Digital Electronic Circuits

Prof. Goutam Saha
Department E and EC Engineering
Indian Institute of Technology, Kharagpur

Lecture – 24
Arithmetic Building Blocks – II

Hello, everybody. In the last class we had looked at adder-subtractor circuit, we defined half adder, we defined full addition and then we looked at 4-bit addition, even 8-bit addition and there we had seen ripple carry addition.

(Refer Slide Time: 00:33)

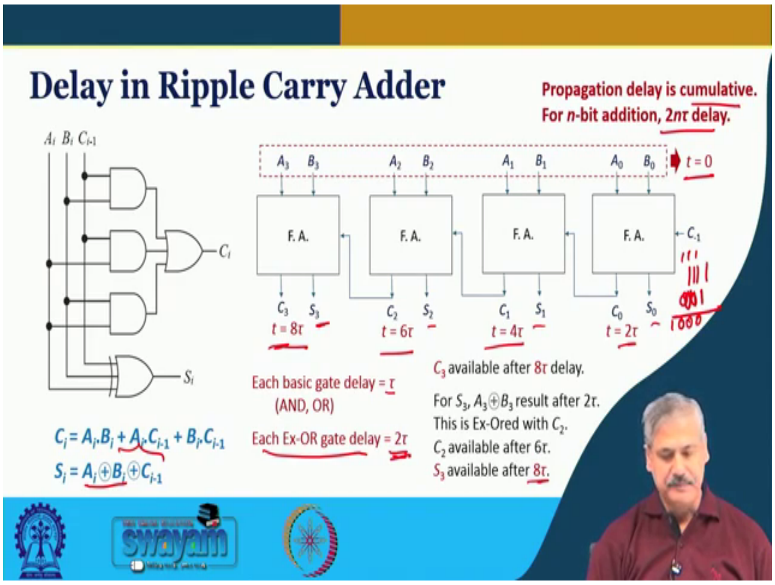
CONCEPTS COVERED

Concepts Covered:

- ❑ Propagation Delay in Ripple Carry Adder
- ❑ Carry Look Ahead Adder
- ❑ Cascading Fast Adders
- ❑ Group Carry Generation and Propagation

So, in today's class we shall look at how the propagation, what is the propagation delay in ripple carry adder, how it is calculated and then we shall look at some circuit which gives us first addition; I mean where the addition done addition is done in quicker time. So, addition is done means subtraction is also done when we talk about twos compliment arithmetic.

(Refer Slide Time: 01:03)



Delay in Ripple Carry Adder

Propagation delay is cumulative.
For n-bit addition, $2n\tau$ delay.

$C_i = A_i.B_i + A_i.C_{i-1} + B_i.C_{i-1}$

$S_i = A_i \oplus B_i \oplus C_{i-1}$

Each basic gate delay = τ
(AND, OR)

Each Ex-OR gate delay = 2τ

C_3 available after 8τ delay.

For S_3, $A_3 \oplus B_3$ result after 2τ.
This is Ex-Ored with C_2.
C_2 available after 6τ.
S_3 available after 8τ.

So, we begin with how what is the delay in ripple carry addition right. So, if you remember the example where in 4-bit addition 1111 that is A was added with B 0001. We found carry that was generated in the first stage, right with that ripples through the subsequent stages to come to get to arrive at the final carry. So, 1111, when it was added with 0001 so, carry generated here; so, this carry is finally, coming to the final stage. So, that is why it is called ripple carry.

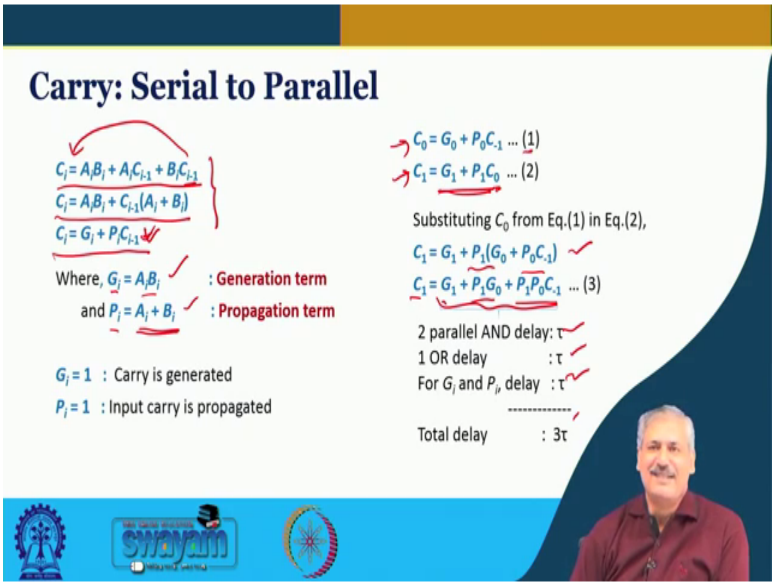
So, if we place the numbers at a times it t is equal to 0, and we assume that the each of the basic gate delay has a propagation delay tau minor variation can be there it is proximity tau for each one of them right and of course, we take note of that XOR gate required a AB prime A plus A prime B that kind of circuit which is which requires 2 tau gate delay to that is the two basic gate delays if you consider the way it is arranged. So, if that is the case then how much is the delay in completing the ripple carry addition for 4 -bit.

So, the maximum delay path the critical path that we shall see so, that is when the number is placed and you have to wait for the carry to arrive at the n most adder. So, that is the time that will consider right. So, t is equal to 0 and we have noted that the carry is generated by AB plus BC in plus AC in. So, that is one bank of AND gate and there is OR gate. So, 2 tau delays is there to generate the first carry C note, right. So, this will go over here, right and then again AND operation and OR operation will take place so, another 2 tau. So, C 1 will require another 2 tau that 4 tau C 2 6 tau and C 3 final equal required 8 tau.

So, if it is 8-bit it will require 16 tau. So, we see the propagation delay here is cumulative and for n-bit addition to n tau delivery be there, for the generation of the carry. And, let us see what is happening for the in the sum-bit generation. So, we considered see S 3; so, S 3 will be XOR B I, right. So, this ex or first part is done in to 2 tau time right, the XOR gate released 2 tau and C 3 C 2 is available at 6 tau. So, after that another XOR operation is required. So, the 6 tau and 2 tau together 8 tau will be the time after which S 3 is also available. So, that is the maximum time it is required.

So, others are available it earlier, but if you to consider as a whole this 4-bit addition will require 8 tau amount of time in maximum case, when carry is there at the n most point.

(Refer Slide Time: 04:46)



Carry: Serial to Parallel

$C_i = A_iB_i + A_iC_{i-1} + B_iC_{i-1}$

$C_i = A_iB_i + C_{i-1}(A_i + B_i)$

$C_i = G_i + P_iC_{i-1}$

Where, $G_i = A_iB_i$: Generation term

and $P_i = A_i + B_i$: Propagation term

$G_i = 1$: Carry is generated

$P_i = 1$: Input carry is propagated

$C_0 = G_0 + P_0C_{-1}$ ... (1)

$C_1 = G_1 + P_1C_0$ ... (2)

Substituting C_0 from Eq.(1) in Eq.(2),

$C_1 = G_1 + P_1(G_0 + P_0C_{-1})$

$C_1 = G_1 + P_1G_0 + P_1P_0C_{-1}$ ... (3)

2 parallel AND delay: τ

1 OR delay : τ

For G_i and P_i, delay : τ

Total delay : 3τ

Now, we said that to you look at some mechanism by which this edition this time can be reduced, right because you have seen if the if the number of bits are more than the delay is you know getting accumulated domain is if it is n, it is 2n tau that kind of thing.

So, one way of doing things is to investigate the carry generation equation and if we look at it; so, this is the basic carry generation equation right and we see that there is a recursive a kind of you know activity within the equation. So, C i generation is waiting for C i minus 1 C i minus will wait for C i minus 2. C i minus 2 will wait for C i minus 3.

So, there is n I mean it is happening in cascade serially one after another and that because of which the delay is getting in accumulated, ok.

So, the method that we shall try is to make this process parallel instead of serial let us see how we can do it. So, to do that, we are defining two terms. So, first of all we are writing C i as A i B i and the C i minus 1 taking common A plus B i and then we are writing in it is in this way, ok. So, it is easier to for notation purposes also every night to do not need to write two variable. So, one is G i which is A i B i another is P i which is A i plus B i, one is obtained by AND gate after one gate, basic gate delay and these obtained by OR gate to input OR gate after again one basic gate delay.

Now, the first one is called generation term, and the second one is called propagation term. What why are they called so? Generation term if G i is 1, if G i is 1 then carry will be generated at that stage it does not matter whether a carry was there in the previous stage or not, if both A and B are 1 A i and B i are 1, so, C i will be 1 it does not really matter whether previous stage is generating any carry or not. So, that why it is called generation term.

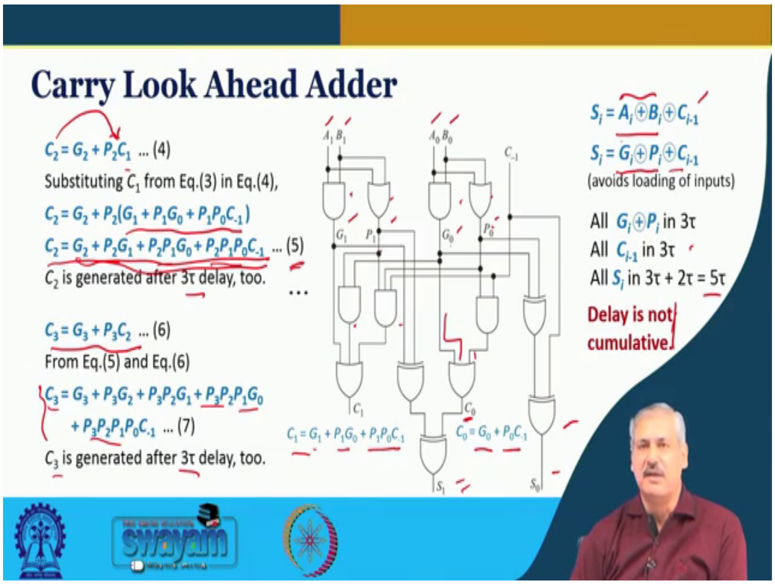
And, what is the propagation term P i, what is its significance? If P i is 0 whether P i is 0 or not we can figure it out come this stage itself. If P i is 0 then when if the carry is there in the previous stage it will not get propagated because C i C i is a minus 1 is ANDed with P i, ok. So, only if P i is 1 propagation term is 1 and if there is carry in the previous stage then that will contribute to generation of the carry in this stage. So, that is why this called propagation term, right.

So, this is the significance of it now with this G i and P i right we try to revisit the equation. So, C naught is G naught plus P naught C minus 1 C minus 1 is the first stage. So, from the previous stage if it there or if it is the very first stage it will be 0, otherwise if it is in cascade so something will come from the previous stage, right.

And, C 1 is G 1 plus P 1 C naught the basic equation this equation only this equation we are implying. So, this C naught, we can substitute from equation 1. In equation 2, so, that is what we are doing here. So, if you do that what will happen C 1 is G 1 plus P 1 G naught plus P 1 P naught C minus 1, this is what we have got and by this by this what have be achieved in terms of delay. So, G i and P i generation 1 tau it is there. So, this is one AND gate this is one AND gate two input and three input AND gates.

So, they are generating output in one tau and finally, this is all; that is one tau. So, 3 tau after 3 tau we get the value of C 1. Earlier using ripple carry mode what was the time required for C 1 it was 4 tau. So, there is some benefit. Whether this benefit is there for larger number of bit addition so, 3 bit, 4 bit let us see in the next slide.

(Refer Slide Time: 09:43)



Carry Look Ahead Adder

$C_2 = G_2 + P_2C_1$ ... (4)

Substituting C_1 from Eq.(3) in Eq.(4),

$C_2 = G_2 + P_2(G_1 + P_1G_0 + P_1P_0C_{-1})$

$C_2 = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_{-1}$ ... (5)

C_2 is generated after 3τ delay, too.

...

$C_3 = G_3 + P_3C_2$ ... (6)

From Eq.(5) and Eq.(6)

$C_3 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_{-1}$ ... (7)

C_3 is generated after 3τ delay, too.

$C_1 = G_1 + P_1G_0 + P_1P_0C_{-1}$

$C_0 = G_0 + P_0C_{-1}$

$S_i = A_i \oplus B_i \oplus C_{i-1}$

$S_i = G_i \oplus P_i \oplus C_{i-1}$

(avoids loading of inputs)

All $G_i \oplus P_i$ in 3τ

All C_{i-1} in 3τ

All S_i in 3τ + 2τ = 5τ

Delay is not cumulative.

So, for that we investigate the basic equation of C 2; C 2 is G 2 plus P 2 C 1 that is a basic question the recursive equation C 1, ok. So, C 1 we substitute from the previous equation as G 1 plus P 1 G naught plus P 1 P naught C minus 1, then we expand it, ok. So, again what we see that each of these P i and G i's are generated after 1 tau. Each of these AND gate which is doing a parallel operation will take 1 tau, ok. So, 2 tau and final one OR gate to combine all this product terms, right. So, another tau. So, it is obtained after 3 tau delay. So, even for 3-bit addition it is 3 tau that is generated after 3 tau.

And, C 3 for 4 bit operation again basic question is G 3 P 3 plus P 3 C 2; C 2 we can substitute from equation 5 here and again will get equation similar to this. There also we can see that G i P i terms will generate in one tau and this product terms another tau and final one OR terms it is total 3 tau. So, irrespective of the number of bits we can see that the time required to generate the carry here is 3 tau by this bringing; this kind of the parallelism instead of the serial mode of carry generation that we are seen before.

Now, one need to be careful about the you know how many gate it is getting connected to the fan in fan out issues. So, as long as that is not balloted it is able to you know provide this much of current or sing current. So, there is no issue, but that is something by way for which we cannot you know in definitely extend it, we have to confined it to certain number of bits ok.

So, now then that is the way the carry is getting generated. So, carry is generated after 3 tau, right and what about the sum bit. So, A XOR B i that is already available through XOR gate in 2 tau and the carry we are waiting for the carry to come which is 3 tau for 4 bit addition or so, I mean which is the same. So, 3 tau and 2 tau together after 5 tau we are getting the final result maximum 5 tau. Earlier it was you have seen that it is 8 tau. So, in this case the delay is not cumulative.

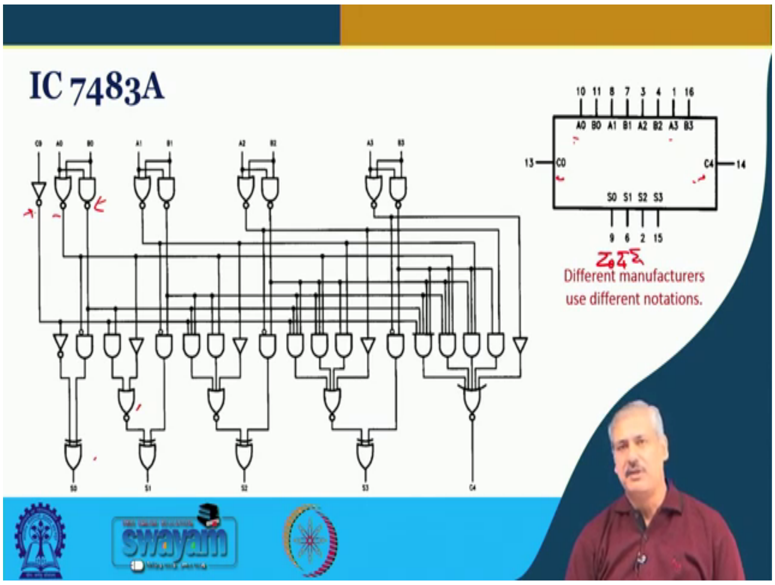
And, how we realize this circuit? Ok. So, the circuit realization is little bit more complex than what was in case of this normal ripple carry addition. So, we are showing you now two such bits here. So, such sum bits that is un carry bits that is getting generated.

So, this is your A 1 B 1 and this A naught B naught and this is G 1 P 1 and G naught P naught are getting generated by AND gate OR gate, and then the C naught will require G naught and plus P naught C minus 1, right. So, this is the G naught is coming over here, this is or bit P naught and C minus 1. So, this is another input of the OR gate, ok. So, this is how C naught is getting generated. and, what about C 1? C 1 will G 1 plus P 1 G naught plus P 1 P naught C minus 1. So, G 1, right and then your P 1 G naught this is your P 1 G naught and then P 1 P naught C minus 1 this is P 1 P naught C minus 1.

So, that is how C 1 is getting generated. Now regarding the sum bits that is to be generated we can generate from XOR B also you can calculate and you can see that G XOR P XOR C i is also giving the same output. So, it is preferred that we generate it from G XOR P i instead of A XOR P i because it is connected to the external world. So, how the it will load and other things that is not much known to us how it is getting driven in the external circuitry. So, here we know how much current it can handle and other things.

So, we when we generate the final sum bit we take the XORing from G i's and P i's instead of A i's and B i, A i and B i.

(Refer Slide Time: 14:34)



IC 7483A

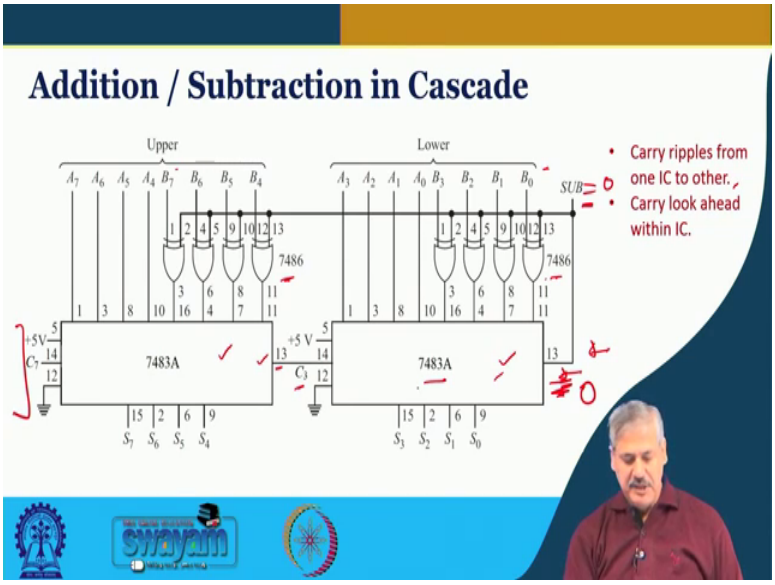
Different manufacturers use different notations.

So, this is a commercial IC; IC 7483 A you can search and you can see there are other ICs is also 7483 and where which thus 4-bit first addition and here this is you know different manufactures has got different way of representing the number. So, the here these A naught, A 1, A 2, A 3 that is mentioned, B naught, B 1, B 2, B 3 is mentioned, output is S naught to S 3, (Refer Time: 15:10) we are followed. But, input we have use C minus C 1 and C 3 the final output carry output they are using C naught and C 4, ok. So, on value more.

So, in some other manufacture you will see instead of some bit they are using the sigma sign summation sign 1 summation 2 and so on and so forth. So, these are certain small small difference is that you have to take note when you are using it practically in lab or in your personal hobby project or mini project, digital electronics project.

The other thing that you take note of is the use of you know NOR gate, NAND gate and here it here again NOR gate, instead of the kind of thing AND and OR that we have used ok, for which again if you work it out then will see the same logic we shall see one example at the end for a carry look at adder circuit ok, group carry generation in relation to that particular discussion. So, this something we take note of when we use this kind of IC.

(Refer Slide Time: 16:25)



Addition / Subtraction in Cascade

- Carry ripples from one IC to other.
- Carry look ahead within IC.

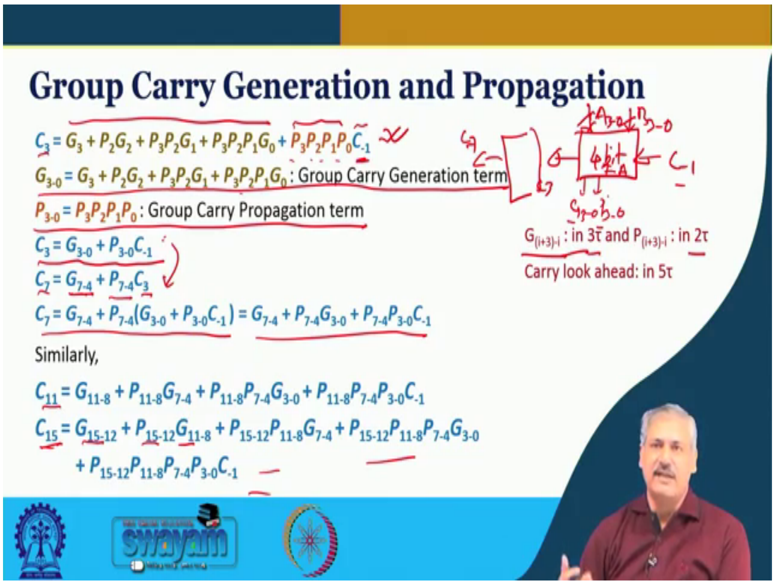
Now, these are in a 4-bit first adder, and we need 8-bit addition, so, how we go about it? Now, here again we shall be using cascading right. So, this cascading forecasting we shall be using one adder and this is another adder. Now, in between how we are connecting? The output carry that is generated here is going as input carry. So, this is nothing, but the mote which is similar to ripple carry addition. Is it not? So, this carry input 13 here carry input of the more significant bits will be waiting for this C 3 2 get generated from the previous stage, which will required 3 tau.

So, if you got another such stage here after this. So, this is 4-bit, this is 4-bit, so, this is another say 4-bit over here. So, it will wait for 6 tau to generate this particular carry right. So, that way this becomes within the you know across stages across these fordable blocks the carry ripples from one IC to another. So, that becomes cumulative with in this block it is not when we are looking for this kind of arrangement. Is it ok? Is it understood? ok, but each of this block is a first adder you know the kind of addition that we have seen before which is called carry look at adder. And, here again we are showing the arrangement of getting both addition and subtraction using first adder and for that you are using the similar approach that we are done before.

So, we are having a bank of XOR gate here and bank of XOR gate here and this sum in put is 1 when we are doing subtraction and which adds to the carry initial carry in the twos' complement of this B 7 to B naught number. So, this is ones' complement at the XOR gate output and 1 that is getting as input carry over here, ok, right and when it is addition this sum will be 0. So, basically only B naught to be 7 will come over here no inversion will be done and this is 0. So, this is normal addition that will take place.

So, we take note that within the, you know across this blocks it is carry is replaying.

(Refer Slide Time: 19:05)



Group Carry Generation and Propagation

$C_3 = G_3 + P_2G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_{-1}$

$G_{3\text{-}0} = G_3 + P_2G_2 + P_3P_2G_1 + P_3P_2P_1G_0$: Group Carry Generation term

$P_{3\text{-}0} = P_3P_2P_1P_0$: Group Carry Propagation term

$C_3 = G_{3\text{-}0} + P_{3\text{-}0}C_{-1}$

$C_7 = G_{7\text{-}4} + P_{7\text{-}4}C_3$

$C_7 = G_{7\text{-}4} + P_{7\text{-}4}(G_{3\text{-}0} + P_{3\text{-}0}C_{-1}) = G_{7\text{-}4} + P_{7\text{-}4}G_{3\text{-}0} + P_{7\text{-}4}P_{3\text{-}0}C_{-1}$

Similarly,

$C_{11} = G_{11\text{-}8} + P_{11\text{-}8}G_{7\text{-}4} + P_{11\text{-}8}P_{7\text{-}4}G_{3\text{-}0} + P_{11\text{-}8}P_{7\text{-}4}P_{3\text{-}0}C_{-1}$

$C_{15} = G_{15\text{-}12} + P_{15\text{-}12}G_{11\text{-}8} + P_{15\text{-}12}P_{11\text{-}8}G_{7\text{-}4} + P_{15\text{-}12}P_{11\text{-}8}P_{7\text{-}4}G_{3\text{-}0}$
$+ P_{15\text{-}12}P_{11\text{-}8}P_{7\text{-}4}P_{3\text{-}0}C_{-1}$

$G_{(i+3)\text{-}i}$: in 3τ and $P_{(i+3)\text{-}i}$: in 2τ

Carry look ahead: in 5τ

Now, we look at a mechanism a method by which that particular aspect can be addressed, ok. If we have got first addition in our mind right and we are having a situation where more than 4-bit addition is required say maybe 12 bit thirty 8 or 16-bit. So, so how we can what can do to (Refer Time: 19:38). So, to do that let us again look at the basic equation of the carry generation.

So, the carry that is C 3 is going as input to the next block that we had seen. So, you are looking at this C 3. So, how C 3 is getting generated? So, this is why C 3 is getting generated in the first carry first adder, ok. So, this term this term ok, we if when it is 1 this C 3 is getting generated, irrespective of there was a carry to the input to this block or not, ok. So, this we define as group carry generation term group carry generation term G 3 to 0, ok, is it clear? And, this particular term in different color, when this is 0, then even if there is a carry present at the input of this block this is not get propagated. This will get is exactly similar you will see the equation and other things that will get is exactly identical to the one that we are seen in case of first adder development is it not?

So, this we say this we say group carry because it is generated from individual A i's and B i's the propagation term the oring of A i's and B i's individual bitwise ORing. So, this is the group carry propagation term right. So, now, we are looking at this as a group. So, this is 4-bit adder first adder. So, this is your C 3 and this is C minus 1 and this is C 3 and this where your A 3 to 0 and this is your B 3 to 0 ok.

So, how what you can write now? Now, that you have defined them, so, C 3 is equal to G 3 to 0 P 3 to 0 C minus 1, it is similar to the equation that we had seen before if you remember, ok. So, this is the kind of equation that we had seen before. So, earlier we are doing at bitwise carry generation now it is block wise carry generation, right.

So, this is what we are doing it and this is group carry generation term and group carry propagation carry term. Earlier it is only G generation term and propagation term here group generation term and group carry propagation term, that was bit was this is block wise, ok. This is the difference. Clear? Now, if there is another block of after this which takes C 3 input and generates C 7 ok.

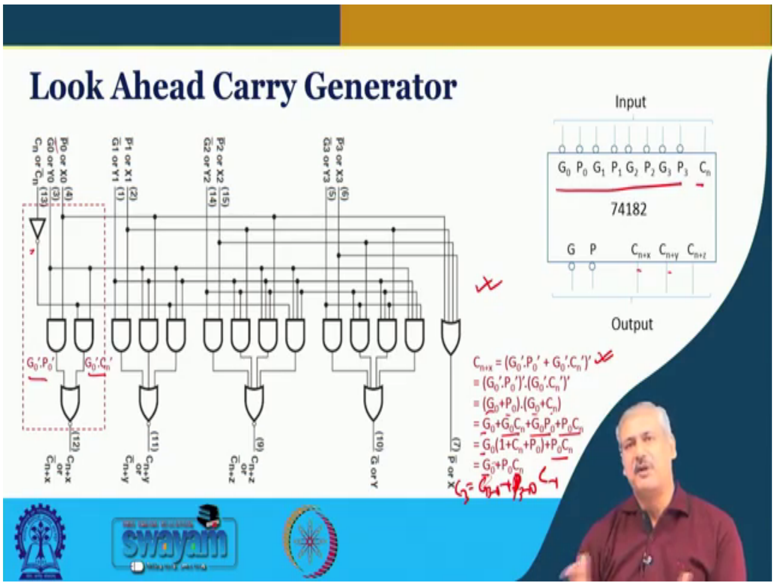
So, this is C 7 can be written from the generation terms and propagation term that is generated at that particular block G 7 to 4 and P 7 to 4 and with input as C 3 this just similar thing, right for the next particular block we can write it in this manner. Now, this C 3 we can substitute from the previous equation over here and then expand it, that is what you will get, the similar again identical to the treatment that we have when we develop first adder.

So, that way we can go on doing it for the next stage and next stage and all. So, for you know 16-bit addition, so, C 15 is getting generated in this manner, right. Now, each of these terms if you look at them. So, that if you are generating them at which of these output if you are additional you are generating this group carry generation term G 3 to 0 and P 3 to 0 each one of them if you are generating and if you are making use of them to generate the C 15, then how much time is required? So, this G 3 to 0, this will require 3 tau. Why? Each of these G i's and P i's requiring 1 tau, this product term is another tau and the final ORing is another tau.

So, G i each of these group carry generation term will required 3 tau and group carry propagation term will require 2 tau. So, the term here these are 2 tau 3 tau, these are 2 tau, this is 3 tau for their generation. So, the higher of 2 is 3 tau, and this 3 tau and then there is product 1 tau then finally, is an another ORing. So, this is after 5 tau getting this particular carry here this particular carry here.

So, you need to have an aggregator to and each of these terms need to generate this group carry generation group carry propagation term. And, and we see that the equation is similar these aggregator can also take individual bits also to generate the corresponding carry this is individual level the carry is, ok. This is what we can understand by comparing the equations.

(Refer Slide Time: 25:40)



Look Ahead Carry Generator

Input
G_0 P_0 G_1 P_1 G_2 P_2 G_3 P_3 C_n
74182
G P C_{n+x} C_{n+y} C_{n+z}
Output
C_n or $\bar{C}_n$ (13)
P0 or X0 (4)
G0 or Y0 (3)
P1 or X1 (2)
G1 or Y1 (1)
P2 or X2 (15)
G2 or Y2 (14)
P3 or X3 (6)
G3 or Y3 (5)
$G_0'.P_0'$
$G_0'.C_n'$
(12) C_{n+x} or $\bar{C}_{n+x}$
(11) C_{n+y} or $\bar{C}_{n+y}$
(9) C_{n+z} or $\bar{C}_{n+z}$
(10) $\bar{G}$ or Y
(7) $\bar{P}$ or X

$C_{n+x} = (G_0'.P_0' + G_0'.C_n')'$
$= (G_0'.P_0')'.(G_0'.C_n')'$
$= (G_0+P_0).(G_0+C_n)$
$= G_0+G_0C_n+G_0P_0+P_0C_n$
$= G_0(1+C_n+P_0)+P_0C_n$
$= G_0+P_0C_n$

So, this is one circuit for look at carry generator IC 74182, and this particular one takes four such terms, four generation term and four propagation terms, ok, four generation term and four propagation terms. So, this could be group carry group carry generation term and group carry propagation term as well or normal in basic numbers, ok.

Then, the corresponding the you know because the basic equations remain the same, they way we have seen it before. And, this is the input carry and this are the output carry for individual stages, individual stages and this is the final again group generation term and propagation term for multilevel use, next level use. And, this is the corresponding circuit and we told that again we are having you know inverter, then NOR gates and other things which make us think like the example that had given before. So, for example; this circuit. So, let us investigate one such particular block and see when it matches with the equation or not, ok.

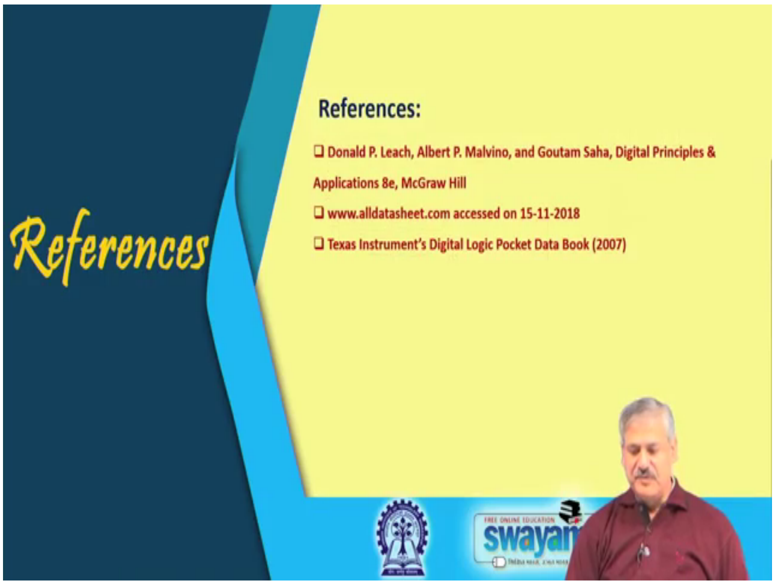
So, this is the block that we you know look at. So, this output over here is the input here is G naught bar and P naught bar that is getting ANDed, right and this input is G naught bar G naught is coming from here and C in and C in here is active high, but there is inverter. So, this is C n bar. So, in put these NOR gate is G naught bar P naught bar and G naught bar C n bar.

So, input to the nor gate which is generating C n plus x, the first level a carry is first is the this one, ok. Then you apply this DeMorgan theorem, right. So, this G naught bar P naught bar the whole bar ANDed with because this is OR sign is becoming a and term and each one of this terms is inverted complemented. So, G naught 1 C naught and this were then you again perform DeMorgan's theorem, ok.

So, this becomes G naught complement of complement that is G naught and P naught complement of complement that is P naught. Similarly, this is G naught and this is C n ok, right, you can follow it? Then when you perform the product get each of the finally, you may some of perform. So, G naught G naught this gives you G naught. So, G naught C n, G naught P naught and P naught C n and if you take common G naught, then 1 plus 1 naught C n or P naught which is nothing but, G naught and P naught C n. So, we get back the basic equation G naught plus P naught C n here is C minus 1 the one that we have use that concept. So, this becomes your you know C 1.

And, if it is group carry this are the group carry term this is G 3 to 0 plus P 3 to 0 and this C minus 1 and this is your C 3, ok, have you understood it? And, similarly for other terms depending on whether it is group carry or normal generation propagation term, it will generate for that particular bit or for that particular block.

(Refer Slide Time: 29:29)



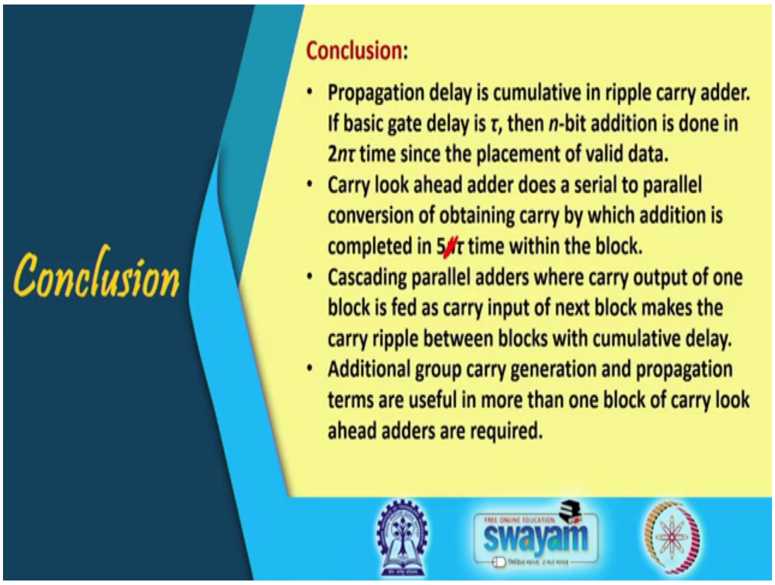
References:

❑ Donald P. Leach, Albert P. Malvino, and Goutam Saha, Digital Principles & Applications 8e, McGraw Hill

❑ www.alldatasheet.com accessed on 15-11-2018

❑ Texas Instrument's Digital Logic Pocket Data Book (2007)

(Refer Slide Time: 29:30)



Conclusion

Conclusion:

- Propagation delay is cumulative in ripple carry adder. If basic gate delay is τ, then n-bit addition is done in $2n\tau$ time since the placement of valid data.
- Carry look ahead adder does a serial to parallel conversion of obtaining carry by which addition is completed in $5\cancel{n}\tau$ time within the block.
- Cascading parallel adders where carry output of one block is fed as carry input of next block makes the carry ripple between blocks with cumulative delay.
- Additional group carry generation and propagation terms are useful in more than one block of carry look ahead adders are required.

So, with this we conclude today's class propagation delay is cumulative in ripple carry adder. If get basic delay is tau, if we consider that as tau then n-bit addition is done in 2n tau time since the placement of valid data. Carry look at adder does serial to parallel conversion internal process for optical carry by which the addition of completed in 5 tau time. Please remember, this is 5 tau time within the block.

Cascading parallel adders where carry output of one block is fed as carry input of the next block makes the carry ripple between blocks with cumulative delay, ok. And, to avoid that we can have additional group carry generation and propagation terms and by which we can get carry look at in by a aggregator and that reduces the time for generation of the carry, ok.

Thank you.