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

Lecture - 04
Algorithm to Efficient Architecture Mapping

Hello everyone. Welcome to the course on Architectural Design of IC's. So, in last few classes, we have seen the basics which are related to this course; that means, the basic like critical path, false path, what is false path, what is multi-cycle path; all these things. Apart from that what are the VLSI design styles are available and how you can chose; that means, and what is the VLSI design flow, we have seen that is FPGA flow or ASIC design flow we have seen. That where just to start or to familiar with what why I need to know about this architectural design of IC's. So, this few classes were on that to know about that these are the phases there in VLSI design flow.

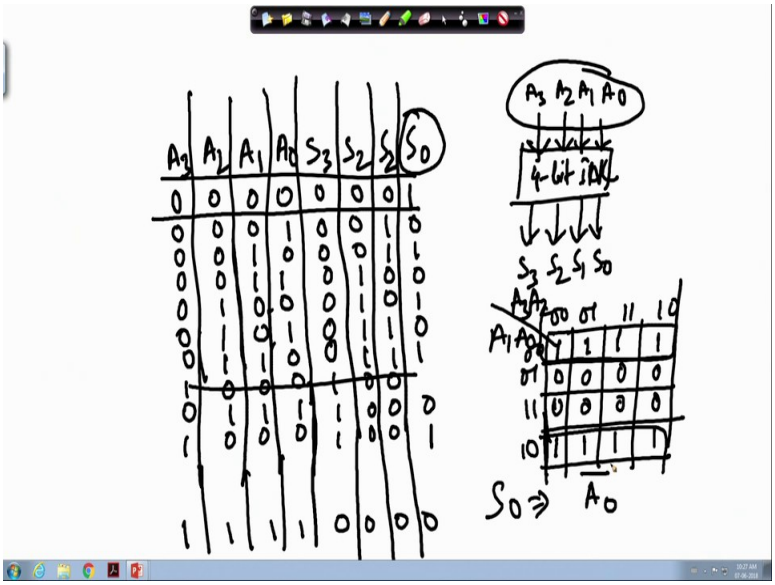
So, from today, we will start this Algorithm to Efficient Architecture Mapping ok. So, here what we will learn? We will see that so, several circuits I need to implement or I need to design. So, at that time how efficiently, I can design those circuits. Because these are the basic operation or these are the basic circuit which is implement anywhere to build up a or to design one small; that means, one big function ok; that means, to express any function I need arithmetic operation right.

So, here we will see or we will focus on to design efficient architecture at the; that means, this component level or this basic components will be designed in such a way so that whenever I will design the bigger circuit so, at that time I can get the benefit of it ok. So, the first is that I suppose I need to design 1 4-bit incrementer ok. So, 4-bit incrementer design means increment by 1. Let us consider first increment by 1; that means I need a circuit which will give if I give any number so, at that time it will in 1 each and; that means, whenever I will put the numbers so, at that time it will give me just plus 1 ok.

So, if I consider, suppose if I consider 4-bit so; that means I can give the values of 0 to 15 ok and I can get the value from 1 to 15 ok. So, as to implement whenever I will put 15; so, at that time I need 15 plus 1 that is equal to 16. So, 4-bit within 4-bit that will not be covered. So, I at that time it will be 0; it will be considered as 0 ok. So, how

efficiently or what will be the architecture for one, 4-bit incremental design so that we will start with this al algorithm to architecture mapping ok.

(Refer Slide Time: 03:57)



So, to start with that, so what I need? I need to design one 4-bit incrementer. So, 4-bit incrementer, suppose at that time if I consider this; this is my 4-bit incrementer. So, here these are the inputs suppose this is A 3; this is A 2; this is A 1; this is A 0. So, at the output I will get this is S 3; this is S 2; this is S 1; this is S 0. So, I need to design these particular, these particular 4-bit incrementer in such a way so that whenever I will put whenever I will give any of these 4- bit input so, I will get increment by 1 in each cycle right.

So, at that time, what I have to do? I have to do the thing is that how to; that means, where from I need to start ok. So that means, first you draw in any that means, whenever we design any circuit so, at that time at the very first what we know, about the gates also what we know? We know the truth table right.

So, here also as this is 1 circuit; so, if I draw, how it basically truth table means what? What is the operating principle of that particular circuit right? So, here I need to design one 4-bit incrementer. So, first I need to know what is the that means, operating principle of this 4-bit incrementer? So that means, now if I consider this 4-bit here, A 1, this is these are the 4 input bits.

So, as this is 4-bit; so that means, total 16 combination I will get. So, if I start with this 0 0 0 and at this particular side, I will be mentioned this output. What will be for corresponding input bit; what will be its output condition? Ok, for 0 0 0 0, if I give 0 0 0 0. So, at that time output will be this 0 0 0 1; then for 0 0 0 1, it will be 0 0 1 0. For 0 0 1 0, it will be 3. For 3, it will be 4. For 4, it will be 5. So, just increment by 1 right.

So, in this fashion, I have to fill this total table ok. So, then 1 1 0; then this will be 7; then for 8. It will be sorry for sorry for this it will be 7. It will be 8. For 8, it will be 9, something like this; that means, up to here I will get all 1, at the time it will be all 0's, correct. So, I have drawn the corresponding truth table for this particular 4-bit incrementer circuit.

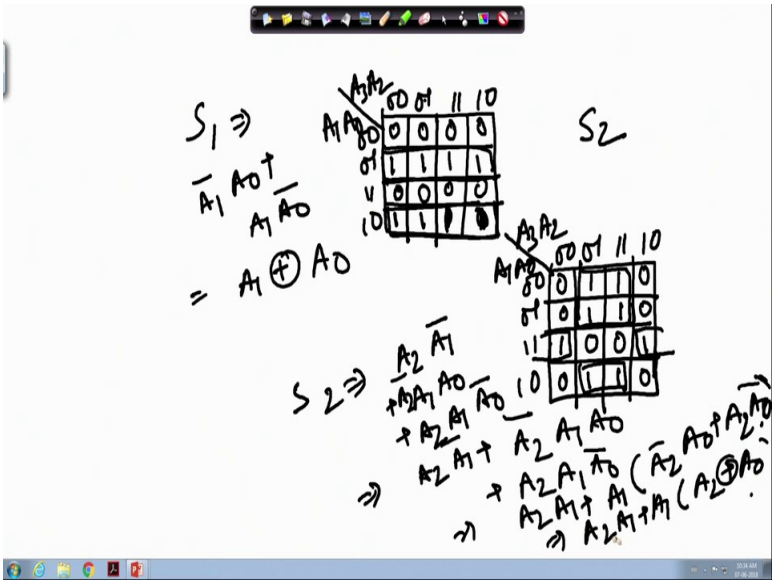
Now, how it will work; that means, this 4-bit incrementer, how it will work? So, from this truth table; that means, for the input changes how the output changes that is the behavior of this particular circuit. So, now, I have to make the connection for this particular output, what is the combinational logical expression in with this particular input, that I have to find out. So, how we can find out that thing? We can find out that thing using Karnaugh map ok. So, I think this is very basics, we have learned in digital electronics circuit design ok.

So, for this particular bit how this A 3, A 2, A 1 and A 0 is basically varying. So, that I have to see. So, for that; that means, for now for S 0, if I want to describe the logical expression for S 0. So, at that time, what I have to do? I have to put the Karnaugh map right ok. So, here at this, this is A 3, A 2; this is A 1 A 0. So, this is 0 0 0 1 1 1 1 0 0 0 0 1. So, for 0 0 this is 1 ok; for 0 0 0 1 this is 0; for this is 1 this is 0; again for 0 1 0 0 for 0 1 0 0 it is 1; for this is 0 for this is 0; for this 1 ok.

Then, 1 sorry for then this one, 0 0 1 0 0 1 ok; so, why for 15 you see for 15 this is 0; for 14, this will be 1. So, after that if I just; that means, from the truth table, if I just make the; that means, logical expression. So, at that time this S 0 value will be if I make this as common. So, at that time it will be what? It will be nothing but A 0 bar ok. So that means, now I can write the expression, I have got the expression for S 0 that is equals to A 0 bar ok

So then, next if I just want to erase this or if I start with a new page; So, for now what I have to do again I have to find out the next expression; that means, what will be the expression for S 1 ok.

(Refer Slide Time: 11:01)



So, for S 1; then again if I draw the truth table sorry for the that means, the Karnaugh map. So, at that time at this is A 3 A 2 this is A 1 A 0. This is 0 0 0 1, 1 1 1 0, 0 0 0 1, 1 1 1 0 ok. So, if I just go back to the previous slide. So, for S 1 if you see for 0 and for 3 that is 0 and 0; for 1 and 2 that is 1 and 1 ok. So, for 0 this is 0; for this is 1, this is 1; for 2 this is 1, for this also 0 ok. Then again, if I just go back, then if you see for 4 what is the value? 0 here and for 5, this is 1; for 6, this is 1; for 7, this is 0. So, again you go back for 4 this is 0; for 5 this is 1; for 6 this is 1; for 7 this is 0 ok.

So, in this fashion if I just go at that time what will happen? If you just; that means, look at these. So, at that time what will happen? That means, now if you just go back and then again if I just; that means, it will be filled like, something like this. So, if I just make this pair sorry this will be 1 sorry. So, if I just make this pair at that time S 1 expression can be written as A 1 bar A 0 plus A 1 A 0 bar. So, this is nothing but A 1 XOR with A 0.

So now, that means the corresponding expression for S 1 that is A 1 XOR A 0. So, this is the; that means, truth table from the truth table, I draw this Karnaugh map and from the from this Karnaugh map now I got the expression for S 1 ok. So, now, again what I have

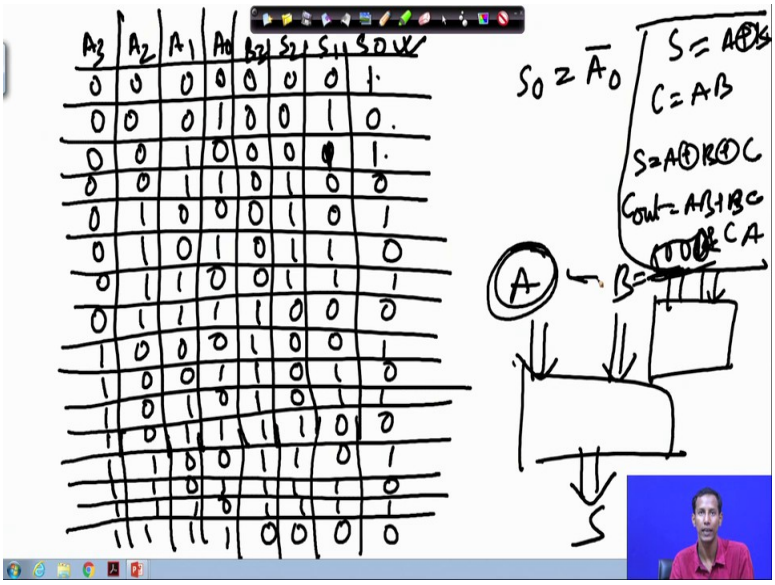
to do? I have to find out the expression for S 2 also. So, what will be the for S 2? If I just draw again the Karnaugh map, A 3 A 2 A 1 A 0, this is 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 ok.

Now if I just go back. So, at that time if you see for 0 0 0 1 and 1 0, for this 3 this is 1. Sorry, for this 3 this is 0; for 3 this S 2 value is 1; for 4 this value is 1; for 5 this value is 1; for 6 this value is 1 ok; again for 7 this is 0; for 8 this is 0; for 9 this will be 0. So, that means, up to 7 sorry up to this 7 value whatever I got that will be again repeated right. So, if I just then again if I just go to the next slide, for 0 0 0 this is 0; for this 0 for this 0; for only for this is 1 ok.

Then again, you go back then for 4 5 and 6; for 4 5 and 6 that is 1 for 7 again that is 0 for 4 5 and 6 that is 1 for this 0 ok. Then again, for this, this will be ok. So, now, if you just make the that means, pair of this 4 and then again this 1 and this 1 ok. So, now, what I can write for this particular pair? S 2 value will be what will be the S 2 value here? This A 2 and A 1 bar ok. For this particular case plus it will be A 1 A 0 plus this will be sorry not this will be A 1 A 0. So, here this is A 1 A 0 and for this that is A 2 bar and for this 2 particular case, it will be A 2 and then A 1 A 0 bar ok.

So, now what I can write that is A 2 A 1 bar plus if I just freshly if I just write the corresponding expression, A 0 bar ok. Now again, if I just make this things common here; then what I can write? A 2 A 1 plus A 1 into A 2 bar plus A 0 plus A 2 into A 0 bar ok. This is A 2 A 1 plus A 1 into A 2 XOR A 0 ok. But this is the corresponding expression for if I just go back to the previous slide; this is the corresponding expression for this particular S 2 ok.

(Refer Slide Time: 18:46)



Now, instead of that what I can see from the previous slide here if you see if I just take one; that means, you just this is the that means, method which you have to follow whenever you are designing the particular circuit ok. Now, instead of doing that there is another way of doing that, what is that? For that what I have to do?

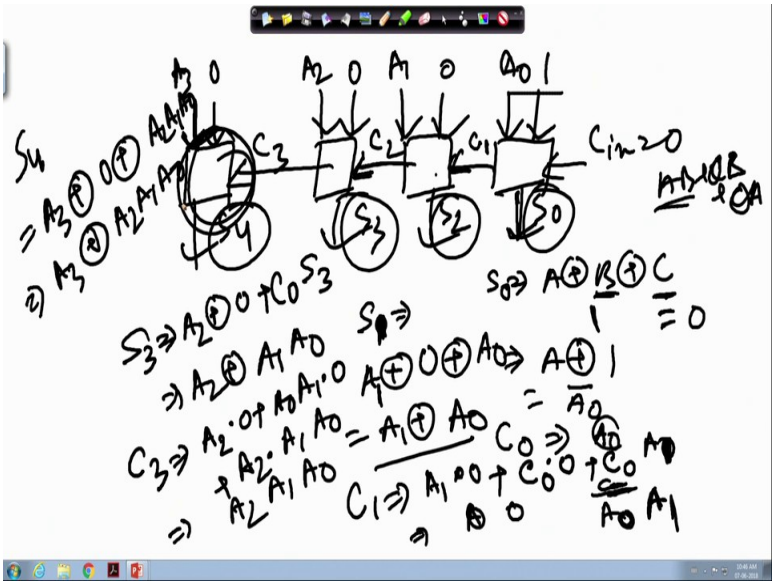
Again, I have to draw the whole table that is A 3 A 2 A 1 A 0 here. This is S 3 S 2 S 1 and S 0 ok; so, now 2 3 4 5 6 7 8 9 10 11 12 13 14 15 and 16. So, 0 0 0 0 0 0 0 1; so, again I am just writing the corresponding truth table, for this particular increment circuit; why I am just coming later. First let me draw this once, then I am explaining ok. Then, this one is 1 1 1 0 and 1 1 1 ok. So, for this is 0 0 0 0 1 0 0 1 0 0 0 0 sorry this is 1 1; for 3 it will be 4; for 4 it will be 5; for 5 it will be 6; for 6 it will be 7; for 7 it will be 8; for 8 it will be 9; for 9 it will be 10; then 11, 12, 13, 14, 15 and then 16 ok.

So, from this particular table visually can I that means, draw any relation for this in that means, output changes with the input. Now if you see, but first target to this S 0 ok. So, S 0 if you follow, that means, whenever this is 0 this is 1; this is 1 this is 0; this is 0 this is 1; this is 1 this is 0. That means, at that time by seeing this table I can write that this S 0 is A 0 bar. Why I can write that? Basically what I am doing here? I am incrementing it by 1 right. So, if you just think about adder; that means, architecture at that time how it works adder architecture? That means, for sum it is A XOR; sorry A XOR B and for carry that is A and B.

If I that means, chose this as 3 input at that time it will be A XOR B XOR C and this carry out will be AB plus BC plus CA ok. So, this is the that means, corresponding logic for full adder circuit. Now here, what I am doing? I know that one of the particular value that is fixed. So that means, whenever I am just if I just draw 1 adder circuit. So, at the time if this is A and this is B. Suppose this is A; this is B and I need to produce the sum. So, here B is fixed to 0 0 0 1; so that means, this one will only affect this changes or whatever the input of A; rest of the B as this is 0. So, it will not affect.

So that means, this one will be propagated to the MSB side whenever I whenever I have designed this particular adder. So, at that time it will be if I just draw this particular circuit using this full adder.

(Refer Slide Time: 24:00)



So, at that time what will happen? Ok. So, C in equals to 0; here this is A 3 A 2 A 1 A 0. But here, what is happening? This is 0; this is 0; this is 0; this is 1 ok. So that means, now for the sum, what will be the expression for this? For sum this is the expression that A XOR B XOR C. So, here B is 0 and sorry B is 1 and C is 0; so that means, if I just make 1 XOR with A. So, at that time it will be just A 0 bar, for this S 0 ok.

Then, next what is this? Here, it will it can produce one carry there is C 1 if I write. So, at that time in the corresponding and what will be the carry out here; what will be the carry out here? There will be A0; that means that is A B plus B C plus C A ok. So, C is 0 in both the case. So, only the AB so; that means, this is A 0 B is 1 C 0 is A 0. So, here

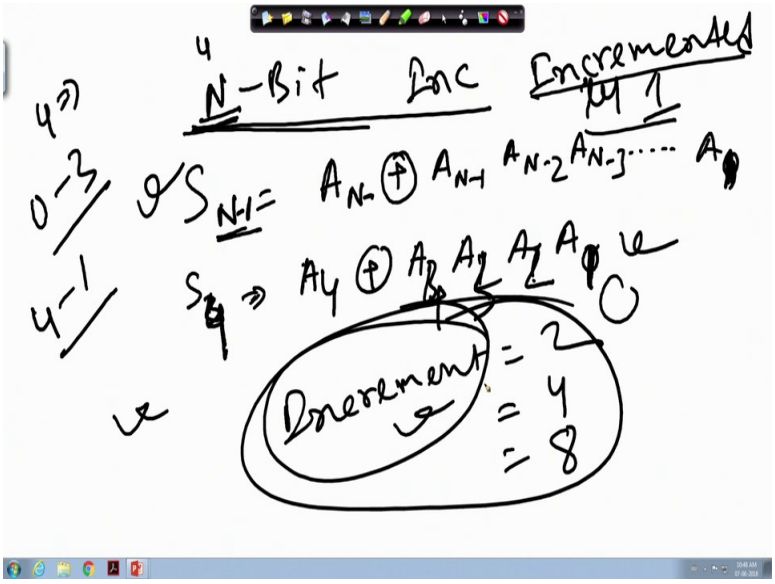
what will be the sum? That is A XOR here A 1 this is A 1 XOR with 0 XOR with A 0 ok.

So, this is now if I just got the expression for S 0, this is A 1 XOR A 0 right. Then again, sorry this is S 1. So, now, for C 1 what will be the expression? If I just write AB plus BC plus CA again. So, here what is AB? A 1 into A 1 into 0 plus what is this? This is C 0 into B means 0 dot plus what is this? C 0 into A 0 sorry A 1 ok. So, now, this is 0 this is 0 and what is this? C 0 is A 0 bar and this is A 1 correct. Sorry not this one is. So, C 0 is A 0. So, this is not bar this is A 0 X A 1.

So, now whenever I try to find out this S 3; so, what will be the S 3? S 3 value is now if I consider at the time this is A 2 XOR with 0 plus this is C 0. So that means, now A 2 will be XOR with A 1 A 0 ok. Now what will be the C 3 value? C 3 value is again A 2 into 0 plus A 0 A 1 into 0 plus A 2 into C 0; that means, this A into C that is A 1 and A 0. So, then C 3 value is A 2 A 1 A 0 correct. So, now, if I want to find out this S 4 value; so then, this S 4 value will be again this will be A 3 XOR with 0 XOR with this; this is C 3 this is C 2; this is C 1 ok. So, then along with this means A 2 A 1 A 0. So, if I write this will be A 3 XOR, A 2 A 1 A 0.

So, for 4-bit now, I can got the expression for each of this corresponding sum values. Now if I want to make it a generic case if you follow the trend; that means, this carry is basically propagating and it is XOR with the corresponding bit which is the input of that particular adder structure or this adder circuit right. So, if I just insist of 4-bit if I just want to make it a generic case that means, for N bit, what will be the expression for sum?

(Refer Slide Time: 29:28)



So, if I just consider the that means, for N-bit for N-bit incrementer what will be the expression? The sum expression will be I can write that as A N XOR with A N minus 1 A N minus 2 A N minus 3 dot dot dot up to A 0. So, now if you just write that for N-bit if you put any values let us consider this is 4-bit. So, if your this will be N minus 1. So, if you this is this S 3 value is sorry this is if this is S 4. So, at that time this will be A 4 not this will be 1, A 0 sorry. This will be 4 3 2 1 as I am not considering 0 here if I consider 0.

So, at that time this I can write, so that means, for 4-bit I am considering 0 to 3 write. So, that is why if I just make it N minus 1. So, at that time this will be this; that means, if I consider N as 4. So, at that time it will be 4. So that means, the value for 4-bit value will be 4 to 1. So, that is why this will be 1 this will be 4 3 2 1 correct. So, for N-bit this is it. So, this is N bit incrementer incremented by 1. In the next class, we will see that in the next class we will see if I want to change these value, I want a circuit this increment by 2 or increment by 4 or increment by 8.

Then, at that time from this particular circuit can I design or can I draw the conclusion that what will be the change; where will be the change or how can I design the corresponding circuit for this particular this incrementer circuit. And we will see the circuit of this particular 4-bit incrementer in the next class.

Thank you.