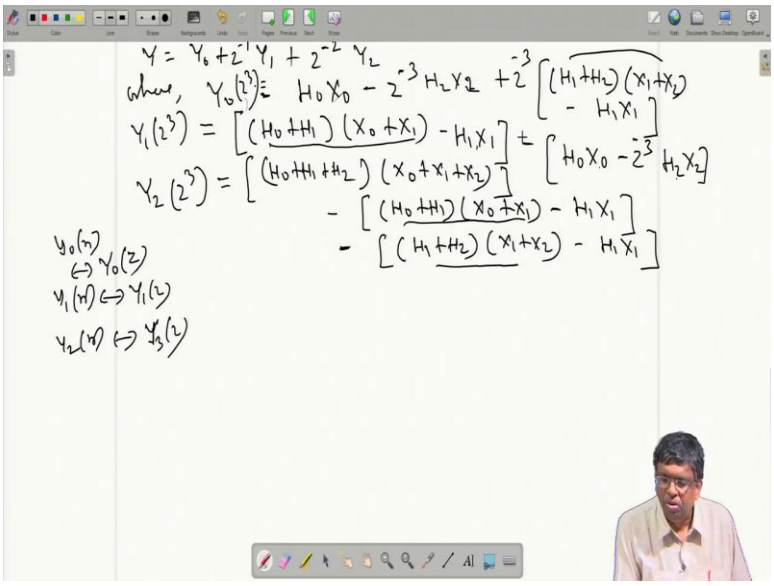
VLSI Signal Processing

Professor Mrityunjoy Chakraborty
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture 32

Introduction to First Level Architectures

(Refer Slide Time: 0:31)



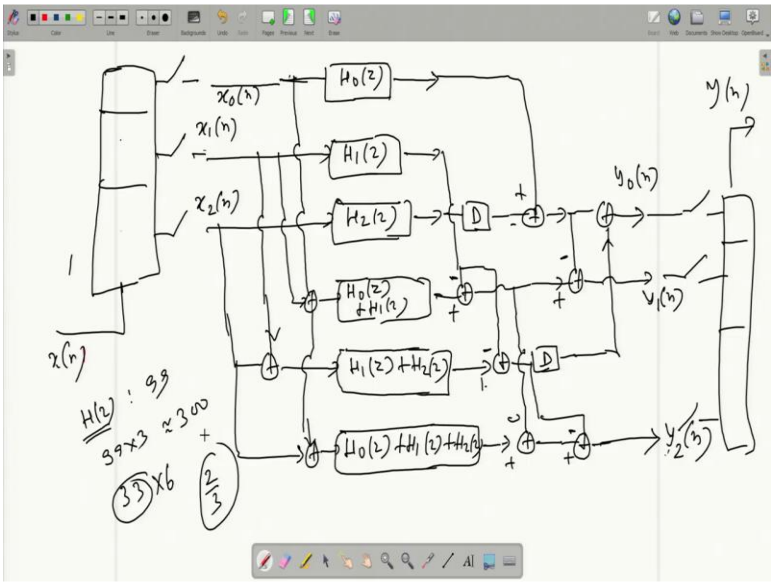
So, suppose X0 n is given to us. I do not think I can draw it the limited space here, so I have to move to the next page. Remember, what we have? X0 n passing through H0 n and since it is a function of instead of Z to the power of 3 I have brought it Z. So, all Z to the power 3 has turned out to be Z.

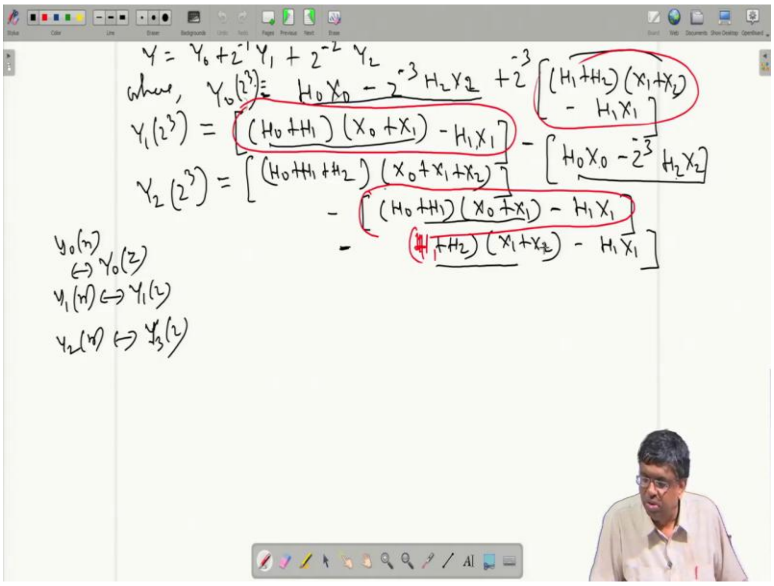
So, Z to the power minus 3 will become Z to the power minus 1. Because I am no longer considering it as a function of Z to the power 3. That is why, you know, Z to the power 3, I am not considering because that means between every pair of samples samples and then Y0 n, there will get two zeros, that I am not considering.

That is why, I am considering Y0 z, there is a polyphase factor 0th polyphase factor. Y0 z, there is no zeros between in every pair of samples here. Y0 z. As a matter of fact, wherever I have, Z to the power 3, like H0 Z to the power 3, now will be called, it will be written as H0 Z. H0 Z to the power 3, which have two zeroes between every pair of samples.

Here, but this sequence, now it will be X0 z, z to the power minus 3. Now, it will become Z to the power minus 1. Here, Z to the power minus 3 will become Z to the power minus 1, so on and so forth. Remember X0 H0, then H2 X2, another filter is you had X1 n and X2 n passing through H1 plus H2. Subtract from the filter output, where X1 is the input, X1 n. Filter is H1. Then, delayed by Z inverse, okay and likewise.

(Refer Slide Time: 2:31)





Or this, you have to implement H0 X0 minus Z to the power minus 3, that is Z to the power minus 1 now, H2 X2. H2 X2, okay. And there is a H1 X1 also used here, so let us first complete these filters. Now let us see, H0 X0 and then H2 X2, which is delayed, Z to the minus 1 and subtract it from this output. So, H2 X2 to be delayed.

So, H2 X2 here to be delayed. So, put a delay here and to be subtracted from here. So, that is what you get here. And now, X1 plus X2 to be filtered by H1 plus H2 and then H1 X1 to be subtracted, then whole thing is to be delayed by 1, Z inverse, it is not Z to the power minus 3 now, Z inverse so, another filter now, H1 plus H2. So, H1 plus H2 are filters. Similarly, H0 plus H1 are filter. Another is H0 plus H1 plus H2.

So, I have one filter here. Another, we keep across. And last one is, now X1 plus X2 to be filtered by H1 plus H2. So, X1 plus X2, let me see. X1 plus X2, H1 plus H2. So, X plus X means these two. So, these two to be added and then H1 X1 to be subtracted from this output and then delayed by 1. H1 X1 to be subtracted. So, H1 X1 is here. This is to be subtracted and then to be delayed by 1. 'D' delayed by 1 and then whole thing to be, this part is to be added with this part. So, this to be added by this. This will be your Y0 n.

Next come to Y1. In Y1, X0 and X1 to be added to be filtered by H0 plus H1 and H1 XI, which is already calculated obtained, that is to be subtracted from here. So, X0 plus X1, X0 plus X1 to be filtered by H0 plus H1. So, X0 plus X1, so X0 plus X1 filtered by H0 plus H1. X0 plus X1 filtered by H0 plus H1 and subtracted, what is to be subtracted? H1 X1. So, H1 X1 is here. This is to be subtracted.

Then there is this part. At this part is what H0 X0, which is already known and then minus it will be now Z to the power minus 1, because everything is a function of Z now. Z to the power 3, instead of Z to the power 3, we have now Z to the power 1. All Z to the power 3 replaced by Z. So, Z to the power minus 3 replaced by Z to the power minus 1.

So, H0 z X0 z, one filter output. H2 z X2 z another filter output, delayed by 1, and this is subtracted from here. This is already obtained here. H0 Z H0 X0 minus this 1 delayed output of, 1 delayed version of what, H2 X2. This is already obtained, already calculated in this part. These two are (())(10:10). So, I will just take that thing and subtract from this part.

So, that is already calculated here. H0 X0 minus 1 cycle delayed of version of H2 X2. their inverse H2 X2. We have already obtained, so need to calculate again. So, I will simply use it. And this is to be subtracted from here. So, minus plus, this is Y1 n. And lastly, I have got this filter H0 plus H1 plus H2 and it processes what signal? Summation of the three components, X0 plus X1 plus X2.

Then from the output, this is to be subtracted. But this is already obtained somewhere. These, these two are same, okay. And sorry, I do not know how it is coming red. Anyway, you can easily see this H0 plus H1 into X0 plus X1 minus H1 X1 that is here also. So, I already I used it so I can take this output, no need to calculate it separately. Similarly, H1 plus H2 into X1 plus X2 minus H1 X1 is occurring here.

So, keeping these things, first you consider this filter, process it; use it to process X0 plus X1 plus X2. So, this is here. So, X1 plus X2 plus X3. So, X1, X0 plus X1 is already here. I pull it here and X2. So, X0 plus X1 plus X2 coming here. From this, I have to subtract this part, H0 plus H1 to X0 plus X1 minus H1 X1. H0 plus H1, X0 plus X1 minus H1 X1. So, H0 plus H1, X0 plus X1 and this, minus H1 X1, this part. So, this I have to subtract.

And further subtraction will be this further subtraction, H1 plus H2 filter, it is processing X1 plus X2 then minus H1X1. So, H1 plus H2 filter is processing X1 plus X2 minus H1 X1, this one. So, this is the structure (())(14:43) from we get X0 n, X1 n, X2 n, this is nothing but as we discussed last time what we do in that unfolding, this is input buffer, X n comes.

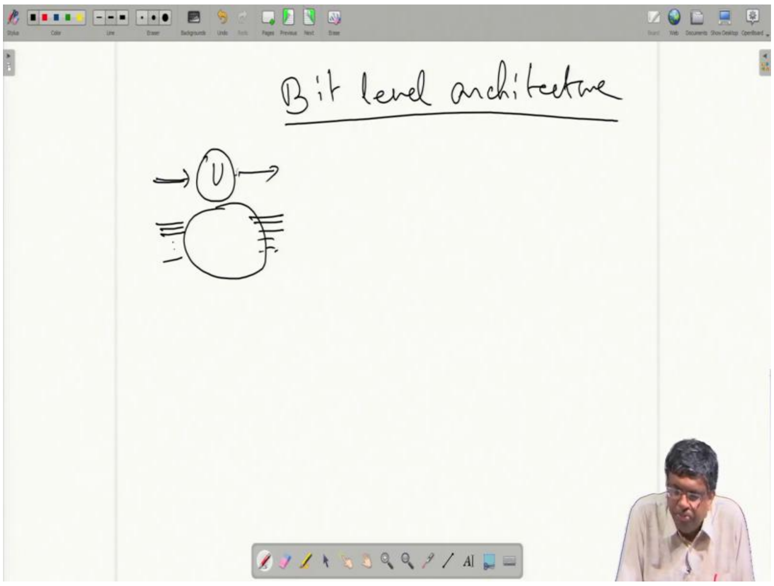
So, X of 0 comes, X1 comes, X2 comes. Then these switches open. So, X0, X1, X2. Next time X3, X4, X5 and so on and so forth, just by this buffering arrangement. And then, here, they get into output buffer. All these were discussed last time. And this buffer is clear, that is three times the rate. So, you get back your Y n, this was discussed in the previous class.

So, here, you see I have got 6 parallel components, but what is happening, suppose H Z has say 99 coefficients. 99 coefficients. If I had gone for ordinary unfolding, then this, I would have had 3 times 99 that much hardware increase. Because, I would have had parallel if I had filter copied thrice in parallel. So, 99, 99, 99. Almost close to 300. But now, every 33 coefficient. So, 33, 33, 33 and again 33, because this duration of H0 Z is also 33 times, H1 also 33. So, this just get added to form a new sequence. So, here 33, here 33, here 33. So, here 33 into 6, okay.

So, roughly two-third. It will be 198, so roughly two-third of the total amount that will be required, which is a big saving. Now, if you are more interested, there is a way of doing it in the general case that is J-parallel. Not just two parallel or three parallel, but J-parallel. So, that I leave to you. In case you are interested, you can, you know, try yourself that, read from (())(0:17:09) book.

But by this we show that how (())(17:12) tricks can be used to, you know, further optimise this implementation that is you are getting the speed increased by a factor of 3 here, but hardware is not really going up by factor of 3 but just by, I mean you are taking, of course, not three but some fraction of it, some fraction of it.

(Refer Slide Time: 17:56)



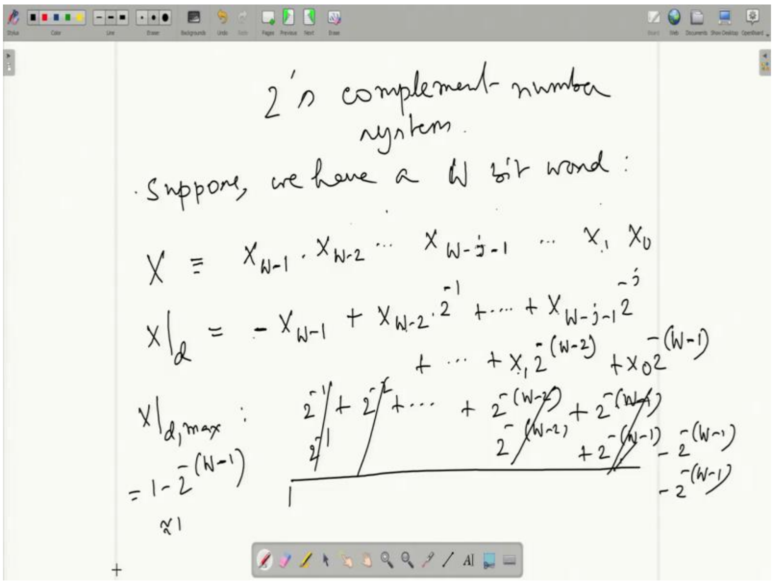
All right, so this covers this topic and we now move to our next topic, which is called ‘Bit Level Architecture’. So, long our (())(0:18:08) thinks like this, a DFG, where there is a bus, there is a line and there is another line and there is a node. It was doing some processing. But this line is actually a bus, it can be a 16 bit bus, it can be a 32 bit bus, similarly this also is a bus.

Now, through each line I have a bit, either 0 or 1. But, so far, whatever it optimising techniques were discussing, like folding, unfolding, in most of the cases except for some particular place in unfolding, we otherwise concentrated only on, I mean this where we take the bus as it is. We do not get into the bit level description, that is, we do not get into each separate line, that is these are bit lines and then do optimization, taking them into account. Rather we can say, that total bundle together is a bus and then whatever architecture, whatever structure, we have, we do optimization there. That is what we have being doing earlier. That is called (())(19:15) level.

But now if you really split a bus that is, resolve the bus into all the bit lines and then start optimizing considering the bit lines, then it will be called bit level architecture, that is you are now getting into the various bit lines at a bit level, you are doing optimization. Not by treating all the bits together, bit lines together is a bus and then whatever structure you get, you are doing optimization. Not that, if you are looking at the bit lines separately and then doing optimization then what we get is bit level architecture.

This again a very important topic, there are lots of optimization techniques and very powerful techniques. So, I will just introduce some of them to you. But the basic thing for this is 2's complement number system.

(Refer Slide Time: 20:13)



2's complement number system. Suppose, we have a W bit. This is one bit. This is called the (())(0:20:56) the most significant bit called X W minus 1, then W minus 2. We have a binary point here. This is in our mind. There is no hardware for binary point. This is a reference point I will tell you the meaning of it. Just put a point, so X W minus 1 is a bit, that is either 0 or 1, binary bit. X W minus 2 is a binary bit, either 0 or 1, dot dot dot dot W minus 1, W minus 2, minus j plus 1, okay. One minute X1 X0.

So, how many bits? 0, 1, 2 up to W minus 1. So, 0 to W minus 1 means W bit word. And I have a binary point in my mind here. In my mind, there is no hardware. It is just a point of reference. Then, under 2's complement number system, remember, you all studied little bit of 2's complement number system but there is a what you studied in your college is actually not that 2's complement number system, rather how to take 2's complement. That is, you know, complement each bit and add 1 at the (())(22:33), that is called taking the 2's complement, 2's complementation. But here, what we are discussing is different. It is the 2's complement number system. So, I am defining a 2's complement number system.

How? That under this, if I am giving a W bit word, X0 X1dot dot dot up to X W minus 1, then its decimal value will be what? I will write a formula. But before I write a formula, let me explain certain things here. When I am considering a binary word. Every bit is binary. Then, this can take 0 or 1 but though at binary bits. 0 or 1, binary bits. 0 or 1 binary bits. And they follow what is called binary arithmetic, where, 0 plus 1 is 1, 0 plus 0 is 0, 1 plus 1 is 0 with a carry 1 and dot dot dot dot.

The moment I write like this I will a decimal expression. That time, binary 0 will become decimal 0, binary 1 will become decimal 1. That is, if it is this bit is binary 0 here, here in the expression, when I write it, I will take that to be decimal zero. If this bit is binary 1 here, if I write it here in a decimal expression, I will take the value to be decimal 1. Because I write an expression where there will be minus symbol.

Now, you see, the binary domain, there is a thing called minus, minus binary 1 has no meaning. Minus binary 2 has no meaning. But minus 2, minus 3, minus 5 that minus decimal number has a meaning. So, using that, then this under 2's complement system, this binary what? X has a decimal value d for decimal. So, its decimal value will be minus of this that is if this was binary 1, this will be minus of decimal 1. If it was binary 0, it will be minus of decimal 0 which is 0. So, minus this, then plus, you start from here. X W minus 2 into 2 to the power minus 1, if it is 2, it is minus 1. If it is 3, it will be 2 to the power minus 2. So, dot dot dot dot X minus j minus 1 that is within bracket j plus 1, so it will be 2 to the power minus j.

1 means W minus, within bracket, W minus 1. So, it will be 2 to the power minus W minus 1 minus 1 and it will be this. This is the definition of 2's complement number system that is, if I say that I am following 2's complement arithmetic then, if you give me a binary word, then the decimal value of the binary word will be this, okay.

Now, what is the meaning of this, let us say what will be the maximum positive value? (()) (0:26:02) what is the maximum positive value? Maximum positive value will be when it is minimally negative that is this minus term has minimum contribution. What will that be? If this is binary 1, then it will be minus 1. That means there will be some substantial minus number. But if it is 0, then minus 0.

So, obviously, in that case only minus will have minimum contribution so that means this image will be 0 and let everybody, every positive term contribute, that is X W minus to be 1, X W minus to be 1, X W minus j minus 1 be 1, X 1 be 1, X0 be 1. Then all these have contribute positively and negative contribution will be minimised if this is 0.

So, if it is 0 and all others are 1, then it will have a maximum value, X d max. And that will be equal to 1 into 2 inverse, then 1 into 2 inverse 2 dot dot dot 1 into 2 inverse W minus 2, 1 into 2 inverse W minus 1. This is (())(0:27:14) series you can work out the sum, okay. I can now get the sum by some trick.

Suppose, to this I add the same thing again and subtract it out, they can (())(0:27:29) basically there is no problem, but if I add the two, I get twice this. Twice this means it will come here. Twice this means it will be here. 2 into this that is 2 to the power 1 so minus W minus 2. Again when you add the 2, it will move here so on and so forth.

Finally, it will be 2 inverse and 2 is a 1. So, 1 minus this. So, its maximum value is, 1 minus see W is large. What size is large? This will be so small, it will be virtually 1. But remember, actually it is not 1, slightly less than 1. Similarly, you will find out its negative values, maximally negative value and then all other properties of this, we will consider that will be the next class. Thank you very much.