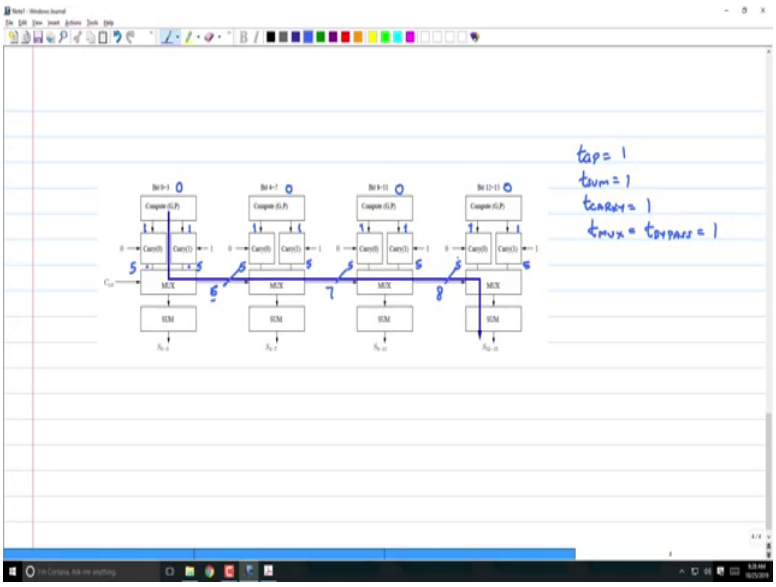
Digital IC Design
Prof. Janakiraman Viraraghavan
Department of Electrical Engineering
Indian Institute of Technology, Madras

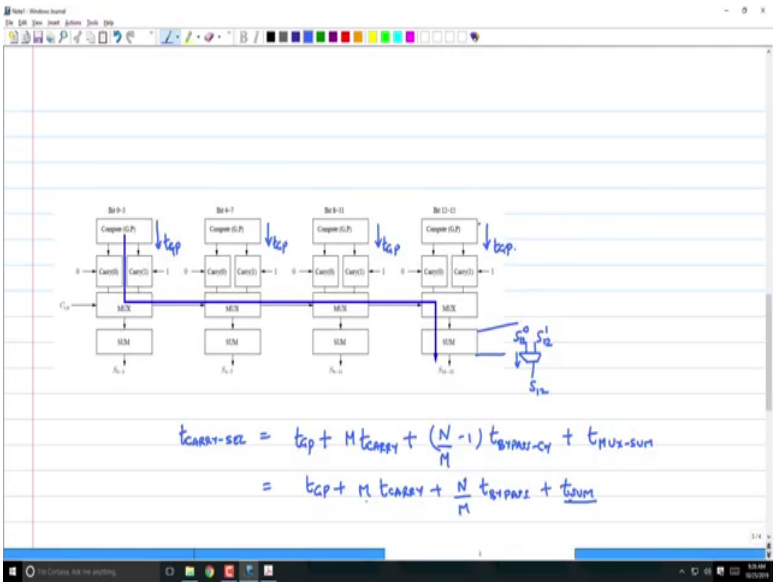
Lecture – 59
Linear and Square Root Carry Select Adder

(Refer Slide Time: 00:15)



Let us now evaluate the arrival times of various signals in this ok. I am going to just evaluate the arrival times of various signals now, fine. So, let us assume the following t GP equals 1, t sum equals 1, t carry equals 1, t MUX equals t bypass equals 1; everything is unit delay ok. So, how long does it take for the signal to arrive at this point out here? Remember A and B, A 0 to 15, B 0 to 15 arrive at time 0 right. So, this arrival time is 0. How long does it take to a generate GP? 1 right, 1 ok. Now, what about for the arrival time at this point? Yeah? No, think carefully.

(Refer Slide Time: 01:58)



Look at this expression here; you have to ripple through that right. The first stage you have to ripple you have no choice. What is that therefore? Yeah. So, it will take 4 units of time for the signal to propagate through that, but the signals arrived at 1. Therefore, this has to be 4 sorry, 5 you are right, 5. What about in the other cases? Bits 4 to 15, arrival time at the output of carry 0, carry 1 block. Yeah, same thing, right. It is all parallel, it has no connection to carry in at 5. What about output of the multiplexer now?

Student: 6.

1 unit. So, arrival time here is?

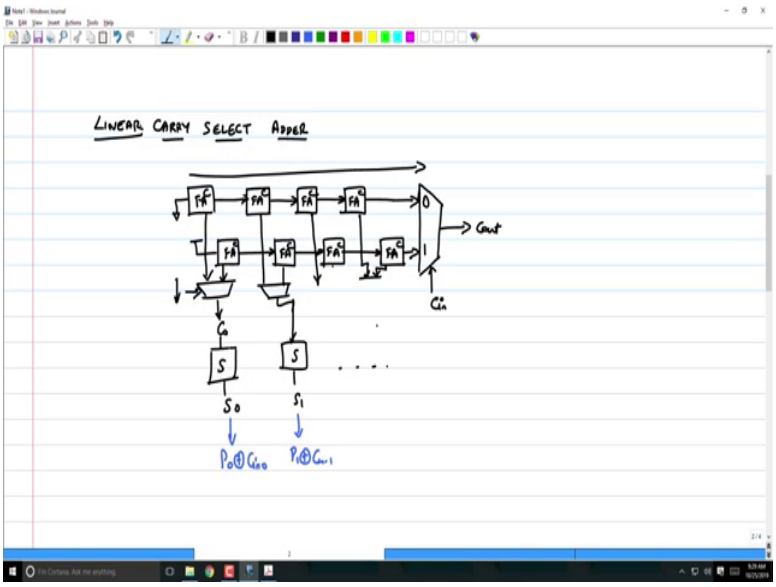
Student: 6.

6. Arrival time at the next stage is? Here? 8 right, you are going to look at MUX of 6 and 5 plus 1, 7; MUX of 7 and 5 plus 1, 8; MUX of 8 and 5 plus 1 whatever you can keep going like that. What is the obvious problem in this case? Is it the delay of MUX? Is it the delay of MUX?

Student: (Refer Time: 04:08).

So, what happens if I take this to 128 bits and the last stage whether it says 4 bits right, it is a 4 bit segment that I have made? The last stage would have generated this carryout carry 0, carry 1 block would have been ready in 5 units of time, but for the carry to arrive it is going to take forever. You can see that this gap 6 5, 7 5, 8 5 is growing linearly. It is going to go 9 5, 10 5 and it will keep going. So, that is and it is just a waste of time. There is signal which is ready to evaluate, but it is just waiting. How do I overcome this problem?

(Refer Slide Time: 05:03)



I will give you a clue. This is called a linear carry select adder. What is it that I have to change? Sorry? So, what is it that is actually causing this mismatching delay? Suppose, I made the segment 3 bits, what would be this delay here? No, what would be the delay here?

Student: (Refer Time: 06:05).

4. What would be the delay here?

Student: 5.

5.

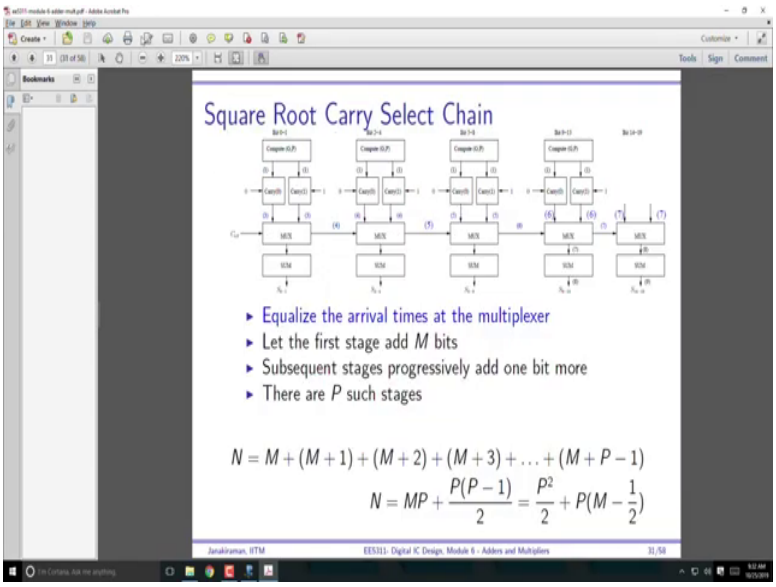
Student: Sir, I got (Refer Time: 06:17).

Exactly. See the first stage is the one through which you have to ripple. So, if you reduce that ripple time then the output will be ready quickly, but now the following stages actually have more time to evaluate other things and be ready. In some sense, if you want the circuit to be very fast you have to make sure that all signals to it is input arrive at a same time. There is no point in making something wait.

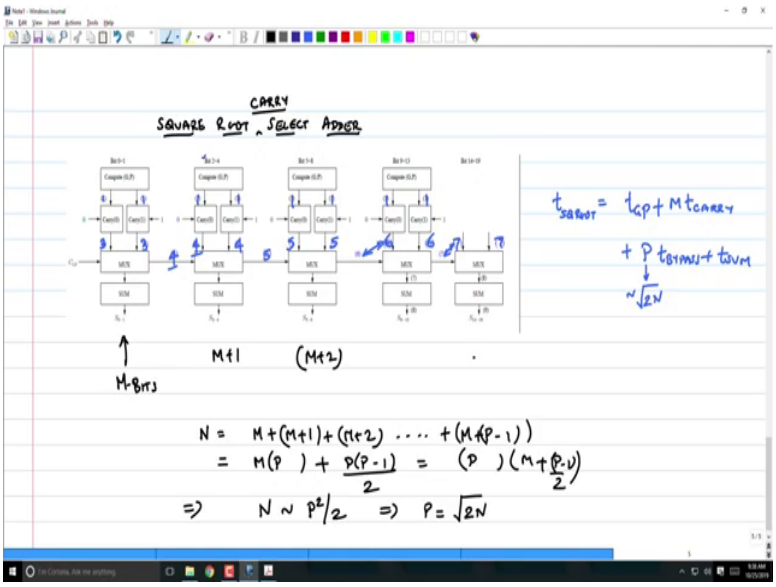
In fact, earlier also in the gate sizing algorithm that I told in the full circuit what happens is when I have multiple paths and all of them let us say have similar around similar delays. Post optimization you will find that all these delays become even closer. All paths basically you are saying that I do not want one path to very fast. So, I am going to make that you know little slower there by making a other path fast.

So, in some sense this similar delays is what optimizes the performance of a circuit in general and that is exactly what we are doing here. Make sure the arrival time at the input to every block is similar; do not waste time waiting ok.

(Refer Slide Time: 07:54)



(Refer Slide Time: 08:13)



So, what do we do? Basically, you just have to you have to do the following. Like he said instead of having uniform bits in each segment have increasing bits. Obviously, you have to start with the least number which is 2 bits right. You have a 2-bit ripple adder in which case the arrival time at after compute GP will obviously, be 1, right. There is no doubt in that, at this it will be 3 because I have to ripple only 2 stages now 2 plus 1. This will be because that is now rippling through three stages; it is going to take 4 units of time. So, this will be 4, 5, 6 right 7 and so on.

Now, the multiplexer will take one unit of time. So, this will arrive at 4 what are we done we have make sure these two arrival times are the same. Earlier that we sharpen at 5, now it is happening at 4. Now, at following stage again arrival time is 5, the arrival time of the other stage is also 5 everything we are matching perfectly like this, correct. So, now, how do you find out you know how to partition this let us say you start with M bits right. Let this be M

bits progressively you increase it by 1. This is M plus 1 M plus 2 and so on. So, if I have a total of N bits clearly this is going to be M plus M plus 1 plus M plus 2 up to let us say M plus P or M plus P. So, what is the sum? Sorry.

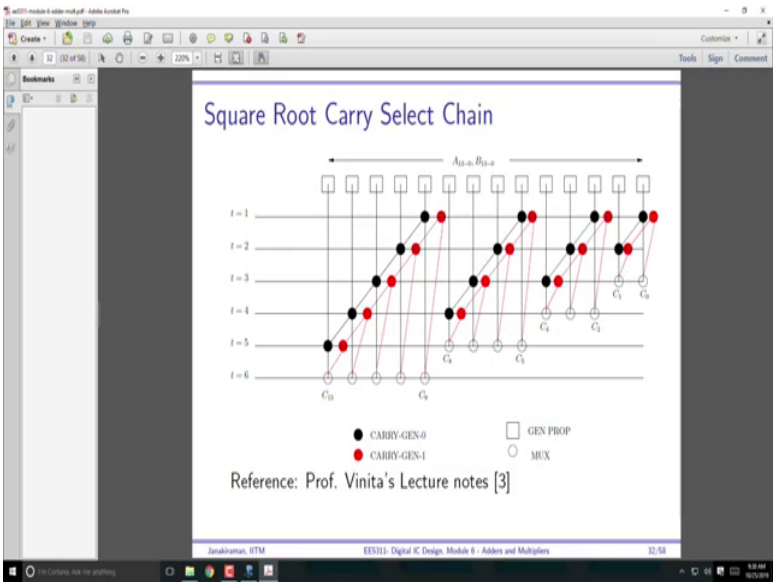
Student: (Refer Time: 10:45) P plus 1 by 2.

M into P plus 1 plus P into P plus 1 by 2 right, if M is very small right. So, let me write this as P plus 1 into M plus P by 2. Assume that I need a 128 bit adder right then you can neglect this M right. I can say n is approximately P squared by 2 of course, P n P plus 1 also I am approximating as P which implies that P is equal to root 2N that is a number of stages right.

So, how many bypasses do I have to do? P bypasses have to do or P plus 1 bypasses I have to do right therefore, I think I should have made this P minus 1 M plus P minus 1 actually you know. So, that let me do that because it makes the P into P minus 1 by 2.

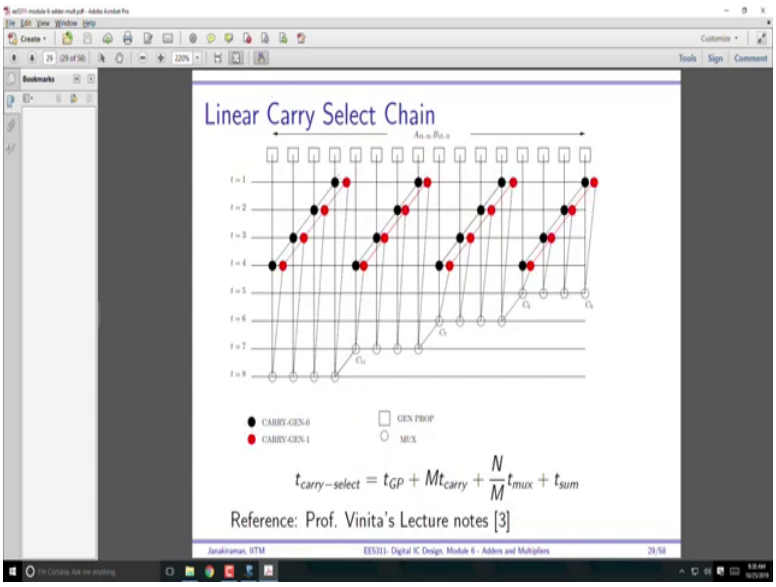
Then what happens is there are P stages right then I can now write the t I will fill out that because the name is in we we will get the name from this expression t GP plus how many ripples M t carry plus how many stages of bypasses now? P stages of bypass plus t sum. What is P now? This is approximately root of 2N and therefore, this is called square root adder ok. This is a that was a linear carry select adder, square root select adder carry select adder, clear? Any questions here?.

(Refer Slide Time: 14:06)



So, what we have done is if you look at this whole you know. So, by the way what is shown here is just another time representation of what happens for signal flow ok.

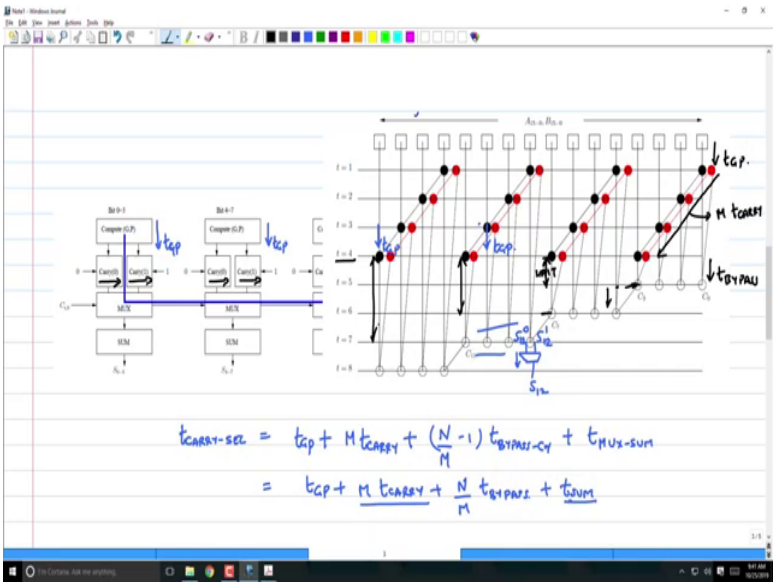
(Refer Slide Time: 14:15)



Let me just show you that. So, because if you see the slides so, if you look at the linear carry select chain right on the y-axis is time t equal to 1, 2, 3, 4, 5, 6 right this is my carry 0 stage right. So, in 1 unit of time the GP signals would have been ready. What is shown on this black and red is the rippling through the 4 bits there in that stage. So, in 1 unit of time the signal goes by 4 units of time right what you see is across all the stages right may be I will just explain this there.

This is this is also another nice way of visualizing this delay right we wrote out the arrival times, but this is another nice way of visualizing the delay. So, what is this?

(Refer Slide Time: 15:19)

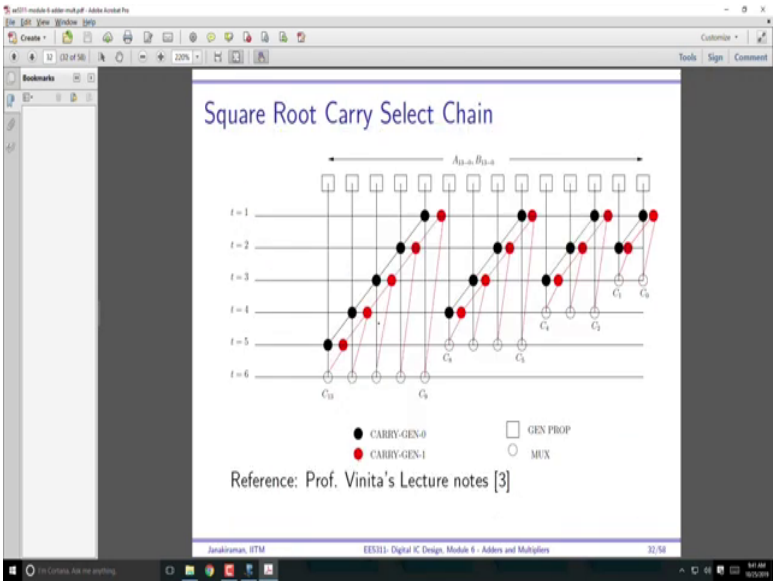


So, what is shown here is essentially the carry is rippling through the stages here. So, this delay is basically the delay through this ripple adder, but the point is all the ripples happen in parallel. That is why you see every stage is rippling in parallel. So, these are nice way to visualize and see that it is all are parallel operation right. Now, you go to go through the bypass MUX. This is the bypass t bypass right. This delay is basically the M t carry. And, of course, this delay is the t GP.

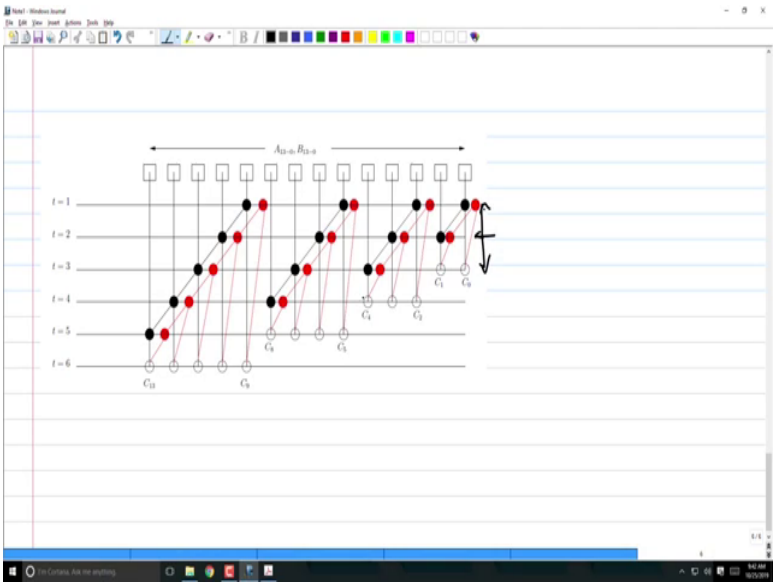
So, now, what you see is the carry has arrived here, my rippling has finished at t equal to 4 you have to wait that 1 unit of time for this carry to be available only then the evaluation can happen, from here it will take one unit of time. So, what you see is even though all the stages have their rippling done by t equal to 4, you are still waiting longer and longer as you progressively go in to the higher bits, longer and longer is this is the waiting time right sorry.

This is the waiting time. Here you are waiting 2 units of time and here you going to wait 3 units of time ok.

(Refer Slide Time: 17:34)



(Refer Slide Time: 17:36)



Now, when you go to the square root adder what you see is now the rippling time itself is different for all the stages. The first stage ripples in one in 2 units of time right this thing; second stage ripples in 3; third in 4; 4 in 5 and so on right and therefore, the waiting time is basically 0. Essentially we have eliminated the waiting time. So, this is another nice way to represent the delays right and so, if you see my slides you should not get confuse as what this representation is.