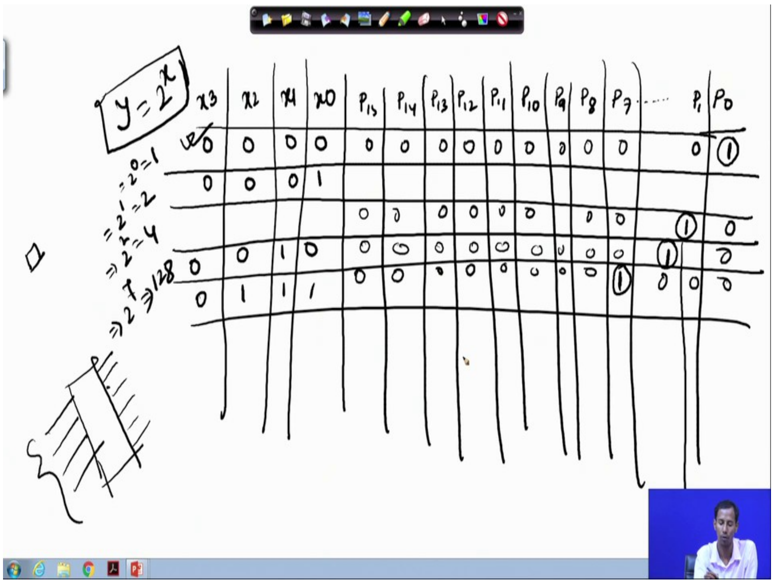
Architectural Design of Digital Integrated Circuits
Prof. Indranil Hatai
School of VLSI Technology
Indian Institute of Engineering Science and Technology, Shibpur, Howrah

Lecture – 12
Algorithm to Efficient Architecture Mapping (Contd.)

Welcome back to the course on Architectural Design of IC's. So, in the last class we have seen that means, architectural design of GCD and then again today, we will see another function which I want to implement that is and then again we have seen the architecture of how we can that means, implement x into 2 to the power of n. So, today again we will see another of a function if I want to implement that is y equals to 2 to the power x; how can I design that ok?

(Refer Slide Time: 00:58)



So, whenever we will just start with this what implement; what function I want to implement that is y into y equals to 2 to the power x. So, these are why I need to learn this basic that means, what is the architecture of this kind of function because this has a basic function which very much required whenever, you are building your final circuit; that means, suppose you are a building one FFT operation or one DCT operation function or one this DST of function.

So, at that time these are the basic operators which you required to implement the final expression or final function to be which to be implemented ok. So, that is why the optimization, it is something like this what happened that means, it is like this what? Suppose, you are building a home and home is basically build by brick by brick right. So, if you can reduce the cost of each of the brick; whenever you are building the home; obviously, you will save more amount of money. So, here also we are trying to do that thing that means, if this building blocks or this operator level optimization if we do whenever you implement the final function which can use 1000 times of this particular operators in the for final implementation of those function.

So at that time I can get more on of the benefit in terms of whether that is in terms of its power or in terms of area. So, that means that is why we are targeting to the operator level optimization and obviously, there are algorithm optimization and in the high levels of abstraction if you just follow. So, at that time the each of the levels has been its optimization technique. But here architectural optimization, we are trying to find out at the basic operators; obviously, we will see some of the that means, functional level or optimization how we can do design or what is the architectural changes we do to get the optimized circuit or efficient circuit for those function to be implemented that you will see in the later of the of this course.

But here at this particular point, where we have been started with is operater level optimization. So, here another example, that means, function which I want to implement this y equals to 2 to the power of x ok. So, how to find out this with that means, what will be the circuit for this y equals to 2 to the power x? So, if I just draw that means, circuit that means, the truth table of this. So, at that time if I consider it 4 bit of x; that means, x is of x is of that means, 4 bit.

So, that means, now x 3 x 2 x 1 and x 0 ok. So, for this 4 y will be of what? That means, it will be of 16 bit for considering this it will be of 16 bit ok. So, let us consider P 15 P 14 P 13 P 12 sorry P 11 P 10 P 9 P 8 P 7 dot dot dot up to P 1 and P 0. So, if I just draw it something like this. So, at that time what will happen? For all zeros of x; so this means what? 2 to the power 0 that is equals to 1.

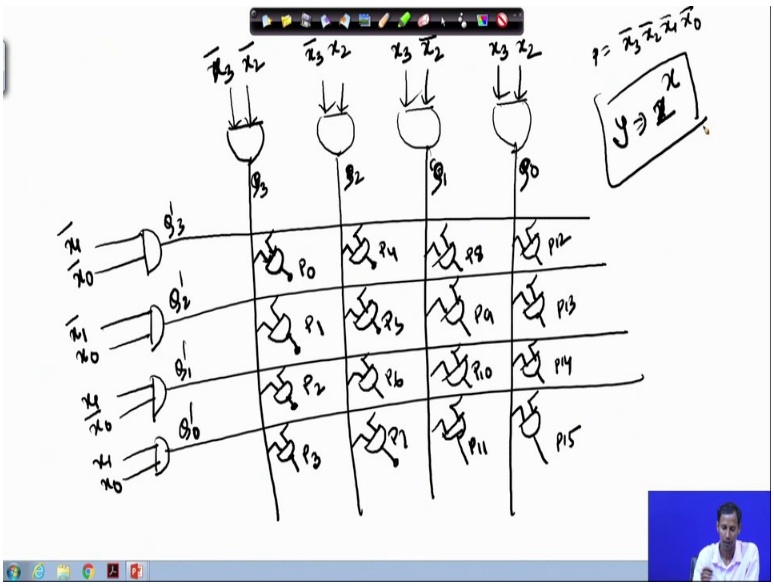
So, that means at that time only this P 0 will be 1 and rest of the all bits will be 0 ok. For x equals to 1; for x equals to 1 y equals to what? 2 to the power 1 means 2; that means, only this P 1 position is 1 and rest of the positions are all zeros. Then again, if I just come to 0 0 1 0; so that means, here for x equals to 2. So, this is 4. So, that means, only P

3 position will be 1 and all other positions will be all other position will be filled with zeros only. So, the same thing if I consider that x value is as that means, 0 1 1 for 7 ok. So, 7 means 2 to the power 7 that is equals to 128. So, 128 means this P 7 will be 1 and rest all will be 0; so that means, if I just draw a connection from this if you just see at this particular position changes of x this only that means, changes.

So, if you just make a connection with the corresponding digital circuit, if you see that this is nothing but a decoder circuit ok. So, in decoder circuit what happens? So, whenever that means, for this particular bit position, we basically decode and that gives me the corresponding 1 value right. So, if I just that means, from this particular decoder circuit means what?

So, I am having these 4 input and I will have the corresponding 16 output over here. The same circuit I am getting for y equals to that is nothing but the function of y equals to 2 to the power x. From 2 triple, we can conclude that particular thing. Now what will be the that means, circuit for this if I want to draw the corresponding decoder circuit from this particular 2 truth table; how it will look like for that reason what I have to do?

(Refer Slide Time: 07:57)



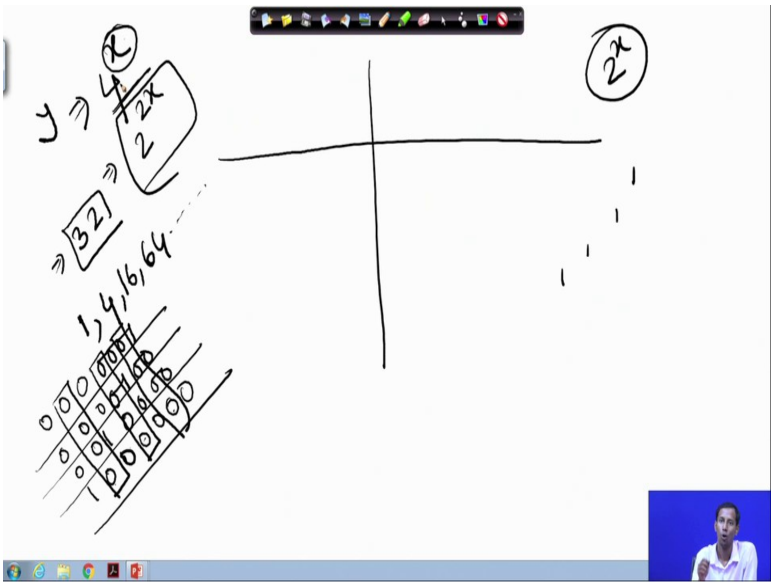
I need the AND gate. So, here also the same thing at this particular position ok. So, this sorry this is x 3 x 2; this is bar, this is bar; this is x 1 and x 0 bar. So, then this is x 3 bar x 2; this is x 3 x 2 bar and this is x 3 x 2 ok. So, that means this is for 0 0; this is for 0 1; this is for 1 0; this is for 1 1 and this is x 1 bar x 0 and then x 1 x 0 bar and this is x 1 x 0 ok. So, now if you just make the lines, draw the lines from this particular point. So, if I am having 1 AND gate over here which is connected with this.

So, what is the values of this? Here what is that means, this and this if it is AND gate and then again, this 2 particular results if this is P 3, this is P 2, this is P 1; this is 0 sorry not again this is Q. I consider this as Q; Q. This is let us consider Q 3 dashed Q 2 dashed Q 1 dashed and Q 0 dashed. So, here what I will get? I will get this is as P 0.

So, why P 0? P 0 is what nothing but x 3 bar x 2 bar x 1 bar and x 0 bar ok. So, if you just go back to that. So, for P 0 only when all are 0; so at that time P 0 is 1 and rest are 0. So, here for this only this is P 0 ok; for this if I just connect to this. So, then this is P 1 ok; then again, if I just this is P 2 this is P 3. So, you see here this is for 0 0 and this is for 1 1. So, this is the case of P 3 ok.

So, actually that means, for each of the case now, you can find out using the same thing. So, this is for P 4 this is P 5 this is P 6 this is P 7 and this is for the other case this is for P 8 P 9 P 10 P 11; then, this is for P 12, this is for P 13, this is for P 14 and this is for P 15; that means, now if I just draw this particular that means, if you implement this particular function, I can get via the function implementation of y equals to 2 to the power of x ok. So, this is the circuit for implementing 2 to the power x.

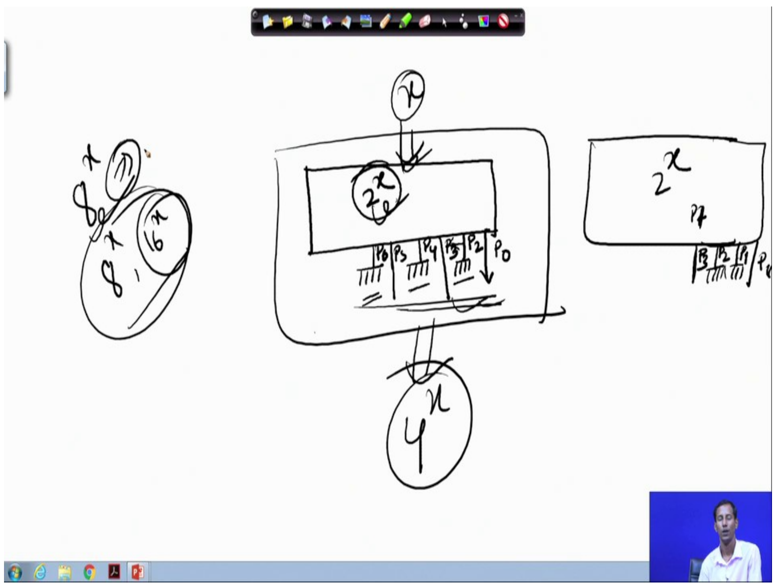
(Refer Slide Time: 13:02)



Now next, if I want that means, function which is 4 to the power x or 2 of 2 to the power x; so at that time how can I do that? So, that means 4 to the power x means at that time what will be the output size of 4 bit, the size will be of 32 bit. So that means, where will be the whenever I will do that means, this. So, 4 to the power x means it will changes from oh that means, how it will change whenever this is 0; that means, this is 1; then 4; then what? Then, 16; then, for 3, 64 something like this; this will change. See if you just see that means, for 0 0 1 for 4 what is that for 16 for 64 something like this it is changing all other bit positions are 0.

So, again that means, from the 2 to the power x function 2 to the power x function we have derived. So, what was the logic there? It is changing 1, then 1, then 1, then 1 something like this; but here, if you see that this positions is 0, this positions is 0, this position is 0; that means, the alternate positions are filled with all zeros for 4 to the power x. So, that means now from the 2 to the power x circuit, I can easily implement 4 to the power x function by putting something like this.

(Refer Slide Time: 15:23)



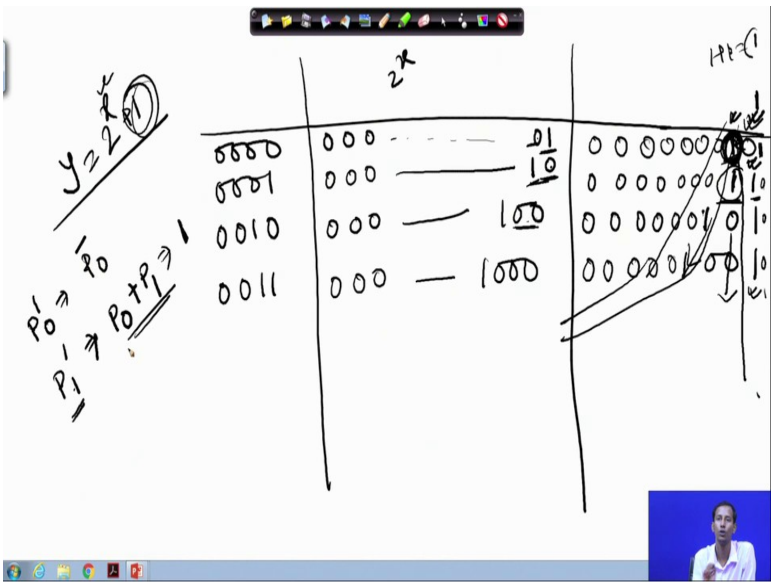
Suppose this is the 2 to the power x function which I have already implemented earlier. So, here what I need is; so that means, initially it was P 0. Now this will became P 2, this P 3, this is P 4, this is P 5, this is P 6 something like this; that means, all the even position that are filled with zeros of the 2 to the power x function and this will give me the whole of this circuit is nothing but the 4 to the power x ok.

So, this is the that means, circuit which I can implement using the that means, using 4 to the power that means, 2 to the power x I can implement 4 to the power x too. The same thing in if I want to implement 8 to the power x at that time how can I design? That means, at that time 8 to the power x means 2 corresponding bit position will be filled with 0's; that means, alternate 2 bit 2 to the power x function. So, at that time what will happen? 4 to the power x from this particular circuit, how can I implement?

So, 2 to the power x; so then, P 0 is P 0. Then, 2 bit positions are filled with 0; then this will be this its P 2 will be connected to this here P 1 sorry this is if this is P 0, this is P 1, this is P 2 this will be P 3 here and it is P 1 will be P 3 ok. So, that means any value of 8 x or 16 x or any values can be implemented using this 2 to the power x; but hardware wise there is no changes. Once you just this using this gate level, you have implemented the 2 to the power x, then you can easily implement the other function of this ok.

So, this is for like the implementation technique for 2 to the power x, then 4 to the power x, then any of this ok. Then using this 2 to the power x as the base function, what are the others function we can implement that we will see in the next that means, in the next.

(Refer Slide Time: 18:42)



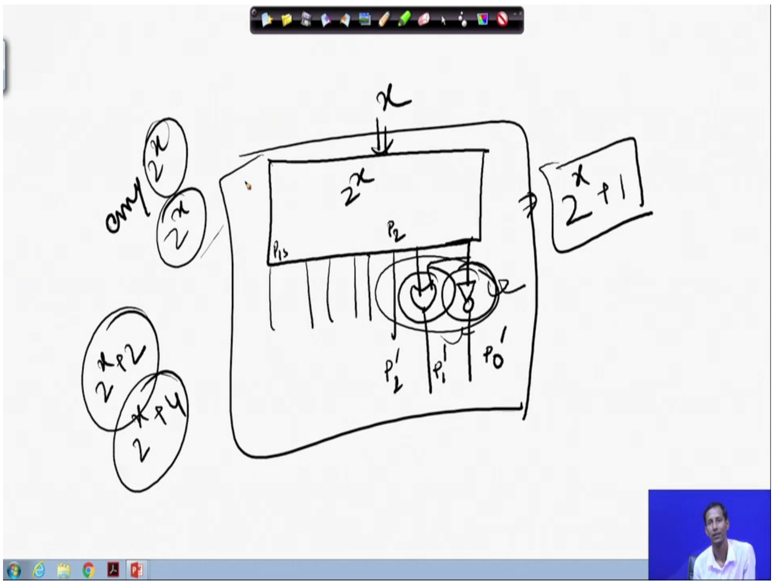
So, now, next you consider y equals to 2 to the power x plus 1; suppose this function I want to implement. So, at that time what will be the changes? Ok. So that means, at that time if I just want to add y equals to 2 to the power x plus 1; so at that time means what for 2 to the power x for this for 0 0 0 0, it was all 1 ok. This is 2 to the power x, then I want to add plus 1 ok. So that means, this will now the this modified will be come and rest of all bit will be 0; then for 0 0 0 1, this is 0 and 1 and what it will be? It will be this ok. Then for 4 it is for 2 this is 4. So, now, it will be 5 ok; then again for 3 it is 8; but here it will be 9 ok. So, you just see that for adding at the that means, adding of 1 to 2 to the power x, how it is basically effecting? It is effecting this last 2 bit position only; this rest of the bit positions are basically remains same, it only effects the last 2 bit position.

So, here what happen? It becomes just inversion of the P 0; that means, if you consider in the earlier case it was 1 and rest of the bit was 0 for this, but here what happened this has become 1, but rest of the all cases becomes 0. That means, from this particular logic, if I just then, this P 0 modified or P 0 of this particular circuit that will be what? If this is a P 0 from 2 to the power x and where are the changes are there; that means, here initially this was 1 sorry 0 this was one; but here now it becomes 1 because of what here for 1 plus 1 that carry has been generated and that has forwarded to this particular equation ok.

So, that means, now to do that what will be the P 1 bar? What will be the P 1 bar now ok? So, that we is now P 1 bar is basically dependent on this corresponding P 1 and P 0 value. So, P 1 and P 0 value is that means, initially this position was 1 right and the in the modified because of this that means, addition of 1, now it becomes 1 initially for 2 to the power x this was 0 and this was 1. But here it is as this is 1. So, it is 1 only. Only for this particular bit position I need to change; that means, for this P 0 for P 0 and this, this will be changed. So, how this will be changed? If I just do it that P 0 plus P 1 that is equals to modified of P 1, then that will be the case for this; how for this is this was 0 plus 1.

So, in the modified case P 1 will be 1; for this what was the value 0 and 1. In this case this is 1, this is 1; in the other case what it is? It was 0 plus 0 so that means, in all other cases it will be 0 and for rest of the case this carries not effected on the third bit; so that means, it will remains same. Only in the last 2 bit position the 1 will effect and that will be the change which I need. So, then what will be the that means, corresponding architecture for this?

(Refer Slide Time: 24:13)



So, if you just draw the architecture of this, then this and you require 1; this is the P 0 modified this is P 1, then rest are just same as this is the 2 to the power x function this is x. So, this circuit is this circuit is basically nothing but your 2 to the power x plus 1; So that means, only in this two particular bit position I am having the change to make it to 2 to the power x plus 1.

So, now if I just want to add this 2 to the power x plus 2, then what will happen? 2 to the power x plus 4, then what will happen? If I just want to add any values with 2 to the power x; So, at that time how it will effect and what will be the change or how easily or how efficiently we can implement that circuit, we will see along with many other new circuit using this 2 to the power x as the base function, how we can implement that we will see in the next few classes.

Thank you for today.