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

Lecture – 14
Algorithm to Efficient Architecture Mapping (Contd.)

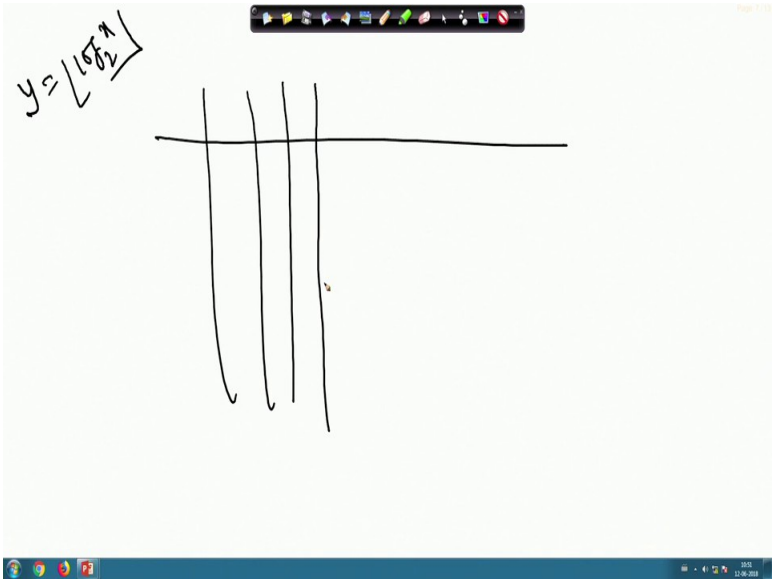
Hello everyone, welcome back to the course on Architecture Design of ICS. So, we are saying different architecture efficient architecture from different algorithm ok. So, in considering or in corresponds to considering design aspects of VLSI design aspects of speed power and area ok. So, in the last class we have seen one function, which I need to implement that is y equals to ceiling of log 2 base x ok.

So, that I want to design, or that I want to implement using the circuit, or that circuit I need to design so, at the time how we can do we have seen that we have we have done, what we have just drawn the operation table and from that operation table. Now, our job will be we have to find out each of the logical expression for each of this bit output bit and, then from that particular after finding out the logical expression for each of this output bits.

Then we will just convert it to the, or we will draw the architecture of that particular circuit ok, or we will get the hardware for that particular circuit ok. So, we have drawn the, that means, this operation table ok. So, you have got how this inputs how the outputs are basically effected by the combination of input that, we have seen in that occupation table that, we have already drawn. So, from that we have concluded that this is nothing, but a priority encoder circuit.

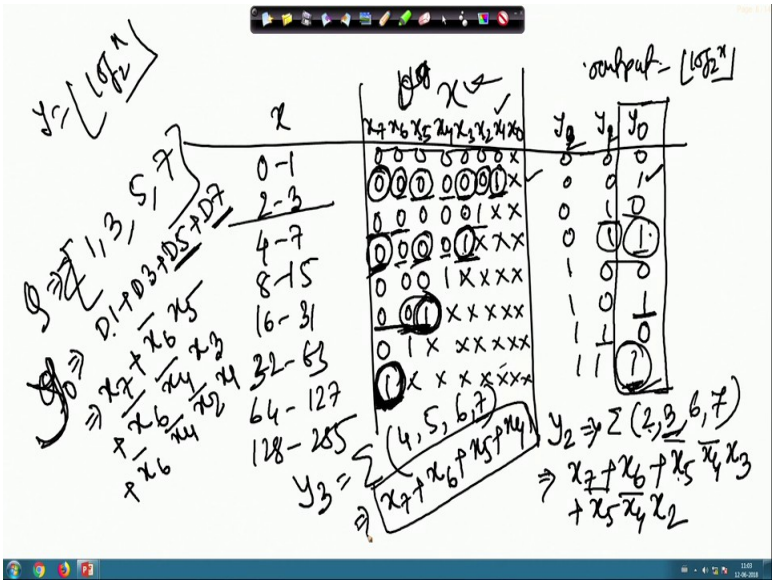
So, now we will try to design one priority encoder circuit which will give me the function of y equals to ceil of log 2 base x ok. So, then again go back to the, that means, corresponding circuit ok.

(Refer Slide Time: 02:23)



So, if I just draw again if I just; that means, quickly draw that particular point ok. So, what I need to do that is y equals to log 2 x ok.

(Refer Slide Time: 02:41)



So, now, what we have seen sorry y equals to log 2 base x that I need to implement so, what we have seen that for x of 0 to 1, there is a range for 2 to 3, then 4 to 7, then 8 to 15 16 to 31 and 32 to 63, 64 to 177, 128 to 255 ok, 1, 2, 3, 4, 5, 6, 7, 8 ok. And, for that what will be the that means, this value or this sorry this is x, if I just write it in another terms, if this is x 7, x 6, x 5, x 4, x 3, x 2, x 1 and x 0.

So, initially it will be something like this and for that the output which is nothing, but implementing this particular function so, as I am considering 8 bit so, that is why I can using this particular sorry y 2, y 1 and y 0 using 3 bit I can implement. So, and then if I just write it in them as we have seen that that the range is 2 2 3; that means, if this bit position is 1 and whatever is the value of this x 0 it does not matter the value will be something like 0 0 1 ok. So, that we have already seen let me draw it again.

So, just quickly I am doing that ok. So, then for 4 this is for 5, this is this is 6 and this is for 7 ok. So, this is my table so, now from this particular point, now what will be the logical expression for this y 0. So, how we basically actually what is the general method for finding out this y 0, I have to draw I or I have to put it to the Karnaugh map and from that I can find out the logical expression. But considering 8 bit variables on that using this Karnaugh map it is became too much complex ok. So, we have to find out another method, or we have to follow another method for calculating the logical expression for that.

But here what we can do, we have seen we can see that for 1 this then, this is for 1, then this is sorry this is for 2, then this is for if I just consider let us me consider and for Q 0 that will be for 1 ok, then this is 1 for which 3, then this is 1 at which for bit position, then 5 and this is 1 for 7 ok, for this particular bit position this Q 0 value is 1 so; that means, if I just write that means, at this particular point at this particular point, for this for this for this and for this y 0 bit is 1.

So, now I have to find out the corresponding expression for this ok. So, how can I find out the corresponding expression for this so, initially at I as see , I told you that this is priority encoder circuit. So, this when this bit position is basically this means, what if I just write it as the a that means, term as D 1 plus D 3 plus D 5 plus D 7 this these are these are the; that means, corresponding expression of this particular bit position. So, if I just write that as a function that D 1 plus D 3 plus D 5 plus D 7 that is the Q 0 ok.

So, if you just see whenever this bit position is 1 no maters what is whatever the value of others, it does not depend on them, it depending only this D 7 sorry x 7 bit position, it basically gives you the value of 1 at Q that means, y 0 sorry this is y 0 so; that means, now for these 7 what I can write, I can write that as a x 7 ok. So, I am trying to find out

each of the expression for this D 1, D 3, D 5 and D 7. So, D 7 is basically nothing, but your x 7, because this is all of these are do not care bit position.

So, only this bit is 1 so; obviously this will be 1, then for D 5, D 5 means this. So, here you see I am having the that means, the be that means, corresponding 1 at D 5 that means, sorry for x 5 what would be these two particular bit position so, at the time these two particular bit position that is 0 and 0, but here you see I have already as this is priority encoder so, I have already got one at this particular bit position so; that means, for this two the expression for D 5, it will depend only this two particular bit position and, it will be something like x 6 bar x 5 ok.

Then for 3 if I just go to or if you just go to this particular bit position. So, at the time you see this is already became mean; that means, for this expression I have already consider this particular bit position, or I have already got this 1 here and I have already got this 1 here. So, for this particular combination what will be the that means case, or what will be the expression for that it will depend on the corresponding bit of x 6 bar, then x 4 bar and x 3 so; that means, as I have already got 1 over here and, this 1 for this 5. So, I do not need to consider this 2 bit at for considering this, why because this is priority encoder so, if this equation is 1 already that will effect this ok.

Then for 1 I have got 1 over here, I have got 1 over here, I have got 1 over here ok. So, then for this only I have to consider this 4 I have to consider. So, for D 1 now I can write the expression as x6 bar, bar x 2 bar and x 1 ok. So, this is the final expression of y 0, by the same method by the same method, if I want to find this y 2 so, y 2 means what y 2 is basically for this is for 2, y 2 is basically for 2 and then 3 and then this is 6 and then 7 ok. So, using the same method for 7, it is x 7 for 6 means, what for 6 means what I have already got this.

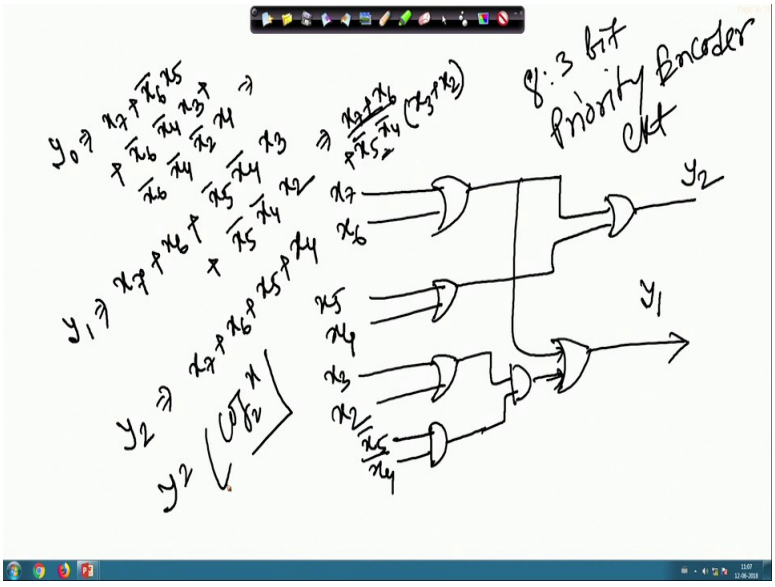
So, here what we are getting 6 so, 6 means this already I got so, only this x 6 plus 3 means what for 3, what is the bit position so, for 3 is this so, this 2 bit position I have already got 1 so, this will be x 5 bar, x 4 bar, x 3 sorry this for this 3 so, this is I already got so, this and this so, for 3 it will be x 5 bar, x 4 bar x 3. And for 2 for 2, what I got I have got 1 over here, 1 over here, and 1 over here.

So, rest of the things I need to consider which is x 5 bar, x 4 bar and x 5 bar x 4 bar and I have got 1 over here and then this is x 2 so; that means, x 7, x 6 and 3 I got only what I

left 5 4 and 2 ok. So, that is the expression for y 2. So, then again what will be the expression for y 3 then so, for y 3 the expression will be this is 4, 5, 6 and 7..

So, for 7 this is x 7 for 6 this is already I got 1 over here so this is x 6 so, for 5 this 2 is already I got 1 over here so, for this is x 5 for 4, 7, 6 and 5 both I got 1 so all 3 I got 1 so for 4 this is only x 4. So, this is the expression for y 3.

(Refer Slide Time: 14:27)



So, now that means for y 0 the expression is something like this x 7, then for 5 the that is x 6, bar x 5, then for x 3, x 3 is x 6 bar x 4 bar, x 3 plus for 1 it will be x 6 bar, x 4 bar, then x 2 bar x 1 ok, for y 1 what was the case for y 1 it was 2, 3, 6, 7.

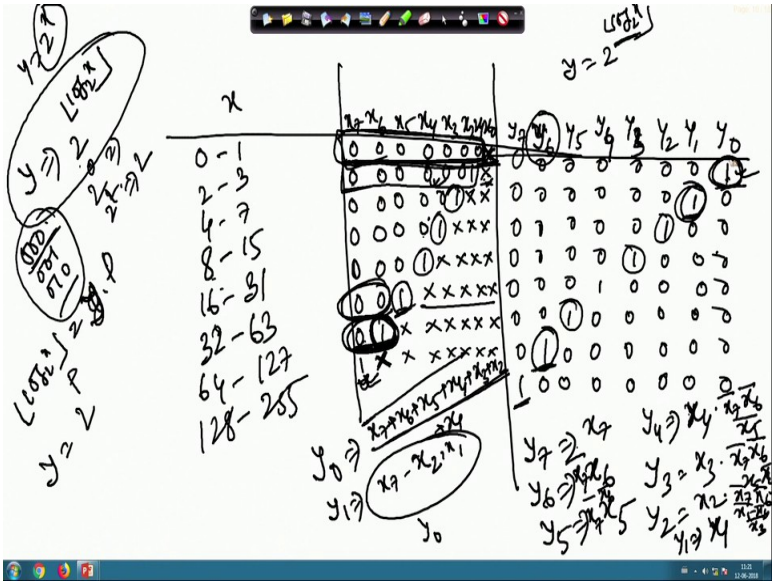
So that means, for 7 it was this for x 6 it was x 6 for 3 x 5 x 4 bar x 3 plus x 5 bar x 4 bar x 2 and for y 2 sorry for y 2, it was x 7 plus x 6 plus x 5 plus x 4 ok. So, now, if I just write this expression as x 7 plus x 6 plus x 5 bar x 4 bar x 3 plus x 2 ok, these also I can just minimize and I can just write and, then for that what I need, I need this OR gate and AND gate. So, in combination of that if I just that means, if I just write or draw particular only 1 that means, this is x 7 and x 6 ok.

So, then again I need 1 OR gate and, here if I just like draw this x 5 and x 4, then x 3 and x 2 so, this will become this, then x 5 bar and I need 1 AND gate, where this is x 5 bar x 4 bar ok. So, this will be then ANDed with this x 3 plus x 2 and, then that will be added

with this particular point so, this is y 2 this is y 1 and for again y 0 I have to just minimize this and used by this method using this gates.

Now, I can implement that 8 is to 3 bit, priority encoder circuit, which will give me nothing but that is y equals to log 2 x so; that means, now you can implement this circuit which will from where you can get this log 2 base x.

(Refer Slide Time: 18:16)



Now next from this particular point let us consider another circuit suppose I need one circuit, which is 2 to the power of this log 2 base x, so, at the time how I actually what will be the circuit for that so, for that again the same thing; that means, for x the range will be same ok, it will be 4 to 7 something like this, then 8 to 15, then 16 to 31, then 32 to 63, then 64 to 127 and 128 to 255 ok.

Then for this x that can again write as x 7, x 6, x 5, x 4, x 3, x 2, x 1 and x 0. So, that again can be written as something like this, in the earlier case what we have seen, the same thing ok. So, where will be the change if I consider this at the time where will be the change, that we will just want to see or that we have to see ok.

So, if I just sorry this is 0 and 0 1, then something like this ok, then this y equals to 2, this log 2 base x that means, if I consider only this at the time it will be give it will give you 3 bit that is 0 0 0, 0 0 1, then 0 1 0 so, 2 to the power of if I consider this as a suppose I am considering this x as let us consider this as a y ok.

So, now sorry if I consider there is a P, now y will be what 2 to the power of P. So, 2 to the power of P we have seen how we have seen y equals to 2 to the power x right that means nothing but a decoder circuit ok. So that means, now in the decoder circuit for 3 bit, how many combination I will get I will get 8 so; that means, now for this particular y y 7 sorry y 6, y 5, y 4, y 3, 3 y, y 4, y 3, y 2, y 1 and y 0.

For this it will be one over here for 2 sorry for what I am getting 0 0 0 for P or for log 2 base x so, 2 to the power 0 means I am getting 1 over here so, 2 to the power 2 1 means I am getting 2 that means, now it will be this something like this 2, then I will get 4, then I will get 8, then I will get what I will get 16, then I will get 32, then 64 and then this 128 ok.

So, 16, 32, 64, 128 sorry 256 1, 2, 4 sorry 1, 2, 4, 8, 16, 32, 64, 128 so, this is that table which I am getting. So, now what will be the expression for each of this y 0 after considering this, if you just consider or if you just want to map from this to this, then what it will be so, y 0 means here that is if that is 1 ok. So, at the time if I consider this so, at the time it means what, if all of this combinations is that means, all of this combinations are considering that all 0s so, at the time it will be 1.

So that means, no matter the values of this, whatever is the values of this all this bit positions, if it is considering for y 0 to b 1 ok. So, for y 1 for y 1 if I consider all of this particular bit positions so, then because this can vary so, I will discuss this particular bit position only for this particular bits, this will indicate 1 ok, because this will vary if I consider this, then I have to consider 0 and 1 for both the cases..

So, considering that whatever is value for this if I consider that other that means, from this particular is x0 excluding that x 0, if I consider the others bits combination of that particular point will give me the output of y 1 over here. So, y 0 means what when all this bits as 0, then it is 1 so that means, now that is what that will be just x 7 bar sorry, not bar of x 7, x 6, x 5 plus x 4 plus x3 plus x 2 plus x 1.

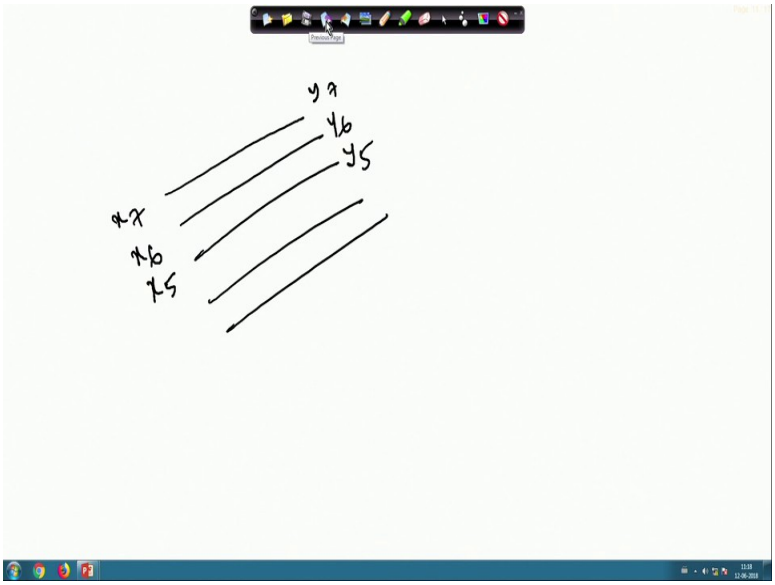
So that means, or of all these bits that you will get y 0 ok, for y 1 what will be the expression for y 1 for y 1 if all this bits, if all this bits are if all of this bits are 1 sorry 0 and along with this is 1, then only I will get y 1 over here so; that means, now I have to or I can write this corresponding expression in the same fashion ok. So that means, now for this particular bit position for this y 7 or x 7 2 x 6 sorry x 7 to x 2 and x 1 I have to

write this. So, what will be there that means corresponding function ok, let us consider for this for y 7, if x 7 is 1 so y 7 will be 1 ok.

For y 6 for y 6 this is 1 x is this 1 and for x 7 if this two combinations are same because in the previous case what was here; that means, I am considering the these bit position whenever I am getting that means, I am getting for multiple 1, whenever I am getting y 7 multiple one at the time I can exclude this particular point. But here I dont I am not getting multiple of 1 over for this y 7 or y 6, only for one particular position this is considering this so; that means, now whenever this x 6 is 1 so, I am getting 1 over here so; that means, now I can write that has x 6 ok.

So, now for y 7 sorry for y 5 no matter the values of others are so whenever I am getting this as at the time, I will get if this is 1 so; obviously, I will get 1 over here. So, in this fashion this is same as this same for this, same for this and only for this particular bit position, it will change which is something like this ok. So, for y 4 that is sorry for y 4 that is x 4 for y 3 that is x 3 for y 2 that is x 2 and for y 1 that is x 1 for y 0 it will this ok.

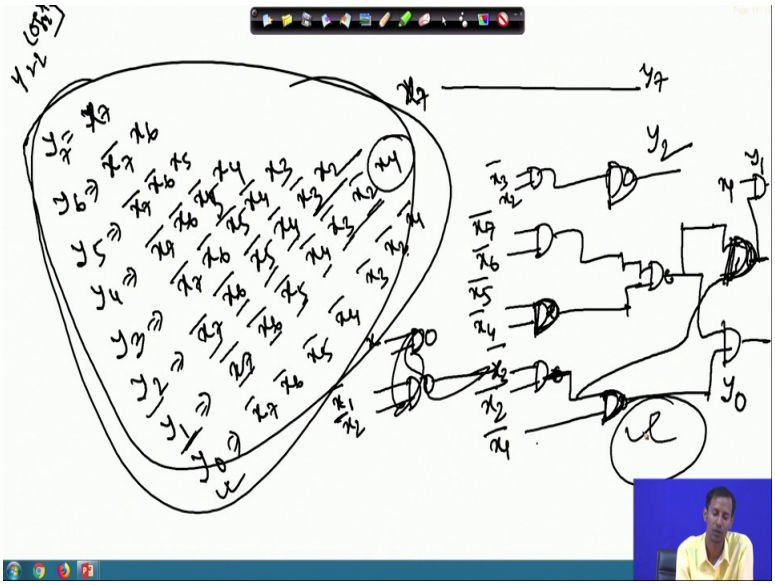
(Refer Slide Time: 28:48)



So, this can be the or if I just implement the logic that means connecting this x 5, sorry x 7 bit, as y 7 then x 6 to y 6, then x 5 to y 5 no sorry sorry sorry no actually we have to consider this, because for y 6 for 1 this can vary right as this position is go do not care so, at the time it will be different it will be some say on the same.

So, as I am getting 1 over here so that means, it will be x 7 bar into x 6 x 7 bar into x 6 ok. So, and for this it will be x 7 bar x 6 bar x 5 for y 4 it will be x 4 multiplied with x 7 bar x 6 bar and x 5 bar, for this it will be x 7 bar x 6 bar x 5 bar and x 4 bar ok. For x 2 it will be x 7 bar x 6 bar and x 5 bar x 4 bar and x 3 bar ok.

(Refer Slide Time: 30:23)



If I just write it in a different that means, for y 7.It will be x 7 for y 6 it will be z, it will it will just depend upon this 0 and 1 so that means, it will be sorry this is x 7 bar into x 6, for y 5 this is x 7 bar x 6 bar x 5 and for y 4 that is x 7 bar x 6 bar x 4 bar, sorry x 5 bar x 4 for y 3 that is x 7 bar x 6 bar x 5 bar x 4 bar and x 3, for y 2 that is x 7 bar x 6 bar x 5 bar x 4 bar and x 3 bar x 2, for y 1 that is x 7 bar x 6 bar x 5 bar x 4 bar x 3 bar x 2 bar x 1.

And for y 0 for y 0 if I consider so, the time what it will be that means, x 7 bar x 6 bar x 5 bar x 4 bar x 3 bar x 2 bar ok. So, when all this or if you sorry this one, if I just consider when all this bits are 0 0 0 0 so, at the time only it will I will get 1 for this. So, whenever I where I will get that means, when all of this bits is 0s ok. So, then I can get one for this particular point ok, or else I can just write in terms of OR and then invert of that the same logic.

Now that means, this is the corresponding expression for this y equals to 2 log 2 x ok. So, now using just AND gate that means, now y 7 sorry x 7 that will be directly connected to x 7, then we have to that means connect something this x 1 bar x 2 bar x 2

bar, then that will be connected with this. So, these now this basically these function, now I can you implement using this gate ok. So, far that x 1 and x 2 bar that I have so, then x 3 and sorry x 2 not this. So, if you just consider this x 3 and x 2 they are common.

So, x 3 and x 3 bar x 2 bar is common, that I can sorry this is and gate AND gate this is not NAND gate x 1 bar ok, then what I need else x 4 and x 5 so, x 4 bar x 5 bar then what I need x 6 and x 7 x 7 x 7 bar ok. So, then this will be this will be so; this is y 0 right ok. So, then for y 1 what I need, I need x 7, x 6 bar, then x 5 x 4 bar and then x 3 x 2 bar ok.

So, here what we are doing x 7 x 6 and from here, we will get another which is this. So, along with this x 3 x 2 then along with this I need to ANDed with x 1, then again this will be ANDed with another gate which is x 1 and from that particular point we will get y 1 over here ok, for this y 2 what I need x 7 x 6 x 5 x 4 so, x 7 x 6 x 5 and x 4, then what I need x 3 and x 2 so, x 3 and x 2 again I have to design, or I have to do this x 3 bar and x 2.

So, then again I can do or I can just implement this y 2 so, in this manner or in this for this particular that means, expression now using this AND gate only, I can implement it ok. So, in earlier case in priority encoder circuit ok, we are basically what we are doing, we are just finding out these expression for 3 bit position, but here for 2 to the power of log 2 base x, we are getting 8 number of expression over here.

So, 8 expression we have to find out and then for each of the logical expression, we have to converted to the corresponding gate level ok. So, thank you for today's class this is the, that means, end of architecture to sorry algorithm to architecture mapping. So, in the next class we will start with different things.

Thank you.