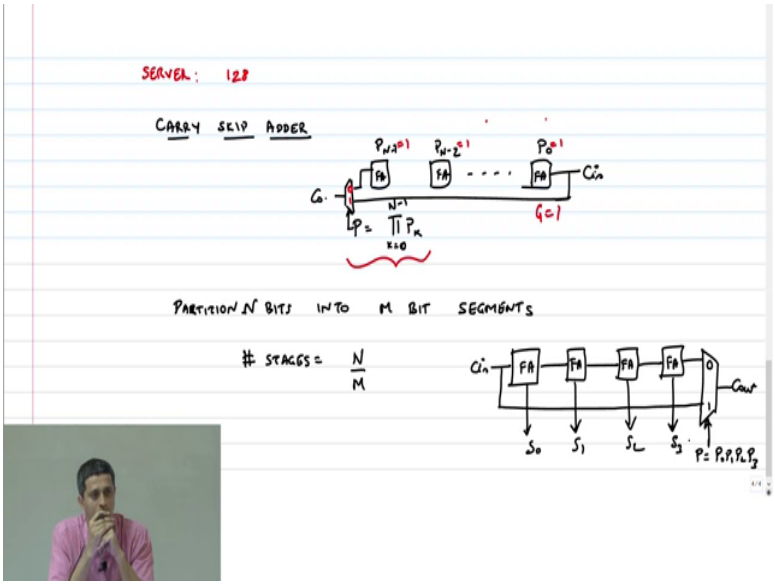
Digital IC Design
Prof. Janakiraman Viraraghavan
Department of Electrical Engineering
Indian Institute of Technology, Madras

Lecture – 57
Carry Skip Adder

The net point is there are many other circuit implementations which will allow you to optimize the delay of this full adder somehow right. Unfortunately there is only a certain amount of gain that you can get by optimizing the delay at a circuit level.

(Refer Slide Time: 00:41)



So, for example if I am doing an addition in a server it is a 128 bit machine, just think about the fact how much ever I optimize my delay in the at a circuit level right. I reduced t SUM and t CARRY to the maximum extent possible ultimately the delay is going to be killed by n.

But I want to do the ripple addition of 128 bits it is in that n is going to dominate the delay term and not t SUM and the t CARRY. So, there is only certain amount that you should spend optimizing the circuit part of a full adder. Beyond the point even if you have to do even a desktop machine 32 bits right. Now, even that 62 64 bits right G P used all of these are going to have larger bit registers and in order to optimize that ripple CARRY delay we have to optimize the architecture of addition, it is not possible to do it through circuit optimization anymore ok. So, we will look at various architectures of adders going forward and some of them we will cover in this class.

So the first thing is called a CARRY skip adder ok. So, the idea is essentially like this, suppose I have an in between ripple adder full adder full adder. Suppose my aim was to just ensure that the final CARRY out came out as fast as possible you should come out as fast as possible. The simplest thing to do final CARRY out ok, remember that if any one of the stages in between generates a CARRY, then you are done.

The CARRY will appear much faster than the worst case critical path ok. So, if I want to make sure that that worst case propagation is killed right, all I have to do is take the propagate signals P N minus 2 P N minus 1 right. And create a signal called P equals product of P k 0 to n minus 1. Which means that if all the adders are in propagate mode, all I have to do to get the C out is to BYPASS these signal BYPASS entire adder path and takes C into the output right that is what propagate means. If every adder that is why we are using an condition there, if every adder is in propagate mode CARRY out equal to c in.

So what you do is you simply put a multiplexer here C in this is my C out and this is multiplexer will be controlled by this propagate signal. If it is 0 this is 1, if it is 0 it means at least one adder in between was not in propagate mode. Which means that the CARRY is being generated somewhere in between or the CARRY is being killed somewhere in between and therefore that you have to wait for the CARRY to propagate from there right. But if P happens to be 1 then C out is simply equal to C in, this way all I am doing is I am not saying that this address is unconditionally faster that worst case delay I have reduced. The worst case delay now what is the worst case delay here by tell me?

Student: (Refer Time: 05:07).

At least in what mode should each of these n adders be in. If everything is in propagate mode ok; if everything happens to be in propagate mode that is this equal to 1 this equal to 1, then and for how long will CARRY out it take to settle. It is going to take one unit of time to create all the propagate signals a XOR b operation will happen in parallel across all the adders, you will get all propagate signals in one shot. Then you just need to do this addition and multiplexing in two units of time or let say three units of time does not matter, in some fixed time which is independent of N I have got my CARRY out ready. So, it cannot be the everything in propagate mode first one is in.

Student: generate

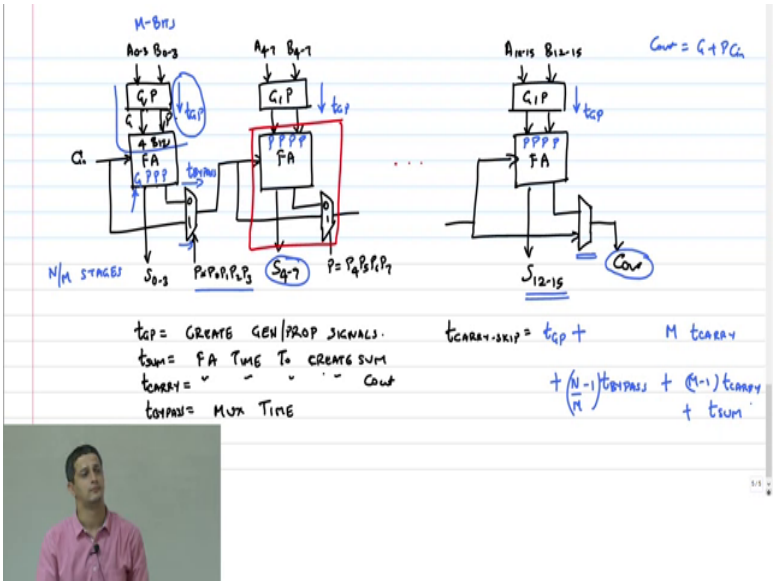
Generate mode exactly. So, the worst case is this has to be in generate mode and then the CARRY has to propagate all through right. Clearly I cannot do this for all the 128 bits, because the ultimately my worst case is if you look at the second worst critical delay time right. It will be the first one being in generate and everything and all that is going to kill my time and in my critical path. Therefore, I have to do something where this even if one stage is generating that propagation should not take enormously long amount of time.

The second key point is if this generates if one stage generates suppose I split my 128 bits into chunks of let say 4 bits. Now, if a CARRY is generated in any one stage, then I do not have to wait for the CARRY to propagate all through this is the key idea. But let me show the schematic then you figure out what I am saying ok.

So, the idea is now to partition; partition N bits into k bit segments and I think I used M slides. So, let me use M bit segments and create number of stages therefore will be will simply be N by M ok. So, what I am going to do is full adder 0 let say I am going to partition this with M equal to 4; 4 bits and I am trying to do 16 bit addition that is the example you are going to consider full adder full adder then a MUX C in.

So, this P is simply going to be P 0 P 1 P 2 P 3, I select for the MUX and if it is in propagate mode I am going to create the C out. The sum is still going to come out by the way like this there is nothing changes there this is S0 S1 S2 and S3 ok. So, what I am going to do is I am going to take this circuit and now repeat it n number of times ok. So, let me just copy this schematic. So, that I saved some time here it may not be a good idea, let me redraw that because I have already indicated the critical part there ok.

(Refer Slide Time: 09:33)



So, each block I first need to create the generate and propagate signals right, that is going to take one unit of time. So, that is the first block that I will put G P and this is going to take 4 bits A 0 to 3 and B 0 to 3. Then I have a full adder and then what I am going to do is it this is my C in, I want to put a BYPASS MUX here CARRY out will come from here and C in will come from here, this is my propagate signal P equals P 0 P 1 P 2 and P 3 ok.

Of course from here I will get the SUM S 0 to 3 as well G and P, now I will repeat this block G comma P A 4 to 7 P 4 to 7. Again my full adder four bit full adder changes then again I will put a MUX here P equal to P 4 P 5 P 6 P 7 right. This is my thing CARRY out that will go here and this will also come for the BYPASS ok. This is 0 1 0 1 this block here is what I showed in the previous slide this one 4 ripple ladders with one multiplexer behind.

So, like this I can take it on the last stage will be basically we assign be G P, A 8 to 11, 12 to 15 right b 12 to 15 again I have a full adder C in BYPASS C out and this is S 12 to 15 this is S 4 to 7 this is the architecture of the CARRY skip adder at every n bits we are going to try and skip right the CARRY out part through that BYPASS multiplexer ok.

So, now, let us define some terms here let us say t G P is equal to time to generate time to create generate propagate signals which means that this operation going from the first block, the time here is t G P all of them happen in parallel. Then of course, I have t SUM equal to basically time to create SUM t CARRY is at that circuit level for a full adder how much time it takes to create SUM right. This is full adder time to SUM full adder time to create C out.

Now, there is a BYPASS multiplexer that I have put in the path. So, let us say that this time is t BYPASS MUX time ok. So, with that I think we have covered all the blocks. Now, only thing we have to figure out is what is my worst case delay and what mode of operation should each of the 16 adders being. Can for example, bit A 5 and B 5 that full adder can it be in generate mode. Suppose it is; suppose it is right then of course from G 5 to bit 7 it has to propagate right that it is going to be in BYPASS mode my worst case.

Basically if A 15 and B 15 that full adder happen to be in generate mode then my CARRY is ready in immediately right. It is the same thing in the ripple adder in the last stage in the generic mode, then I can there is no there is no question of that being the worst case delay CARRY out is ready immediately. So, it has to be somewhere earlier, how much earlier? We said it has to go all the way to the first bit. So, if that I am asking similar questions here can any bit in between 0 to this thing be in generate mode, yeah?

Student: (Refer Time: 16:48).

You agree that it is the very similar to their ripple adder here, any bit in between being in generate mode cannot be the worst case delay because that will have a lesser time to propagate through now correct.

So therefore, all these adders in between have to be in propagate mode right. I am I have 4 adders, so I want to put all of that here propagate mode ok. Now, what about the first stage? Obviously three adders A1 to A4 I mean A3 have to be in propagate mode. What about the first adder?

Student: (Refer Time: 17:40).

A0 can it be in propagate mode, why?

Student: (Refer Time: 17:52).

Yes if the A0 B 0 adder happens to be in propagate mode, then this signal will go high and BYPASS the CARRY in. Therefore, that cannot be in the propagate mode that has to be in generate mode. My question is why cannot the other guys in the other stages why is that not the worst case, why is GPP not the worst case for the other stages? Because, if you generate in between there then the CARRY has to propagate must a lesser distance just like a ripple adder, get it. So therefore, you have to first learn to identify this worst case condition propagate or generate after that we can write out the critical path delay very easily ok.

Student: (Refer Time: 18:48) but output is not like depended and all of that then if main thing the arrival time of output also will not come like if (Refer Time: 16:58) input is changing, but the output will stay (Refer Time: 19:00) is not telling for their (Refer Time: 19:05).

Correct.

Student: So, the arrival time of output that we do not include that is.

Correct because it is max kind of thing right. So therefore, the worst guy will worst case guy will determine yeah ok. So now, can we write out the delay of this the critical part delay of this circuit t CARRY SKIP? Of course, first thing is you have to basically create this t G P, right.

It takes time to create the generate propagate signals. What about the delay through this first adder? The delay has to literally if you look at the critical path it has to go through this adder like this. Because it has to propagate it has to ripple the CARRY through that first stage correct. So, what is the ripple delay there? This full adder block here is a regular ripple full adder, it has only four bits that is all. How long does it take to words a rippling delay?

Student: (Refer Time: 20:32).

Yeah so basically t SUM plus.

Student: (Refer Time: 20:39).

Yeah so we said in general this is going to be M bits and the number of stage is will be N by M right. So therefore, this has to be what now? M minus 1 t CARRY sorry plus you have arrived till here. Now, I have to whether the MUX is by passing the CARRY or it is taking the CARRY from the rippled thing, you have to go through the BYPASS MUX right. In the first stage you are rippling to the thing and you to go through the BYPASS MUX, from second stage onwards there is no it is there is no rippling it is only bypassing right.

So, that is t BYPASS and how many bypasses N by M go let us write that down. Now, question is if you BYPASS the last MUX this MUX, what are you doing? You are basically telling me when C out will be ready this time that you told me t GP plus t SUM plus M minus 1 plus t CARRY plus this.

Is basically the time that is going to take for, it CARRY out to be ready the what about that these SUM bits? All the bits have to ready right here I did not worry about that because while the CARRY was rippling through the other stages these SUM bits will be evaluate. So, there is no problem there I do not want to count that, but question is for the last stage alone how will you handle this correction?

Student: (Refer Time: 23:05) I will make (Refer Time: 23:06).

Exactly you have to allow these to go through N by M minus 1 BYPASS right and now what should be the delay for the last stage? Yeah, plus?

Student: (Refer Time: 23:36).

Plus t SUM.

Student: (Refer Time: 23:51).

So, plus t SUM in to t ok, BYPASS yeah.

Student: (Refer Time: 24:15) first part (Refer Time: 24:17) it should be (Refer Time: 24:21) because they do not need to wait for the SUM that (Refer Time: 24:25).

That is also a very good point.

Student: (Refer Time: 24:31).

Good point. So, this point that in the first stage we do not have to the ripple the ripple worst case time, that we derived earlier for the ripple CARRY adder was for both SUM and CARRY to be ready t SUM and t CARRY that is why. Therefore, that is why we said it

should be t SUM plus N minus 1 in to t CARRY. But in the first stage when I am rippling through I do not have to wait for t SUM to evaluate.

I just have to wait for the CARRY out to be ready while that propagates this t SUM will evaluate. So therefore, this also worn be it has to be M times t CARRY, no M t CARRY it is not n minus 1 bits all the M bits are propagating.

Student: (Refer Time: 25:29).

First bit no no that is it is first bit it is generating the CARRY right. It takes t CARRY time to generate that CARRY. See whether you generate or propagate you have to perform a logic operation to get that. See the CARRY out will be G in to C in right I mean it will be it will be G in.

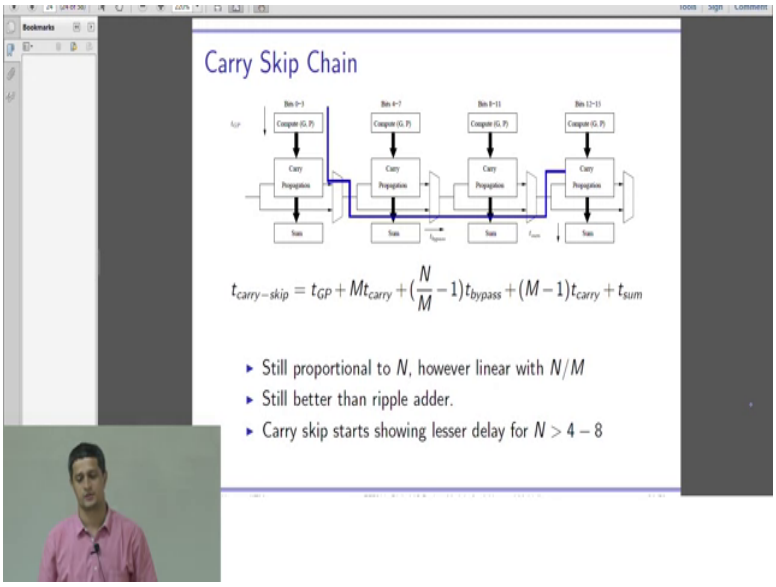
Student: (Refer Time: 25:59).

No no what I am saying is hold on what I am saying is that CARRY out is equal to G plus P into C in correct. Now, this is going to logic gate it does not matter if G this just appearing at C out, but it still has to go through the logic gate and that we are saying is going to take to t CARRY. Whether you are doing propagate time C in or just taking G to the output right. You have to go through that inversion logic right you have to add an inverter you have to do. So, that is going to take t CARRY. So, therefore, this has to be only M times t CARRY right plus now the last stage has to be what?

Student: (Refer Time: 26:55).

Yeah N minus 1 t CARRY plus t SUM. Now, we will check if you are right.

(Refer Slide Time: 27:13)



Yes t G P M t carry plus N minus N by M minus 1 t bypass plus M minu1 t carry plus t sum clear. Since this is the first architecture I am doing it very slowly; I am doing it really slowly. So, that we go through the obvious mistakes and correct that that is the aim ok.