Architectural Design of Digital Integrated Circuits

Prof. Indranil Hatai
School of VLSI Technology
Indian Institute of Engineering Science and Technology, Shibpur, Howrah

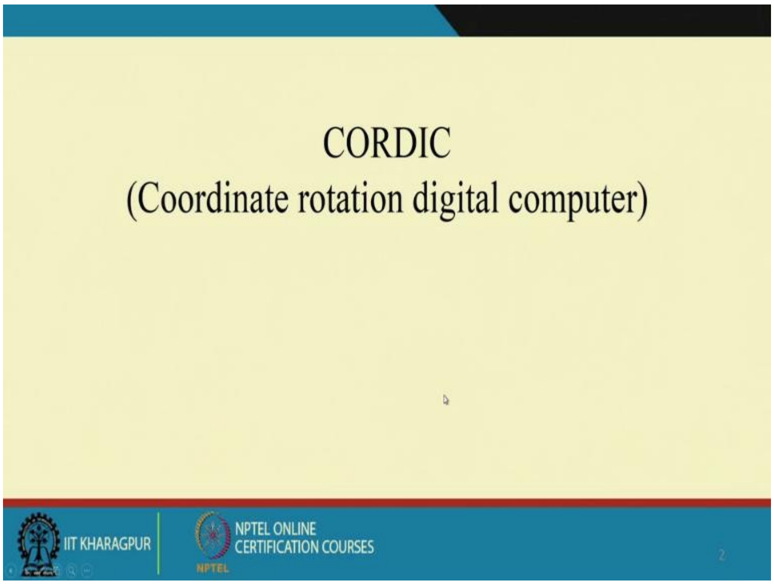
Lecture - 41
Cordic Architecture

Hello everyone, welcome to the course on Architectural Design of ICs. So, in the last class we have seen different kind of architecture of adder circuit, multiplier circuit and then we have seen that, how fix point representation can be done. So, these are the basic operators design, till now we have already learnt how to do that? It is not that only this much of the architectures are available on the literature or available on the market. This is just the few examples, there are many more, this fields is so much rich there are so many architecture, so many people have worked on this particular to find out the best possible architecture for their particular application ok.

So, there are several architectures available on the literatures, but I have just give you some brief about those architectures that, how we can improve or what are the techniques or how we can proceed to get different architecture for different application aspects ok. So, that we have seen till now. So, now, we have learn the basic operators design, it is not that only this much operators are there, but still there is so many left; that means, sometimes we need square rooting circuit, sometimes we need divider circuit, sometimes we need logarithmic function to be done. So many others arithmetic operations are also required to implement the function ok.

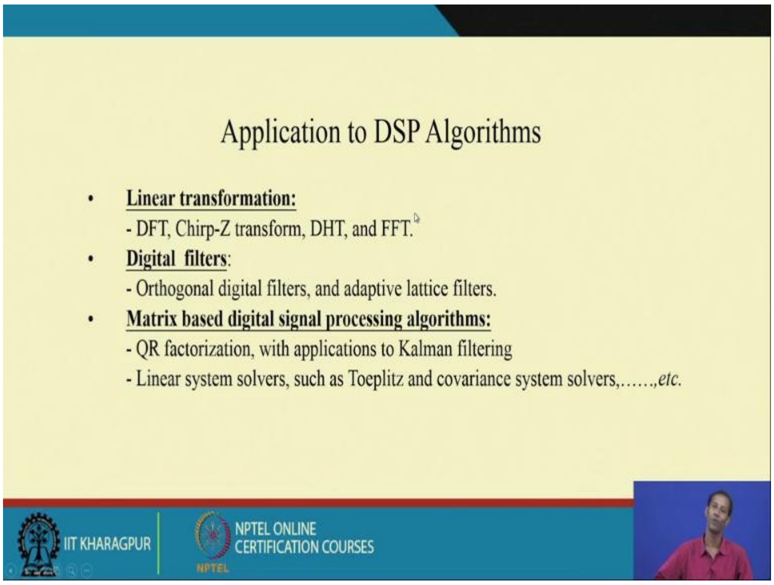
So; that means, in the function there are several other operators too which have been used. So, considering this fact here, today we will learn one interesting; that means, algorithm to implement different kind of arithmetic operation ok. So, the structure is this or the architecture is Cordic architecture. So, first we will see, what is the meaning of Cordic ok? So, what is the algorithm of Cordic and how we basically calculate different operators, arithmetic operations, how it does? And what are the types of operation it can do? Then from that particular point, we will see that different kind of architectures to develop the corresponding hardware for Cordic ok. So, let us start with today’s topic, which is this Cordic architecture ok.

(Refer Slide Time: 03:09)



So, the full name of Cordic is basically one acronym. Cordic is basically the full form of Cordic is coordinate rotation digital computer ok. So, that is why here, you see this from coordinate co from rotation this r and from digital di comes and from computer this c comes.

(Refer Slide Time: 03:41)



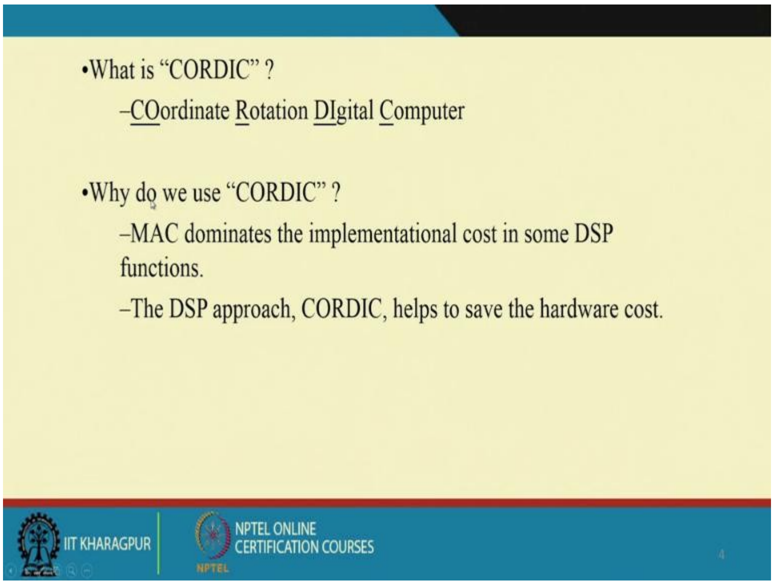
So, the full name is Cordic ok. Then where are the applications or what is the application area of Cordic? In many DSP algorithms or to implement many DSP function, we used this Cordic algorithms, because what I said in the beginning that using Cordic, we can implement different kind of arithmetic operations ok.

So, it is being used in any linear transformation like this DFT then, Z transform then, DHT then, FFT. So, all this transform, they used this as the linear transform. So, in the linear transform basically, Cordic is being used mostly, then not only these specific to this linear transform, it also used in digital filters like this orthogonal digital filters and adaptive lattice filters ok. So, these are the filters, which are available in multi rate digital signal processing, which have been used in multi rate digital signal processing or either in digital signal processing, then matrix based digital signal processing algorithms like, this QR factorization with application to Kalman filtering, then linear system solvers such as this Toeplitz and covariance system solver.

So; that means, for this matrix based whether, there is multiplication or this factorization. So, for this purpose also we use Cordic or we have noticed a plenty of use of Cordic for deriving the hardware architecture of this particular functions, which are like this QR factorization or FFT or DHT or DFT so; that means, this application area itself, it shows that Cordic is being used for various applications in DSP algorithms mainly not only, it is specific to DSP it for a wireless communication also we use mostly Cordic ok.

So, then what is the advantage of Cordic? Then how, what is the algorithm behind that? That we will see in this later of this course ok.

(Refer Slide Time: 05:55)



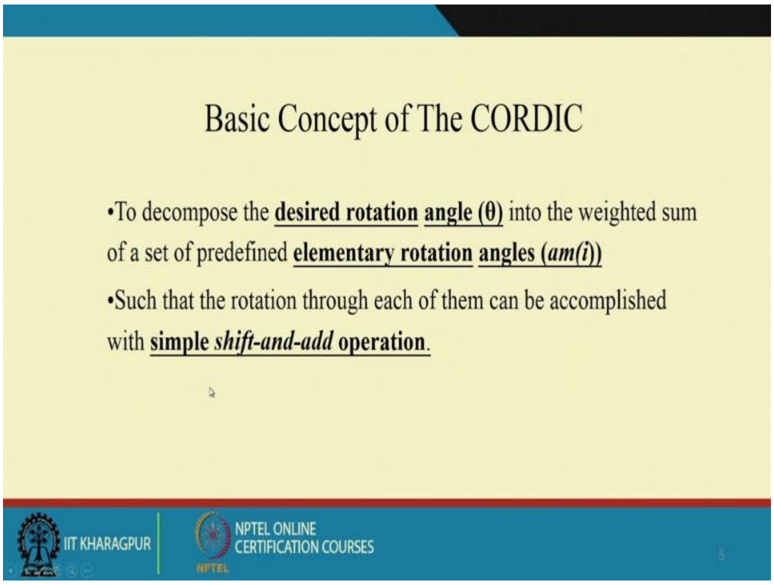
So, the thing is that, why do you use Cordic? Why or what is Cordic? That we have already seen why do you use and when do you use COERDIC. Is that this is not only specific to why do you use, when do you use? Basically all the DSP algorithms, they are mainly this MAC based, MAC based means the operational mostly most likely multiply and then accumulate.

So, multiply and accumulate means. So, some multiplication like, if I say that this digital filtering, so, what we do? Basically that is nothing but the summation of the or multiplication of the input and the coefficients, so in FFT also what we do, this is twiddle factors, their being in each stage, their being multiplied with the corresponding inputs in each of this each of this particular DSP function, they are most likely, they do this multiply first then they add those results and then they again process for the further stage.

So, that is why we need this multiply and accumulate architecture. So, instead of this MAC based operation they are mostly used for this, DSP algorithms. So, this can be replaced easily. So, in multiplier and accumulate for multiplier and accumulate what I need? That means, I need 1 multiplier and then I need 1 adder; that means, accumulator means either along with 1 registers so; that means, now if the word length for actually, what is the problem of multiplier? Is that, if I consider n bit into m bit. So, at the time the product will be, m plus n minus 1 that much bit, I require to hold the product results.

So; that means, the word length that is basically increased. So, after that as I am using this, adders circuit adder and then registers so; that means, automatically it is increasing the hardware for that. So, if use that kind of unit let us say, 100 times or 1000 times so; that means, the area will be much higher. So, that is why this MAC basically, dominates the implementation cost in some DSP function so; that means, where this cost is mostly; very much critical. So, at that time we can replace this MAC by using of Cordic and then in the DSP approach, Cordic helps to save the hardware cost. So, that is why we use. So, when do you use? When we have that hardware cost is one of the major; that means, concern or major constraint in my design. So, at that time we can replace this MAC unit by Cordic ok.

(Refer Slide Time: 09:13)

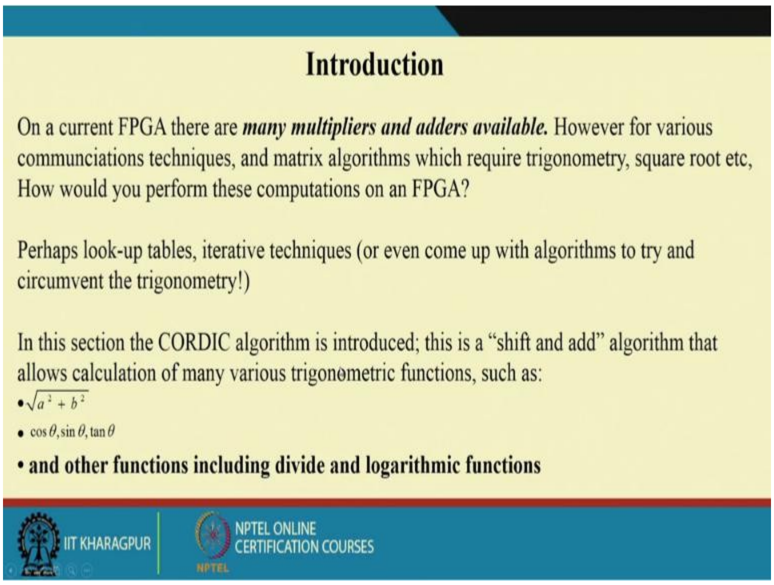


So then this, what is the basic concept of this Cordic algorithms ok. So, in Cordic algorithms, it says that to decompose the designed rotation angle theta into the weighted sum of the set of predefined elementary rotation angles, that means, what is the meaning of this line? Is that suppose basically, we use to calculate mainly, we use there are different kind of that arithmetic operations you can do, but mainly Cordic is being evolved or Cordic is being produced to generate the data of sin and cos ok.

So, sin and cos means related to or if I give any of the angles as the input I need the corresponding the results of cos theta and sin theta mainly. Cordic is being used or developed for use of this function in the calculator and that has been proposed by I think by J Volder ok, in I forget the year, I think I am having it. So, at that time I will say that, what is the exact year it as it is being proposed? this Cordic algorithm and here that this is the basic concept or this is the concept, that the if I need the rotation angle or the desired rotation angle, if I say that is arbitrary ok, that is user dependent.

So, if I say that ok, I need 90 degree, if I need 45 degree or if I need 30 degree, 60 degree any degree, any degree if I mention. So, at that time based on some predefined elementary rotation angle, it has to calculate the desired corresponding angle ok. How? So, we will see later and then what is the beauty of this? That this basically, this while we are calculating this sin and cos function, only this shift and add operation is required means what, I do not need any multiplication and; that means, the operation are much more simpler, it is not that much complex because, it can add based; that means, operation is the more similar one ok. So, that is why, we use or the use of this shift and add operation in Cordic, makes it lesser complex, that is lesser complex than the other circuits ok.

(Refer Slide Time: 12:03)



So, this is the basic concept of Cordic. So, in FPGA, this is not only that based on the FPGA, FPGA as well as you can go for the ASIC design or which is this standard cell based design. However nowadays in the current FPGA, this multipliers and adders, they are pre built, actually, we know this FPGA, we have we are having the pre built function right. So, at the introduction I think, I have already discussed that.

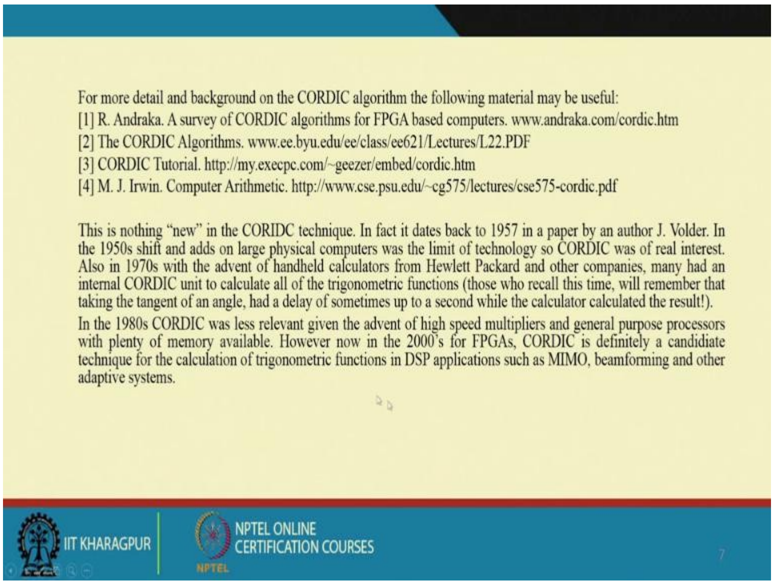
So, but nowadays, this FPGA are so much advanced, they itself contains the MAC unit from the beginning or that is pre defined in the corresponding hardware. Pre defined or pre designed from the beginning, it is being introduced in the FPGA itself, because at that time the performance of the MAC unit will be higher side ok.

But however, for various communication techniques and matrix algorithms, which requests trigonometry, the square rooting function, we need to perform this Cordic implementation then how we can do? That means, suppose I need to do this matrix multiplication or this matrix inversion, actually, if I go for the image processing algorithms; that means, if I am developing image processing algorithms, the hardware for image processing algorithms. So, at that time image processing means, it is basically 2 dimensional ok.

So, at that time I need the processing in the matrix form. So that means, I need the operation, which is not only sufficient to or it will be costlier, whenever we will try to implement those function using only this multipliers and adders. Then the other thing also for this square rooting, we have to develop the circuit for square rooting. Basically, this division and square rooting, we have seen multiplication and then squaring. So, division and square rooting, there the algorithms are very much similar; the operations are very much similar ok.

So, how easily we can do that? So using Cordic, we can do that. So, the Cordic algorithm based; that means, which requires only this shift and only this shift and add operation, it can calculate the various trigonometric function like the square root of a square plus b square then, sin theta, cos, theta, tan theta and then other function like this, division then logarithmic function implementation. So, all this can be done using this Cordic ok.

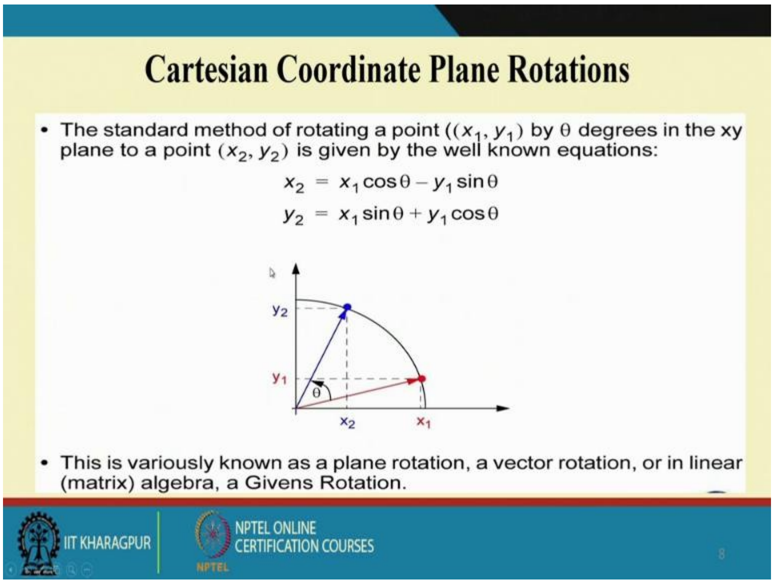
(Refer Slide Time: 15:24)



So, how can be done? So, here you can find some of that means, literature on this Cordic algorithm ok, which have been referenced here.

So, this is not only that, only this much are available, if you find on Google scholar or in IEEE or any other scientific journal, if you just search this Cordic, you will find various works of different hardware architecture for Cordic ok.

(Refer Slide Time: 16:10)

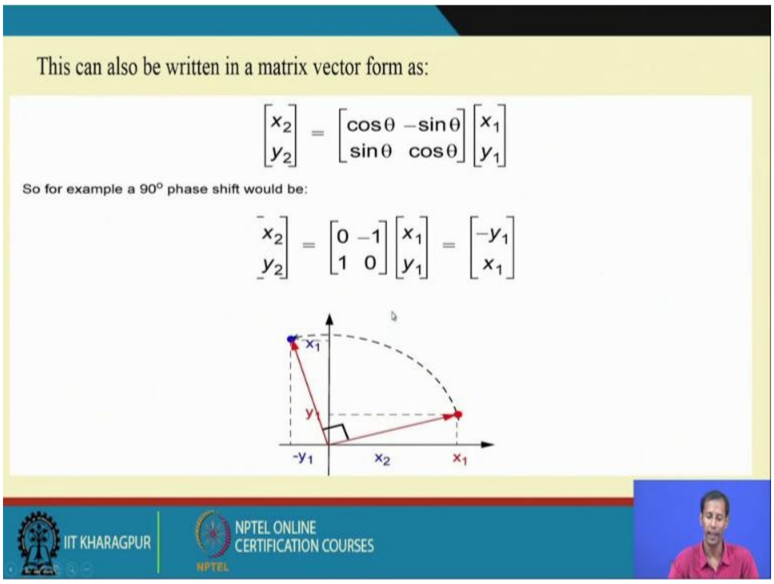


So, here you see in 1957 by the author Jack Volder, it is been introduced this Cordic technique and then how we can; that means, what is the; that means, how it works? What we are saying that, we can calculate this square root function, we can calculate sin theta, cos theta, tan theta. We can calculate this division operation, we can calculate logarithmic function, but how it works basically? Ok. So, mainly if you consider this, Cartesian Coordinate Plane Rotations ok

So, suppose I am having this x 1 and y 1 one vector and another vector is basically, that is shifted y the angle theta, which is basically this x 2 and y 2, the corresponding positions of these vectors is x 2 and y 2. So, then x 2 can be written as x 1 cos theta minus y 1 sin theta and y 2 can be written as x 1 sin theta plus y 1 cos theta. So, this is basically, this is just from the vector algebra, we can calculate the corresponding or we can get to know about the corresponding values of this y 2 and x 2, which are nothing, but dependent on that. So suppose, this is the initial vector, which is being rotated by the angle theta. So, this now after the rotation this is the new set of values, I will get.

So, what will be that value? So, that value will now; that means, represented or I can get the values as x 1 cos theta minus y 1 sin theta and y 2 can be written as x 1 sin theta plus y 1 cos theta ok.

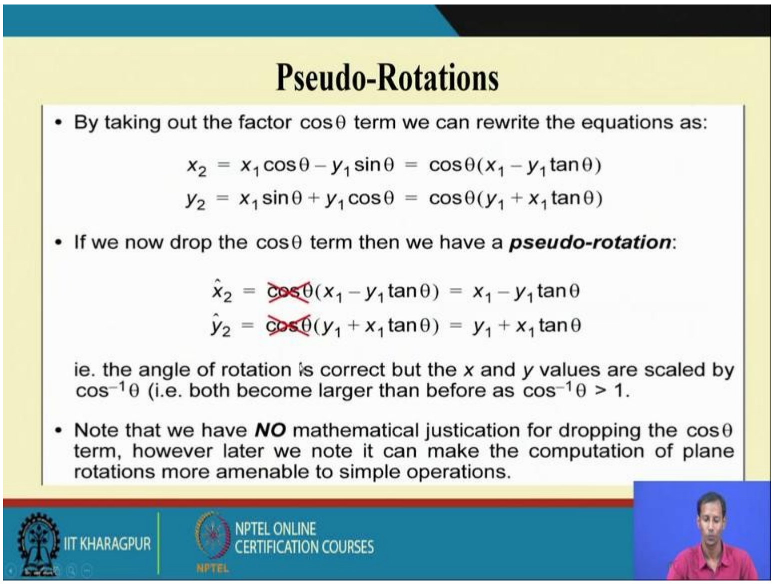
(Refer Slide Time: 18:09)



So, based on this fact, now in the matrix form, how can I write x 2 y 2? That can be written as this cos theta minus sin theta, cos theta minus sin theta here, sin theta cos theta ok. So, why? Because this will be, x 2 will be what? X 1 cos theta minus y 1 sin theta. So, x 1 cos theta minus y 1 sin theta so, and then y 2 will be x 1 sin theta plus y 1 cos theta, the x 1 sin theta plus y 1 cos theta. So, this is written ok.

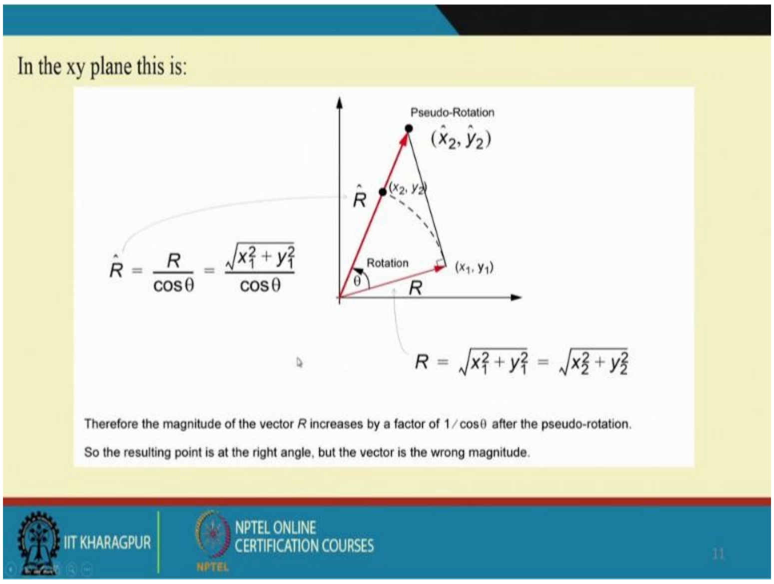
So, if I just rotate or this rotation is basically 90 degree so, at that time what will be its value? It will be 0 minus 1. So, if theta is 90 degree so that time cos theta value is 0 sin, sin theta value is 1 so, 0 minus 1 1 and 0. So, x 2 and y 2 will be minus of y 1 and x 1. So, here you see, if this is the initial vectors, which is x 1 and y 1. So, this is the after 90 degree shift, the position will come over here. So, in this case the corresponding this one is minus y 1 and this one is x 1 ok. So, that is why the corresponding values for this x 2 and y 2, that will be dependent on this x 1 and y 1; that means, in this particular direction, it will be shifted by y 1, this x 2 direction; that means, in the x direction, it will be shifted by y minus y 1 and in y 2, it will be shifted by x 1 ok.

(Refer Slide Time: 19:54)



So, then in Cordic what we do? We do actually this is the corresponding function after rotation here, what is that? Theta is my desired angle of rotation so; that means, I do not know, the exact value of theta. So, we have to calculate the sin theta and cos theta for any desired values of or any; that means, use a defined values of theta. So, how? That means, then x 2 and y 2 that can be, the equation for representing the x 2 and y 2 that is x 1 cos theta minus y 1 sin theta and x 1 sin theta plus y 1 cos theta. Now if you take common of this, cos theta it can be written as x 1 minus y 1 tan theta. So, here it will be y 1 plus x 1 tan theta ok.

(Refer Slide Time: 21:07)



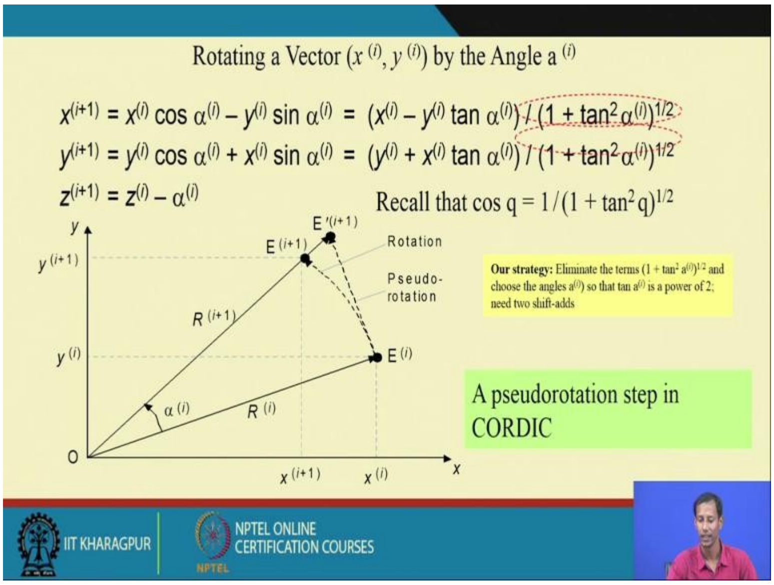
So now, we have to calculate this corresponding function of x 2 and y 2, how we can calculate? So, here actually we do pseudo rotation, what is pseudo rotation? So, pseudo rotation means here you see what is there; that means, this is basically, this particular vector is basically rotated by the angle of 90 degree and you see this is basically circular ok, but if I do not take this circular one, if I take the pseudo rotation, which is just straight of this particular vectors from this particular point; that means, which is basically covering over here, which is 90 degree; that means, just orthogonal to this particular. So, where it is basically meeting with this?

The original value was what? X 2 and y 2 x 2 and y 2 means. So, this is the values of x 2 and y 2 depending on this particular function ok, but here instead of doing that, what we are doing? We are doing another rotation, which is this pseudo rotation and the corresponding values for that is x 2 cap and y 2 cap. So, x 2 cap and y 2 cap, how it can be? That means, this pseudo rotation can be written that is if I just cross this cos theta. So, x 1 minus y tan theta plus y 1 plus x 1 tan theta, that will be the corresponding pseudo rotation values of these, x 2 cap and y 2 cap ok.

So, now here you see the angle of rotation is correct, but the x and y values are scaled by this particular factors means what? Now here actually, there is a multiplication of this function which is basically multiplied with this cos theta. So, we are removing this cos theta means, it is basically scaled by the 1 by of this cos theta. So, which is nothing, but this cos inverse theta. So, cos inverse theta, we can also write that is a cosec theta then, cosec theta can be written as this 1 by square root of 1 minus tan square theta so that, we will see now ok. So, then what will be the value of this? What will be the magnitude value of these particular R? So, initial the value of this particular is magnitude this if this is R. So, here what we are doing? We are basically scaling that by cos inverse theta.

So; that means, now R dash will be, because of this pseudo rotation this R cap, sorry R cap will be R by cos theta. So, R is nothing, but your square root of x 1 square plus y 1 square divide by cos theta ok. So, what is the; that means, corresponding values of this? That is x 1 square plus y 1 square and this R value R value is R cap value is now that, because of this psuedo rotation the corresponding value is now square root of x 1 square plus y 1 square divided by cos theta.

(Refer Slide Time: 25:01)

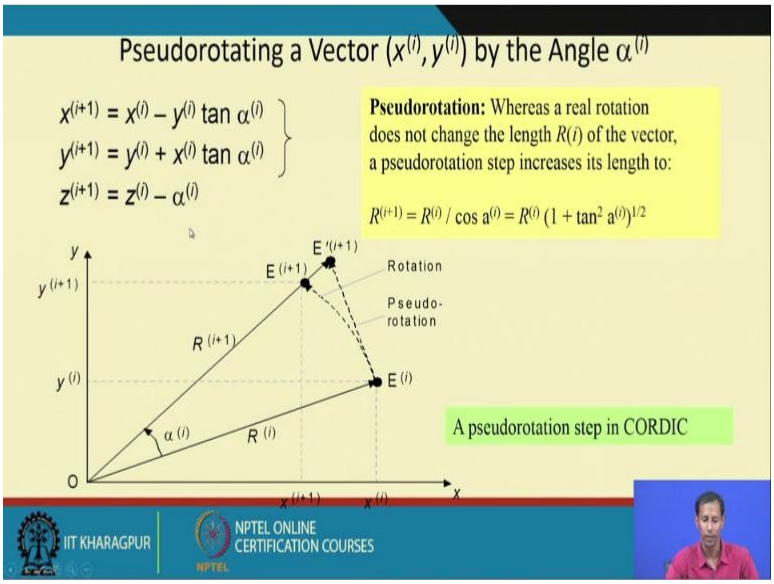


So, here actually this is because of this increase in the vector R which has been increased by the factor 1 by cos theta ok.

So, this is just another example of this pseudo rotation is step in Cordic, what is that? That is here x 2 and y 2 x 2 and y 2, that is being; that means, written as this that is x 1 cos theta plus y 1 sin theta and if I take common then, what I said that is that cos theta can be represented as what and cos theta means what? Cosec theta sorry, cos inverse theta means cosec theta.

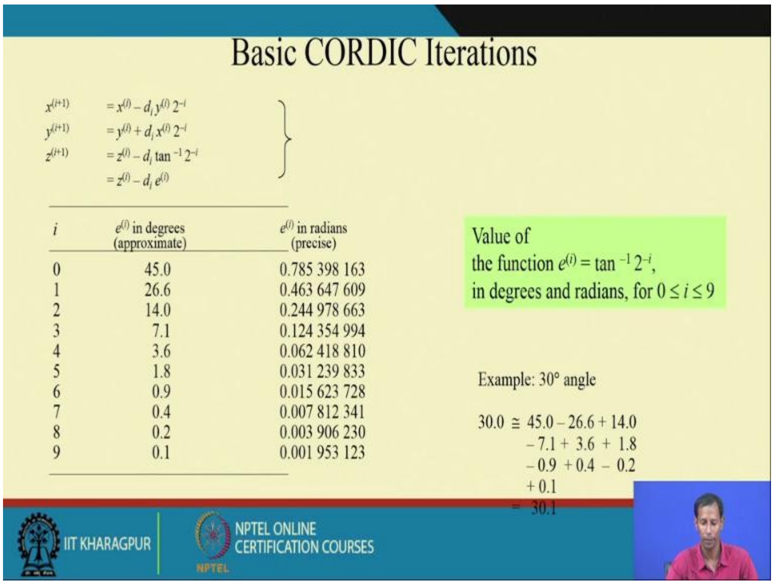
So, cosec theta can be again written again square root of 1 plus tan square theta ok. So, 1 plus tan square theta square root of that is being written over here, no sorry this is the; that means, the 1 by cos theta, so 1 by cos theta, cos theta is now basically, written as 1 plus tan square theta square root of ok. So, the in the same manner, we can write this y 1 2, in the case of y 1, this is basically here minus this is plus. So, this is the initial vector and this is the pseudorotatic vector ok. So, this is the corresponding that trignomatric function if; that means, cos theta that is equals to 1 by inverse of 1 plus sorry, square root of 1 plus tan square theta ok. So, actually what is why I need this pseudo rotation or what is the meaning of this or why is this intentionally, we are doing or introducing this pseudo rotation ok?

(Refer Slide Time: 27:07)



Yes we are basically intentionally introducing this pseudo rotation, why? What for we are using? Because and about this z, I am coming or I am discussing later. So, we have seen till now x and y, but where from this z that is that that I am coming later ok.

(Refer Slide Time: 27:26)



So, here you see, because of this tan of this tan theta, what we can write? Ok. So, what we can write? That can be actually first let me tell you, this z is the corresponding actually, this equation I have to write after the rotating the vectors by an angle of alpha, here it is mentioned as alpha ok. So, we have to whenever, we are just rotating and what I said in the beginning Cordic is basically based on some pre defined angles, it basically decompose the desired rotation of angle. So, how it will decompose? So; that means, in each of the step, how much is the angle? I am basically calculated; that means, traversing, that I have to somewhere store that particular information.

So, that particular information is stored in this equation, which is z equals to initial vectors, which is z i minus alpha i ok. So, this is the; that means, this particular z term is basically used to calculate or it is being used for mentioning the corresponding, how much rotation I am doing of this desired rotation angle? Ok. So, based on these based on these, if I use some of the pre defined set of angles, which is basically, where this pre defined angles is basically dependent on some in the power of 2 in the power of 2 means, this tan alpha i tan alpha I, this is basically now replaced in terms of this tan inverse of 2 to the power i so; that means, the angle is basically rotated on the based on this particular equation, which is e i equals to tan inverse of 2 to the power minus i ok.

So, for what reason I am using this? And what is the meaning of doing this? So, that we will see or that we will cover in the next lecture ok. So, whatever we are doing for this

Cordic implementation or whatever we are changing in this equations, they are basically intentional and there is a particular reason for that and because of this pseudo rotation and considering this fact this Cordic algorithm that becomes, much more; that means, it becomes less complex than, the other things or this or this the hardware architecture that becomes, very much simpler or less complex so.

Thank you for today's class.