Architectural Design of Digital Integrated Circuits
Prof. Indranil Hatai
School of VLSI Technology
Indian institute of Engineering Science and Technology, Shibpur, Howrah

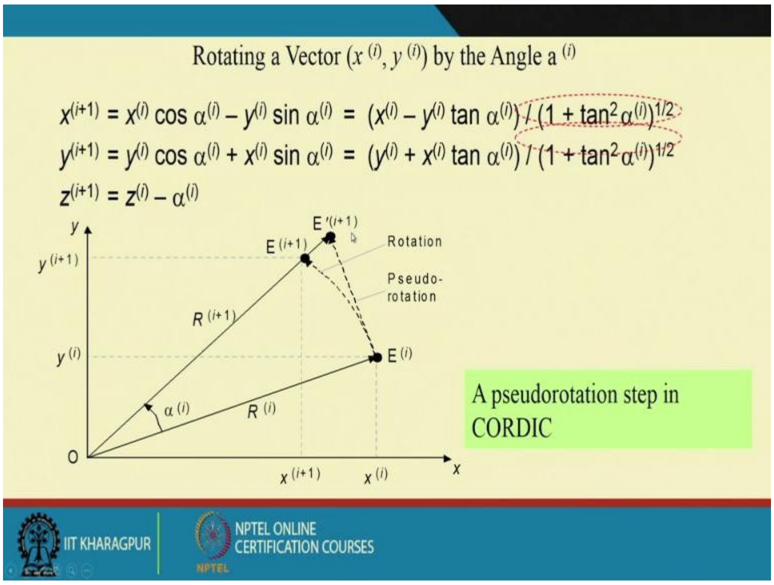
Lecture - 42
Cordic Architecture (Contd.)

So, welcome back to the course on Architectural Design of IC's. So, in the last class we are seeing this CORDIC Architecture. So, CORDIC has a vast application for the DSP algorithm. Basically, whenever this cost or the area hardware cost area is a major constraint. So, at a time we mostly preferred to use CORDIC instead of the in general MAC based approaches ok.

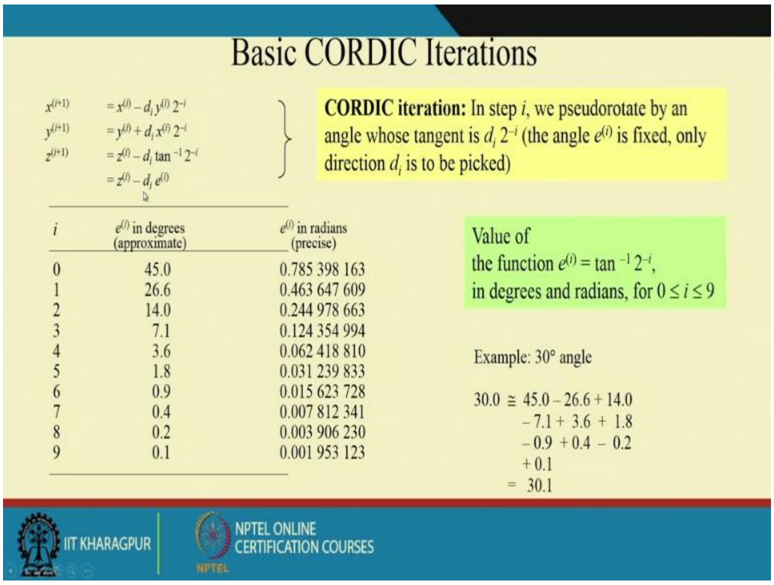
So, when to use and why we will use CORDIC and then what is the concept of CORDIC, that we are basically seen in the last class ok. So, whenever we are doing this is the nothing, but the initial vectors that we have to rotate by some pre-defined angle to get the desired this rotational angle ok. So, how to do that? So, we have seen that suppose we are we are having 1 initial vector which is x 1 and x y 1.

So, if I just rotate that if I want to rotate that another vector which is being rotated by the desired angle theta. So, that value can be written as this x 2 as in terms of x 1 cos theta minus y 1 sin theta. And, then this y 2 value can be written as for this y 1 cos theta plus x 1 sin theta. So, that we have already seen ok. So, based on that what we have not used the original rotation.

(Refer Slide Time: 02:03)



(Refer Slide Time: 02:11)



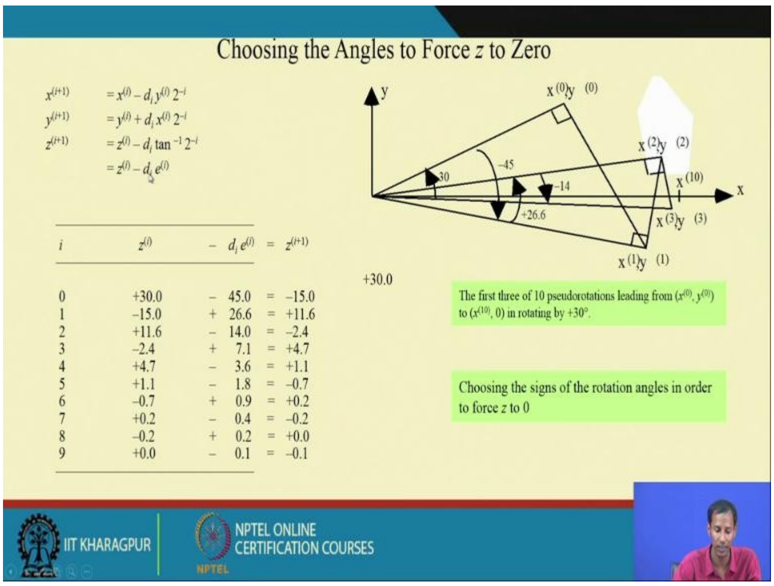
We have used or we have introduced the pseudo rotation. So, what is the meaning of pseudo rotation? Pseudo rotation basically they are being introduced; that means, keeping this fact the values will be in terms of this 2 to the power minus i. Means, what actually what was my initial equation that was x i minus cos if x i plus 1 if I write it, or x 2 there we are basically writing in terms i, I if write.

So, at the time x i plus 1 that can be written as x i minus initially it was y i into tan inverse of alpha i right. But, this alpha i, that has being replaced by the values of this d i into 2 to the power minus i. So that means, now what I said predefined values of the angle. So, how we can calculate the predefined value of the angle so; that means, initially or from the beginning we are using this particular set, where the alpha or the tan inverse values tan; that means, tan of this alpha i. Those values will be depended on the 2 to the power i means, it will be depended on the 2 to the power; that means, this 0, then 2 to the power minus 1, then 2 to the power minus 2, then 2 to the power minus 3, then 2 to the power minus 4, something like this.

So, then what will be the corresponding weight; that means, values of this tan inverse of 2 to the power minus i. So, we can easily get; that means, as this number is basically fixed so; that means, from the beginning. So, this I know. So, this number I can write in terms of degree. So, tan inverse of 2 to the power minus i, when i equals to 0 means this is tan inverse of 1 means that is 45 degree.

So, tan inverse of 2 to the power minus 1 means that is tan inverse of 0.5, the values is 26.6 degree. So, these are the basic predefined elementary angle, which I have to store somewhere to calculate any values of the theta. So, how we can calculate? Suppose this 30 degree angle is my desired angle of rotation. So, how we can do that? So, initially it will be rotated by 45 degree. So, then based on actually if I just go to the next slide.

(Refer Slide Time: 05:11)



So, initially this is my desired angle which is 30 degree. So, initially it is being; that means, the first set of this 2 to the power minus; that means, tan inverse of 2 to the power minus 0, which is tan inverse of 1; that means, it will be shifted by 45 degree ok. And, basically what will be the direction of that basically depended on the values of this z i sorry z i plus 1 ok.

So, what I said in the previous class, I said z is basically holds the value, that in each of the iteration how much value or how much is the angle I am basically moving. So, depending on the Z, if the values are coming in the negative so, this now this di value is the sin of this particular angle ok.

So, if this di equals to; that means, if this z is negative. So, at the time di will make this particular equation positive and this will become negative ok. So, how to do that? So; that means, this is the equation, that is x i minus d i 2 to the minus i of y i and i the y i plus 1 equals to y i plus d i x i minus into 2 to the power minus i and z i equals to z i minus d i of e i. So, e i in each step it is being calculate.

So, initially what is this? This is 30 degree so; that means, 30 degree means this is positive and d i is also positive. So, this is 1; 1 means 30 is the z i, z i minus d i into e i, e i is the elementary angle. So, first elementary angle is 45 degree. So, from 30 degree 45 degree will be minus ok.

So, z i plus 1 will be initially 30 and after first iteration it will be minus 15 degree, because 30 minus 45 is 15 degree right. So, then this 15 degree will be the next corresponding values of the z i ok. So, minus 15 comes over here. So, minus of these now it will became this equation plus or positive, because this is mean negative means the sign of this will be negative means negative, negative it will be positive.

So, what is the next secondary elementary angle that is 26.6. So, then this 15 degree will added 26.6, which is this 11.6 this results 11.6. So, now, 11.6 is my in the next equation z i value current z i value ok. So, then as this is positive, what is the next elementary angle that is 14 degree.

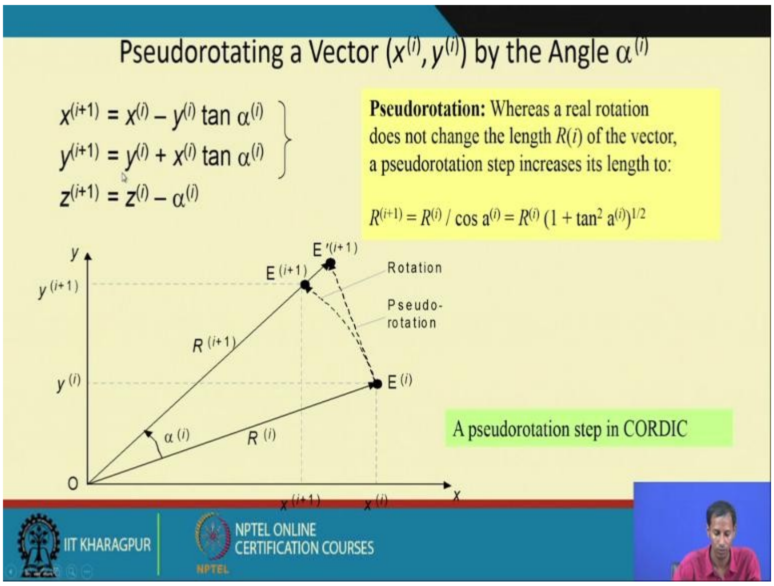
So, now, 14 degree will be minus. So, minus of this again indicates minus 2.4. So, 11.6 minus 14 that is 2.4; so, 2.4 again so, that will be added with; that means, next sorry this fourth one fourth elementary angle, which is 7.1. So, 7.1 is plus 4.7, then plus 4.7 there is minus with this 3.6 3.6 1.1 in this manner. So, unless until I am getting these particular thing as 0 ok.

I will try to move the corresponding vector's position in this manner ok. So; that means, what is the; that means, the final values are like this. And, if I consider the; that means, the word length of that is 10 bit. If, I consider more than this, these value will be more and I have to do this iterations more on this, but at the time as these value becoming very much less over here or it is nearer to 0.

So; that means, I can basically say that whatever value or the corresponding; that means, this x i value and y i value I will get at this particular location. So, that will be my desired sin theta and cos theta value. So; that means, we have started with the initial angle and then we are just rotating in this manner. So, here you see initially this is 30 degree, then this is minus 45, then this plus 26, then again minus 14, then 7.1, in this manner it will this vector will just rotate.

And finally, it will just lie on this particular line ok. So, it says that the first 3 of 10 pseudo rotation leading from x 0 and y 0 in rotating by plus 30 degree. And, this choosing of the sign of the rotation angle in order to force z to 0 so; that means, here what is my corresponding low case; that means, why this z is intended to be 0. Because, here what is happening we are considering the; that means, whenever if I just see this equation ok.

(Refer Slide Time: 11:16)



If, I just see this equation. So, at the time what is z i equals to z i plus 1 equals to y z i minus alpha i, when this z i equal to alpha i. So, at the time it will become 0. At, that time only I can say that ok. So, this is whatever value I am getting. So, that value is same as the corresponding desired angle, which is alpha here.

So, that is why we are basically trying to set this value or we are just making this z value to 0 or nearer to 0. So; that means, whatever x and x i plus 1 sorry x and y value, we are getting at this particular iteration, those values are not correct, because this corresponding z values are not nearer to 0.

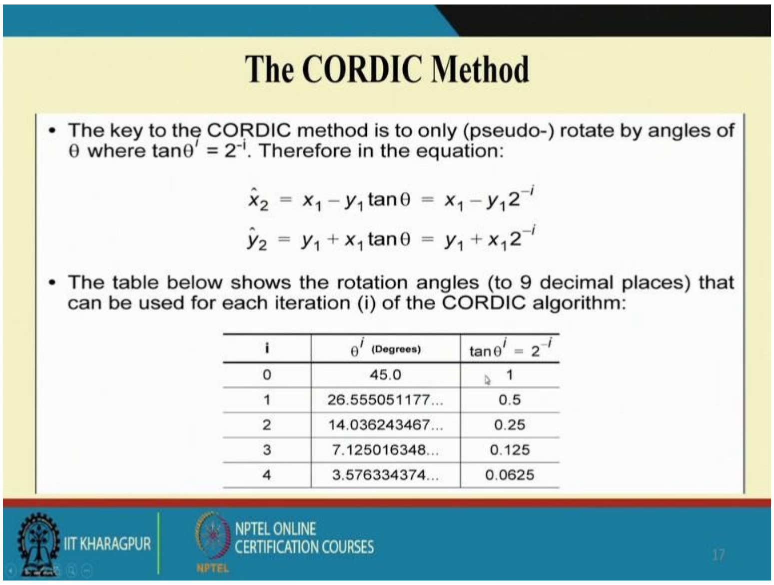
So, here you see these values is basically currently holding the values of 0 and whatever the x and y value I will get. So, that is the final values of x theta sorry sin theta and cos theta. So, in this manner and why, we have intentionally doing this. This d i introduce this system d i into 2 to the power minus i, because 2 to the power i means what it is being divide by 2.

So, in 0 means what this is 1. So, it is just directly x i minus in the first iteration is it will be x i x 1 sorry x 0 minus y 0. In the next it is what x 1 minus this is 1 means by 2. So, it will be y 1 by 2. In the third it will be x 2 minus y 2 by 4, then the fourth it will be x 3 minus y 3 divide by 8. Why, because this power of this is also increasing.

So, divide by this means what? Divide by this 2 to the power I means just nothing, but the shift of the LSB. So, shift in the; that means, in the right direction if you just to; that means, if just discard the LSB position or shift the data in the right; that means, in the right direction.

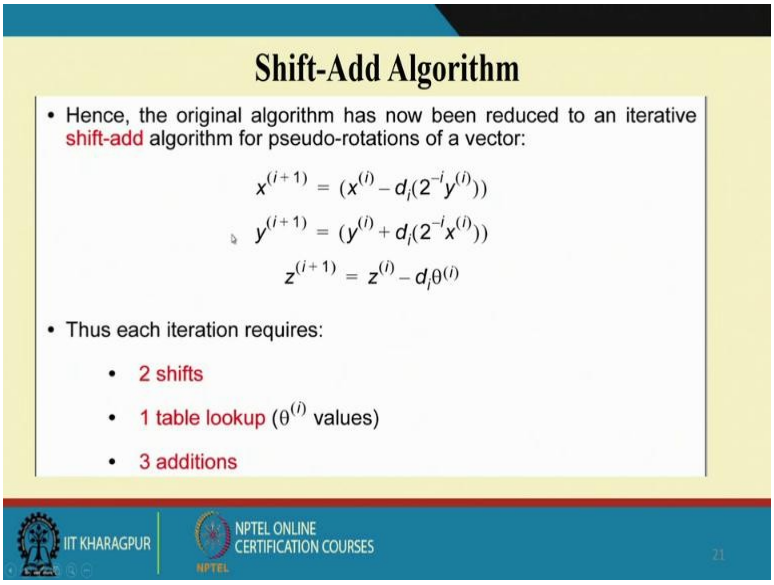
So, you will get divide by 2 very easily, you do not require any hardware shifter at that time. That shifter means any shift register to do that only hardware shifting or just; that means, discarding of the LSB, you will get it you do not need any extra logic gates to do that particular operation ok. So, here only one, what thing you require that you require only the subtractor or addition operation.

(Refer Slide Time: 14:22)



So, that means, so this is the; that means, the angle.

(Refer Slide Time: 14:34)



So, if I just that means, summed up this algorithm. So, at the time the algorithm is basically written as x i plus 1 there is equal to x i minus d i into 2 the power i minus y into y i y i plus 1 that is y i plus d i into 2 to the power minus i of x i, where z i equals to z i plus 1 equals to z i minus d i of the alpha i or theta i whatever is the desired angle we are choosing.

So, these 3 equations what it require specifically? It requires this 2 shift, this 2 shift operation because of this 2 to the power i in this 2 equation. And then 1 lookup table, lookup table means this is the norm based kind of structure, where we have to store this elementary values, which is nothing, but this theta i over here.

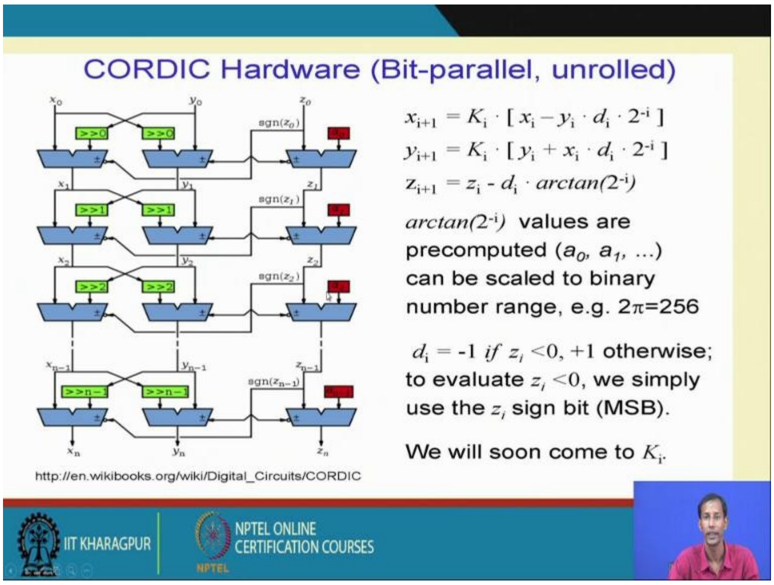
Elementary angle means whatever the values of are like is, 45 degree, then 26.6 degree, then 14 degree, then 7.1 degree. So, this value of this angle value has to be stored somewhere. So, these you need for this theta i value you need to need 1 lookup table to store all this values. And, then what you need you need adder or subtractor operation. Why adder or subtractor, 3 adder or subtractor, because based on this d i it will be either addition operation or shift; that means, subtraction operation.

If this d i values is; that means, that means the corresponding value of this is positive; positive means this will be minus if, this value is negative. So, at the time this will become; that means, addition this will become subtraction ok. And, this is also addition. So; that means, I need 1 adder subtractor circuit.

So, where this control will be done through the corresponding sign of this so, let us consider so, here z i plus 1 when can I recognize this z i plus 1 equals to that is minus negative or positive, if I considered the MSB of the z i plus 1. So, MSB of z i plus 1 equals to 1 means this indicates this is one negative number, because that is the sign bit considering that sign bit it indicates that if it is one. So, it indicates that this is negative. So, at the time it will be positive it will be negative.

So, if this is 0. So, at the time this will be the negative and this will be positive and this will be negative too. So, based on these I need this is not 3 additions I need 3 addition come subtraction operation.

(Refer Slide Time: 17:27)

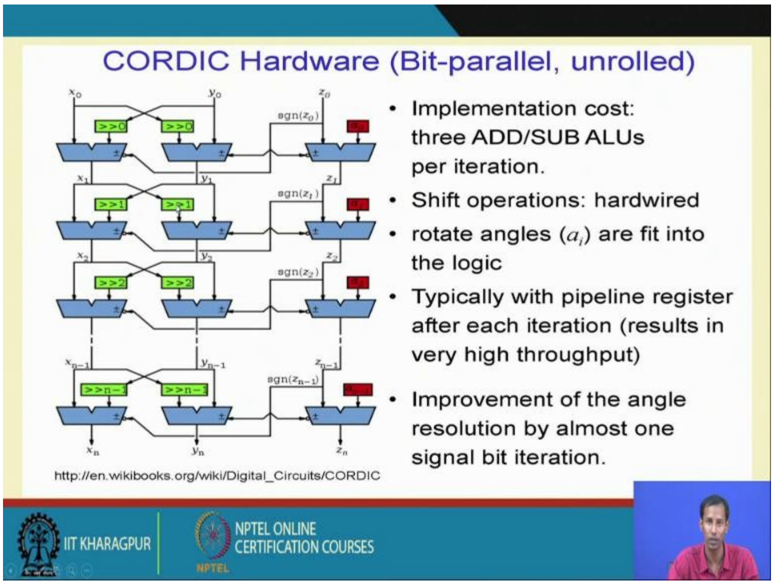


So, if you just see the architecture of this of this particular equation. So, here what is there; that means, x i plus 1 equals to x i minus d i 2 the power minus i of y i. Now, what is i, i is the number of iteration I will use. So, number of the iteration how it depends? It depends on the; that means, what is the length of the; that means, theta I am using.

So, if I use the theta length of let us consider 10 bit so; that means, the tenth is the iteration of what I have to do? Ok. Then if I considered; that means, suppose the CORDIC length is of 24 bit so; that means, 24th number of iteration I have to do. So; that means, it will be done as this 2 to the power minus; that means, 2 to the power minus 0 first then 2 to the power minus 2, then 2 to the power minus 3 2 to the power minus 4. At the time a lookup table entry will be of 24 numbers and each of this tan inverse of 2 to the power minus i ok.

So, here you see so, so, whatever is the; that means, with this length we are using. So, based on that there will be so, at the very first it will I goes to 0 means this stage will be done. Then, it will be x 1, then it will be x 2, then it will be x 3; up to how much up to if I consider for n bit. So, it will be up to x n minus 1, y n minus 1, and z n minus 1. So, what is this is basically parallel architecture? Ok.

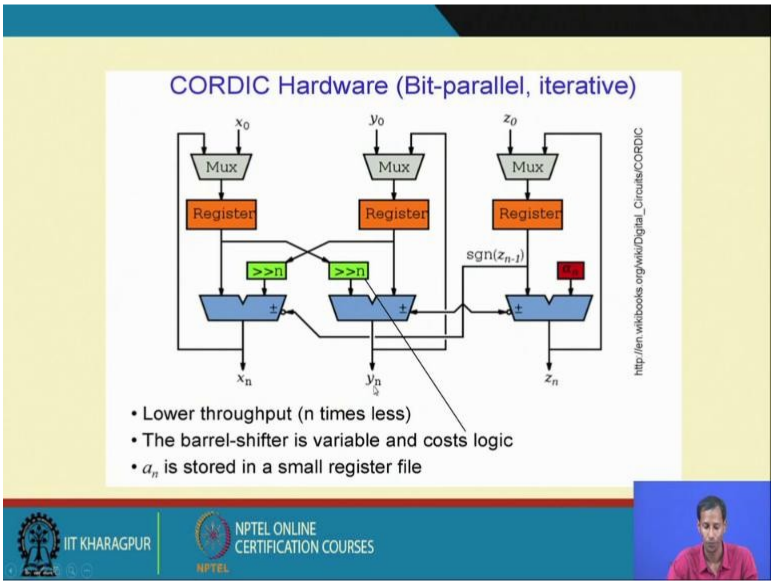
(Refer Slide Time: 19:25)



And, then if you considered this parallel architecture. What it requires? It requires this adder subtractor unit ok. So, adder subtractor unit, then this shifting operation in each of this how much bit it will be shifting, it will shift 0 at this particular, 1 at this particular, 2 at this particular why, because you see this 2 to the power i means, i is for 0 then for in the first stage it will be require 2 to the power minus 1, then 2 to the power minus 2. So, here you see at this is shifted by 0, then this is shifted by 1, this is shifted by 2, and here it is shifted by n minus 1. So, what I said this is nothing, but the hardware shift ok.

So, you just truncate the LSB. So, you will get the results as divided by 2 divided by 4 divided by 8 something like this. And, then what I required, I need to store these corresponding values of this alpha. So, at the alpha 0 then alpha 1 alpha 2 in the same manner of alpha n minus 1 so, if I consider n bit increase of lookup table will be there ok.

(Refer Slide Time: 20:39)



So, then, if I consider this iterative architecture so, here you see each of this stage is very much similar. And, as I am considering this is as a parallel structure. So, that is why it is all the stages are being implemented in 1 particular that means, in 1 flow, but in iterative in one of this one of this particular stage. Now, it can be re-runned, for this n time n minus 1 sorry n times, if I run this particular one stage n times I will get the same results, whatever I am getting the from this circuit. So, at the time what will happen at the time the hardware requirement that will be less? Ok.

So, hardware requirement will be less, because I have to use only 3 addition operation and this registers and muxes to select, whether I have to choose this intermediately results or I have load the initial x 0 and y 0 value ok. So, this muxes is basically once it takes the z; that means initial values of xyz. And, then it takes another value of that; that means, in the intermediate stage; that means, from after 0 then again it will choose this then 1, then 2; that means, it runs in a loop.

So, then what it; that means, the advantages of this particular circuit is that, hardware wise it takes very less amount of; that means, component. Why, because here if I need this n minus 1 number of stage; that means, so how many number of additions subtraction I required? That is 3 into n minus 1 so; that means, 3 n minus 3 number of this add or subtractor.

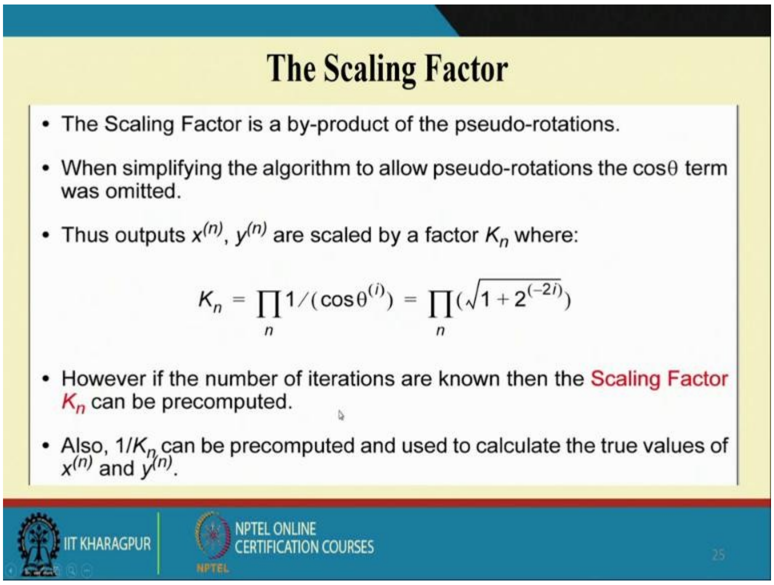
I require over here, but here only I required 3 add or subtractor circuit and apart from that I need extra 3 mux which is not required in this case ok. So, that is why, but at the time where I am loosing; that means, this throughput wise it is basically less. Why less because, I have to wait for the tan cycles suppose here in this particular example we have seen that 10 iteration.

So, after that means tenth iteration only I will get the output at this particular location. So, for tenth clock cycle I have to wait. So, that is why this; that means, that is the latency time or the there is a latency of this particular circuit. So, up to that much time I have to wait to get the results of that though I have put the inputs ok.

So, that is why this and throughput means what that, it is the time to get the output after applying the input; that means, at what time I have applied the input and what time getting the input the difference between that that is the throughput of that. So, here this throughput is basically less. And, how many times it is n time less because n times, it has to the same loop has to run for completion of the whole operation ok. And, here for this actually the shifter here it is not hardware shifter, because in each stage it will be changed.

So, I need barrel shifter ok. And, the barrel shifter is basically in the hardware shifter what I require I do not require any logic ok. Only the truncation of the LSB, that is sufficient, but the barrel shifter it requires some of the logic to do this; that means, here what is the in each of the stage this is barrel shifter or I can say that is a variable shifter; that means, depending on the stage the number of shift operation will be changed ok, so based on that now this question.

(Refer Slide Time: 24:52)



And so, this is the; that means, the advantage of iterative circuit cost by with this advantages, but throughput wise or this it is not greater than this. And, here what is the problem in this particular circuit? As, you see there are n number of stage in any of the path; in any of the path ok, not any of the path, because the sign of this again it is depended on this ok. Means, what basically it will; that means, depended on this the operation of this again this; that means, is depended the output of this or the sorry this is input is basically output from this particular operation ok.

So; that means, and then it is again passing through n number of stage ok. So, that means, the as the stage or the iteration number increased in this so, the number of; that means, the delay amount is also increased. So, the delay amount increased if the n is more. So, the number of delay or the; that means, amount of delay also more.

So, at the time what I can do? If, I put you see this one very regular structure. If, I make pipe line of this structure, pipe line means in each of this point if I put registers actually what is happening? Here this is basically totally data driven, means what whenever there is change; that means, whenever there is change whenever there is this x 0 and y 0 and this value is available.

So, it will immediately; that means, after whatever is the delay for this particular adder substractor after that time, the output will be available over a, then again it will start it is operation, then again it will start it operation after this much of delay. Then again this, then again after this much of delay again this something like this will run sequentially, considering only the delay of this, but from the data driven of this particular circuit. So, that is why this is purely comminatory circuits. So, that is why this is, this circuit is basically data driven.

So, instead of that if I put in this particular intermediate point, if I put register; that means, this point if I put registers, in this point if I put register, in this point if I put registers. So, at the time this will be broken, because of this particular register and at the time from the data driven property of this circuit now it will dependent on the clock driven.

So, in each of the stage now at they will not be adjacent to each other, they will be independent to each other; that means, this particular stage will execute at if that is a zeroth clock cycle, this will be execute at next clock cycle, which is 1 clock cycle or the first clock cycle. Then, this will be executed at the third one, this will be executed at n minus 1 time; that means, 1 clock cycle. So, in this manner this will be changed and at the time the delay will be lesser. So, delay will be just only this adder substracor; that means, that is only dominated by mostly this add or substractor delay ok.

So; that means, at that time my area will be more. How area will be more, as I have to introduce again 3 n minus 1 number of extra registers to do that, but speed wise it will be on the higher side ok. So, that is why for it is depended upon you, whether you have to use this pipe line CORDIC. So, at the time this structure will be named as pipe line CORDIC. So, pipe line CORDIC architecture you have to use. So, pipe line CORDIC architecture means cost wise it is not that much; that means, constant is not that cost if you want to improve the that proper; that means, the speed of operation.

So, at the time you make it pipe line CORDIC, but if you want your cost as a major constant. So, at the time this architecture is very much efficient you follow this architecture. Obviously, the throughput will be lower, but area wise this is has the most advantages ok. So, for today so, this is not only the fact that after this is the hardware I am getting. So, this is not only the fact after doing the pseudo rotation or this equation I am getting it, but there are several other teams also and what I said that many of the arithmetic operation we can performing using CORDIC.

So, how we can do and what are the other thing which are associated with the CORDIC so, that we will see in the next class.

Thank you for today's class.