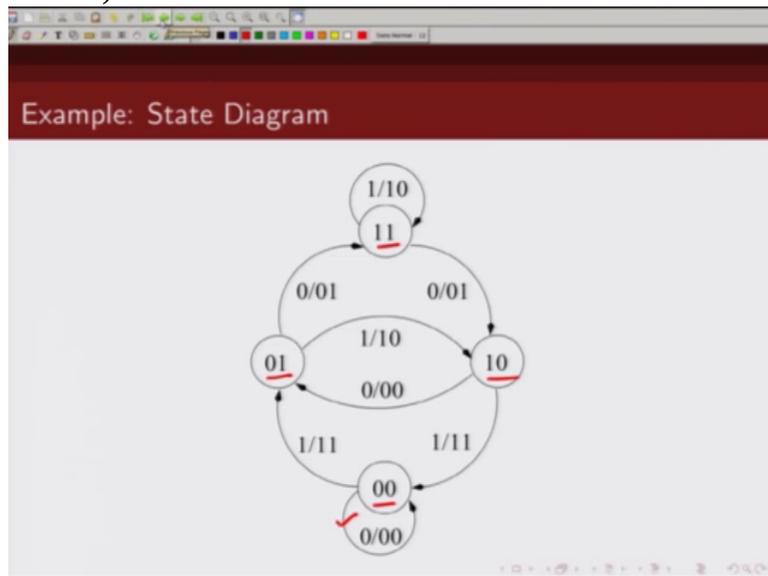
this transition. You get a 0,

(Refer Slide Time 20:52)



output is 0 0, and next state is 0 0. Now next if you

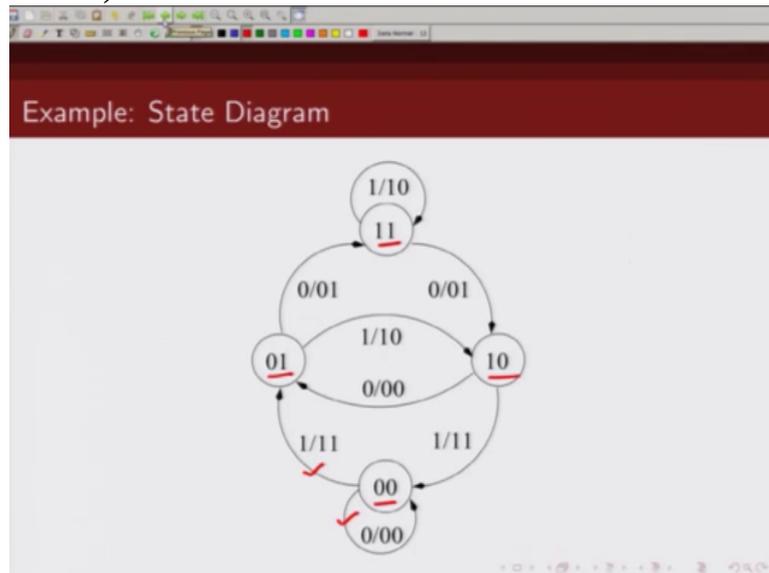
(Refer Slide Time 21:00)



get a 1, you move to state 0 1, output is 1 1. You get a 1,

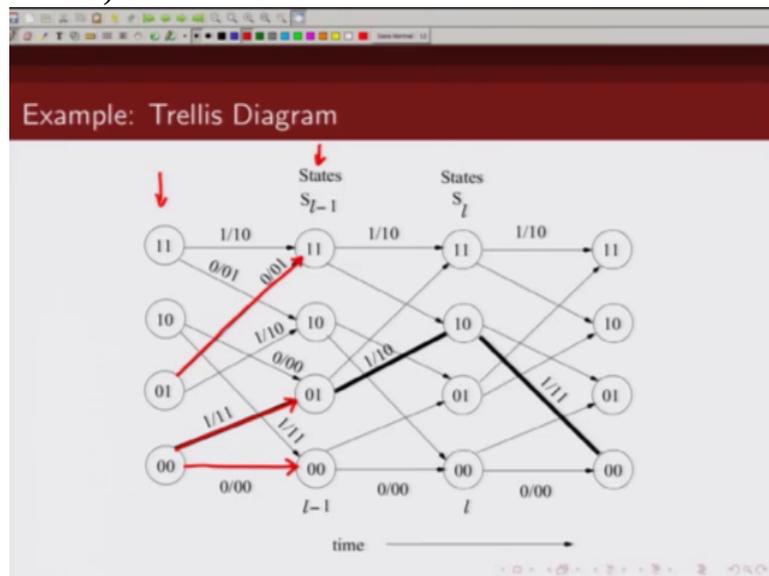


(Refer Slide Time 21:16)



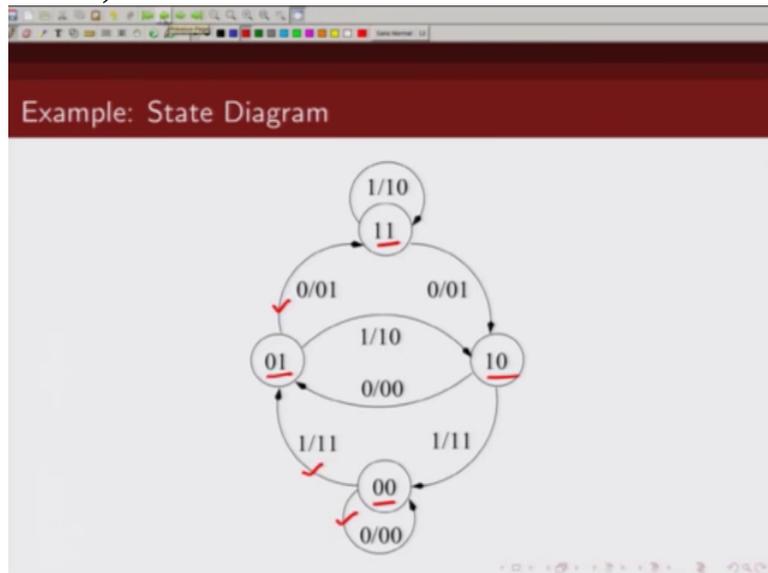
Let's say you are in state 0 1 and you receive a 0, then you move to state 1 1 and output is 0 1. So how is that denoted? You are in state 0 1. You receive a 0, you move to state 1 1, the output is

(Refer Slide Time 21:35)



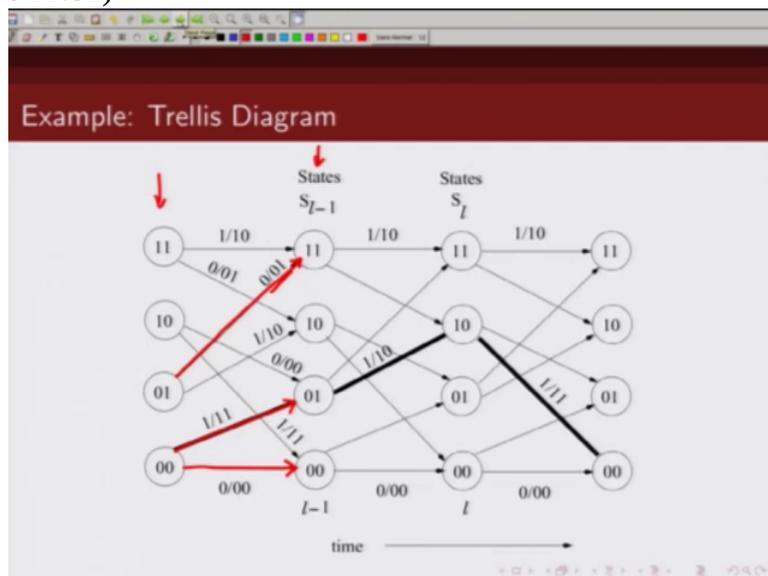
0 1. Similarly when you are in state 0 1 and you

(Refer Slide Time 21:44)



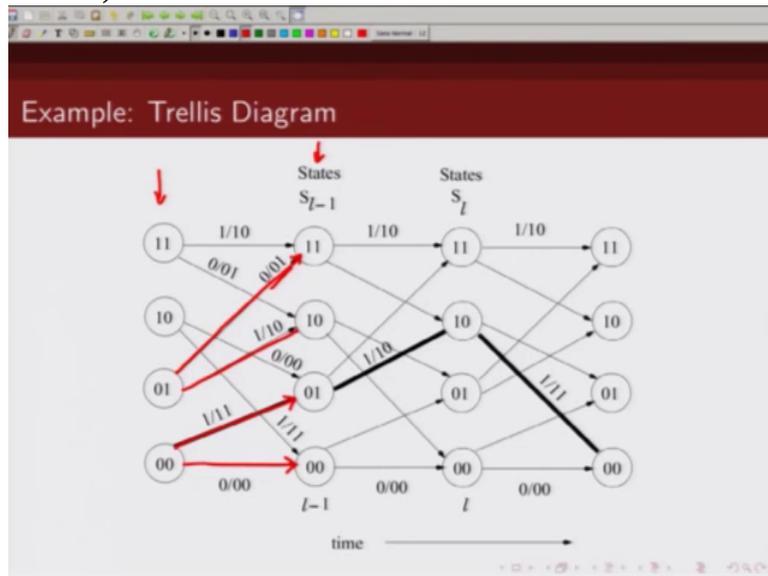
get a 1, you move to state 1 0 and the output is 1 0. So this is denoted by,

(Refer Slide Time 21:52)



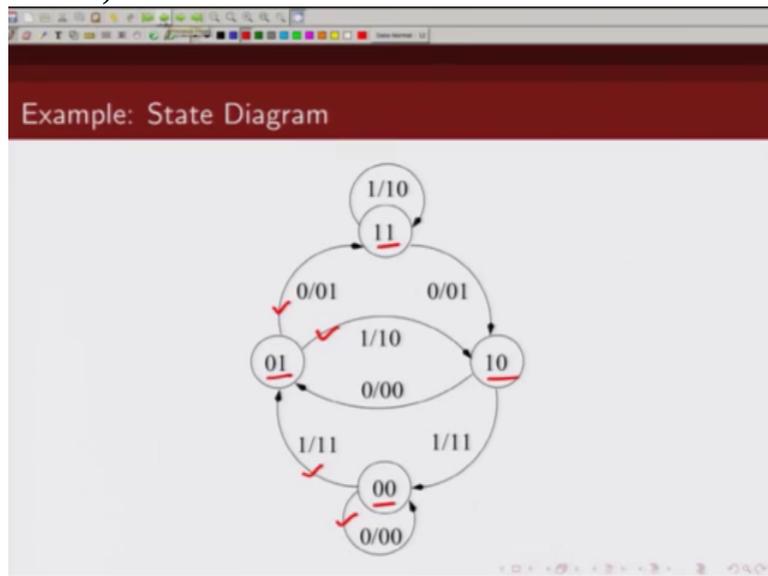
so you get a 1, you move to state 1 0 and output is

(Refer Slide Time 21:57)



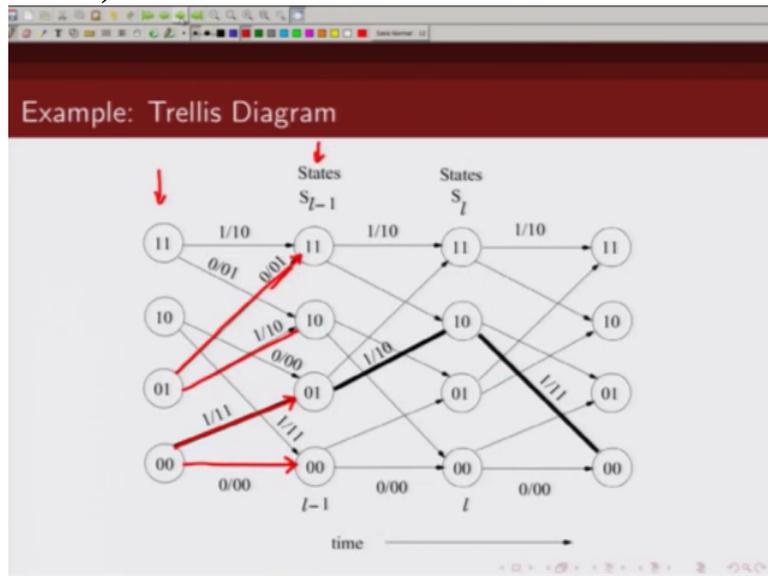
1 0. We repeat the same for other states.

(Refer Slide Time 22:01)



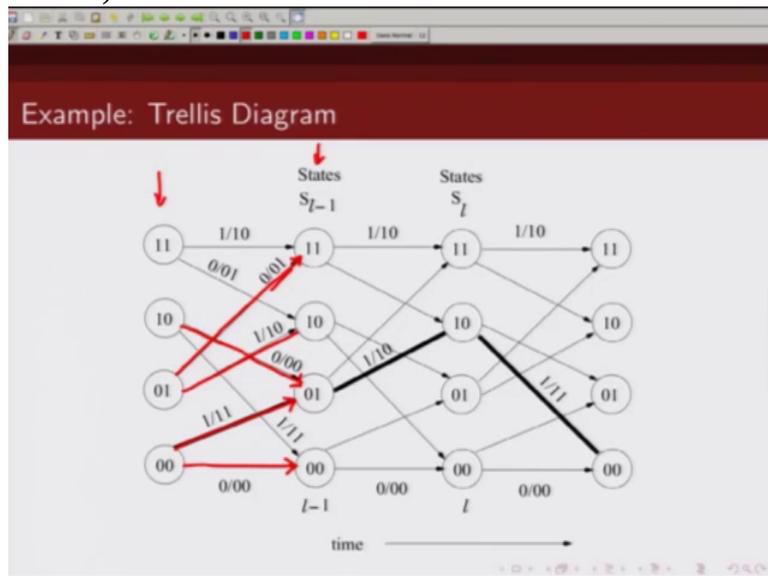
So let's say you are in state 1 0, when you get a 0, the output is 0 0 and you move to state 0 1.  
When you

(Refer Slide Time 22:11)



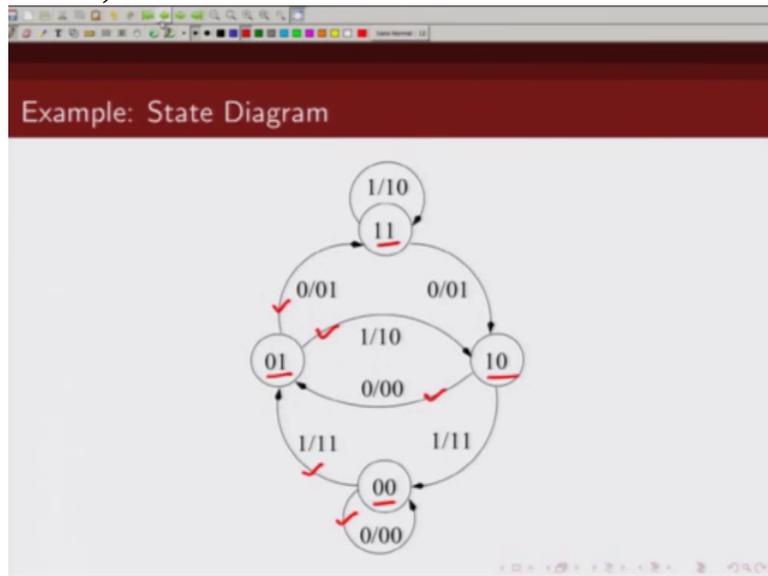
are at 1 0, you get a 0, you move to state 0 1 that is this transition.

(Refer Slide Time 22:17)



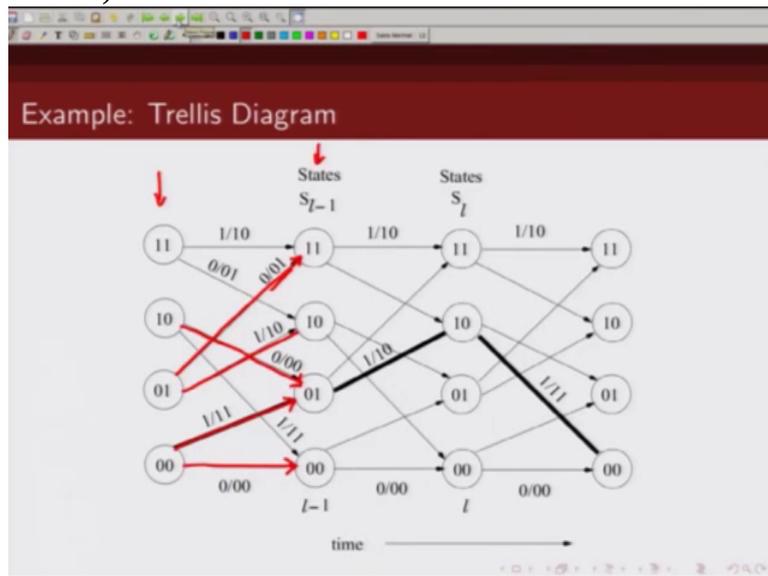
And the output is 0 0.

(Refer Slide Time 22:22)



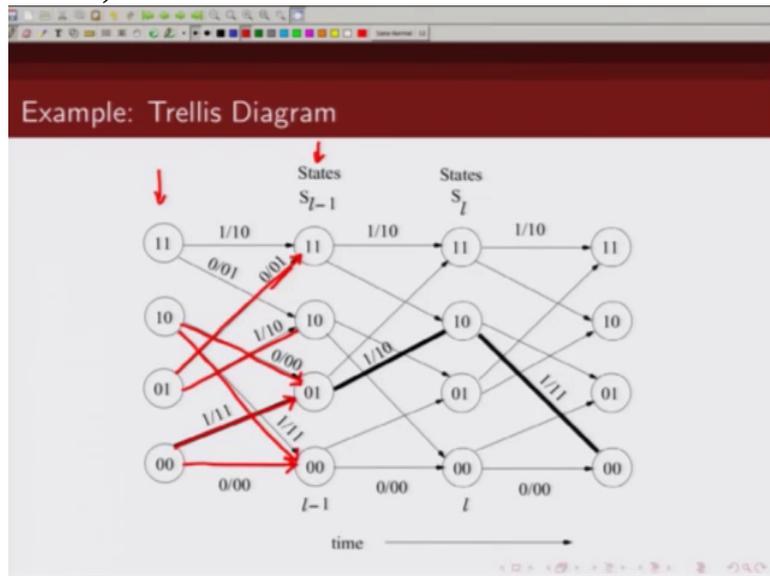
And when you get a 1, the output is 1 1, next state is 0 0. So when you are at

(Refer Slide Time 22:31)



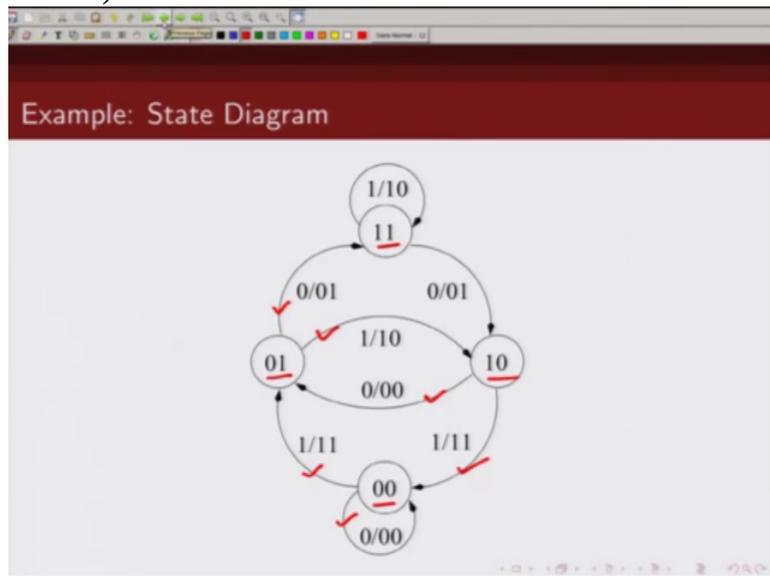
1 0 and you get a 1, then next state is 0 0 and the output

(Refer Slide Time 22:36)



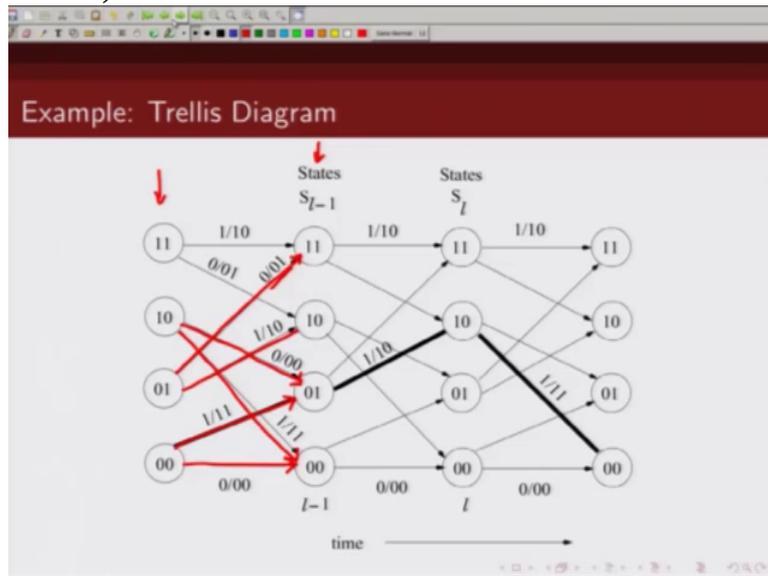
is 1 0.

(Refer Slide Time 22:40)



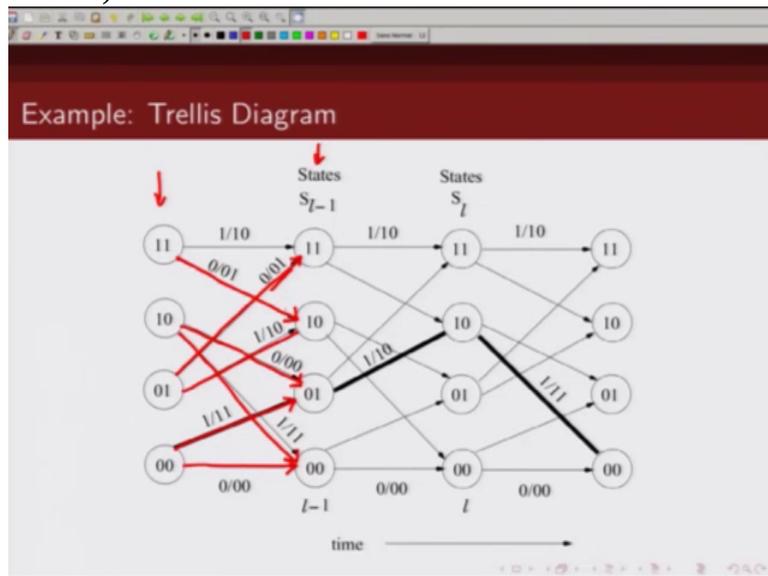
So this we have all enumerated, so only thing left is when you are in state 1 1. When you are in state 1 1, you get a 0, you move to state 1 0 and output is 0 1. So

(Refer Slide Time 22:54)



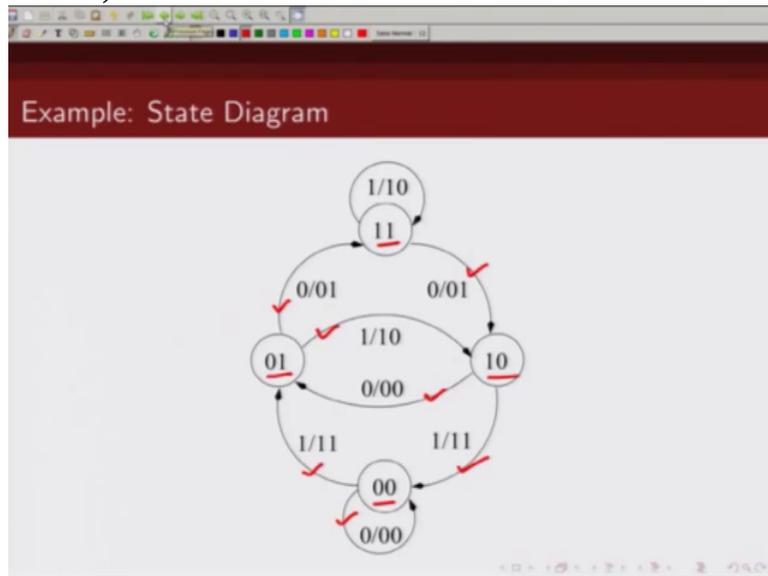
state 1 1, you get a 0, you move to state 1 0 and the output is 0 1.

(Refer Slide Time 23:01)



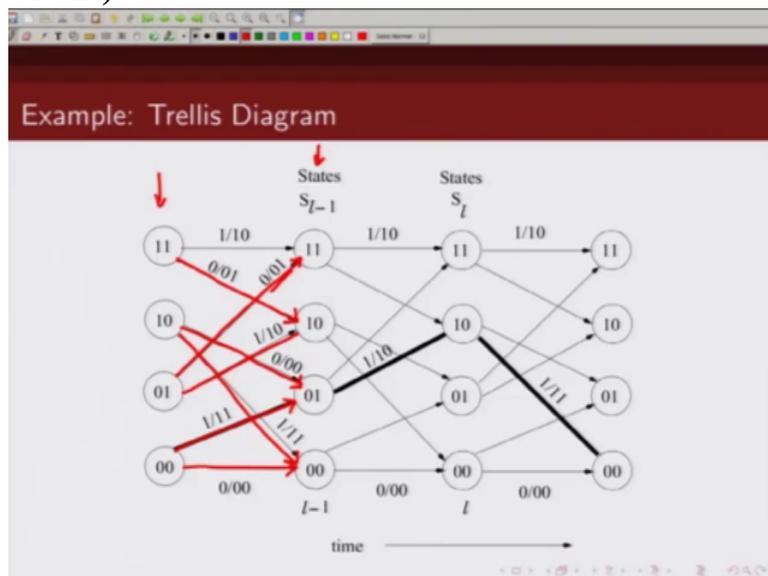
And

(Refer Slide Time 23:03)



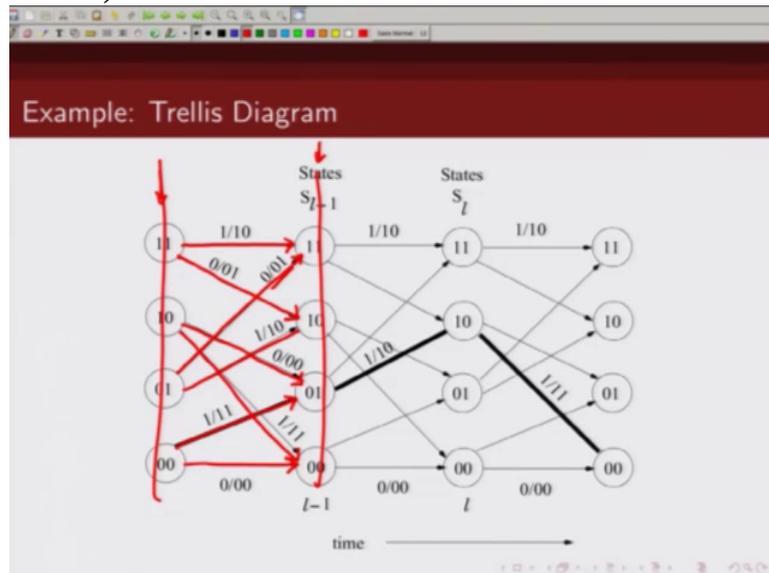
finally when you are in state 1 1 and you get a 1, you remain in state 1 1 and output is 1 0. So that's this line.

(Refer Slide Time 23:15)



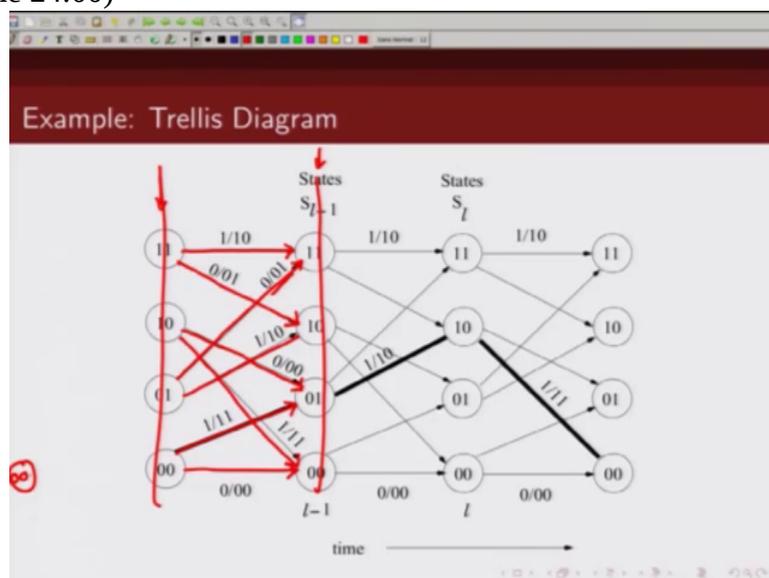
So note that here this Trellis section. I have

(Refer Slide Time 23:22)



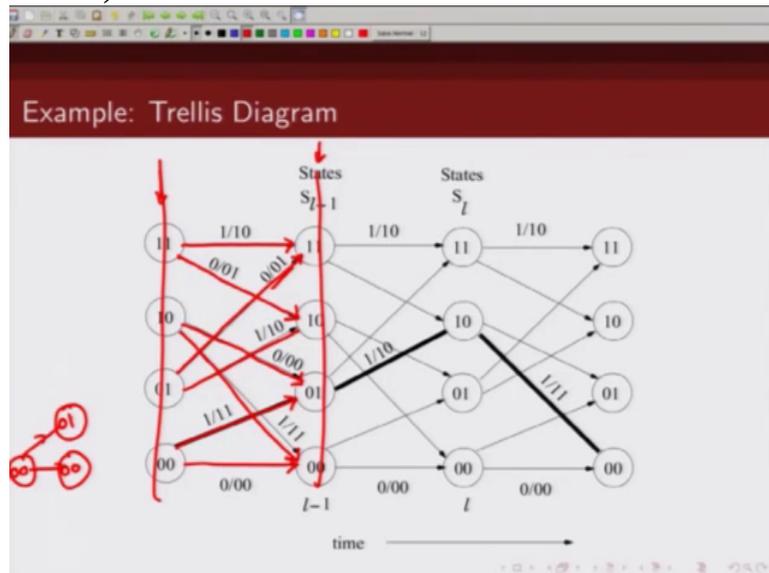
drawn the same information as I had in my state diagram; the only thing now you have this, so in Trellis diagram what you do, you just combine all these Trellis sections. So let's say this is Trellis section at time  $l$ , this is Trellis section at time  $l$  plus 1, this  $l$  plus 2, that's how we add this time dimension to the Trellis diagram. So for example, initially let's say your convolutional encoder is all zero state. So in that case what is going to happen, you are in all zero state, all zero state is 0 0. So if you get a 0

(Refer Slide Time 24:00)



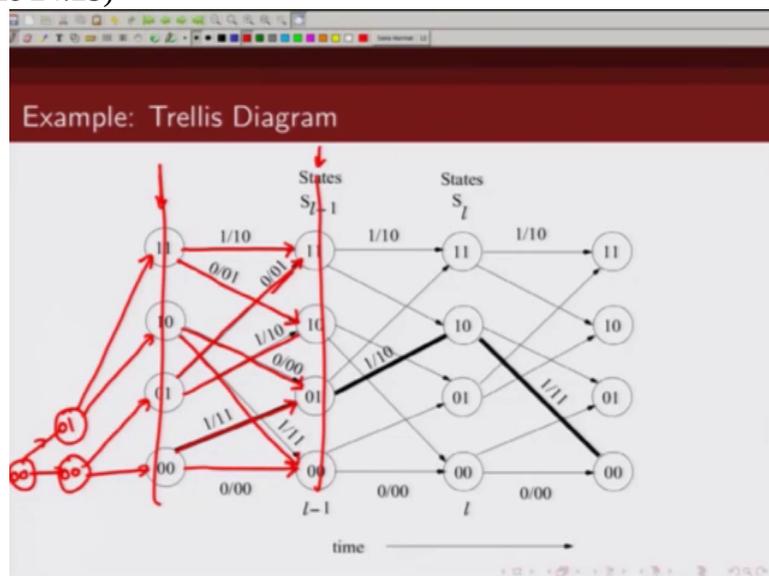
you move to 0 0, and if you get a 1, you move to 0 1. And next time instance

(Refer Slide Time 24:09)



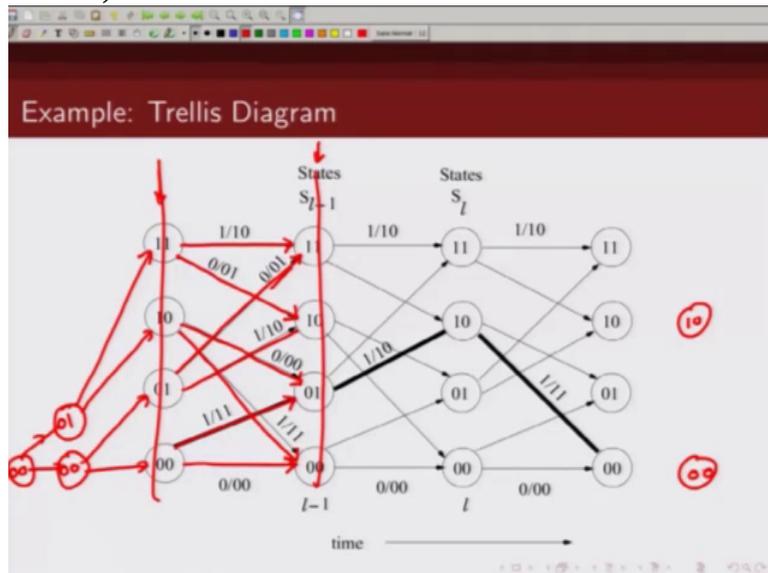
if you are in 0 0 state you move here and if you get a 1, you move here. Similarly if you are in 0 1 and you get a 0, you move to this state. And if you get a 1, you move to this state. And let's say

(Refer Slide Time 24:25)



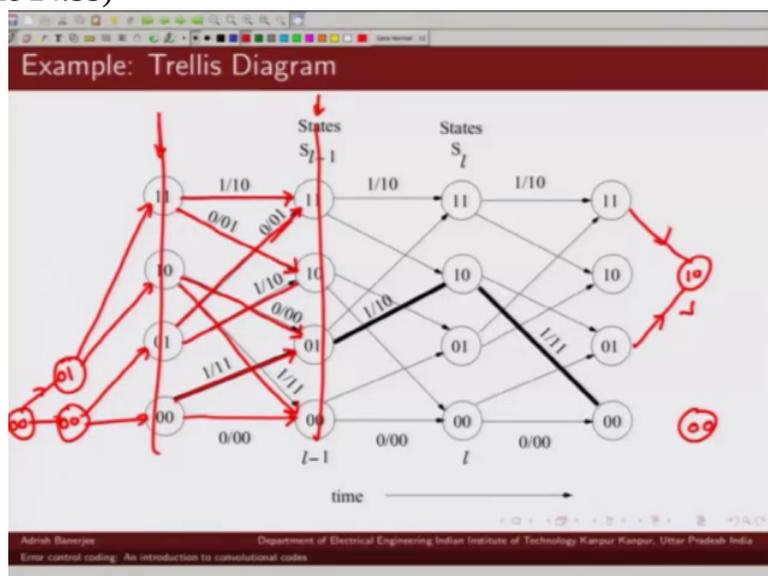
I want to terminate this encoder. I want to terminate meaning I want to bring this encoder back into all zero state. So then in this step what I am going to do is I am going to bring it back to state 0 0 and 1 0

(Refer Slide Time 24:40)



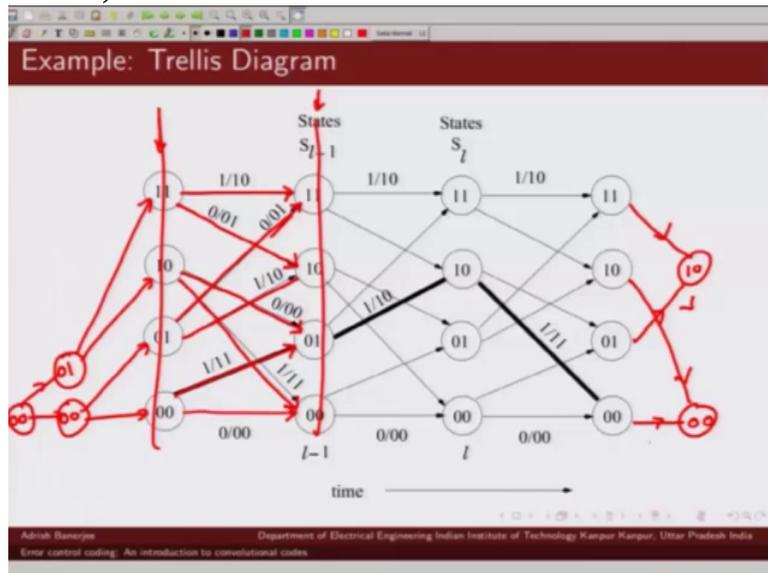
from state 1 1, if I get a 0 I move to this state, and from state 0 1, if I get a 1, I move to this state. Similarly from 1 0

(Refer Slide Time 24:55)



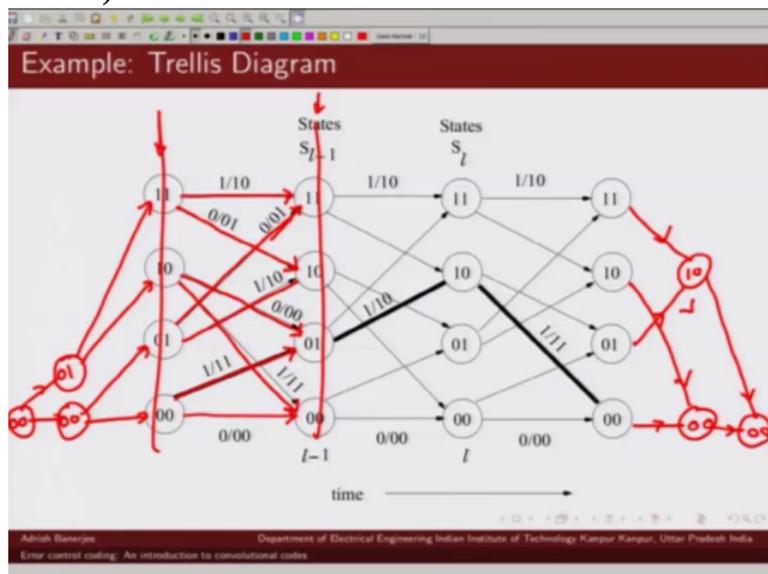
I can move to 0 0 state if I get a 1 and from 0 0 I can move to 0 0 state if I get a 0.

(Refer Slide Time 25:05)



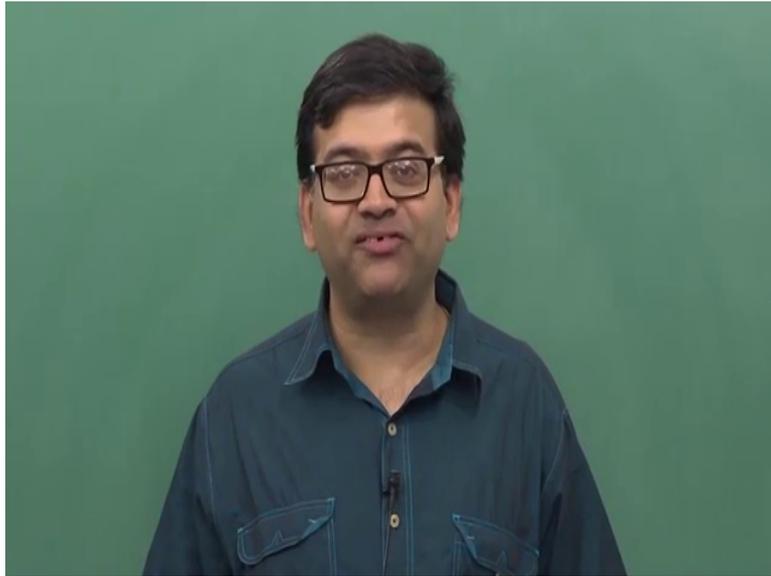
Finally from 0 0 state if I get a 0, I move to all zero state and from 1 0 I can move to zero state if I

(Refer Slide Time 25:15)



get a 1. So you can see how the states are evolving with time; that I can see in the Trellis diagram. With this I will end my discussion on state diagram

(Refer Slide Time 25:29)



and Trellis diagram, thank you.