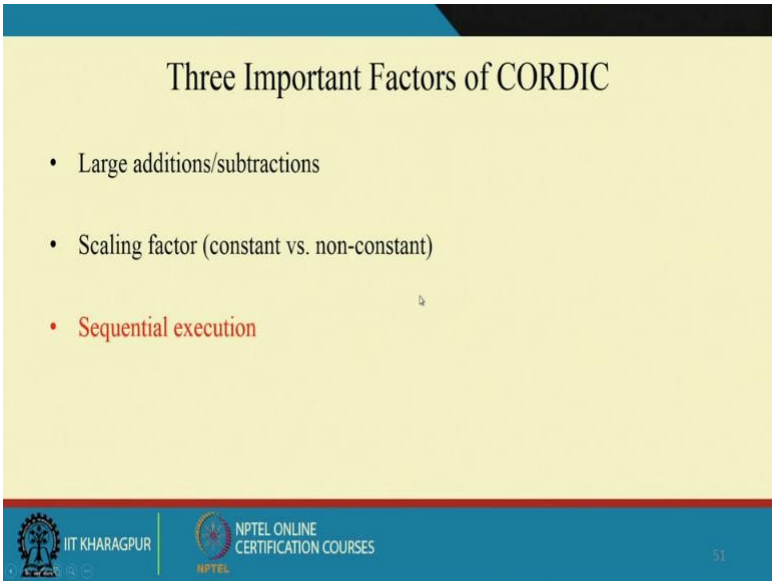
Architectural Design of Digital Integrated Circuits
Prof. Indranil Hatai
School of VLSI Technology
Indian Institute of Technology Science and Technology, Shibpur, Howrah

Lecture - 44
Cordic Architecture (Contd.)

So, hello everyone welcome back to the course on Architecture Design of IC's. So, we are we have seen different CORDIC Architecture. So, we can implement various kind of functions or operations we can do. We can learn that CORDIC in six different mode we can done. Using six different mode various kind of; that means, this arithmetic operations we can do or can perform. It is not only limited to that based on that particular function calculated again we calculate some more of the function that we have already seen. We have seen the; that means, this parallel architecture or this serial architecture that we have seen. And what I say that there are; that means this is also 1 very rich field of research.

So, people have find out as this a CORDIC has a fast application area. So, that is why people have tried so much to get more high performance architecture for this CORDIC algorithm. So, like here what we are doing what makes it; that means, what is the critical point, if we see at the way of very fast and then from that point if we try to analyze that how we can improve the corresponding performance of the CORDIC algorithm. So, I think that will be step by step research methods. So, though very first what we have to do the important factors of CORDIC we have to concentrate or we have to notice.

(Refer Slide Time: 01:55)



So, the important factors which are basically related to CORDIC, that is I need additions subtractions operation which is if we choose the number of bits are more so at the time the corresponding delay will be more. Then there is another thing which is that scaling factor. So, scaling factor is basically that is there is 0.6073 that is in the fractional domain. And then again the thing is that the sequential execution; that means, the CORDIC algorithm is basically has been done based on what that? X i plus 1 equals to x i minus d i into 2 to the minus i of y i. So; that means, the i plus 1 rotation that is depended on x i plus 2 that is depended on i plus 1 so; that means, each of the cycle is basically dependent on the previous one.

So, can I use so; that means so as this algorithm itself it is sequential. So, at the time how can I make it faster so; that means, if it is sequential; that means it will take time. Suppose like if I take 1 example of that suppose if I have to walk a road if 1 step by 1 step if I am going then how much time I will take to cover up that road, if I take 2 step at a time, or 3 step at a time, or 4 step at a time so at that time the time to cover that particular path that will be a lesser. So, here also the same thing; that means, to compute the whole task or to perform the whole task if 1 step by 1 step, if I do so at the time it requires, if I run it for 10th iteration so 10 clock cycle I required.

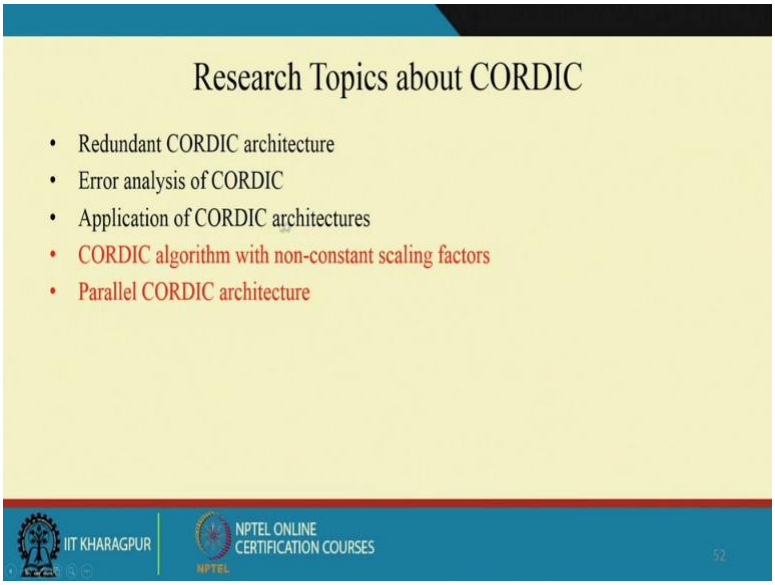
But if I that means, do 2 iteration at a time, or 3 iteration at a time, or 4 iteration at a time if we can perform so at the time the time requirement that will be lesser. So; that means,

what we can do in the parallel what we are doing the; that means, the input and output they are basically dependent on them. That means, whenever the inputs are available at the time whatever is the delay and how many levels of that particular adder shaft or (Refer Time: 04:36) is there dependent after that much of a time I can get my output. So, instead of doing that what we are doing the sequential one is basically taking less time. So, if we can make radix two or three; that means some kind of mixed architecture. That means few of the steps if we can do sequentially and few of the steps if we can do parallelly. So, at that time I can get the area benefit as well as the time benefit both the way, so that kind of architecture also we can think of.

So, that is why people are developed there are if you find in the later lecture. Definitely you will find the later lecture; that means, this radix 4 CORDIC algorithm, then radix 8 algorithm. That means increasing the radix or; that means, increasing in the steps per iteration I can reduce the time though the complexity wise it will be more. So, again what I said the last point here is that sequential execution of the algorithm itself. So, can I make that make it parallel kind of change? So, one of the; that means, literature is there or one of it is not that the only this particular architecture is there.

So, this is just one example of that, if I have to think that if I want to make this if I want to get the time; that means, to the competition time will be lesser. So, at the time I have to make that algorithm parallel by itself. So, how can I make that? If I can make that then what will be the change in the architecture. So, that we will see, and if I make the change so at the time what is the area as well as what is the; that means, delay wise how many or what is the advantage I am getting that we will see. This is just only for an example this is not that only this architecture is available on a market, there are several architecture available on the market for CORDIC algorithm this is just to show you the example how you can proceed in this particular direction.

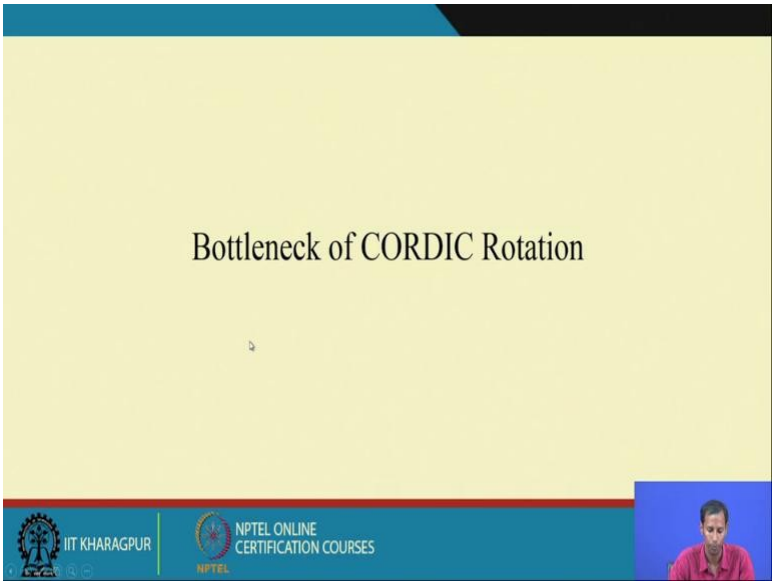
(Refer Slide Time: 07:00)



So, this research topics about this is little bit; that means, I am giving you just one overview of doing research in this particular area of the CORDIC architecture. So, there are like this redundant CORDIC architecture, then this error analysis of Cordic, then application of CORDIC architecture, then this CORDIC algorithm with non constant scaling factors, and then parallel CORDIC architecture.

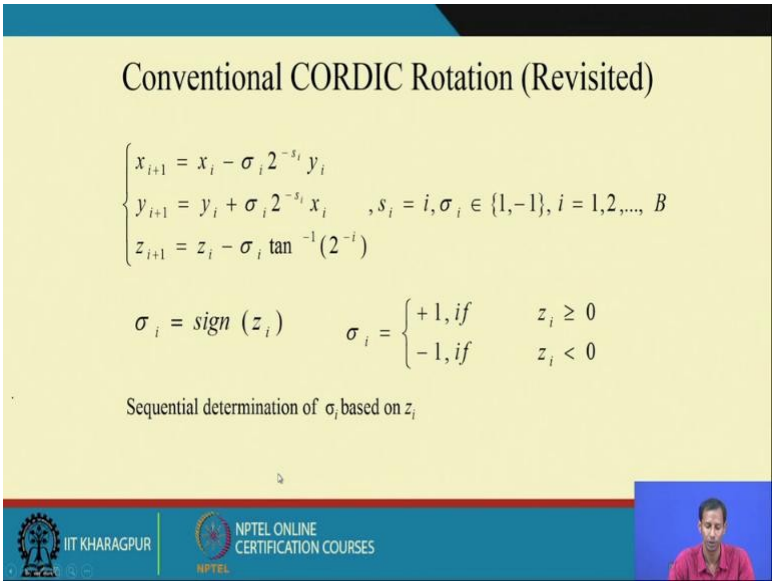
So; that means, what in CORDIC what we do we basically get the some of the errors whenever we are doing iterative if we gather n 1 stage if we gather some of the errors. So, in how many iteration I will continue so that much error I will accumulate. So; that means, error free CORDIC is also architecture is also available. And all these because of fast application of the CORDIC architecture. So, now, what we are trying to focus on that is the parallel CORDIC architecture.

(Refer Slide Time: 08:05)



So, what is the bottleneck of CORDIC Rotation?

(Refer Slide Time: 08:07)

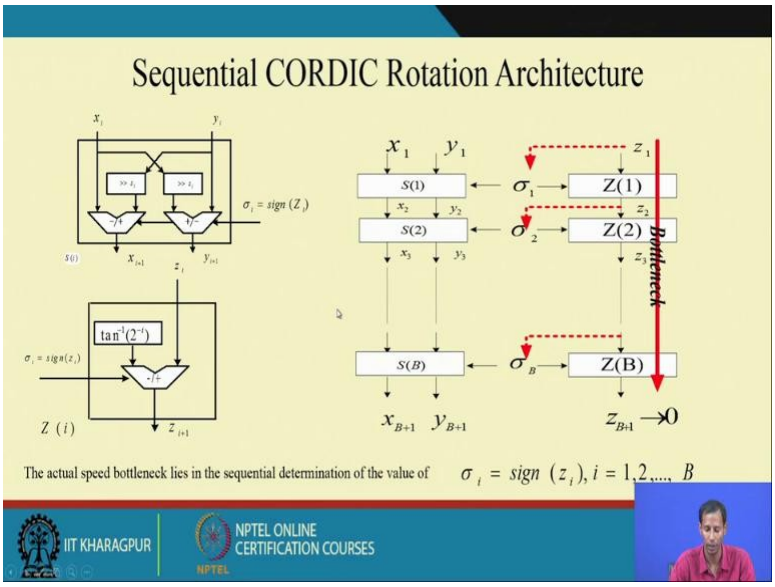


This is the basic equation of conventional CORDIC that is x i plus 1 that is depended on x i. This is sigma here means the sign of z and then 2 to the power i way of y i. So, the conventional; that means the corresponding bottleneck is that the sequential determination of sigma i based on this z i.

Because dependent on this sign we can compute each of this iteration whether that will be plus or that will be minus. In this case also whether that will be plus or in this minus it

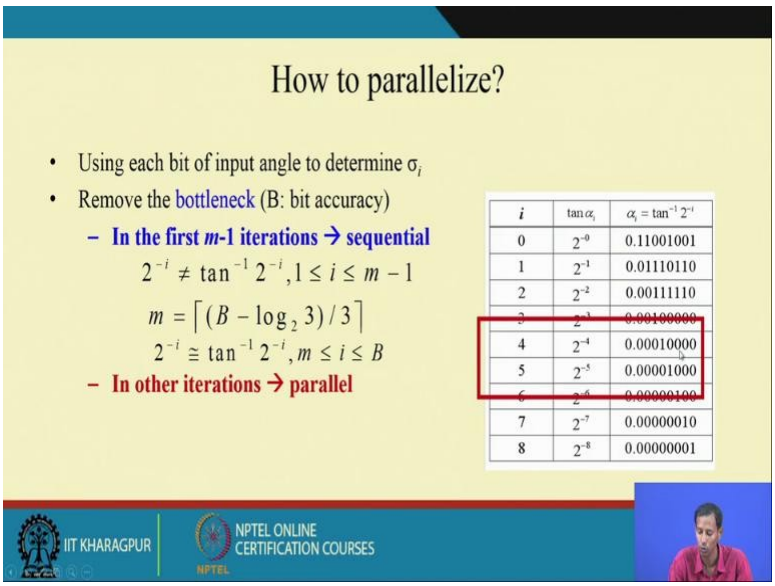
depends on the sign of this. So, this particular things sign of calculation of this sign of this sigma i based on this z i that makes the cordical algorithms sequential.

(Refer Slide Time: 09:05)



So; that means, as this path is running 1 by 1 so; that means, this is basically taking the time for that that is why we are not making this CORDIC as a parallel one. So, at that is why what we can do this is the new method.

(Refer Slide Time: 09:27)



i	$\tan\alpha_i$	$\alpha_i = \tan^{-1}2^{-i}$
0	2^{-0}	0.11001001
1	2^{-1}	0.01110110
2	2^{-2}	0.00111110
3	2^{-3}	0.00100000
4	2^{-4}	0.00010000
5	2^{-5}	0.00001000
6	2^{-6}	0.00000100
7	2^{-7}	0.00000010
8	2^{-8}	0.00000001

So, what we are trying to do, we are basically taking each bit of input angle to determine this sigma i, means what? Whatever input angle we are giving we are considering each of

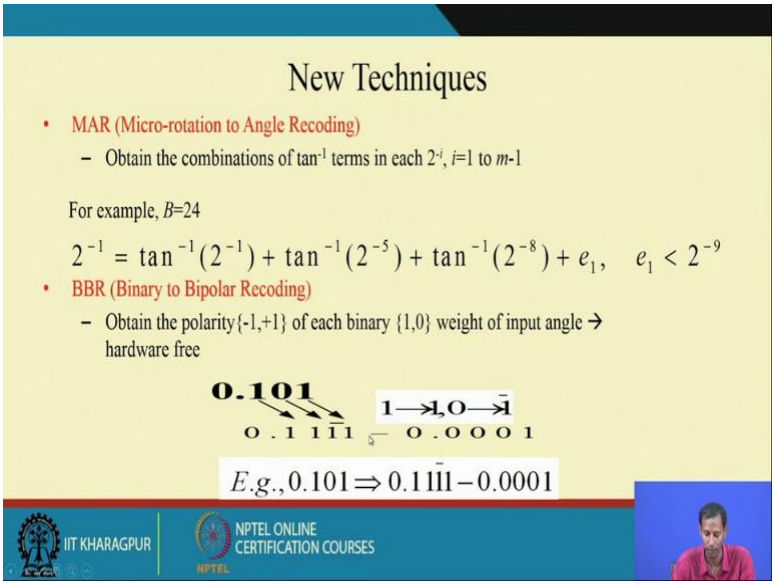
this bit position, whatever is the bit values of this input angle based on that we are basically trying to determine this sigma i. Because the elementary angle that are pre computed right. So, pre computed means if that is for suppose 30 degree right. So, 30 degree means that is lesser than if I consider bit wise that is lesser than the 45 degree.

So; obviously, if the; that means, if it is below 45 degree so; obviously, the sign will be negative one. So, the input angle is given to me and the elementary angle is also known to me. So, on that considering those facts can from the beginning can I calculate each of this sigma can I do that? So, I can do or based on that considering that particular fact this parallel CORDIC architecture has been evolved. So, what it does it basically in the first m minus 1 iterations it basically follows this sequential. Sequential whatever is the conventional method it follows for the first m minus 1 iterations, but for the other iterations if B is the number of bits for the other iteration it basically performs parallel operations. So, how means what.

So, this is the elementary angles so whenever we are considering this that for 2 to the power minus 3 to the power minus 4 to the power minus 5, 2 to the power minus 6 there mostly; that means, only 1 or 2 position of this is 1 and rest of the positions are like mostly filled with 0. We have to consider this in the binary to represent this corresponding that the values of for this 45 degree tan inverse of this 45 degree; that means, which is in the radiant. So, those values can be represented in terms of this. So, in the actually if you just follow that particular you have to; that means, connect with the table. So, in the table you will notice that for the z calculation it is basically that theta, but it is recorded in the radiant mode.

So, those values now you can basically recorded as this, but whenever it will come to; that means, this minus 2 to the power 3, minus 2 the power 4; that means, for the lower values of this the number of ones in this particular bit that will be lower. Here it will be mostly dominated by the number of 0 so that means, for the what it sake for the fast where there is 0 combination of 0 and 1 are; that means, still almost like similar or there still it is; that means, it is not that only 1 factor is dominating. So, at that time we can follows this sequential approach. But where from this mostly 0 will dominate over 1 so at the time we will use this parallel architecture.

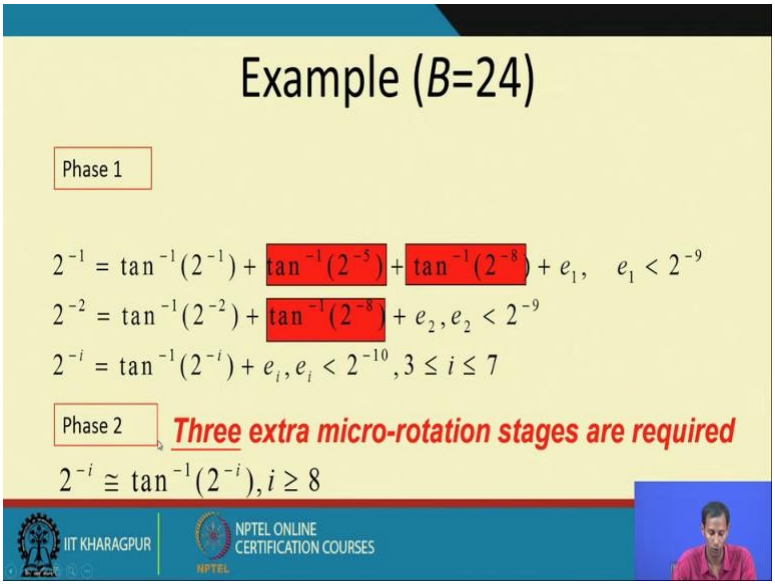
(Refer Slide Time: 13:35)



So, based on that the two; that means, techniques which is this micro rotation to angle recording and then there is binary to bipolar recording. So, how you can do? Suppose this 2 to the power minus 1 so, at that time it will be written as tan inverse of 2 to the power of minus 1, tan inverse of 2 to the power minus 5, tan inverse of 2 to the power minus 8 plus error of this which is nothing, but this 2 to the power of minus 9.

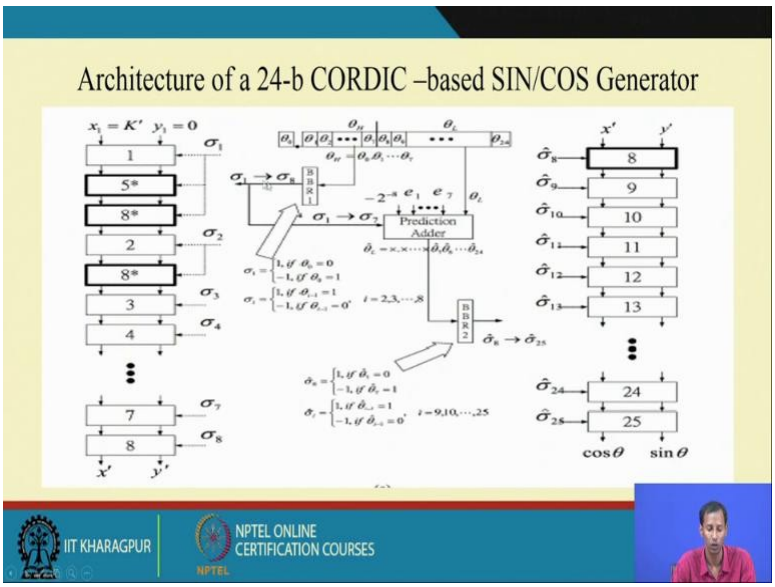
And that in binary to bipolar recording, how it will be recorded? This polarity will be different; that means, for each of this 1 will be recorded as 1, but 0 will be recorded as 1 bar. So, this you can see for 0.101 this will be recorded as this 1 1 1 bar 1. So, 1 for this 0 for this and 1 for this something like this we have to do.

(Refer Slide Time: 14:41)



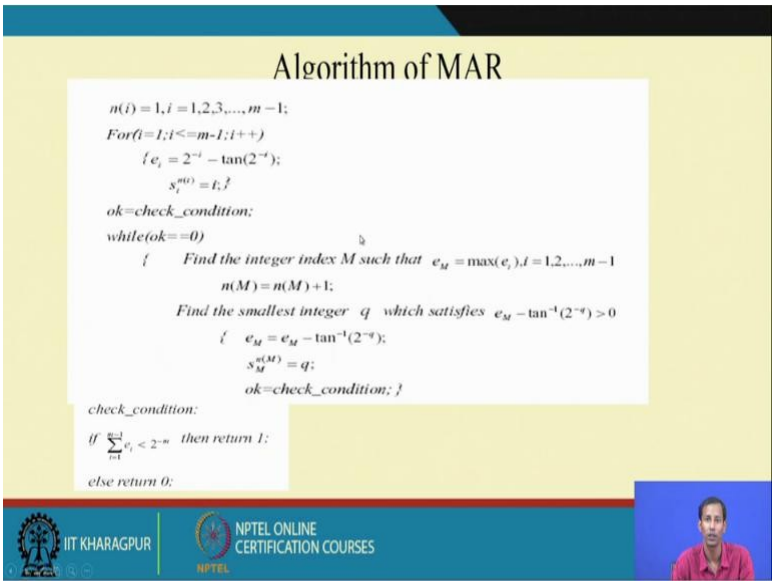
So; that means what? This there will be having 2 phase, what I said? According to the algorithm there will be 2 phase, in 1 phase it has to pass for m minus 1 iterations it has to pass sequential. For other iterations which is if I consider B number of iterations so, B minus m minus 1 sorry, B minus m plus 1 number of iterations that will be considering parallel. So, in phase 2 we have to do that.

(Refer Slide Time: 15:25)



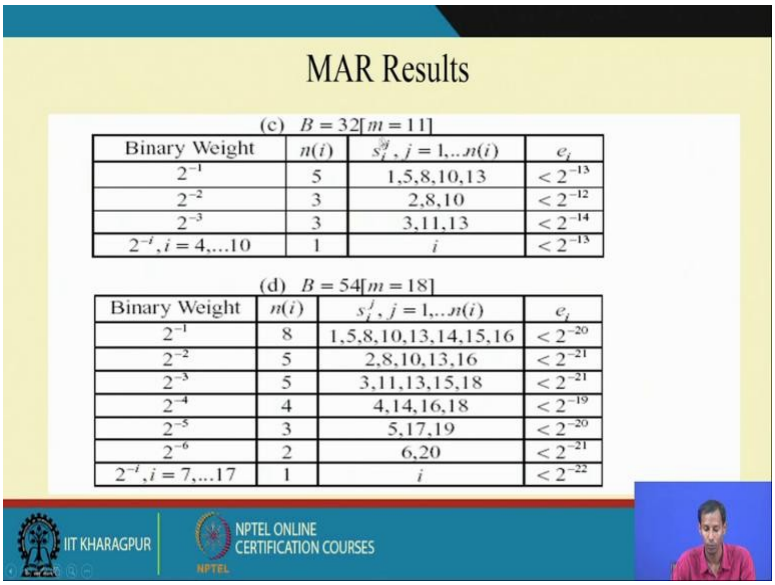
So, considering this here you actually some extra elements or extra; that means, things you have to be or extra circuitry you have to modify or introduce in this particular circuit.

(Refer Slide Time: 15:45)



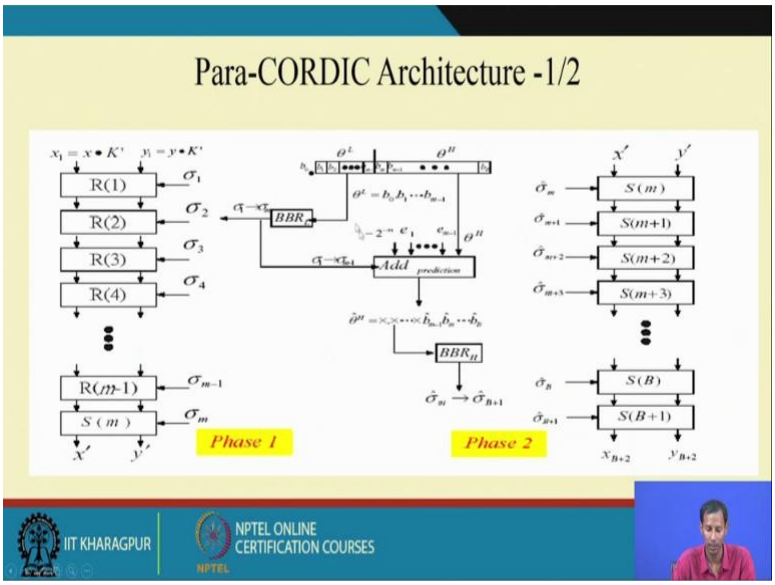
So, this is the algorithm for this micro angle rotation technique. And actually this is the results what we are getting; that means, for different different of number of bit, where n means the first 11 iteration are sequential, rest 21 iteration they are done in parallel.

(Refer Slide Time: 15:59)



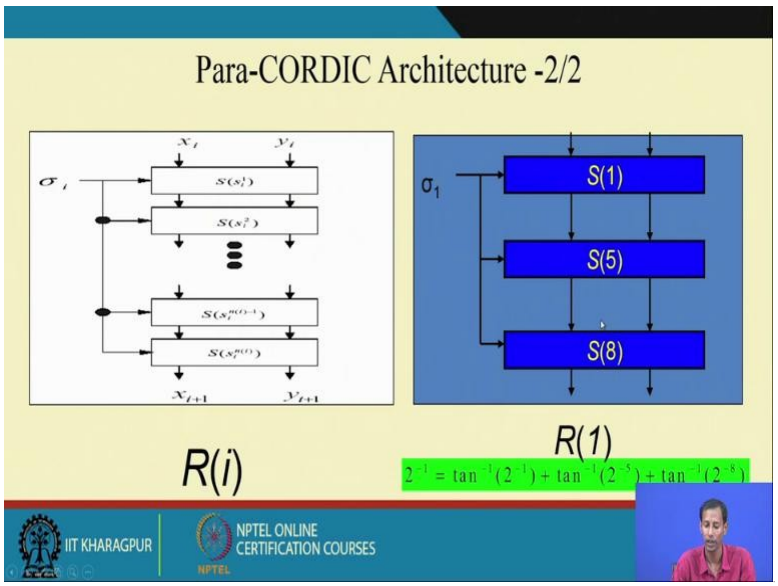
So, if I run that in parallel so at the time the corresponding values of this what is the values of; that means, this sign that is calculated based on the algorithm of this flow chat which we have already mentioned over here.

(Refer Slide Time: 16:35)



So, this is the architecture for that; that means, based on that particular algorithm. Now we are getting this architecture this is the architecture 1.

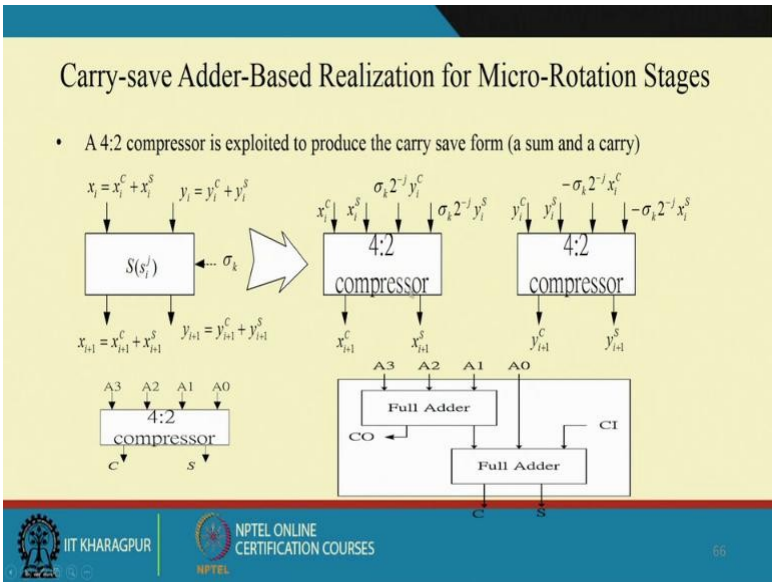
(Refer Slide Time: 16:45)



And here you see how we are calculating that sign that has been calculated considering these three things that is S1, S5 and S2, S8. Based on what? That if 2 to the power 1, then tan inverse of 2 to the minus 1, tan inverse of 2 to the power minus 5, tan inverse of 2 to the power 8 these three has been considering together and to find out the corresponding sign for that.

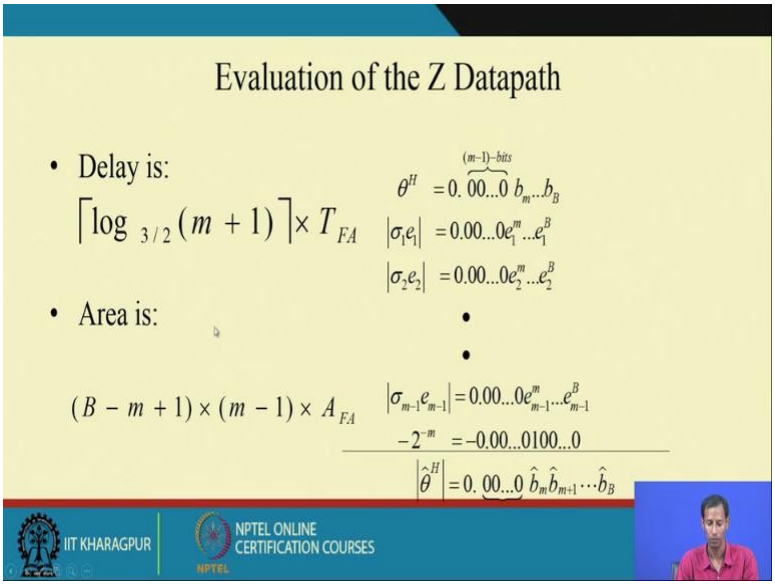
And each of these case now considering this in each of this sign has been computed from the beginning; that means, whenever for the lower part the sign has been calculated from the beginning. When where I am basically doing the parallel architecture or I am running that in the parallel mode.

(Refer Slide Time: 17:39)



So, this micro rotation of this is being realized using this carry save adder so carry save adder we have already seen in the adder chapter. So, carry save adder basically that is very much useful for; that means, this delay wise compression. So, it takes lesser time than the normal one normal (Refer Time: 18:04) carrying 1. So, that we use to reduce the to calculate this micro rotational angle; that means, to find out the sign based on these parallel; that means, algorithms.

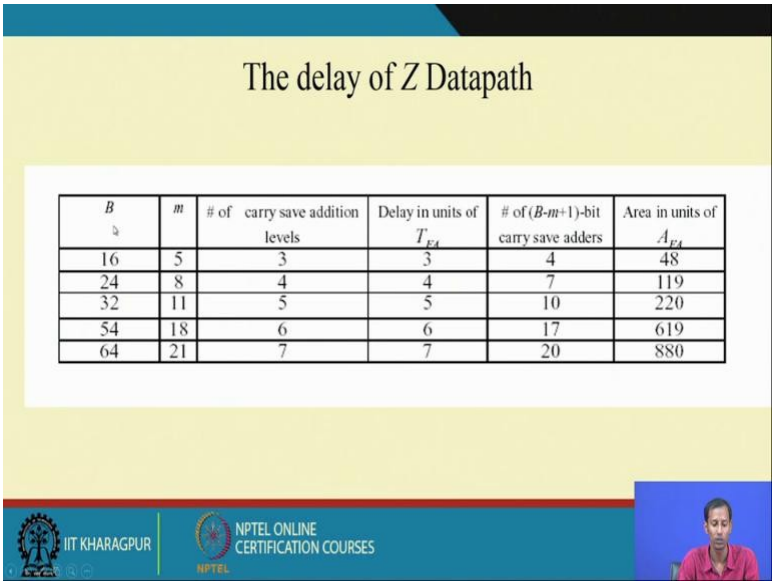
(Refer Slide Time: 18:21)



So, whenever we evaluate this particular structure so this Z Data path means whenever we evaluate this. So, at that time this corresponding delay is this log 3 by 2 m plus 1 into T F A; that means, T F A is the delay for the 1 full adder save. So, m is what? M is the number of iterations initial iteration where I am running there CORDIC in the parallel mode. But area wise it is B minus m plus 1 into m minus 1 into A of F A; that means, A of F A is the area of 1 full adder set.

So,; that means, delay wise it is taking this, and area wise it is taking this much of that area so this much of area means. So, for m minus 1 stage is there and then again B minus m plus 1 stage is there. So, both are basically considering together and then that is multiplied with 1 unit of full adder area to get the total area. But delay wise it is more less because I am considering; that means, these iterations; that means, the CORDIC only for the m number of iterations in a sequential mode. So, that is why this delay is reduced over here.

(Refer Slide Time: 20:09)

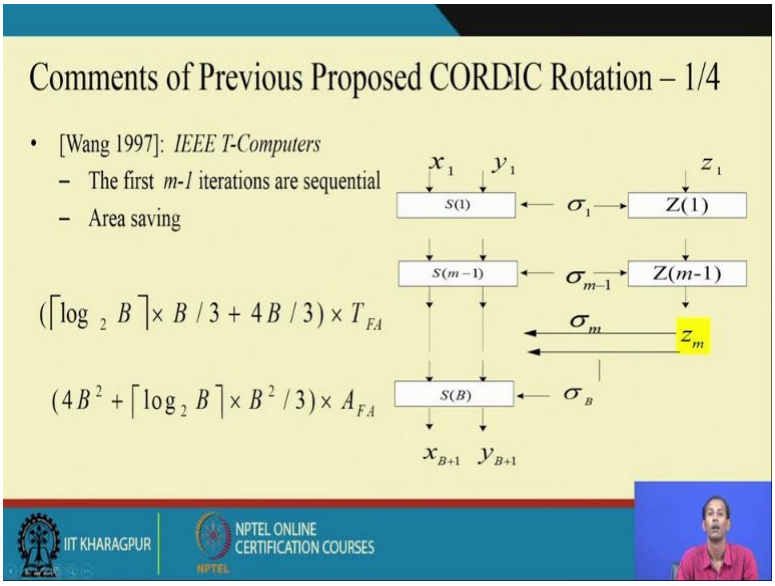


B	m	# of carry save addition levels	Delay in units of T_{FA}	# of (B-m+1)-bit carry save adders	Area in units of A_{FA}
16	5	3	3	4	48
24	8	4	4	7	119
32	11	5	5	10	220
54	18	6	6	17	619
64	21	7	7	20	880

So, considering B different B and m so what I said B is the number of stage. I will consider among which m is the number where how many numbers the CORDIC will run for sequential, rest of the case it will run in parallel mode. So; that means, 5 is the case where it is running for sequential, 7 is the; that means rest, 11 modes they are running in parallel.

So, the carry save addition levels it is required that is 3 and then the delay in unit that is also 3. And then this carry save adder number of carry save adder required that is 4 so total area will be 48. So, why 48? Considering these facts if you just calculate and then we will get this. So, considering different values of B and m, now we will get different in delays and different in terms of area.

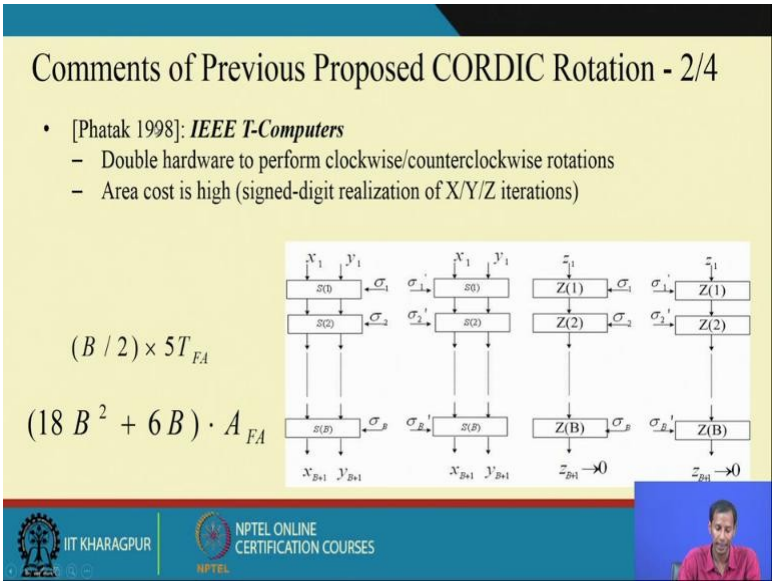
(Refer Slide Time: 21:23)



And if we consider that previously introduced CORDIC rotation; that means, first CORDIC architectures. So, this is one of this; that means architecture which has been published in 1997 in IEEET, transaction on computers. Where they have followed this first m minus iterations are in sequential.

So, considering this fact this is getting the delay amount is of this log 2 base B into B by 3 plus 4 B by 3 into T F A. Whereas this area is 4 B square plus log 2 plus B into B square divided by 3 into A F A. So, this is one of the; that means, architecture which has been proposed in 1997.

(Refer Slide Time: 22:19)



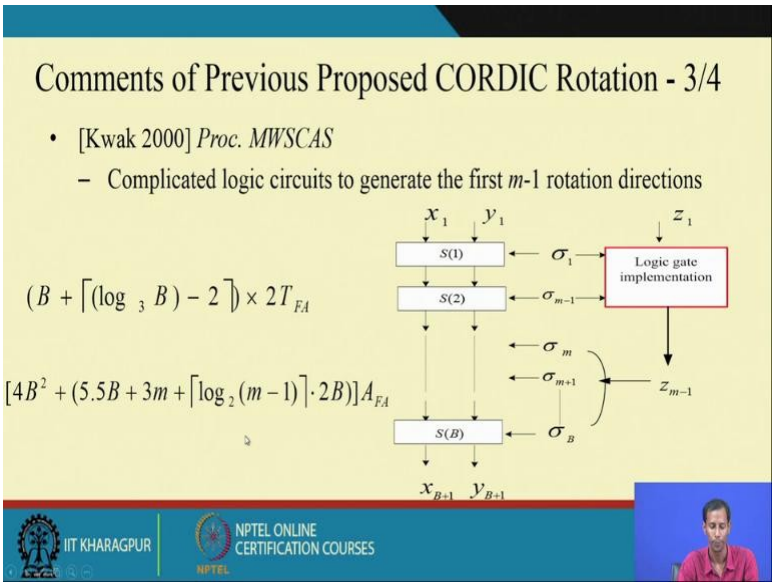
So, in 1998 another architecture has been proposed which, basically based on this double hardware to perform clockwise and counterclockwise zero rotations. That means, here what we are doing; that means, in the positive domain; that means, suppose we are doing positive. So, that means, we are running in that as a counterclockwise whenever we are coming down in the negative so; that means, we are running in that; that means, clockwise.

So, that clockwise rotation and counterclockwise rotation from the beginning using double hardware from the beginning if we can try to pre compute and then if you store this information and then again if I have to use that to reduce the time so we can. That means, here this particular architecture is mainly to reduce the time requirement or the computation time of the algorithm; that means CORDIC algorithm to reduce that focusing on that particular aspect this particular architecture has been developed. So, here what is that; that means, as we are introducing or we are using that double hardware to perform this clockwise as well as this counterclockwise rotation so; that means, area was area cost will be higher.

Because we are doing parallel things in counter clockwise rotation checking as well as counterclockwise rotations checking. That means, we are computing the serially computing the sign from the beginning by considering both clockwise rotations as well as counterclockwise rotations. So, here you see the delay wise it is much more (Refer

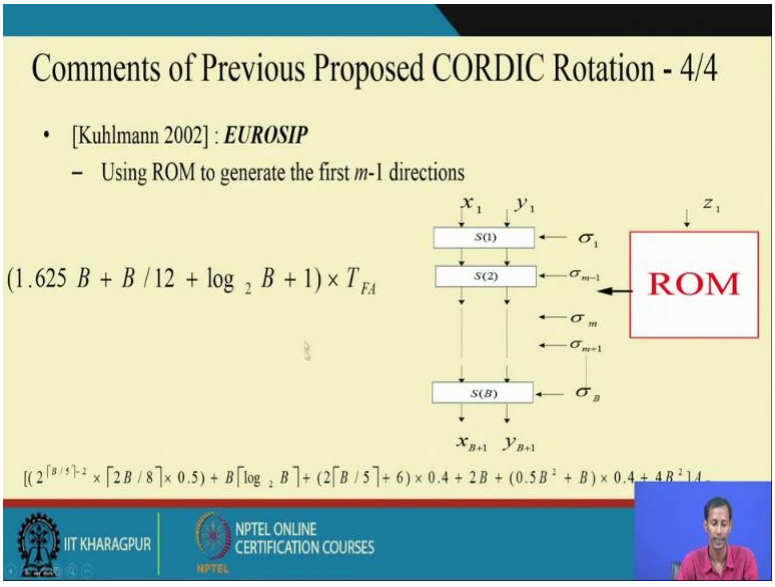
Time: 24:04) that is B by 2 into 5 into T F A, where as that area wise it is 18 B square plus 6 B into F A; so area wise it is much more costly.

(Refer Slide Time: 24:13)



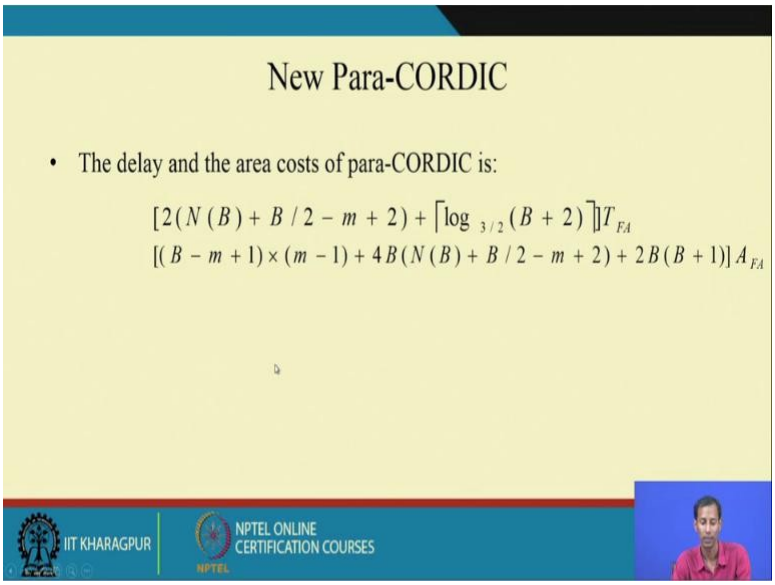
So, then again in 2000 again one work has been produced where this complicated logic circuits to generate the first m minus 1 rotation direction. So; that means, here all the things from the beginning considering the input rotation; that means, input angle given based on that we are trying to pre compute the sign values and then we are trying to use the in the iteration mode. So, here the delay is B plus this and the area is this.

(Refer Slide Time: 24:49)



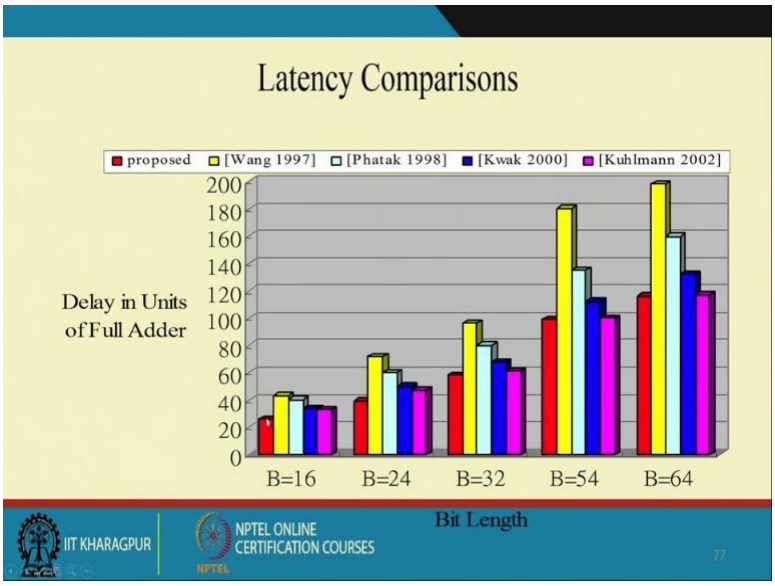
Then again in 2002 this ROM based; that means, the corresponding iterations; that means, the sign that also been stored in the ROM. And then again it is used for this each of the iterations and then this is the corresponding that delay for this is the area for this.

(Refer Slide Time: 25:11)



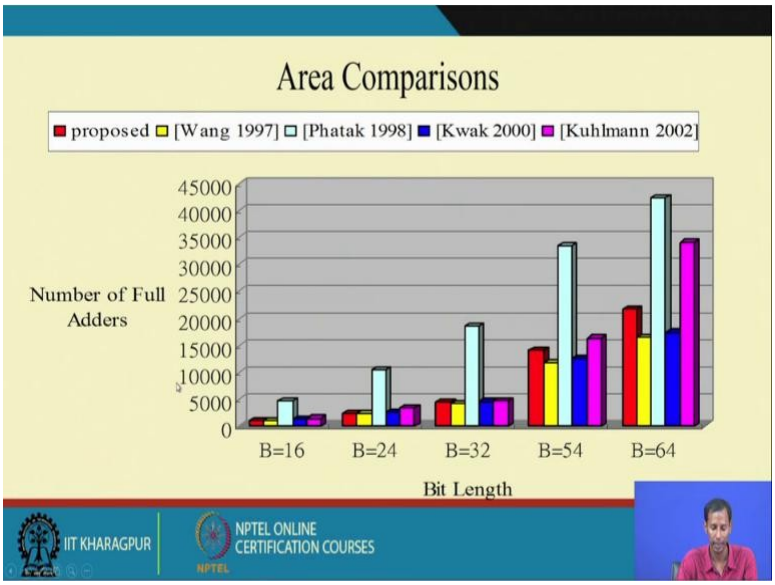
And this parallel CORDIC architecture whatever we have seen just now based on this m A R algorithm. Which is this which is micro architecture; that means, angle rotation and technique. If we see the delay and the corresponding area cost is this much.

(Refer Slide Time: 25:33)



So, if we just compare this 4 4 for a war. So, if you see different considering the different size of B. If you are getting the proposed one means this is a parallel CORDIC one. So, parallel CORDIC is basically having lesser delay in all these considerations.

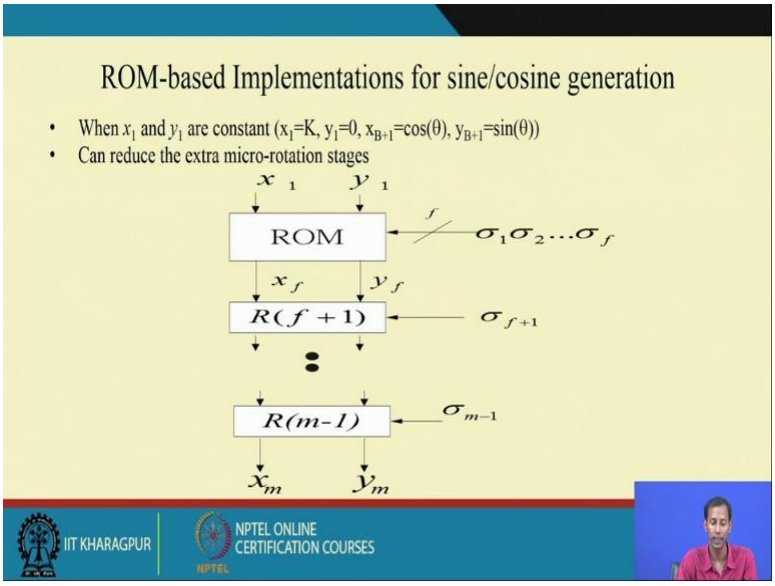
(Refer Slide Time: 25:59)



And area wise consideration also you can see in some case it is getting higher. The yellow one is the lower one, yellow one is the one yellow is the this one so area wise it is having the savings. So, area wise it is having the savings means that is why it is lesser than the parallel CORDIC one.

And this double hardware based area wise it is having the maximum one. So, this 1998 work, this work as double hardware for clockwise and counterclockwise rotations. So, this is having more area than among all these architecture. But if you just see, if you just see that that, but delay wise it is not that much where as the this work area wise it is saved, but time wise it is more on each of this case.

(Refer Slide Time: 27:09)



So, then this what is the application area of this, application area of this is for ROM based implementations of sine and cosine generations. So, sine and cosine generations why we required suppose you want to implement one this frequency synthesizer, this digital frequency synthesizer or suppose you are you have to develop 1 this transmitter or receiver. So, for the highest frequency mixing you need that digital frequency synthesizer which will be of different frequency; that means, sine theta and cost theta. So, that sine value and cos value we have to generate so, at that time you can use CORDIC to generate that sine and cos.

So that means and whenever we are putting that thing or you are you are using that for those type of applications. So, at the time competition time will be; obviously, it requires lesser time because in wireless communication or the transmitter or receiver specification. This your hand of time or the time competition for doing the whole transducer operation that is a very much important factor which is basically related to hand of signal in your cell phone or this wireless signal processing blocks.

So, on that case computation time it is required there is very much lesser that type of application required this kind of architecture to be developed. So, this is one of the particular aspects or one of the particular example which I am showing you to show the research direction for this particular area. This is not the fact that, again I am saying that this is not the fact that this is the only architecture which is available. And this is there

are now into 2000 I think we are in 2018. So, there are several architecture which people have introduced or people have already proposed or invented. So, if you are interested in this direction research or your work on this particular direction so please go through those or please let me know.

So, thank you for this; this is the end of this CORDIC chapter. And the next we will try to see this first transform architecture of the real (Refer Time: 29:48) architecture of the first Fourier transform. And then that is means when again we will try to see that different kind of this timings strategies what are the hazards and then; that means, what are the glitches which occurs in the circuits individual circuits. And then again if time permits then; obviously, we will try to show you this what is the meaning of static timing analysis what is the need of; that means, just the basic of static timing analysis.

But mostly if I do not cover in the lecture those things if you ask then; obviously, surely I will shows; that means, provide you the documentation which through the via the discussion forum you have you just request me; obviously, I will try to share those so. Thank you for today's class again, we will meet on the next class.