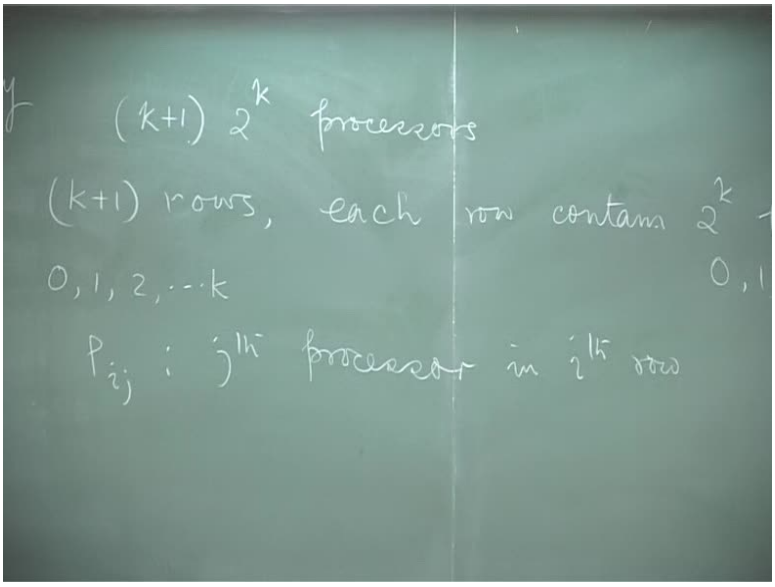
Parallel Algorithms
Prof. Phalguni Gupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture - 5

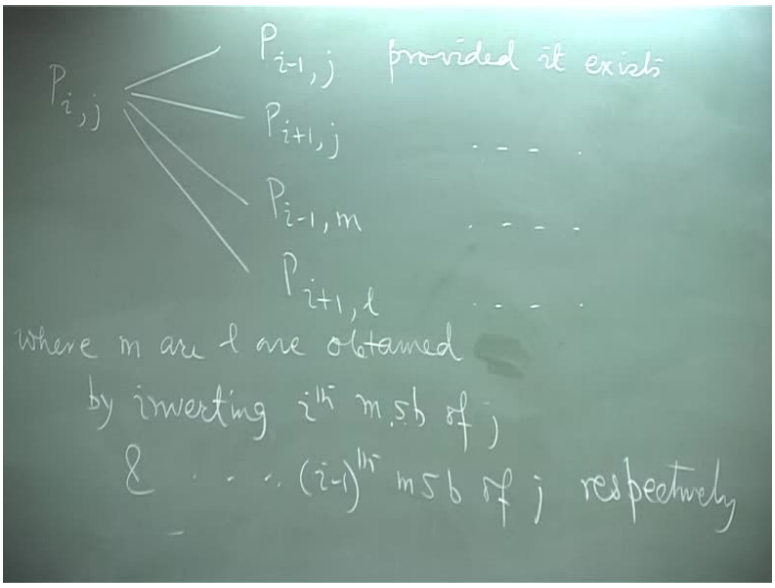
In the last class we discuss about the model sub share memory and then we also discussed all the next connected computer and (( )).

(Refer Slide Time: 00:42)



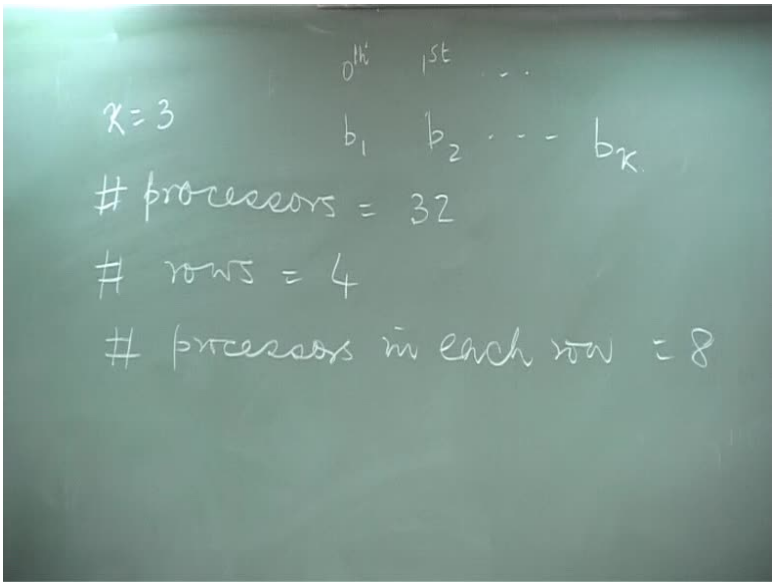
Today the first model what we will be discussing between butterfly model. It consist of k plus 1 2 to the power k processors and this processor divided into k plus 1 rows and each row contains 2 to the power k processors. Now, the rows of number 0, 1, 2, k and also the processor in the in any row index as 0, 1 2 to the power k minus1.

(Refer Slide Time: 04:10)



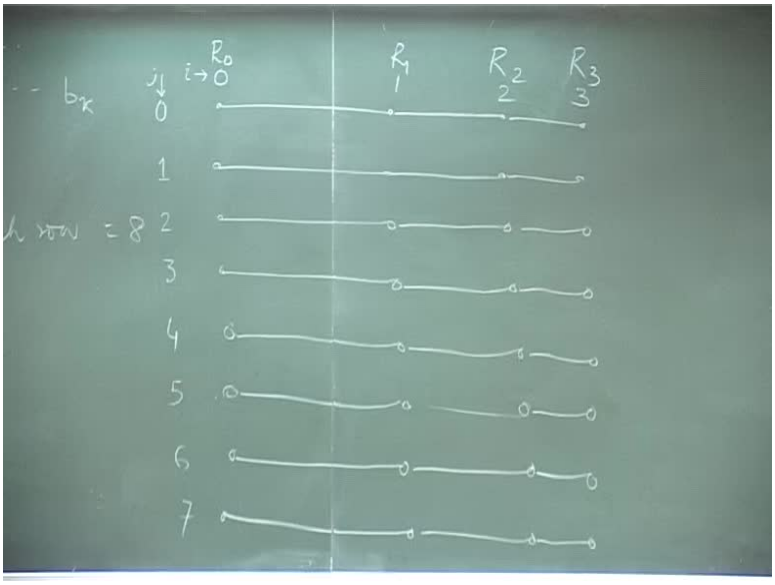
Now let us assume that Pij is the jth index processors in ith row. Now this processor, Pi,j, has the 4 connection. 1 connection is Pi-1,j another one is Pi plus1,j provided it exists, Pi-1,m and Piplus1,l provided it exists, where, m and l are obtained by inverting ith msb of j and inverting i-1th msb of j, respectively. So. Pi,j is connected with at most 4 processors Pi-1,j, Piplus1,j, Pi-1,m and Piplus1,l provided it exists, where, m and l are obtained by inverting ith msb of j and inverting i-1th msb of j, respectively

(Refer Slide Time: 05:39)

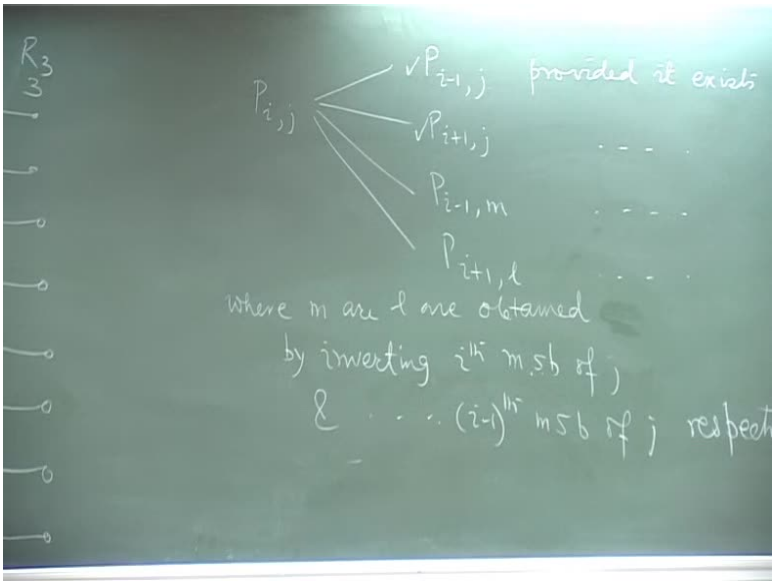


Let us start with ith msb bits. We start with that b1, b2 ….b k. These are the k bits you have because, it is a 2 to the power k processor we have, so, k bits. This is a 0th msb, this is a first msb and so on.

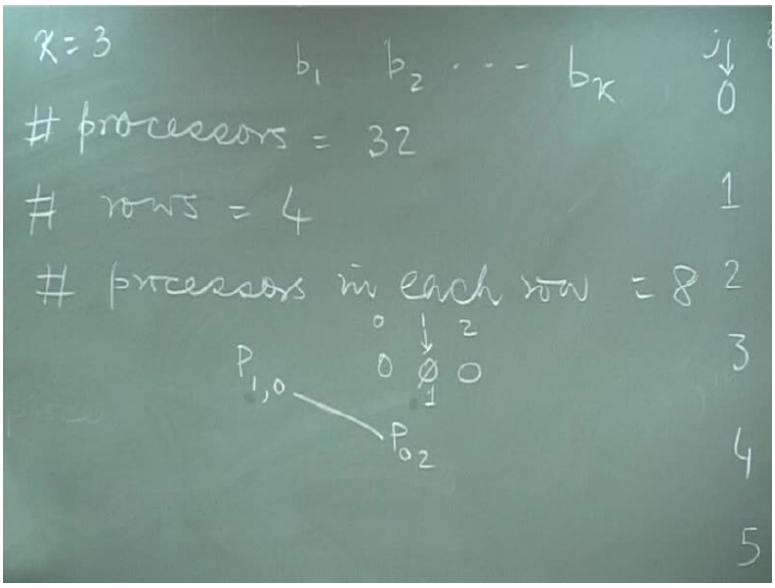
(Refer Slide Time: 07:32)



(Refer Slide Time: 07:43)

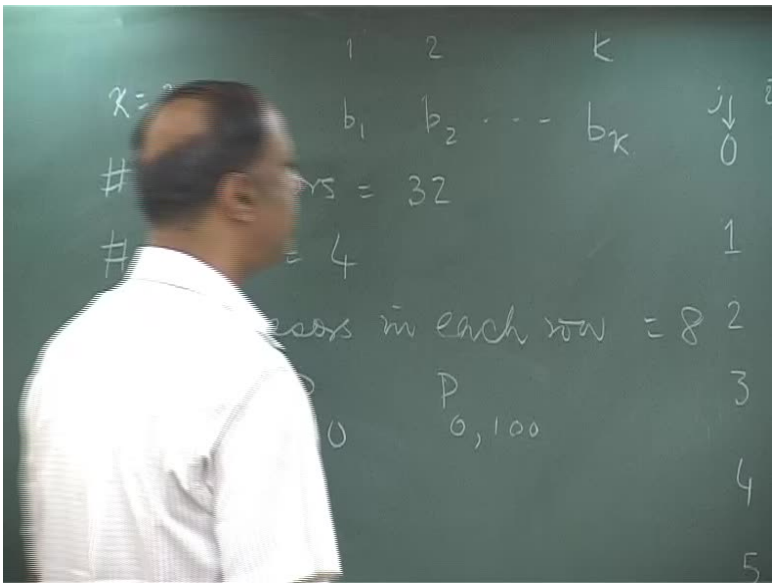


(Refer Slide Time: 09:15)

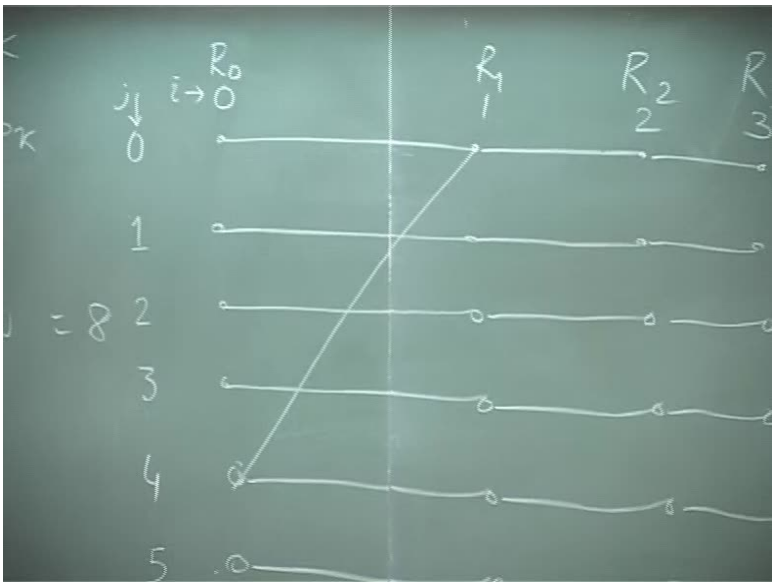


Now let us considered we have say, k equals to 3, the number of processors is 32, number of rows is 4 and number of processors in each row is 8. So, we will assume we have row 0, row 1, row 2 and then another row 3 and then, 0 1 2 3 4 5 6 7. So, Pi is connected. So, you have P0. This is 0. Row 0 index is 0 1 2 3. This side is 0 I and this side is your j. So, P0,0 is connect to it P1,0. So, this is connecting to this, this is connecting to this. So, this links are established based on these 2. Now think about a row 1, so, this one. P1,1. P1,0 is that. So, by inverting this, we will be getting 1 because 1st this is the 0-th bit, this is the 1st bit, this is the 2nd bit. So, here ith msb is replacing and P will be connected to P1,0 which is connected to P0,2. First it is ith, ith i-1, so, this would be connected to P0,2.

(Refer Slide Time: 09:44)

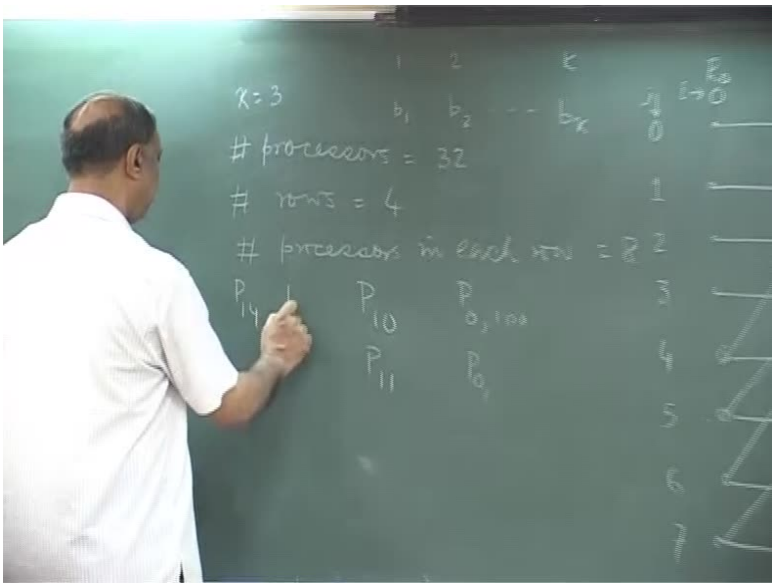


(Refer Slide Time: 09:51)

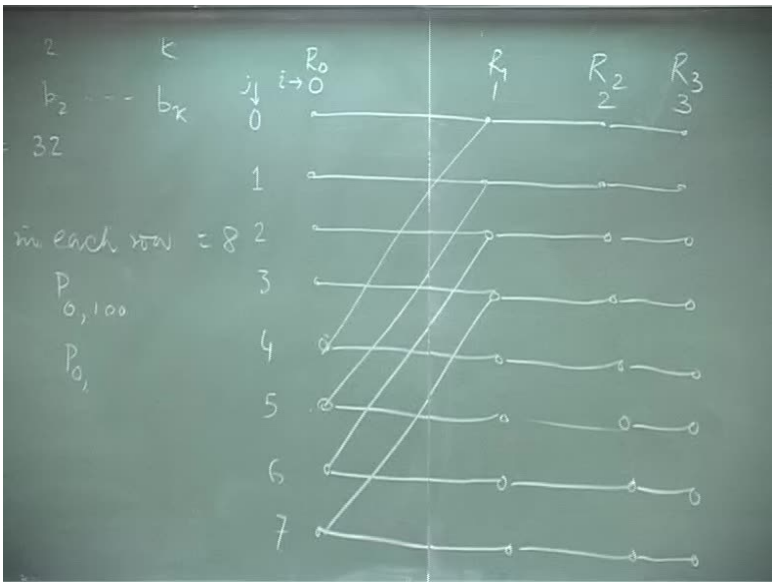


So, this is one thing you have to remember, this would be first second msb bit k msb, is k. So, in that case P1,0 will be connected to P0 and here the first bit will be converting 1 0 0 0. So, P0,4.

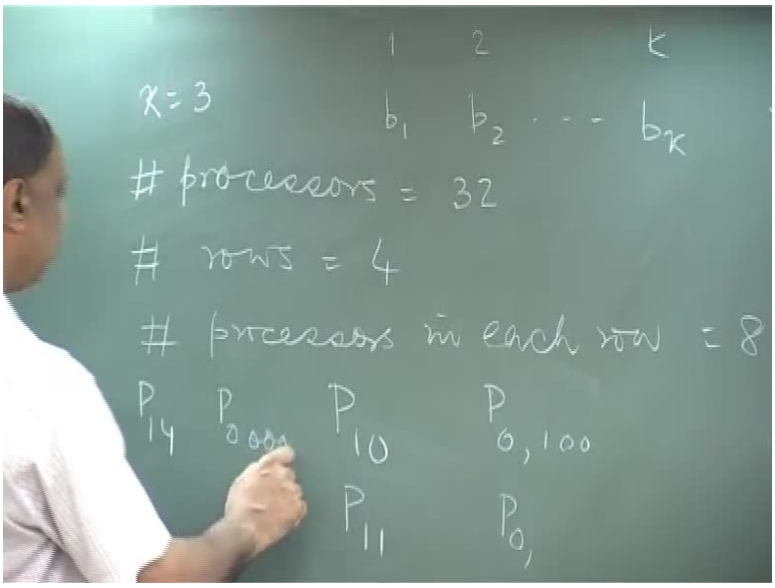
(Refer Slide Time: 10:20)



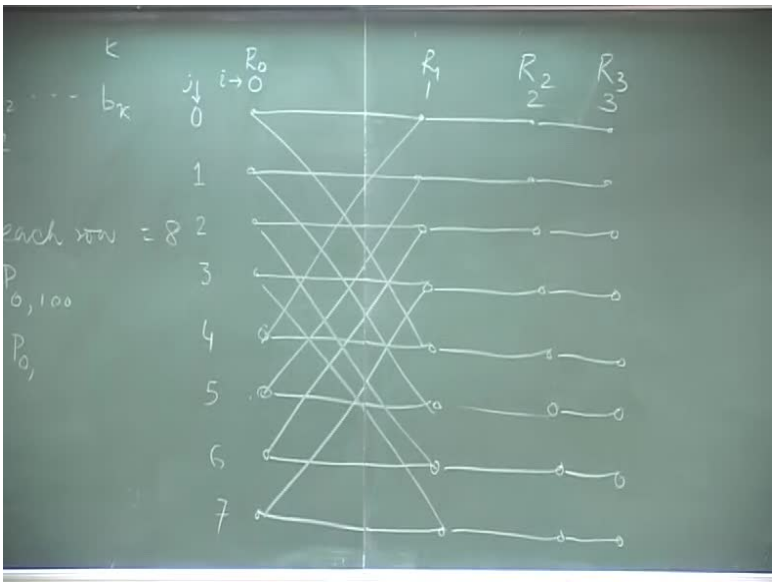
(Refer Slide Time: 10:14)



(Refer Slide Time: 10:23)

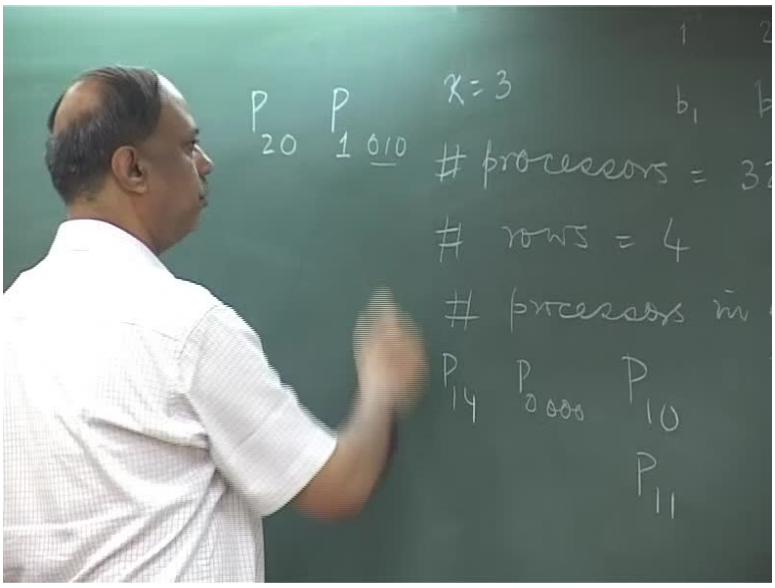


(Refer Slide Time: 10:37)

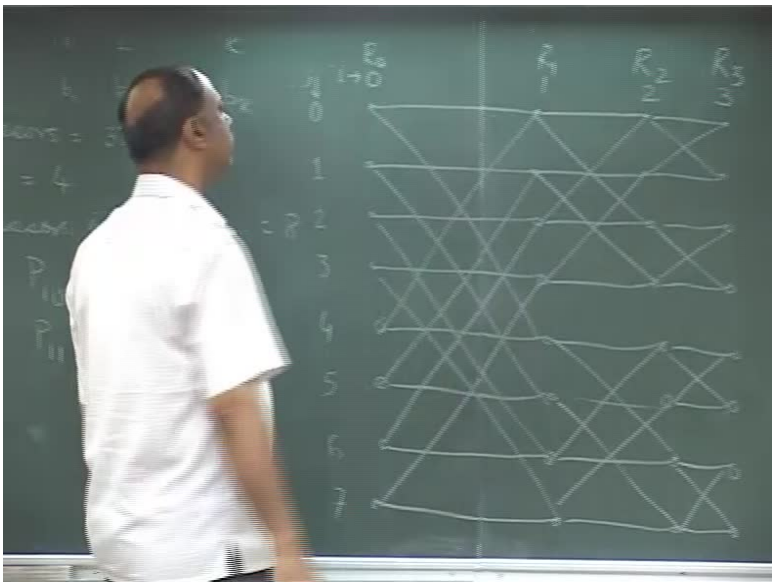


Ok then you have P1,1 and it will be P0,5. Similarly, you have P0,6 and P0,7. Now, what happens, P14 is P0000. So, P1,4 will be connected to this. Similarly, this will be connected to this, this will be connected to this and this will be connected to this, ok.

(Refer Slide Time: 11:09)

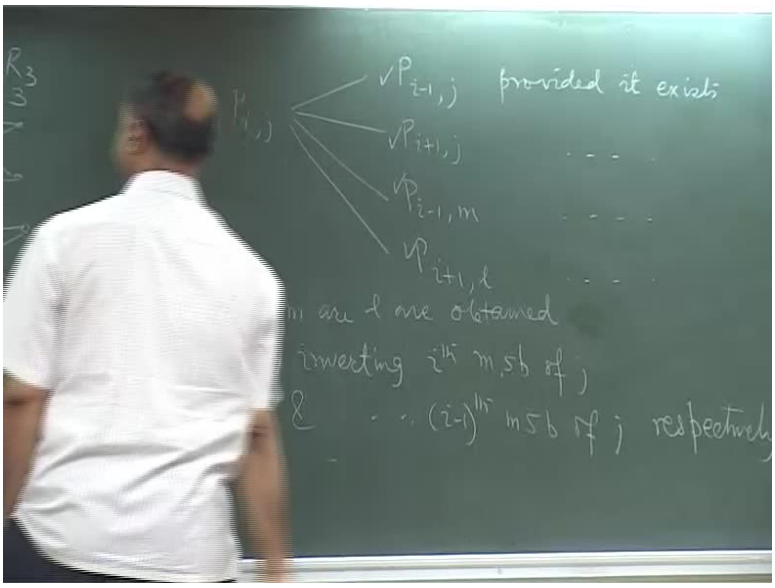


(Refer Slide Time: 11:51)

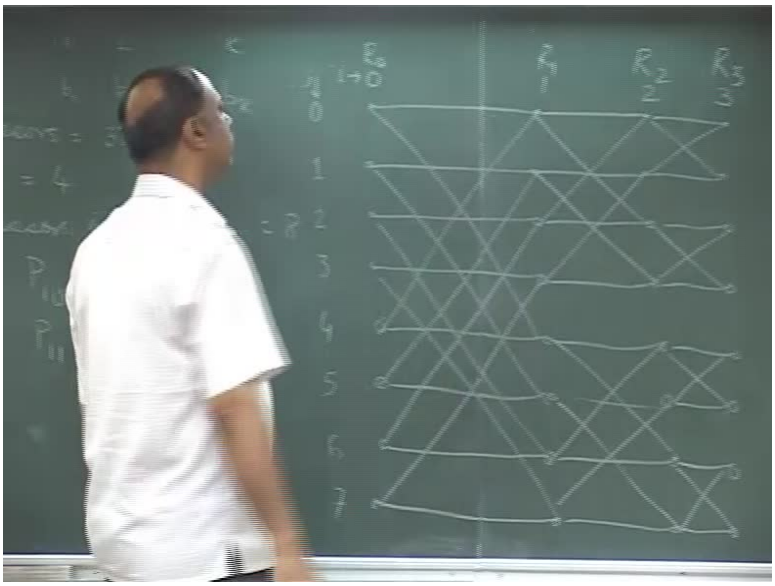


So, this is based on this connection. Now, Piplusl to give you the reverse connection. Basically, while I consider of this. So, this will be give you that. So, this is a by directional thing which will show, and then you have suppose, P21, P20, it becomes P1, P1 and 010, P1 to P20 is P12. Similarly, will getting this one, this will get this one and last one will get, that is why like this, ok.

(Refer Slide Time: 12:26)

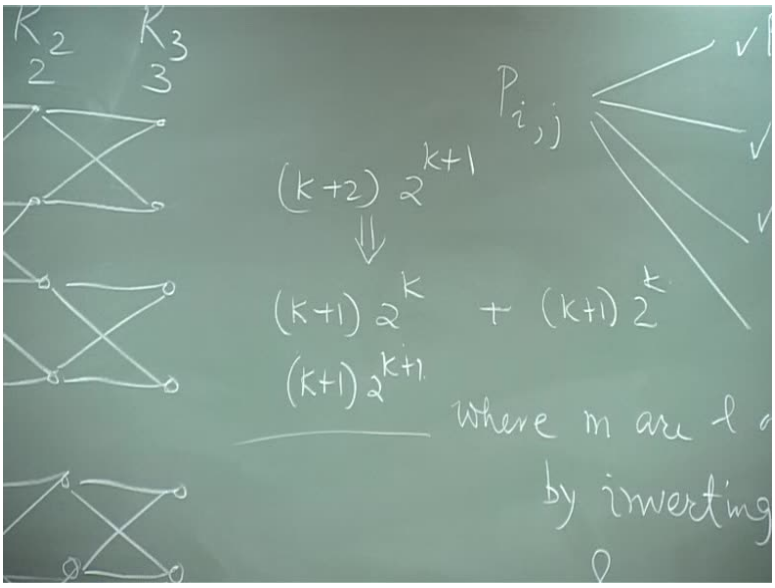


(Refer Slide Time: 12:50)



So, these two are basically to indicate the bidirectional link. So, the structure of this butterfly, you observe this, this looks like a butterfly wing, the h8 of this is the block is nplus1, and this side you have the 2 to the power n processors, and here you observed that to transfer data from one corner to another corner you will not take more than for a login time. And another thing is there this type of network still is upgradeable.

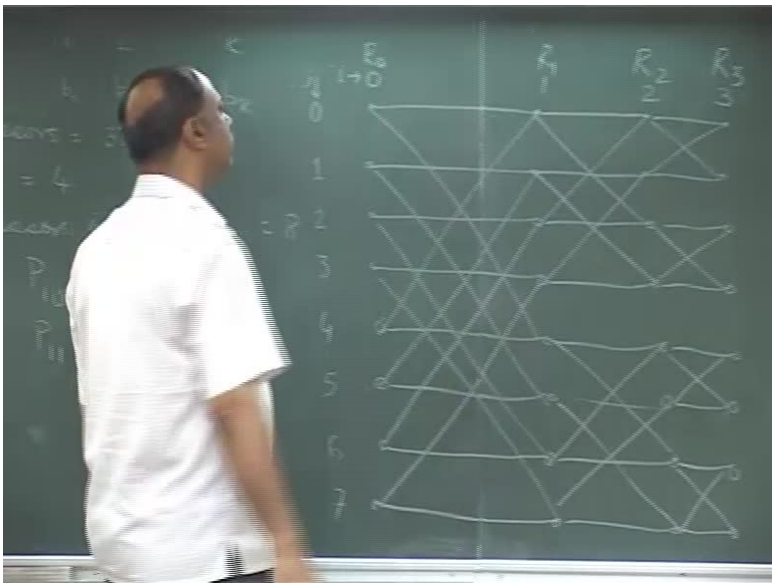
(Refer Slide Time: 13:35)



You can add another butterfly here and you can flag into that. Only this condition is that if you build it in the form of kplus1 2 to the power k. So, suppose we given a kplus2 2 to the power kplus1 number of processors, and you first obtain kplus1 2 to the power k one cluster, the other cluster is kplus1 2 to the power k, and then at the top you put kplus1. So, you get basically, here kplus1 2 to the power kplus1, right. So, basically you need additional 2 to the power k, 2 to the power kplus1, and note to be fixed for additional thing.

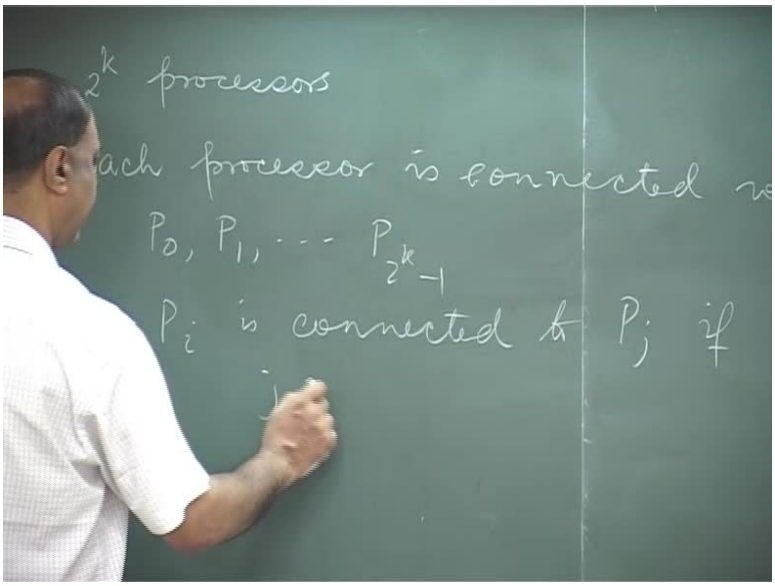
So, the thing is that converting for given to 2 by 2 butterfly networks and there is 1,2 break it the next grade upgradability thing we have to do little homework or book reading to do that, ok.

(Refer Slide Time: 13:58)

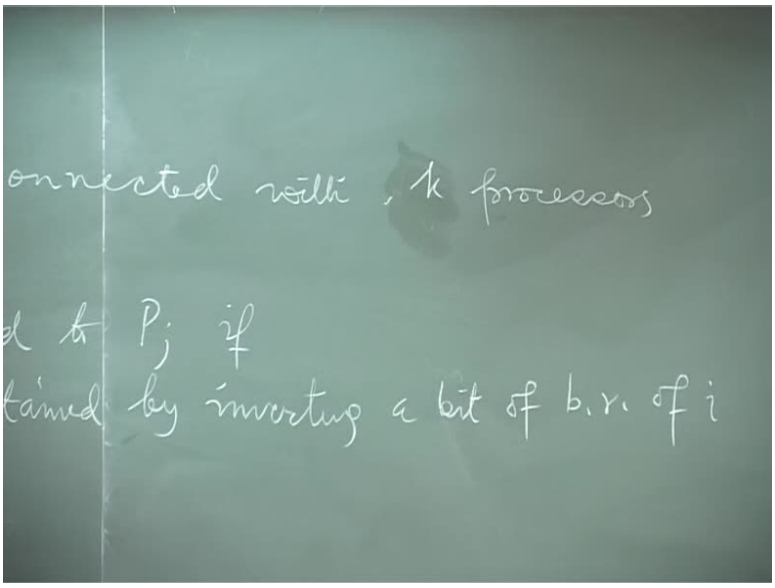


The type of problem which you can solve on butterfly network is that, if you where to meet the question of pipe lining is coming there you find that this network is very useful.

(Refer Slide Time: 15:47)

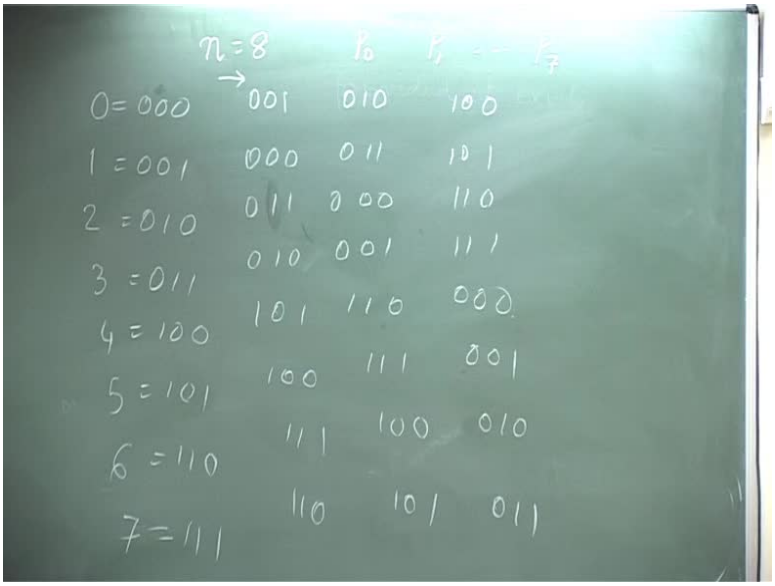


(Refer Slide Time: 16:10)



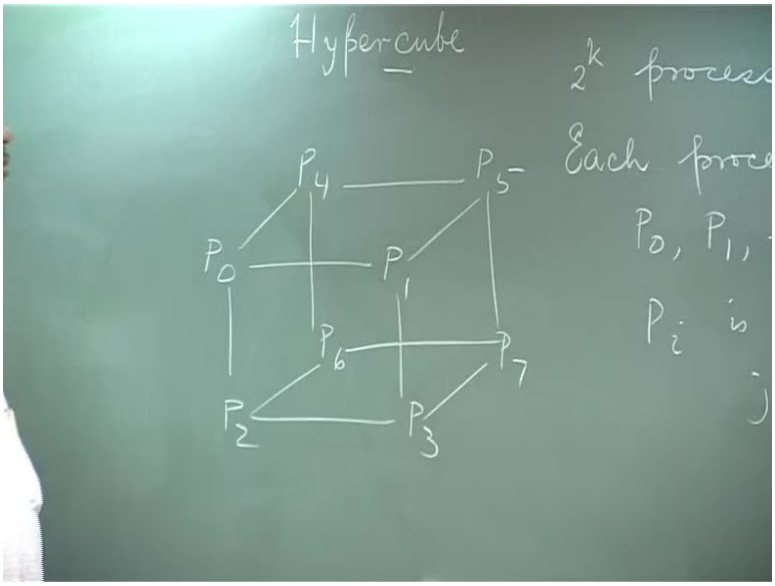
Next model number discussing is hyper cube. Here, you have 2 to the power k processors and each processors is connected with log n, each processor to k of other processors. Now Pi is set to P0, P1, P2 the power k-1 are the indices of the processors and Pi is connected to Pj if j can be obtained by inverting a bit of the binary representation of i. P i connecting to P j if j can obtained by inverting n in by their k bits you have of the binary representation of i.

(Refer Slide Time: 18:19)



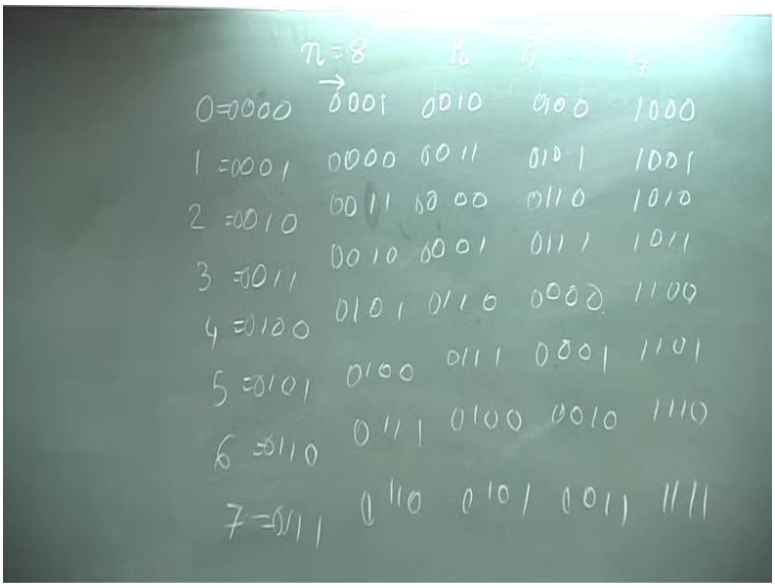
Ok let us assume that suppose you have n equals to 8, so, you have P0, P1, P7, so, P0 is connected with, so you have 0 1 2 3 4 5 6 7, that is the binary representation of this the 000, 001, 010, 011, 100, 101, 110, 111. Now P0 is connected with, by inverting is 001 010 100 000 011 101 011 000 110 010 001 111 101 110 000 100 111 001 111 100 010 110 101 and 011.

(Refer Slide Time: 20:03)



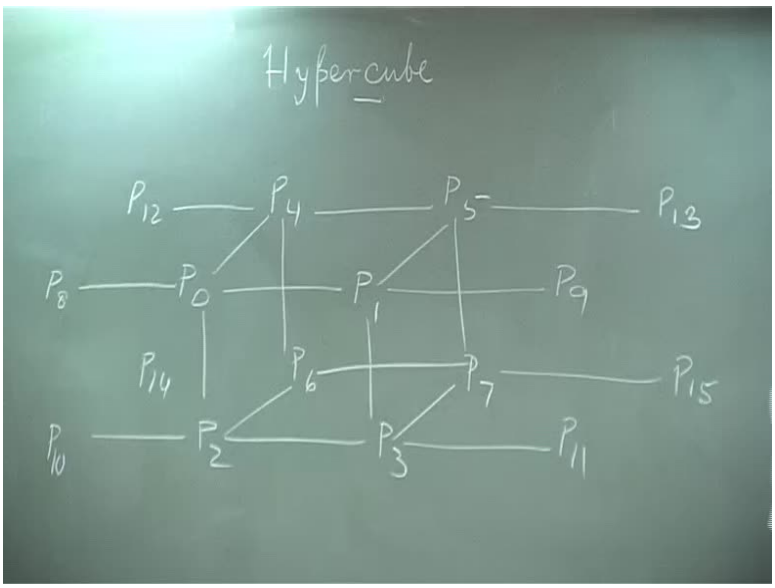
So, P0 is connected to this 3 processors because there are k connections. So, if I draw this P0 P1 P2 P3 P4 P5 P6 P7. P0 is connected with P1 P2 P4. P1 is connected to with P1 P0 P3 P5. P2 is connected with P3 P0 and P6. P3 is connected with P2 P1 and P7. P4 is connected with P5 P6 P0. P5 is connected with P4 P7 and P1. P6 connected with P7 P4 and P2. P7 is connected with P6 P5 and P3 these are careful.

(Refer Slide Time: 21:20)



So this is a structure your hyper cube with 8 processors. Now what happened when we have 6 processors? In case of 6 processors, we will be adding another zeros they are and here also there will be 0there also will be another 0 and you have another 0, we have, 1000 1001 1010 1011 1100 1101 1111. This is called 8 bit processors. Similarly, you can have remaining 8 processors.

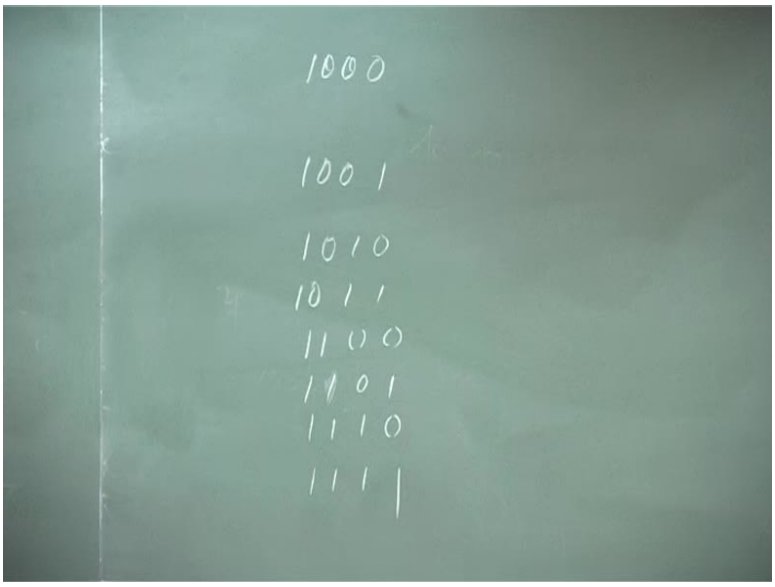
(Refer Slide Time: 22:54)



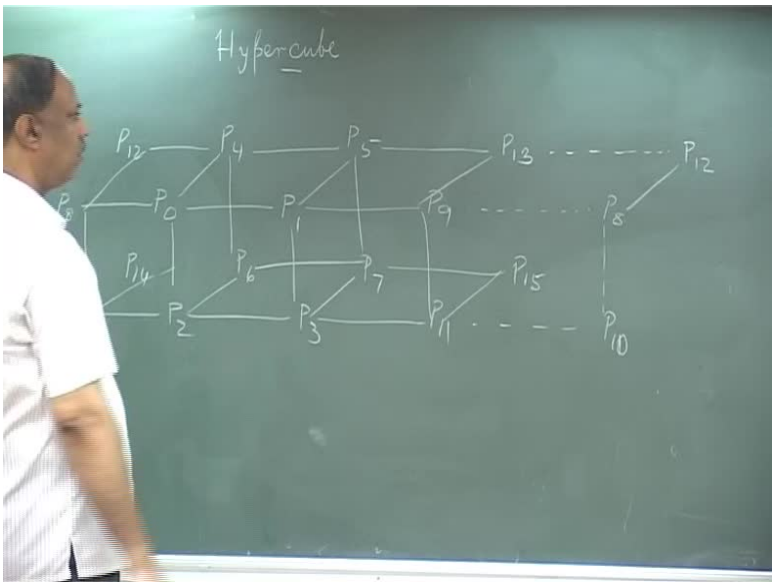
So, here what happened that you are connected already 8. Now it is P0 connected P8, so, I am drawing another one, P8. P1 is connected with P9 and P2 is connected with P10. P3

is connected with P11. P4 is connected with P12. P5 connected with P13. P6 is connected with P14 and P7 is connected with P15, ok.

(Refer Slide Time: 23:32)



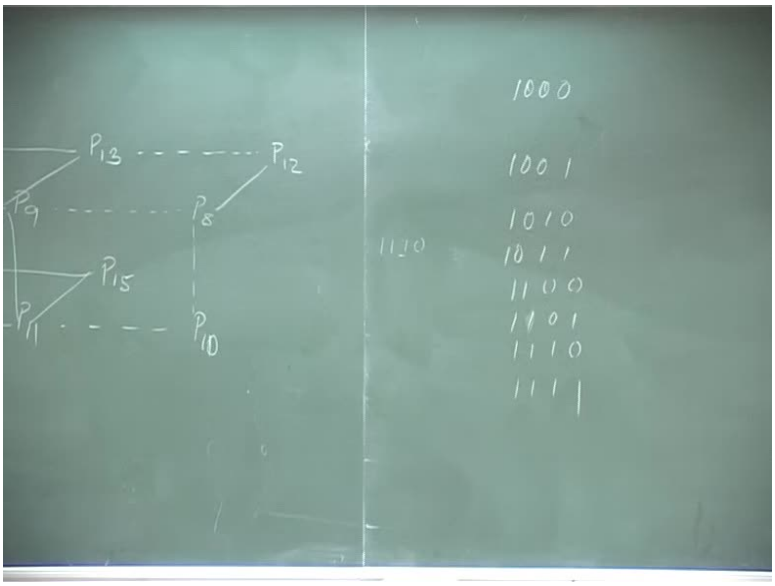
(Refer Slide Time: 26:06)



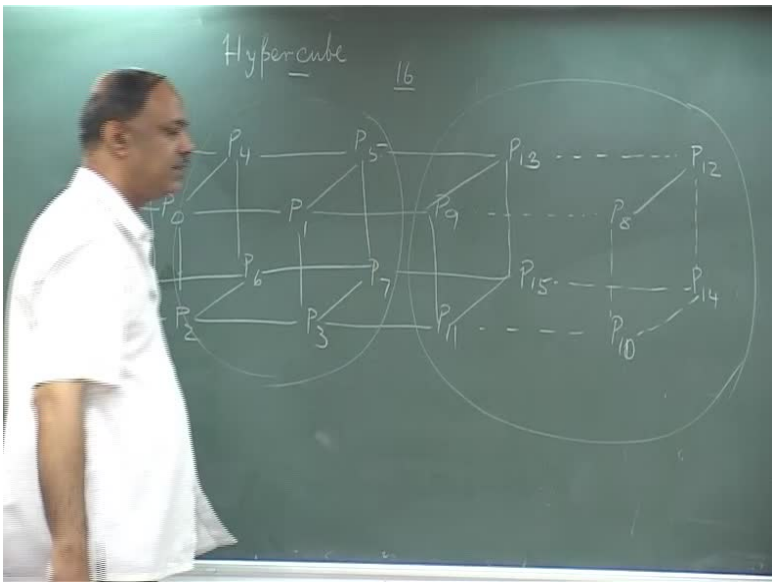
Now think about the 8 onwards. 8 is connected with 0 that is done. 8 is connected with 9. 8 is connected with 9 and 8 is connected with 10 and 8 is connected with 14. 8 is connected with 12. 8 is connected with 12, ok. Now think about next one 9 connected with 8. 9 is connected with 11. 9 connected with 13 and 9 is connected with 1. Next one

is 10. 10 is connected with 11 10 is connected with 11 and then 10 is connected with P10 is connected with 8. 10 is connected with 8 plus 4 plus 2 14 and 10 is connected with 2.

(Refer Slide Time: 26:36)



(Refer Slide Time: 29:35)



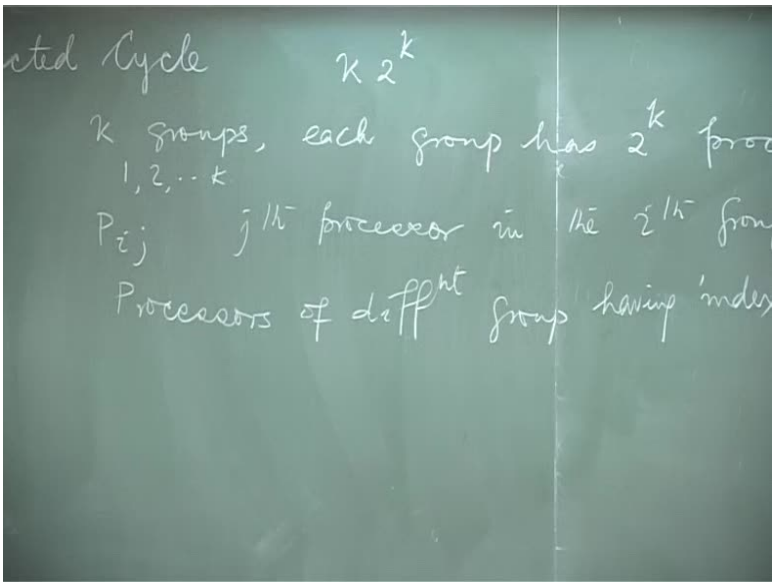
Then 11 is connected with 10. 11 is connected with 10 and 11 is connected with 9. 11 is connected with 15. 11 is connected with 15 and 11 is connected with 3, because then 12 is connected with 13. 12 is connected with 13. 12 is connected with 13 and 12 is connected with 14. 12 is connected with 14. What is 12? 12 is connected with 14 and 12 is connected with 8 yes that 12 is connected with 4.

Ok then 13 is connected with 12. Yes, then 13 is connected with 15. And then, 13 connected with 9. 13 is connected with 5. Next one is 14 is connected with 15. 14 is connected with 12 yes. And then 14 is connected with 8 plus 2 is 10 and 14 is connected with 6.

Ok then 15 is connected with 14. 15 connected with 13. 15 is connected with 8 plus 3 11. 15 is connected with 7. So, this is a structure this is a structure of hyper cube when n equals to16. If you observe that this is basically round and one good thing of this is that given a hyper cube of size 2 to the power k I can marginally value we additional mix of size to the power kplus1. The one beauty thing is that say, so, you heard you had this is initially you had this circle cube initially you had a this circle cube of size 8, this is another hyper cube of size 8 and you observed the 4 8 bit processors. There is one additional link you have a established and can margin get the hyper cube of size 16.

Now if I have another 16 processors another similarly, I can add this 2 hyper sub hyper cube to get a hyper cube of size 32 and you will observe the revaluating. Only one additional link between the 2 processors and these 2 processors are those one is a binary representation msb First msb is of binary representation of the 2 processors they are equal.

(Refer Slide Time: 35:19)

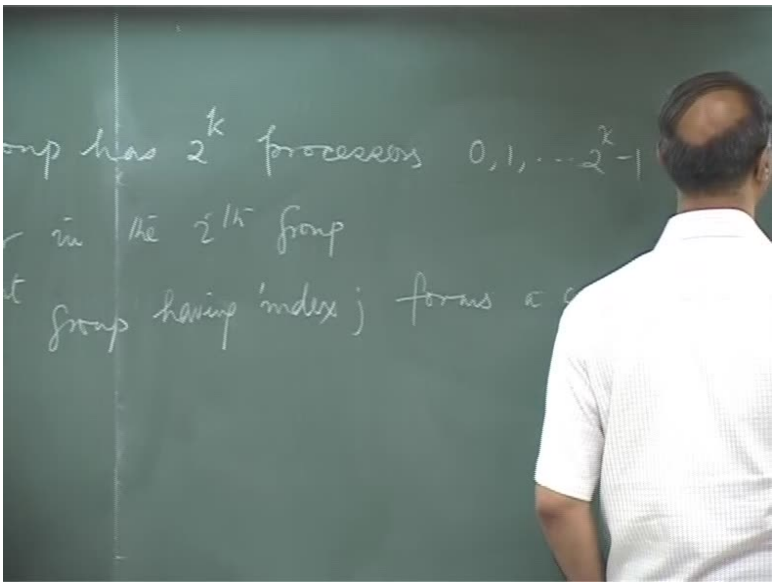


Ok so, you have every processor is having log and connection if you have the n processors the case of butterfly every processors having a at most 4 connection in the

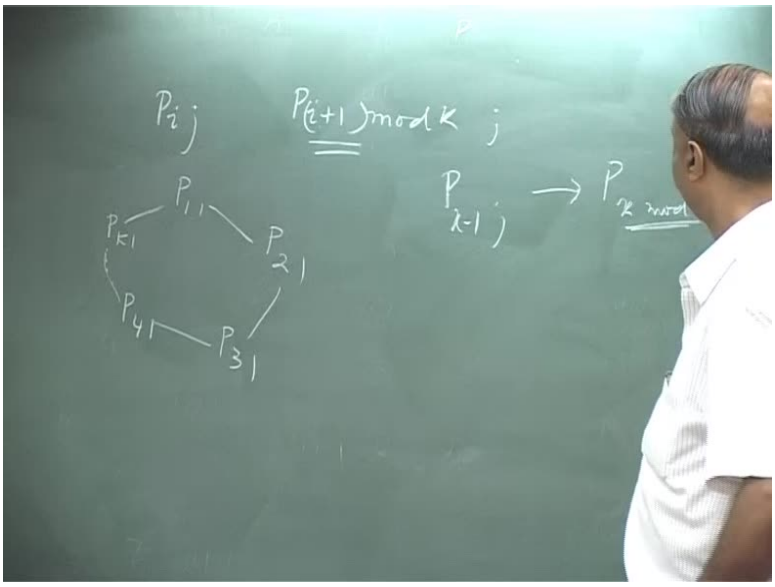
phase of 2 dimensional, net connection every processor is a having 4 connection at most and in the case of perfect shuffle, every processor has at most 3 connections.

The next 1 is cube connected cycle it is the combination of butterfly and hyper cube. You have k in to 2 to the power k processors. This processors are divided into k groups each group has 2 to the power k processors this processors are index 0 1 2 to the power k minus 1 and groups of number dials 1 2 k P i j is a processors is the jth processors in the ith group

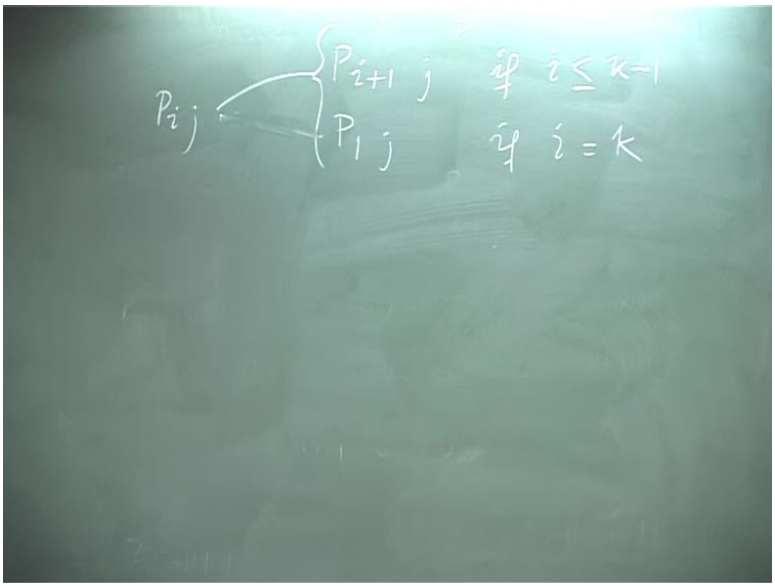
(Refer Slide Time: 32:50)



(Refer Slide Time: 34:03)



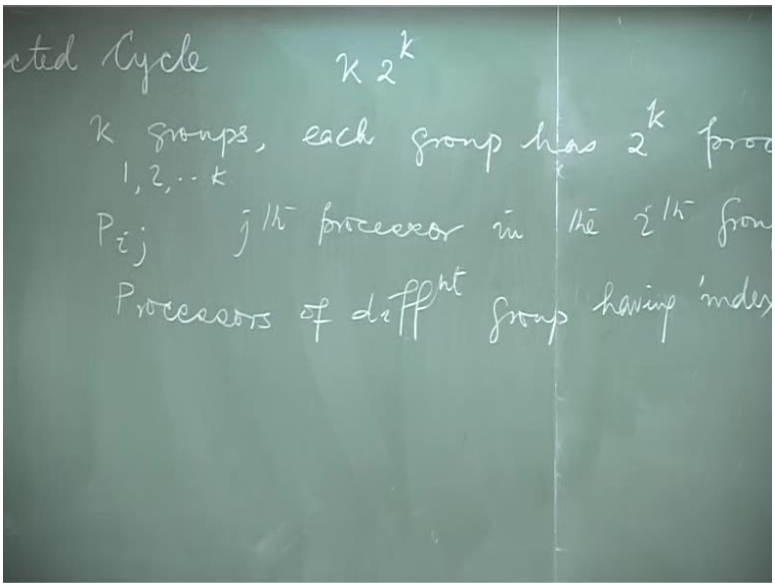
(Refer Slide Time: 34:56)



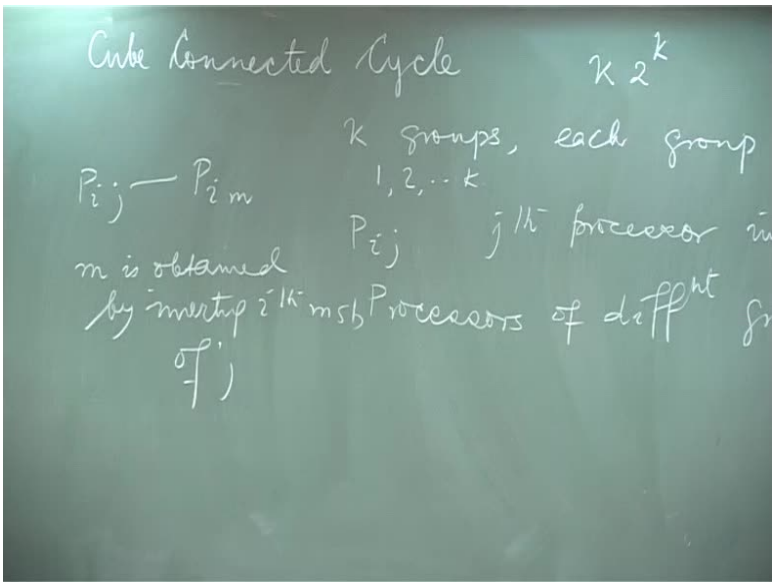
The processors of different groups, processors of different group having same index j forms cycle. That is the P i j connected to P iplus 1 mod k j. I have P 1, P 2 1, P 3 1, P 4 1 and P k 1. So, this forms size here now in this operations we observe there is a problem where it reaches k plus when it reaches k plus 1.When i is P k minus 1 j is connected to i will write P k mod k j. So, this becomes 0 actually P 0 is basically your P k. So, what I have to write here cost of module operations. We have to write it in different way it connected to P iplus 1 j if i is less than k minus 1 if it is i is less than equal to k minus 1and it is P j, if i is 1 i is k.

It is i is k then P k j are P k j is connected. P 1 j so that is the structure we have to write easier of module operation. Because I have started I made the group 1 to k so I have to write P i j is connected with the P, i 1 plus j if i is less than equals to k minus 1 and P 1 j if i equals to k

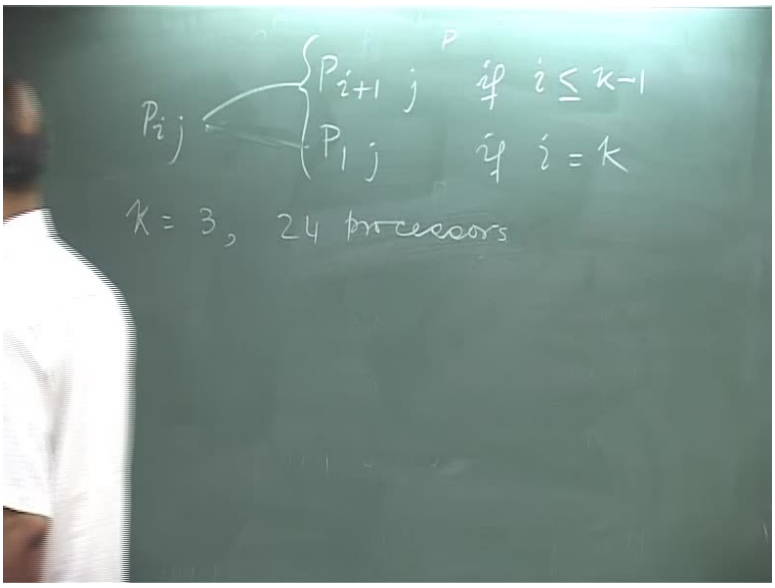
(Refer Slide Time: 35:19)



(Refer Slide Time: 35:59)

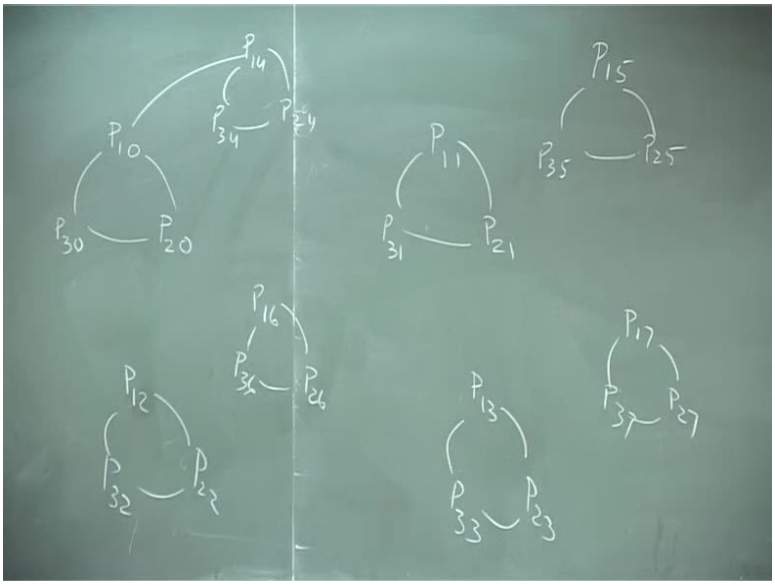


(Refer Slide Time: 36:23)



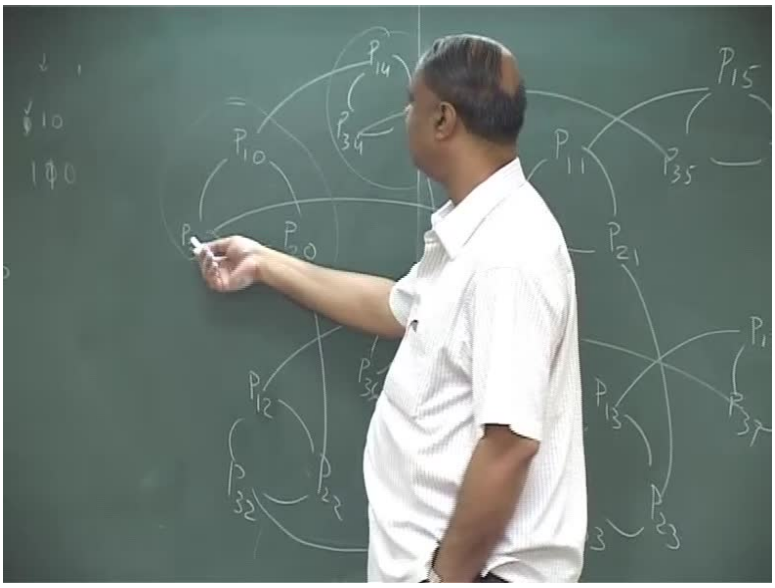
So this forms of cycle and besides this cycle Pij is connected with Pim; where m is obtained by inverting ith msb of j. m is obtained by inverting ith msb of j. So, let us see when you have 24 processors there and k equals to 3 k equals to 3 your 24 processors and the number of groups is 3, and is group is away 8 is group is away 8 processors

(Refer Slide Time: 38:45)



So, additionally write liquidate P 1 0, P 2 0 and you have P 3 0. You have P 11, P 2 1, P 3 1, you have here P 1 2, P 22, P 3 2. You have here P 1 3, P 23, P 3 3; P 1 4, P 24, P 3 4; P 1 5; P 2 5, P 3 5; P 16; P 2 6, P 3 6. P 1 7, P 27, P 3 7.

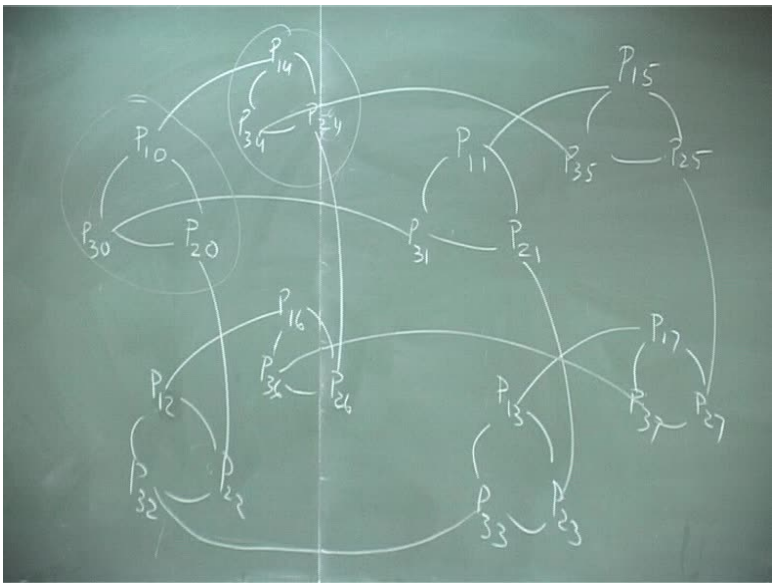
(Refer Slide Time: 41:05)



So this is our paint based on the initial collection that P i j connection with P iplus 1 j, if i less than equals to k. Otherwise P 1 j if i equals to j and P 1 0 is connected with P 0. 0 is connected with P 1 4 because this our first msb deep 2 0 is connected with P 2 0 is connected with there will be changing the second one P 22 and P 3 0 is connected via with P 3 1.

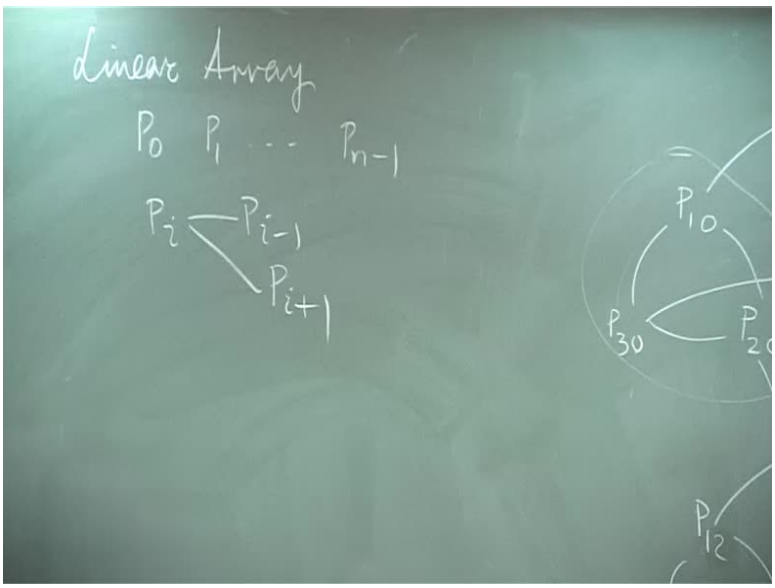
Ok similarly, P 1 1 is connected with P 5 1 5, P 2 0 is connected with P 23 and P 3 1 is connected with this now P 1 2 is connected with P 1 6, P 3 2 should be connected to this now P 1 3 connected with P 1 7. Now P 3 4, P 24, P 24 4 is P 2 6. Similarly, it will complete to 5 P 3 4 to be connected with P 3 5, P 3 6 will be connected with P 3 7.

(Refer Slide Time: 41:16)

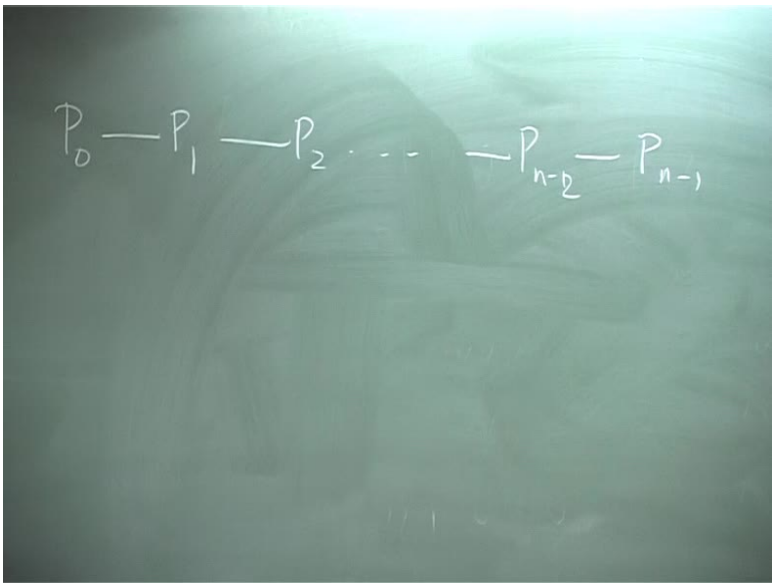


Ok so, we observe that every P processors if I considered this is a single mod then it becomes a hyper cube by consider this whole thing as a single mod this is a single mod it become a hyper cube and every processors is having 3 connections one is a 2 is a with 2 form the cycle and 1 is connected with another cycle so this all about the quick connected cycle.

(Refer Slide Time: 41:27)

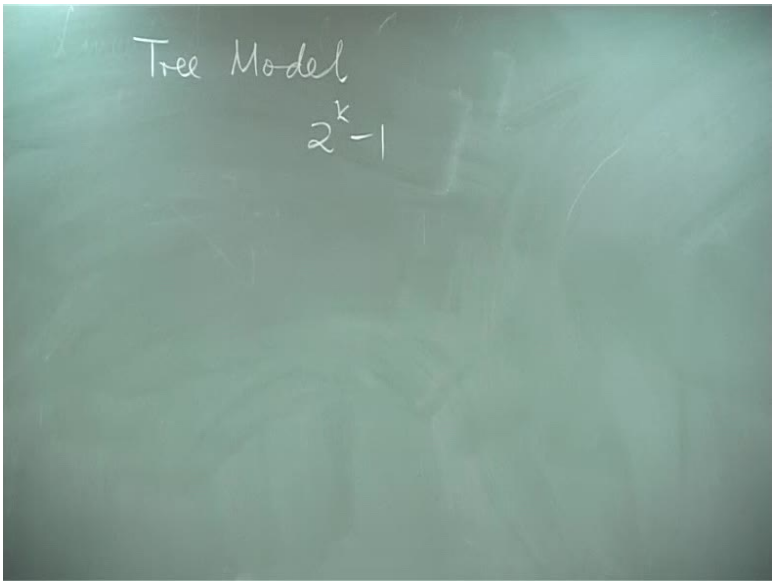


(Refer Slide Time: 41:59)

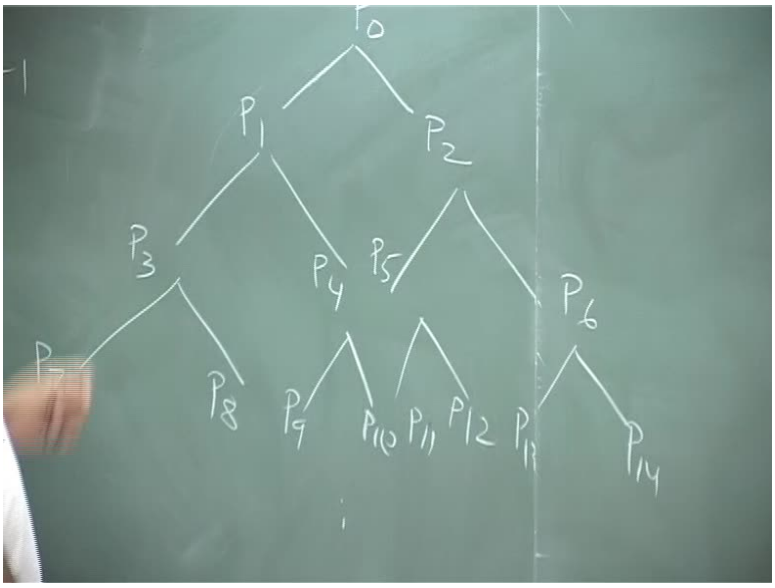


Now the next model is known as linear array. Here if you have n processors the P 0P1, P n minus 1, they are linearly connected. It means Pi is connected with Pi minus 1 and P plus 1 provided they exists. The structure looks like here P 0 P 1 P 2 P n minus 1 and this are these connections by direct.

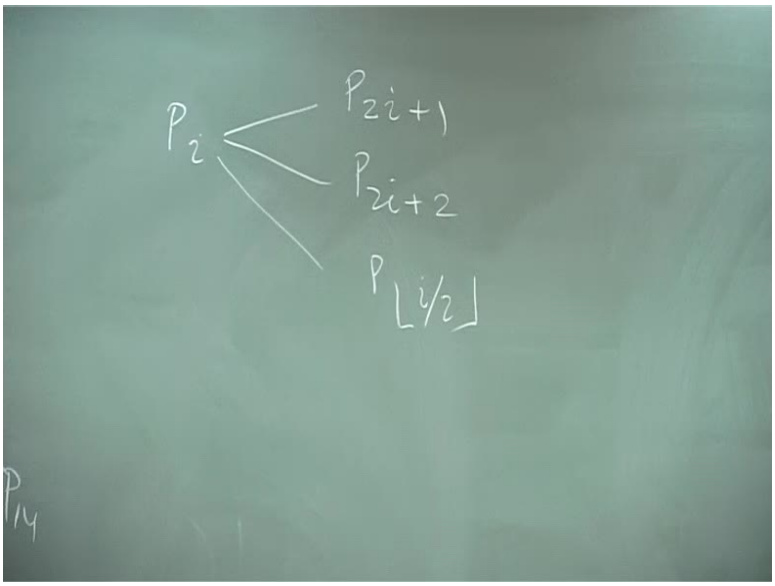
(Refer Slide Time: 42:14)



(Refer Slide Time: 43:14)

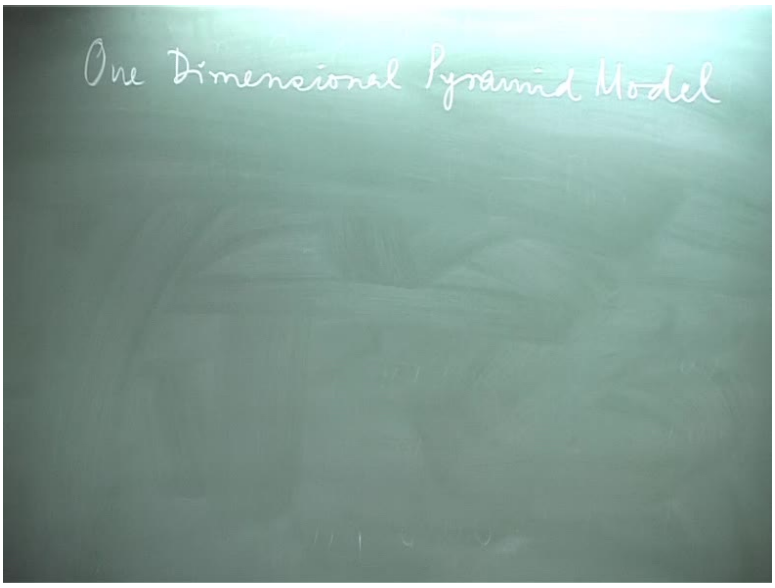


(Refer Slide Time: 43:44)

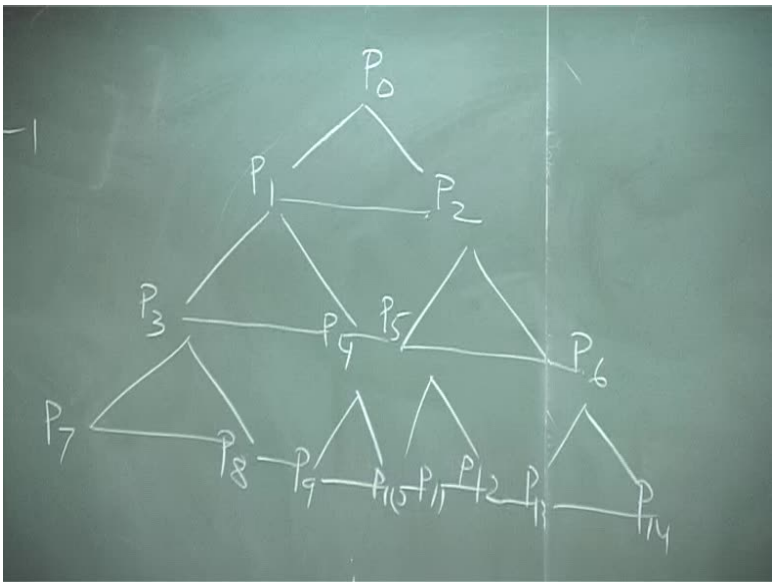


So, next model is tree model here you have 2 to the power k minus 2. Now this 2 to the power k minus 1 processors are raised in such a way that it forms full binary k and this binary k have index. Where from the bit first searching number the d, is P 0, has roots P 1 P 2P 3 P 4 P 5 P 6 P 7P 8 P 9 P ten P 11 12 13 14 and so on. There is the Pi connected with P 2iplus 1 and P 2iplus 2and also it is connected with Pi by 2, this is a parallel of this provided their exists.

(Refer Slide Time: 44:12)

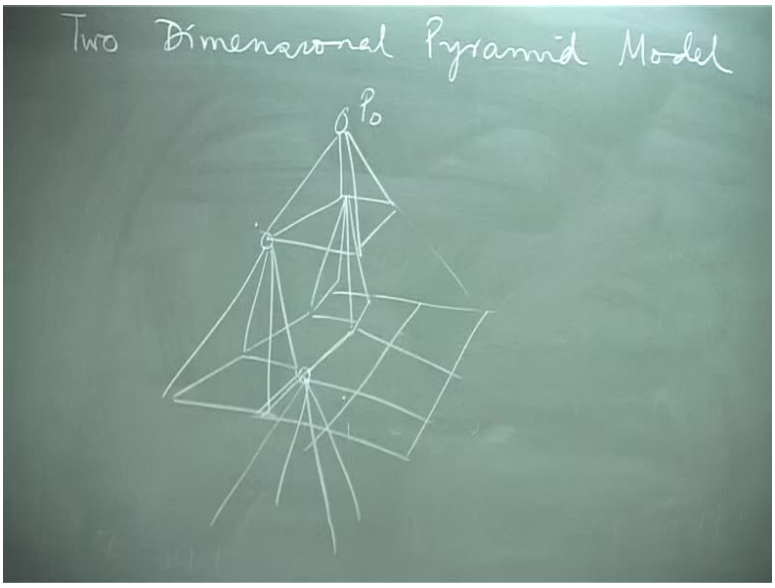


(Refer Slide Time: 44:43)



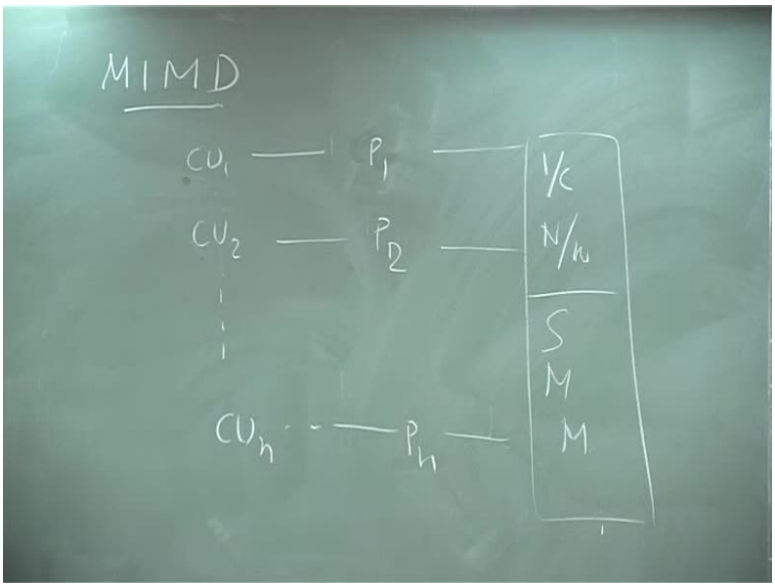
Now height of this tree log n, that is height is k. It indicates the 2 times till the date of from bottom to up you need out of lock it out of log in time. Now the next model is one dimensional pyramid model. There are 2 types of pyramid one is one dimensional pyramid model. It is the combination of linear array and tree. You have the usual tree and these sibling are there from the linear array and it gives you the one dimensional pyramid.

(Refer Slide Time: 46:31)



Now next model is known as 2 dimensional pyramid and it is the combination of mesh and tree and every processor Pi is connect as a post manly, remember that you have P0 and these are 4 children. Now for this node you have again, so every node you observe that this is a mesh this is another mesh so on. And this a node of this mesh is connected through the 4 delivering processors on the mesh. It is connected to it is parallel and now it is connected through 4 children. So, a mesh is having 2 dimensional pyramid. It is a combination of tree and the mesh and furnished node it has a cost 9 connection, 4th connection of the same level then checking the mesh connectivity. 1 and 2 is parent and 4 is to children. So, this is these are the major a SIMD model.

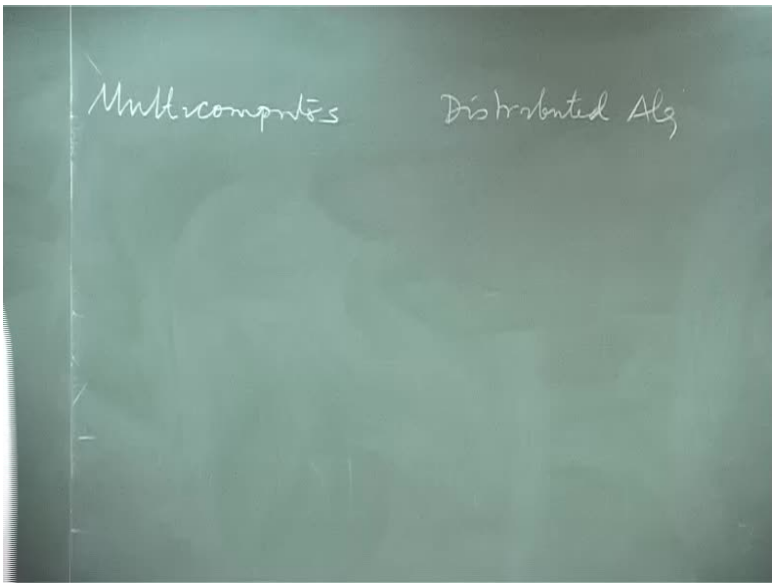
(Refer Slide Time: 47:50)



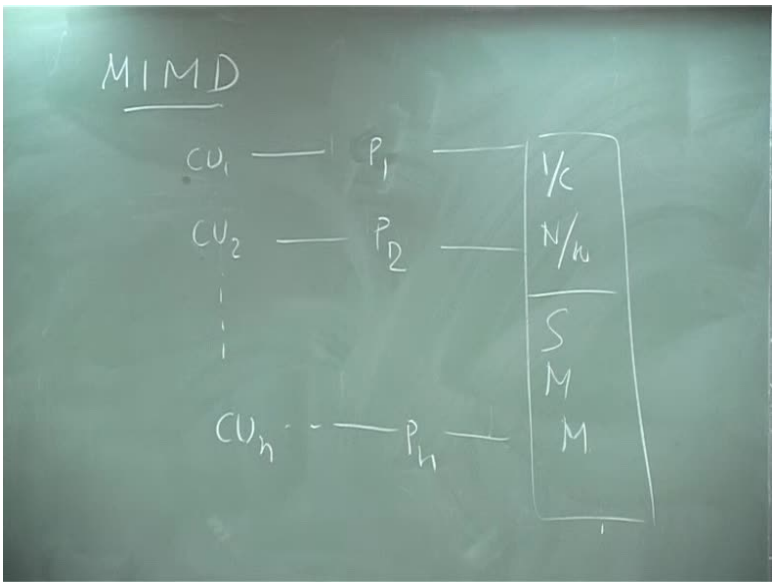
Now let us come to the definition of MIMD multiple instructions stream and multiple data stream. Here you have several a independent machine. Each machine will have the controlling units CU1 CU2 and CUn this n control unit and you have this processors of P1 attest to controlling unit, one processors Pn is attest to controlling unit n and you have this processors are connected either through pure memory or interconnection network or short memory, ok.

Now these processors are capable enough to solve a problem. Now here only issue is that why this is a controlling one but, this one streamer is a instructions control chip 2 can but, has the different set of a instructions to be performed and as a result the synchronization plays the major role. So, the delay becomes very less and since these processors capable enough to solve the problems, so each control unit broadcast us the set of instructions as much as you can. So, that relation there is the communication between the 2 processors becomes minimum.

(Refer Slide Time: 49:06)

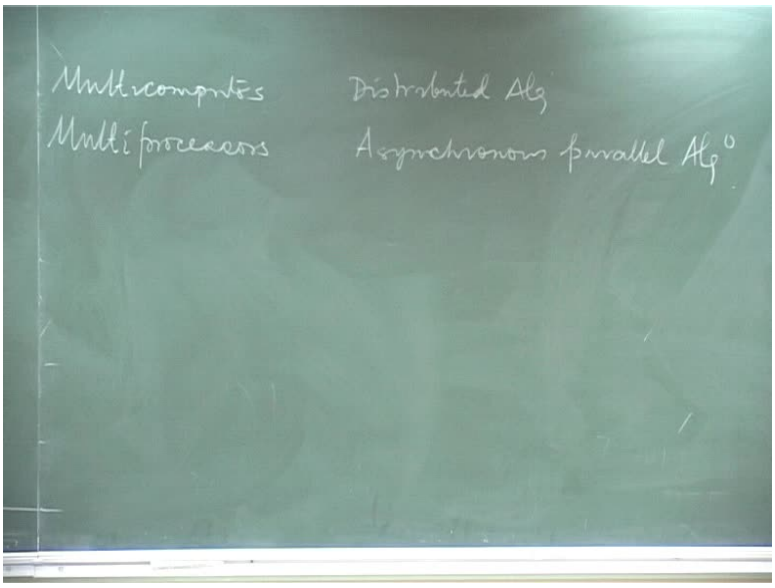


(Refer Slide Time: 47:50)



Now once you have the model based on the interconnection networks, we tell it is multi computed and they algorithm design for this is known as distributed algorithm, and in this case mesh is passing plays an important role because once you pass the message we should see that the other processors are active or it does not get disturb. So, minimum number of mesh is to be distributed or passed amongst there.

(Refer Slide Time: 50:07)



Then so you not mention the complexity that you remix of based on the number of messages transmitted between the 2 processors. Now once we have another type of thing that we have processor can communicate amongst them to short memory and in that case the model name is multiprocessors and algorithm is known as asynchronous parallel algorithm.

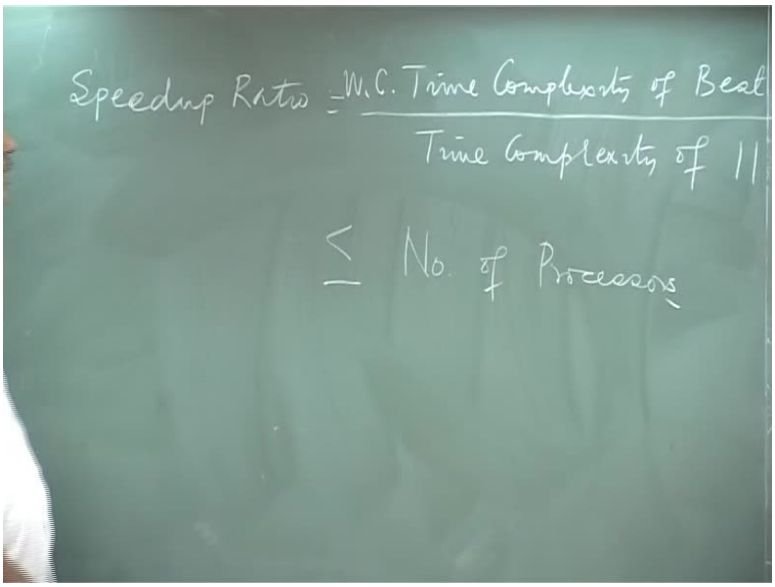
(Refer Slide Time: 51:21)

We have the asynchronous parallel algorithm and here that time complexity is of as basic parallel algorithm a role to measure its complexity. So, these are the various types of

various modules you have in part of you for designing the parallel algorithm. Now, how to measure the complexity of algorithm? In the case of sequential algorithm we measure the complexity of algorithm based on the two factors. They are time complexity and the space complexity and there exist kind of relationship between this two and if you have the most space time possibly you can take this to solve a problem or similarly, if you have less space time can work. So, there exist pair of relationship between the 2 parallel limitations for example, to find the sum of n number whatever the space you have but, you have to give the n minus 1 addition, ok.

Now in the case of parallel algorithms you have the other factor the number of processors you are using. Now here you have the trade of relationship between the 3 factors time, space and number of processors. It will so happen that if we have some more number of processors time may be less or if you have less number processor and you find the time is taking up but, there is a limitation here again that whatever the case may be you have to pay additional cost for using more number of processors.

(Refer Slide Time: 53:30)

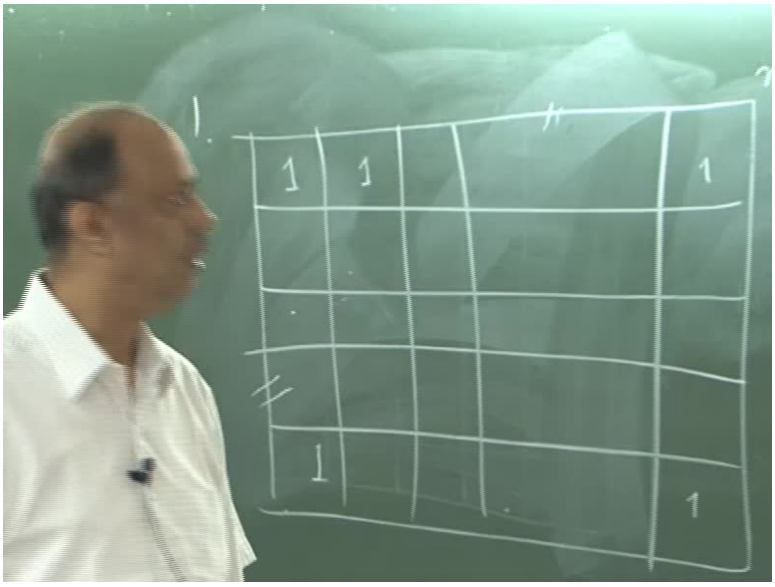


Now there is a here to measure the k 2 to the power we have the parallel algorithm there is term known as speed of ratio. It is defined as the time complexity of the best known sequential algorithm this is a it should be worst case time complexity of best known sequential algorithm divided by time complexity of parallel algorithm so, more the value

of speed better is your algorithm again but, the speed of is there any limitation will you explain that explain limitation

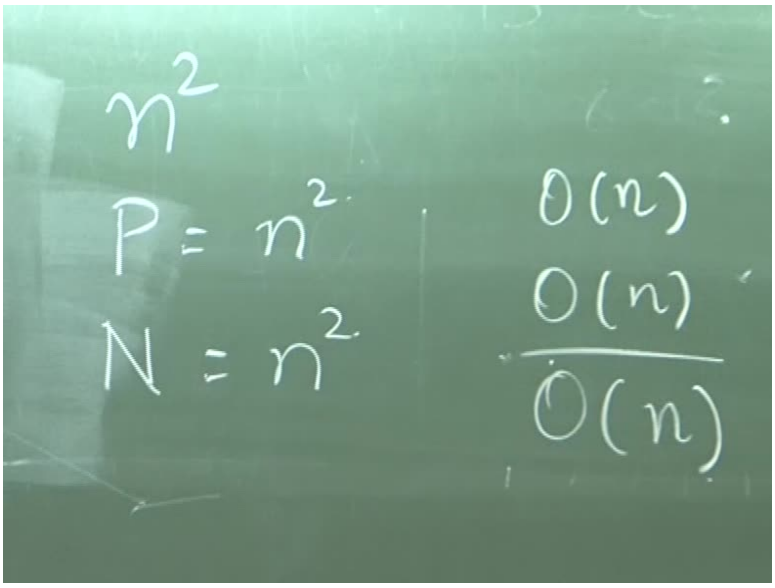
That what we have defined their worst time bases, of best known sequential algorithm by time complexity of the parallel algorithm and I can write can I write that this one less than equal to the number of processors used.

(Refer Slide Time: 54:59)



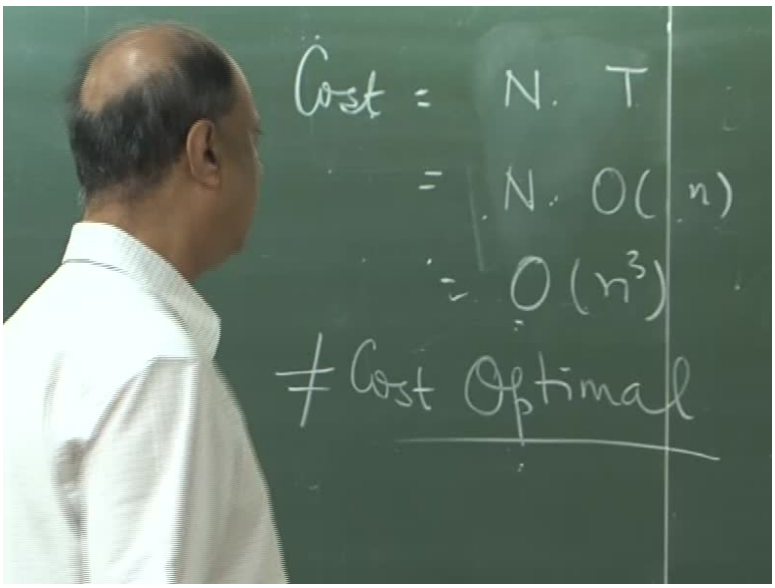
Let us considered the problem of finding the summation of numbers on mesh connected computers and as we know that I have a mesh of second cross and cross n mesh and as suppose the number of elements I have n square so, n is n square P is number of processors is also n square. So, observe that n square elements and n square processors what I can do is I can distribute the n square element amounts each n square processor each is having one element right.

(Refer Slide Time: 55:35)



Now in order to find the sum of this n square elements first I will use or I will assume that this is a linear array and I will add. This can be done order n time. So, you will observe they all the summation of respective row they are available in the last columns. So, again I will add this assuming the this is a linear array I will combined them so this can be done in order n time.

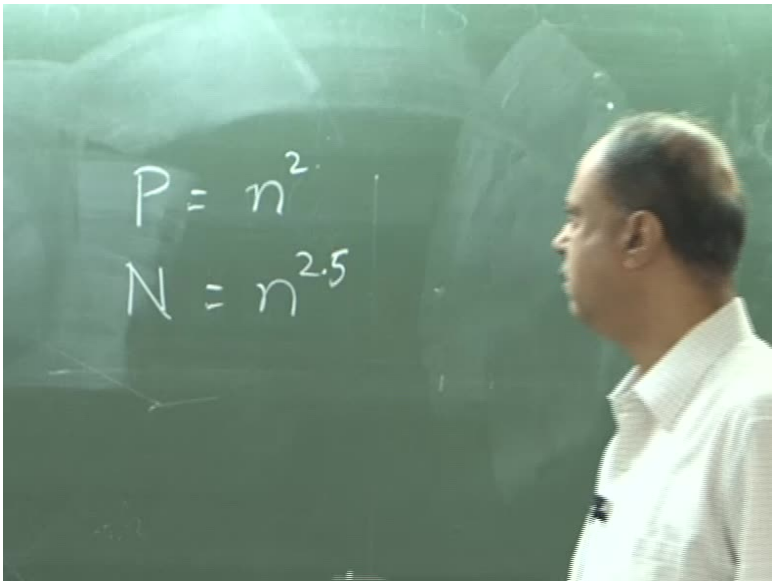
(Refer Slide Time: 56:20)



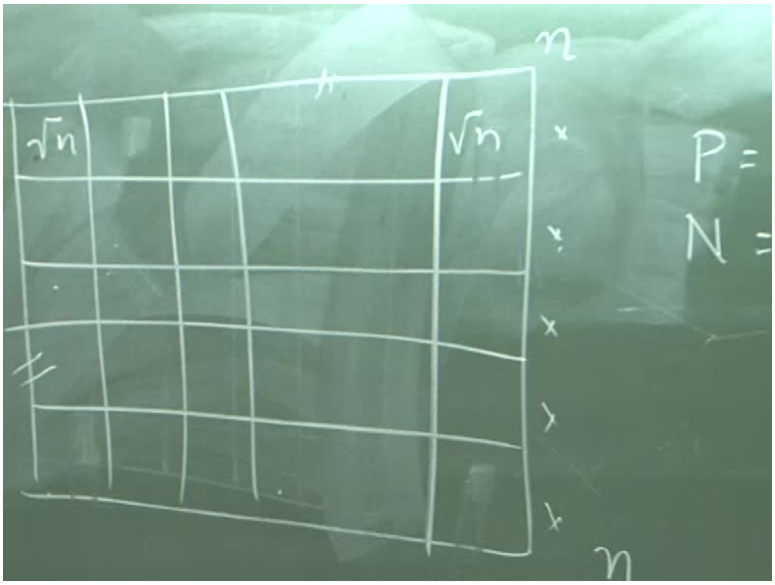
So the total time becomes order n using n square processors to find the sum of n square elements. But, if I estimate the cost of this method it becomes as you know the first is

number of processors and the time required to find the sum of an numbers so which is nothing but, any 2 order n so it becomes order n cube. To find the sum of n square elements.

(Refer Slide Time: 56:57)



(Refer Slide Time: 57:24)

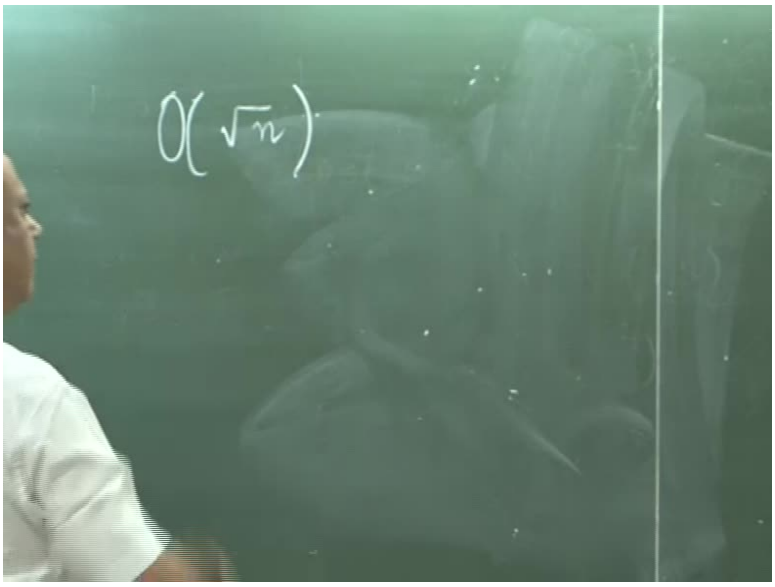


So, it is not cost optimal because to find thus some of n square elements on using the sequential processors is order n square. So can you obtain the cost optimal parallel algorithms? To do that before doing that let us assume that I have n to the power 2 point five number of elements 2 point 5e number of elements. So, what would assume or we

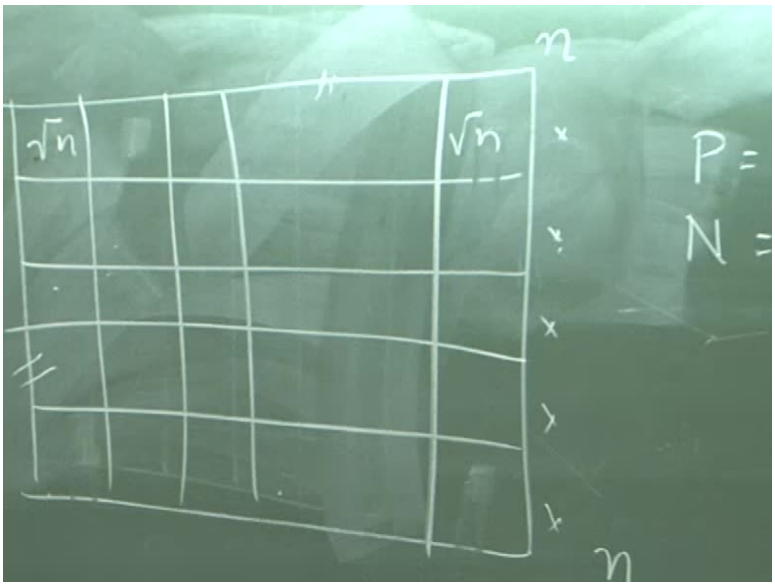
what we do we distribute these n to the power 2 to the n to the power 2point five elements among this n square processors such that every processors contains square root of n.

Now if you observe that a processor is having a square root of n elements so, there are n square processors n square into n to the power half which is n to the power 2point five elements you have distributed among the n square processors. Now I employ each processors to find the sum of yields n square elements each square root n elements

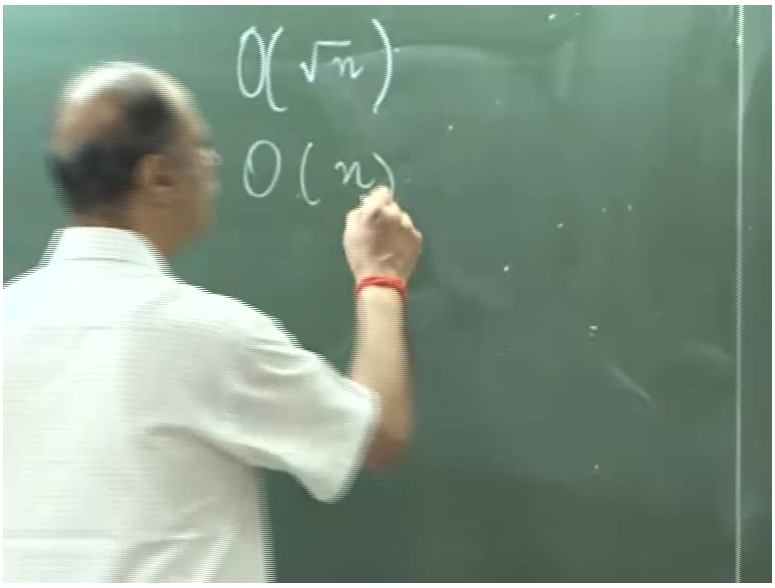
(Refer Slide Time: 58:00)



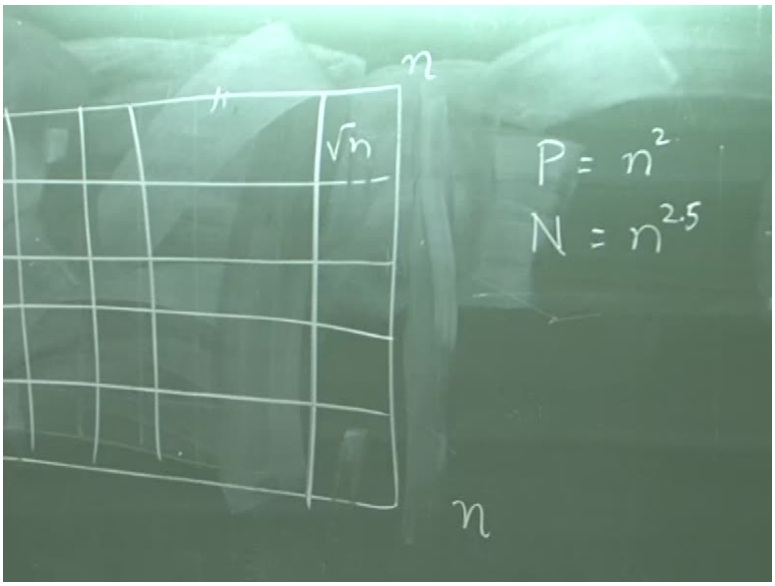
(Refer Slide Time: 58:24)



(Refer Slide Time: 58:28)

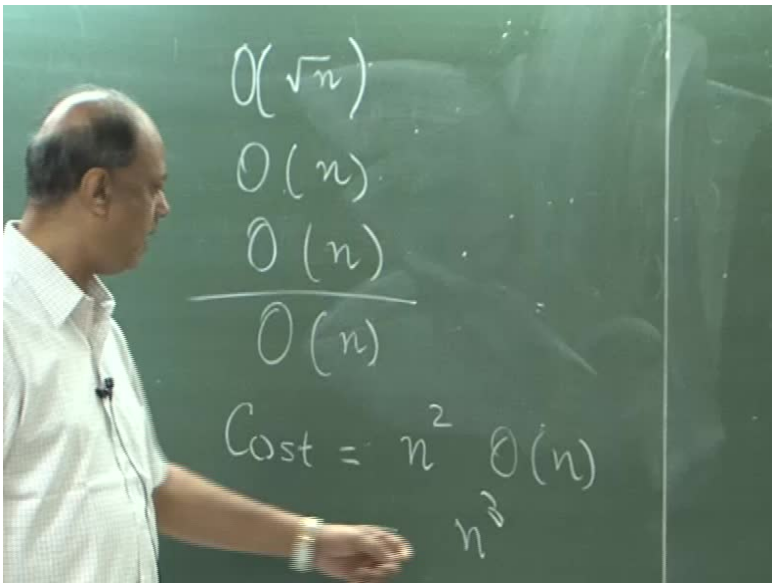


(Refer Slide Time: 58:38)



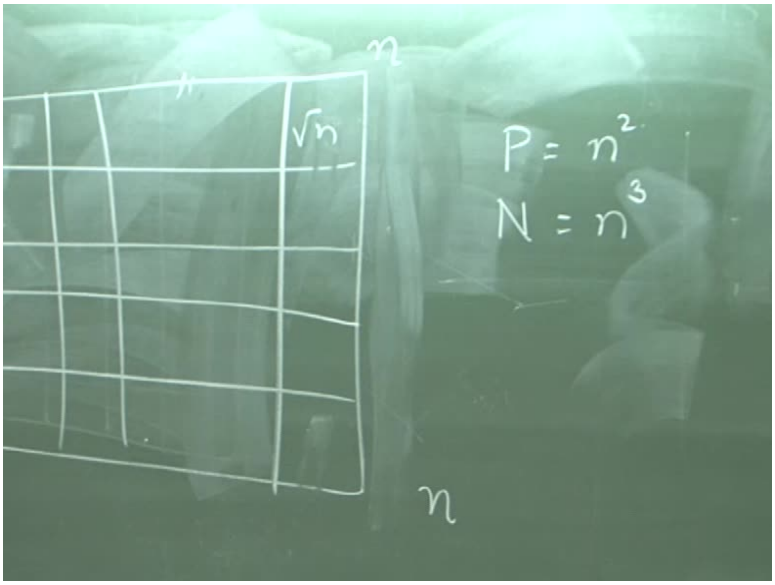
Write which can be done in order n time sequentially which I have done it. Now every processors will retain a sum of square root n elements order square root n time now these some elements. Now I can some row wise in order n time you get again order n time to find some of some of n sum of sum of square root n elements row wise and in the last column you will find that sum of n square root n elements n square root n elements in each row

(Refer Slide Time: 59:51)

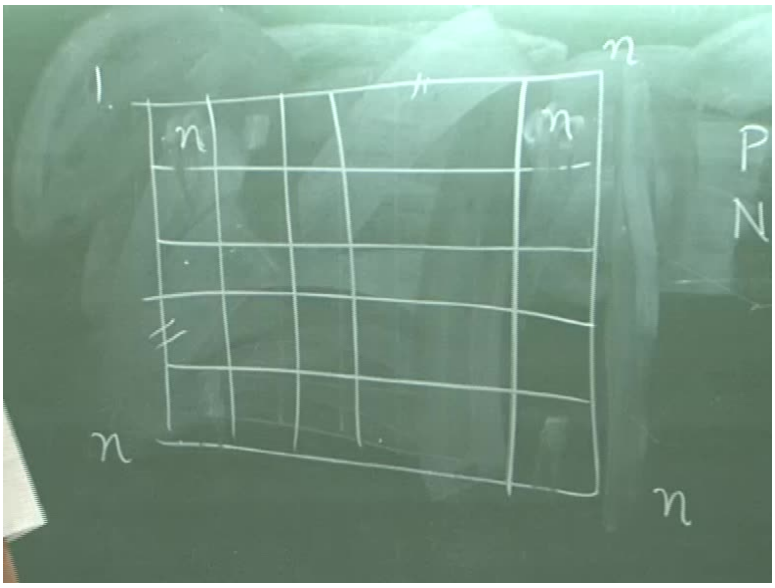


Now this sum this sum can be added in order n time to get the using the array processors collaborative processors to get the sum of n to the power 2 point 5 elements. So, the complexity becomes order n time so cost becomes number of processors n square and time to solve this problem so, n to the power 3. Now you observe that if I have n to the power 2 point 5 number of elements, to find the sum of this n to the power 2 point five elements using n square time it takes order n time and cost is greater than the sum of n square elements.

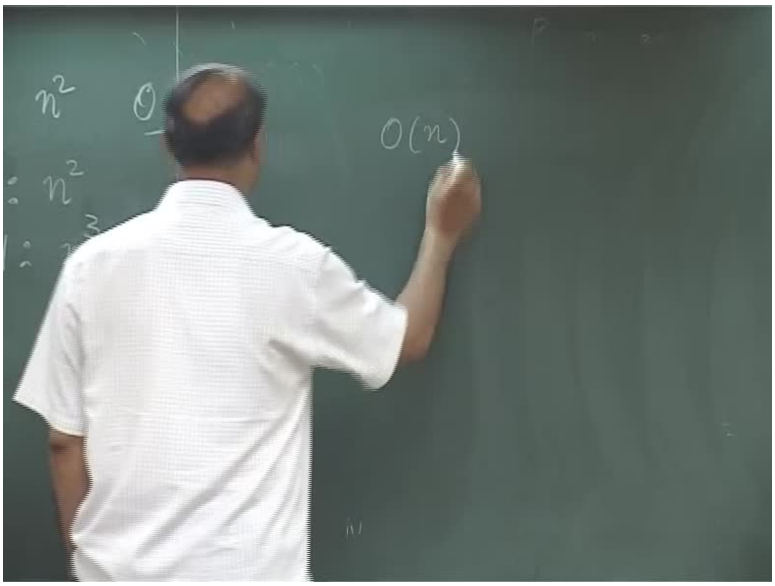
(Refer Slide Time: 57:24)



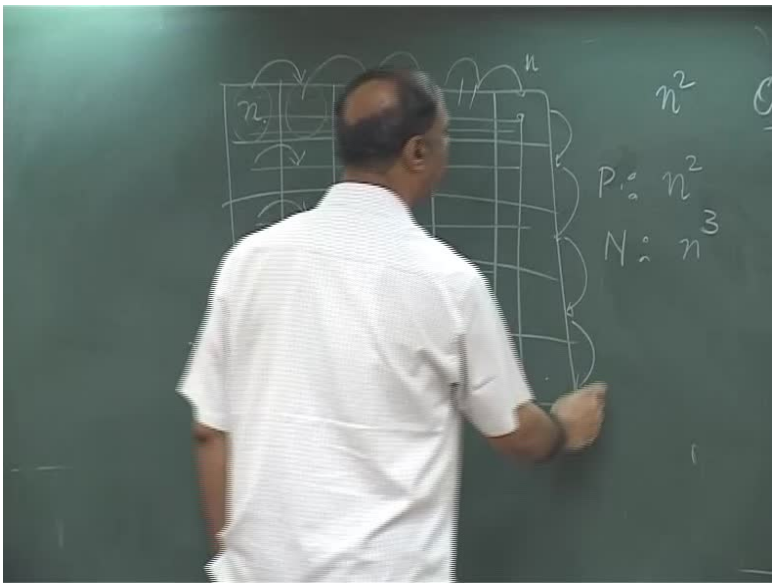
(Refer Slide Time: 57:24)



(Refer Slide Time: 01:01:07)

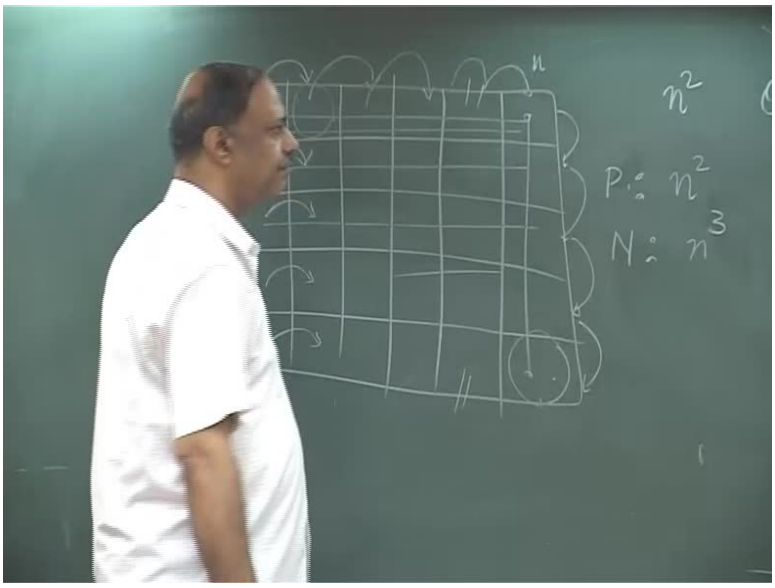


(Refer Slide Time: 01:01:37)

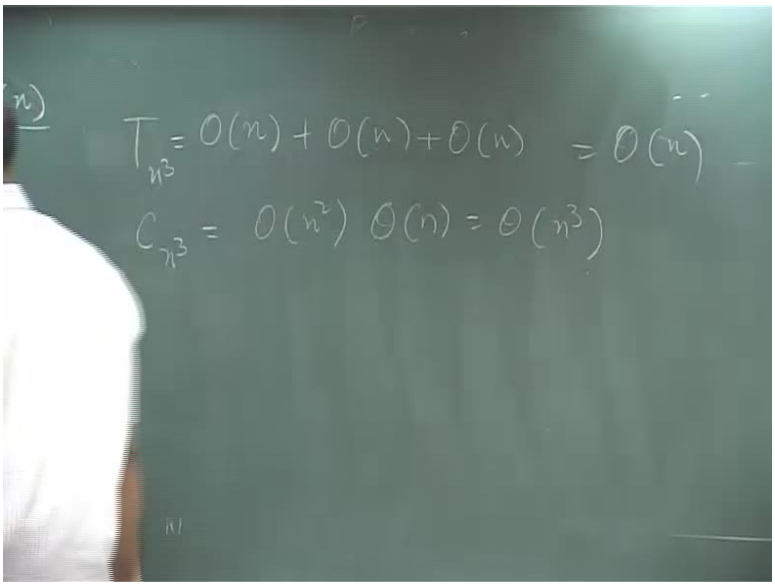


Can you do better than that can you get cost of parallel algorithm using this idea? So to do that let us assume there are n cube elements. Now this n cube elements are distributed among this n square processors. So, each processor contains order contains it n elements so the problem can we redefined as you have n square processors and n cube elements this n cube elements distributed in such a way that every processors contains n elements. Assign to it which takes order n time and then this sums to be added here one by one while you are adding this you can move the data as if they are linearly connected. The next page you add this, next page you add this ,and so on so after order n time you will be finding the sum is lying in this column then you move the data here then this linearly you follow it and you get another order n time.

(Refer Slide Time: 01:01:52)

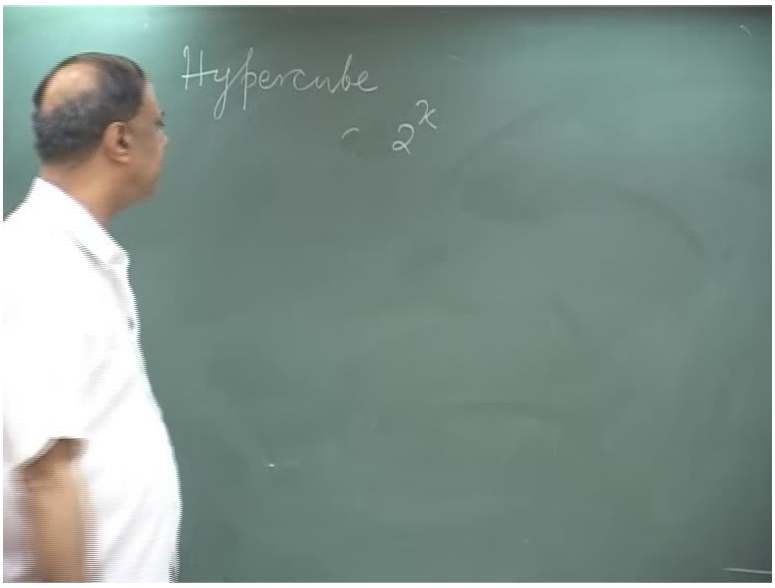


(Refer Slide Time: 01:02:15)

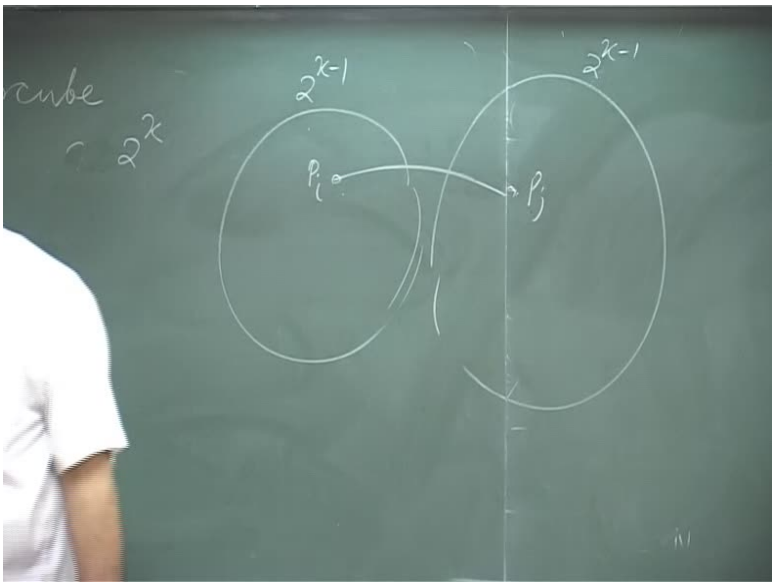


So, you have to do that you basically you need order n time to find the sum of n cube elements which is stored here. So, this is the time complexity to find the sum of n cube elements using n square processors and the cost to find the sum of n cube is becoming order n square processors we have used order n so, which is order n cube, which is the cost optimal which appoints to the sequential algorithm, because it will takes to find the sum of n cube index you need order n cube addition.

(Refer Slide Time: 01:02:40)

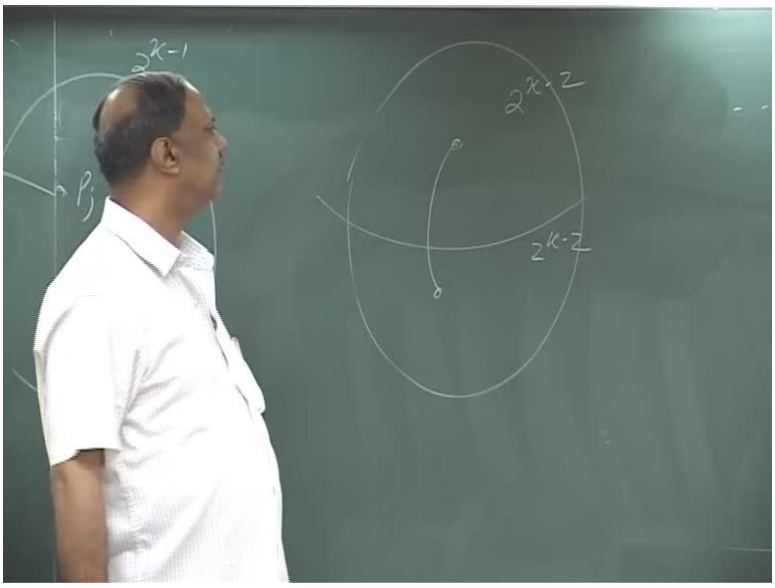


(Refer Slide Time: 01:03:34)



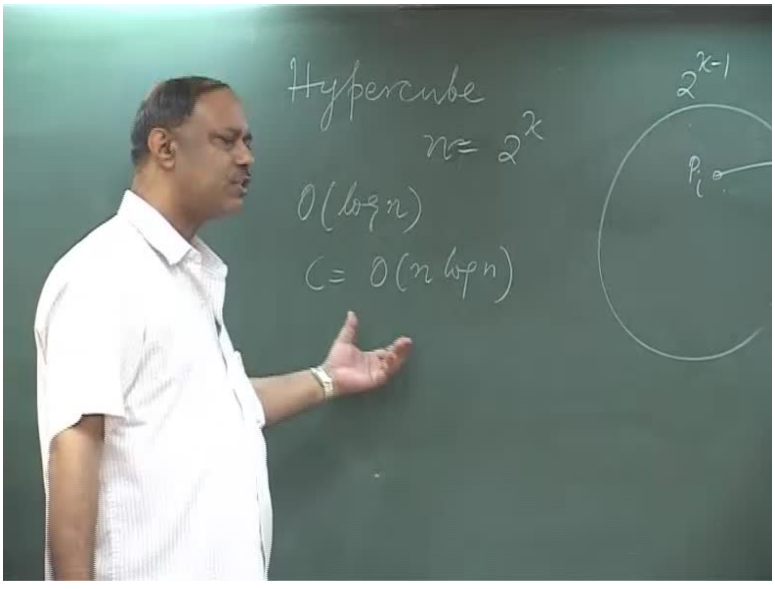
Next model is hyper cube. That is the very simple idea because we know that if you have hyper cube of size or dimension of hyper cube of 2to the power k processors. This can be done as a combination of tools are hyper cube each of size 2to the power k minus 1 and for one Pi is connected with Pj where that msb of this 2binary representations would be deferring.

(Refer Slide Time: 01:04:00)

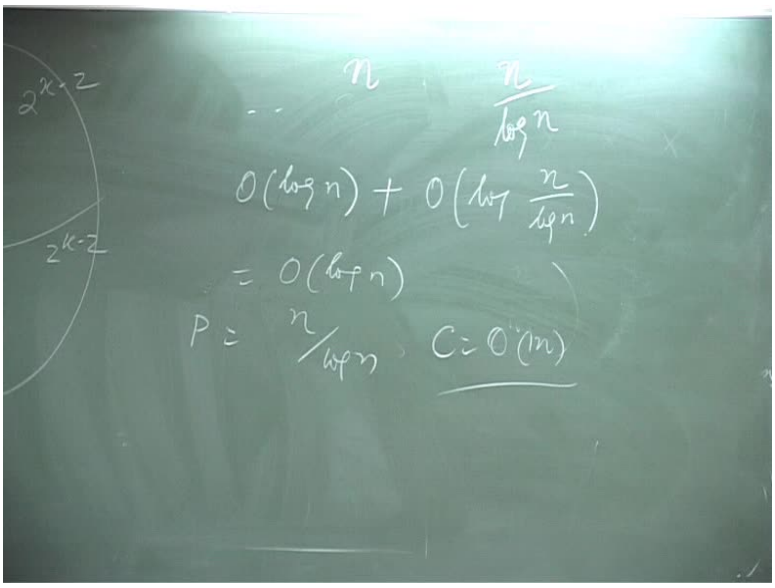


So what I can think about that that you can bring the content of Pj into Pi and add it. Similarly, for all P is of this mod i get the data from the other is counterpart and add it then what happens the dimension is reduced from 2 to the power k to 2 to the power k minus one and now you have hyper cube of size 2 to the power k minus one. This is divided in to the 2parts each of size 2 to the power k minus 2 and again there exists the connection between Pi with Pj. The 2nd msb is different and again the data will move off and add it. So, the size is reduced to the hyper cube 2 to the power k minus 2 and so on.

(Refer Slide Time: 01:04:43)

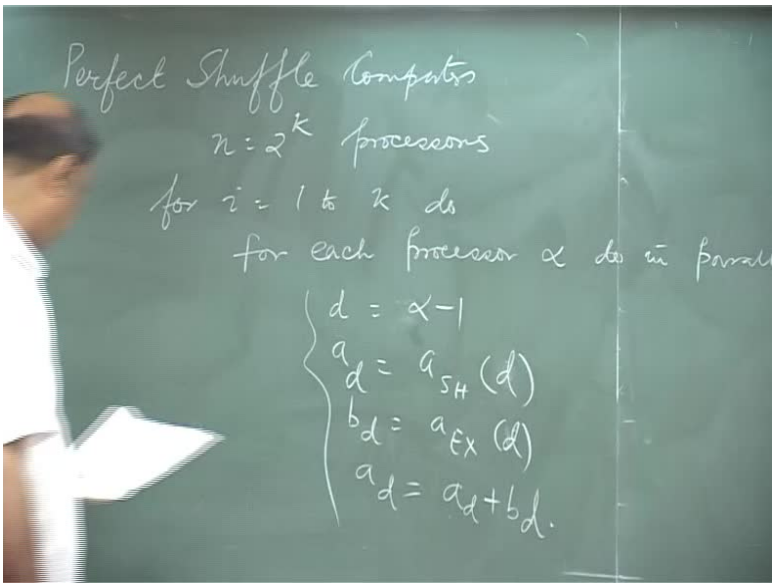


(Refer Slide Time: 01:05:41)



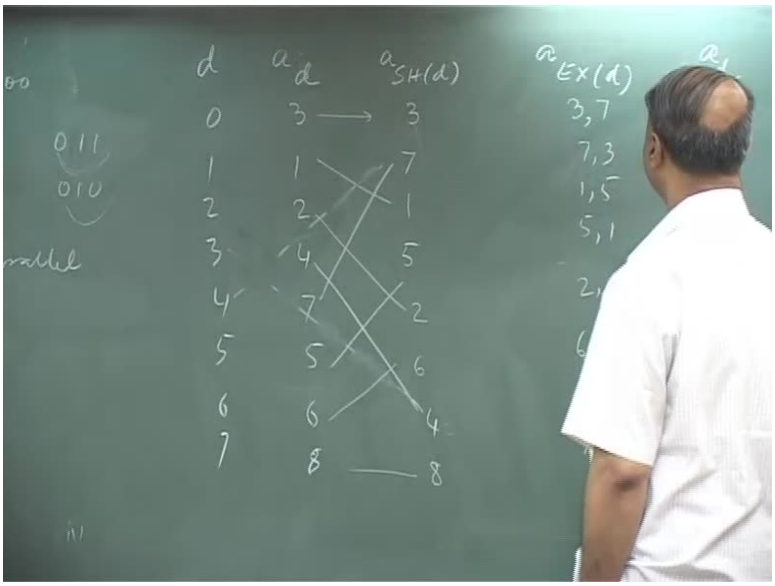
After log after k intention you find the data is a base level in the first processors P0. So, here you need order if n is equal to 2 to the power k then you need order log n time to find the sum. But, cost it becoming order n log n because n processors log n time. So, first is order n log n. Now in order to find the cost optimal algorithm again here idea is same. You divide suppose you have n elements you define your hyper cube of size n by log n each processors but, initial log n element and sequential with the find some of this log n elements so which take order log n time plus you need find the sum which take order log n by log n time. So, this things what are the log n time and, this is order log n, and you have used the processors is n by log n so cost is order m which is first optimal to find the sum of n elements using n by log n times on hyper cube.

(Refer Slide Time: 01:08:08)



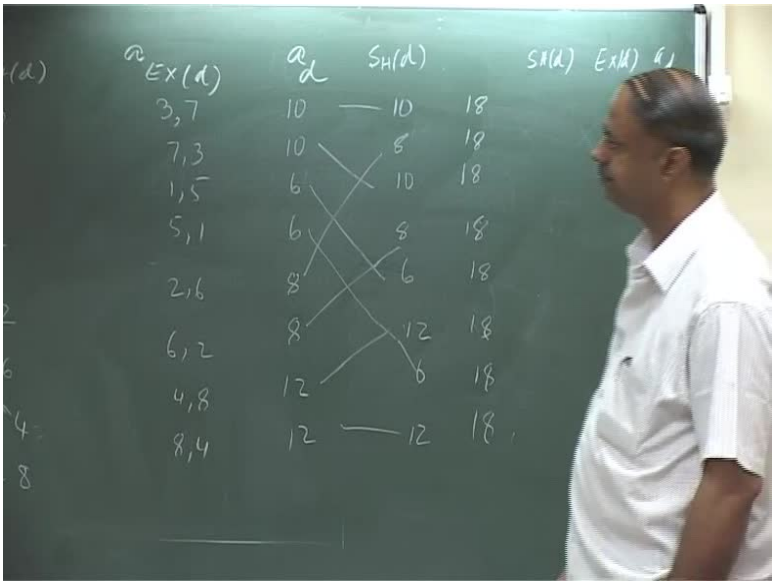
Today we will be finishing our lecture by considering another considering the sum of n numbers on another model which is found as perfect shuffle computer. Remember in the case of perfect shuffle computers it has a 3 connection for each processor it has 3 connection and add here will be using that shuffle and exchange operation to perform this a addition of n numbers. Suppose we have n is equal to 2to the power k processors. Here for i equals to one to k so, for each processors alpha to it parallel d is equal to 2to the power or this equals to alpha minus 1 sh equals to a shuffle of I, ex equals exchange of i sorry it should be d.

(Refer Slide Time: 01:10:55)



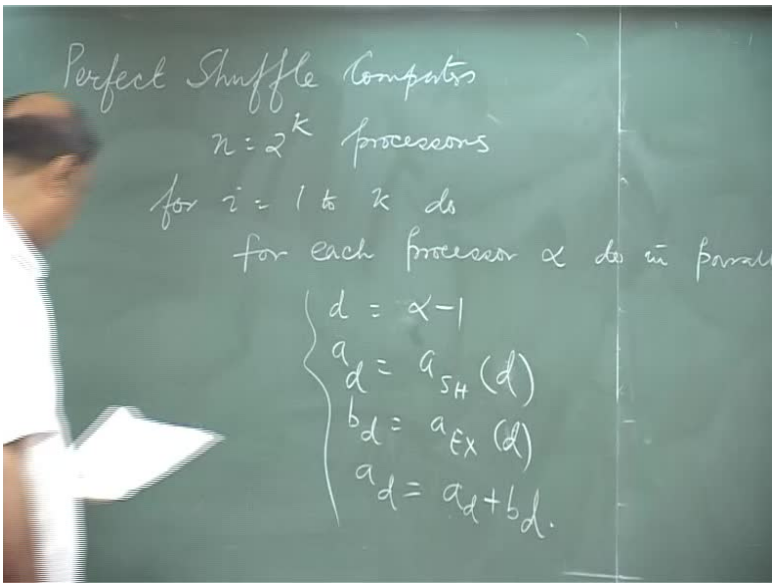
A d is equal to shuffle of d is equal to the exchange of d and ab equals to ad plus bd. So, basically if you have a j or j is equals to or d equals to d is equals to 0 1 2 3 4 5 6 7 and say the number is 3 1 2 4 7 5 6 8. Then a shuffle d, a exchange d and then a d, so, a shuffle d is nothing but, shuffle of 0is 0. So, it moves here shuffle of 1 it moves 2, then shuffle of 2 moves to 4, then shuffle of 3 moves to 6, shuffle of 4 moves to 1, shuffle of 5 moves to 3, shuffle of 5 moves to 3, shuffle of 6 and shuffle of 8 is here shuffle of 7 is 8. Now exchange it here you get 3 7 and 7 3 1 5 5 1 2 6 6 2 4 8 8 4.

(Refer Slide Time: 01:12:14)

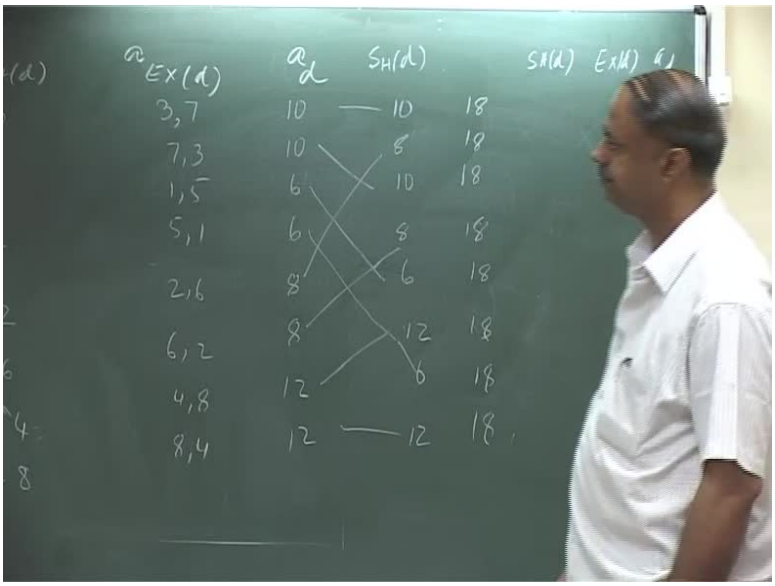


So, if I add it you get 10 10 6 6 8 8 12 12. Next one again you do the shuffle of d so it is becoming 10 it will come here, then it will come here 6 and this will come here no 8 will move up this 8 will be here and this will be here and this will be here. Now you perform the exchange of per shuffle in this become 18 this becomes 18 this becomes 8een this becomes 18. This is the next Sybil for which the extent of remainder you get the shuffle of d again and the exchange of d you will find it after you add, you will get 36.

(Refer Slide Time: 01:08:08)



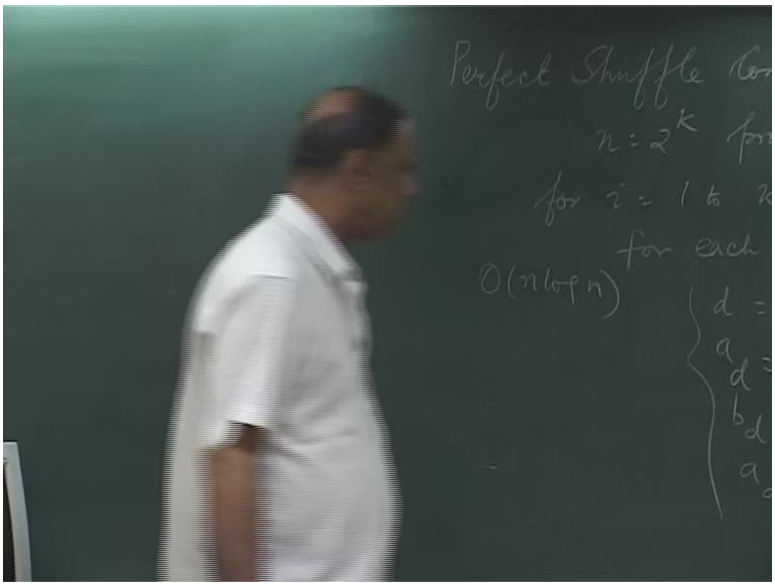
(Refer Slide Time: 01:12:14)



Ok so, this is the way we can do because you observe that it takes order k times to find the sum of 2to the power k elements and this can be also obtained as a cost optimal parallel algorithm. You have this format it is not cost optimal. It is order n log n algorithms. You do not get the cost optimal algorithms, you assume that you have n by log n number of processors and n elements. Initially, you make it n by log n groups each group is having log n cost. log n elements sequentially equal each processors obtain log n sum of log n elements and then you proceed it. You can easily show that it take common

there in time. by it dictates order n cost to find the sum of n numbers using the perfect shuffle computers.

(Refer Slide Time: 01:13:12)



So you can try there is a time to find the sum of n. Usually c r e w model c r e w model here the condition is little different that instead of finding the a i sum of a i for all i , I want to find out that d i has to be replaced by summation over a k. k is 1 to i which is known as a cumulative sum. Basically, I want to find out you have a 1 a 2 a 3 and so on. I want to replace a1 this by a1 plus a 2. This is by a1 plus a2 plus a3 and so on. So please try at home. It is possible to obtain to finding the sum of file given the elements you want to find out the sum of the cube or finding the cumulative sum of this n numbers.

Thank you.