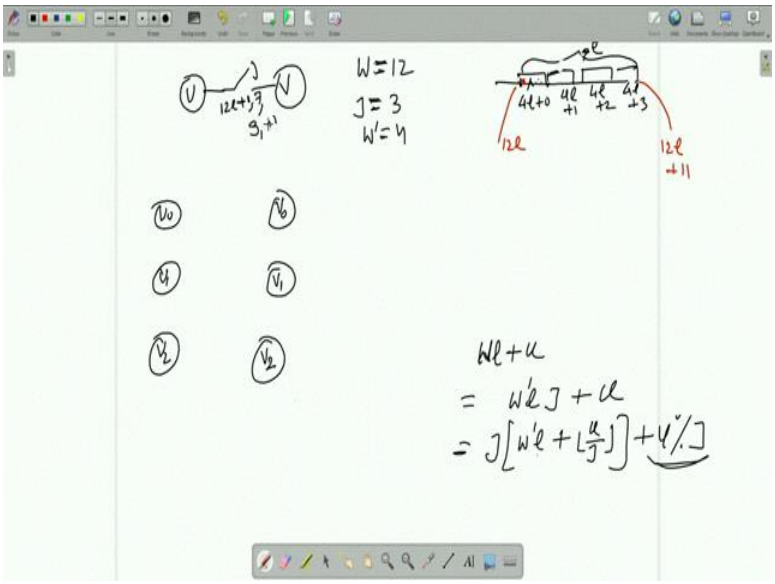
VLSI Signal Processing
Professor Mrityunjoy Chakraborty
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur
Lecture 20
Unfolding Bit Serial Systems

(Refer Slide Time: 00:25)



Yeah, so the previous class we have discussed this, now will consider some example to start with as I told you. So, let me read some part. Suppose it is given U. I am taking an example from Parhi’s book. It is giving W equal to 12, 12 bits constitute one word and you are parallelizing by J and J is 3. Which means out of every 12 bits that constitute a word and they are coming serially because it is a bit serious system. You take first three you form a first digit, they are parallel.

Next three again make the parallel that is next digit, so and so forth. So, everyone has got four digits, each digit of size three, four digit cycles. So, l th word, l th word will be as you know, l th word cycle will have one digit, another digit, another digit. So, this digit will have three points, these three points, they will have three points three and three.

So, first digit cycle will be what? W prime L I told, W prime here is 4. So, 4l plus 0, 4l plus 1 4l plus 2, 4l plus 3. In 4l plus 0, three bits of the l th word sorry, they will be parallel in 4l plus 1, next three bits will be parallel so and so forth. So, 4l plus 0 is the first digit of the l th word, 4l plus 1 next digit cycle. Then 4l plus 2 next digit cycle and so and so, this is what is given. That is W is 12 I have to fold by three, so W prime is 4.

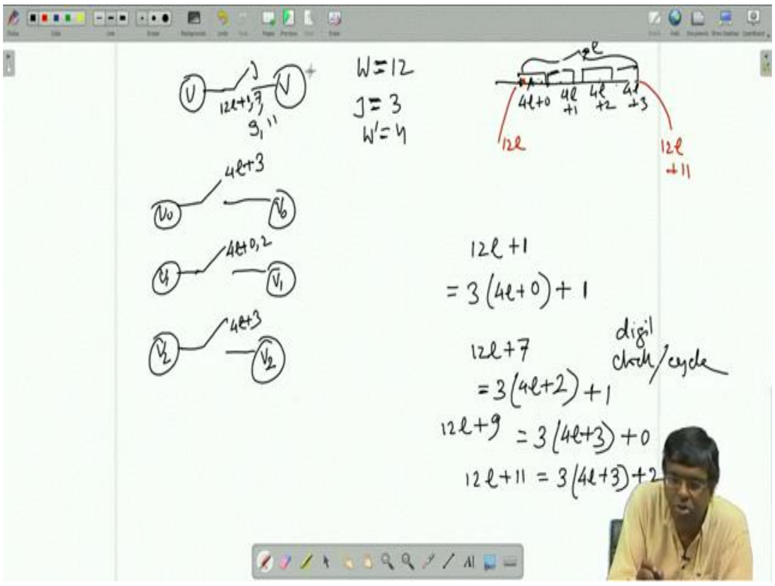
So, since this is parallelized by 3, U will be copied three times U0, U1, U2. V will be copied three times and it is given, this is closing not only at once, it is closing at this in this is. This is 4l plus 0 was the digit index but in terms of bit position, in terms of bit position this was 12 l is not it. Because every word cycle has got 12 bits. So, first one was 0 to 11, next was 12 to 23, then 24 to 33, 24 to whatever.

So, 12 l 12 l plus 0 up to 12 l plus 11, last one is 12 l plus 11. When I write in red here, these are the bits 12 l, 12 l plus 1, L plus 2, L plus 3 like that. This is the digit index, digit cycle index 4l plus 0. For the l th word is a 4l plus 0 th digit cycle, 4l plus 1 th digit cycle and like that. But in terms of bit indices 12 l, 12 l plus 1, 12 l plus 2 up to 12 l plus 11. Because every word cycle has got 12 bits I have divided into blocks and making them parallel that is why block are coming in terms of 4l.

Because totally is 12, I am 3 at a time, so 4 into 3, 12. So, I have got 4l, 4l plus 1, 4l, 4l plus 3. But in terms of bit, actual bits in the bit serial system 12 l, 12 l plus 1. There it 12 l plus 0, 12 l plus 1, 12 l plus 2 dot, dot, dot up to 12 l plus 11. And it is given that at certain positions within the cycle, certain positions not only once because earlier I consider only once, now certain positions given as, 12 l plus 1, 7, 9, 11 at this points of time the switch closes otherwise not.

So, earlier I considered only one point Wl plus U, Wl is 12 here, so Wl plus U either 1 or 7 or 9 or 11. So, whatever I did for one case Wl plus U that have to do for 1, U equal to 1 separately for U equal to separate 7, separately for U equal 9, separately for U equal to 11.

(Refer Slide Time: 05:46)

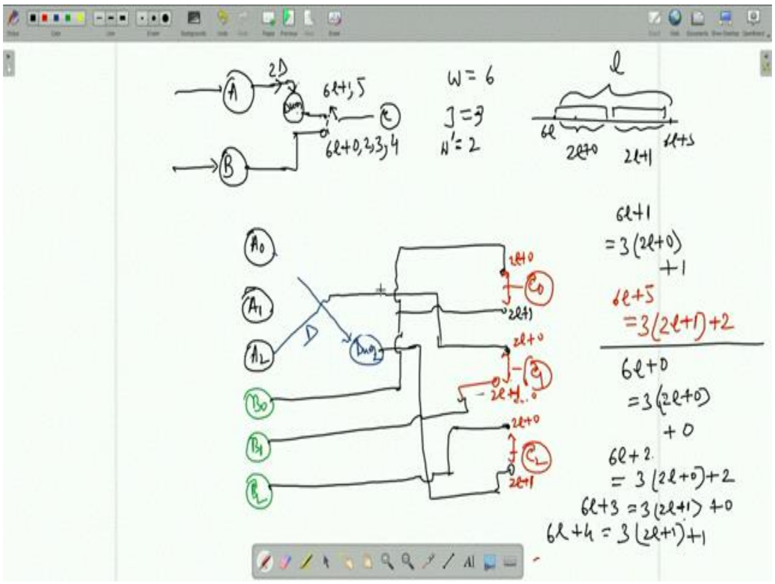


So, start with 12 l will plus 1, take J common, J is 3, so 12 l, 4l, then define 1 by 3 cossent is 0. And whole 1 is left out as reminder. So, in 4l plus 0 th digit cycle that into 3 and which point within that not the starting 0th point but point number point 1, I have got 012, 0l2, 0l2. So 1 that means U1 that will get connected to V1 no delay in this path. So, U and V go to at what clock, 4l plus 0 that digit clock. This is digit clock, digit sometimes as a cycle. 4l plus 0 th digit clock.

Next index is 12 l, 12 l plus 7 again take 3 out, 4l divide 7 by 3, so you got 2 and 1. So, 4l plus 2 at 4l plus 2, I get U1 because reminder is 1. So, this point 1 which goes to U1, U1 will go to V1. So, U1 goes to V1 not only in 4l plus 0 that digit cycle also at 4l plus 2 these digit cycle.

So, 4l plus 0 and 2. Then cosset 12 l plus 9, take 4 common, sorry 3 common, I am sorry, take 3 common., 4l divide 9 by 3. So fully divided 3 and 0. So, at 4l plus 3 that digit cycle U0 connects to V0. So, U0 will go to V0 at 4l plus 3 and last one 12 l plus 11, 2. So, U2 goes to V2 at 4l plus 3.

(Refer Slide Time: 08:40)



Next example another one. Now, I introduce delay, this is a 2D delay here. In this edge, there is a switch, switch either closes to this or closes to these. So, in some cycles, this path is in these bit serial system. So, we need every word cycle at certain bit points or bit clocks, this exists these ribbon is connected, this does not exists, this is disconnected and vice versa. It is given, so let me show by dot, dot, dot, it is given that or word cycle here is 6.

So, it is given a fully bit serial system or original word cycle was 6, so 6 bits constitute one word, now using the faster clock, 6 bits are coming serially down them, every 6 bits constitute one word, every 6 bits constitute one word. Clock is 6 times faster than the original input clock because in one input clock, input word clock all the 6 bits are packed one after another. Anyway that is a (()) (10:25) story.

So, this a situation, the switching indices are obviously here as you know l th, l th word cycle will be having word points, 6l plus 0, 6l plus 1 up to 6l and plus 5 is not. Because total size is 6. So, out of this bits points, this will be closing at 6l plus 1 and 5. And the other side will be closing the other indices 6l plus 0, 2, 3, 4.

Now here you see, so I have to unfold by J equal to 3. So 3 bits will form one digit cycle, 3 another digit cycle. So, W prime is 2, so in to 2 is 6, so this will be 12 plus Wl, W prime l, W prime l plus 1 like that, so this is 12 and there is 12 plus 0, this one in terms of digit cycle and this is 12 plus 1. Here I left 3, 3 bits, here I left 3 bits, this 3 bits parallel, this bits parallel, so 12 plus 0, one digit cycle, next digit cycle 12 plus 1. I have to unfold it by J equal to 3.

Now, you see here I have got a delay 2D so, what I will do, I will introduce a dummy, dummy node which only takes data and gives does nothing else? So, what I will do is I introduce a node called dummy, D, U, M dum or dummy. So, dummy node takes from A and gives it to this side. So, now you see the, if there is an edge between dum and C at this edge has no delay. Delay I have kept to the left top dum between A and dum.

So, now I unfold by a factor how many? Unfold by a factor J equal to 3 because 3 bits will be kept parallel and 2 digit cycle 12 plus 0, 12 plus 1. So, now let be, so A0, A1, A2, then B0, B1, B2, dum 0. Because there will be every node will be copied parallelly three times because J is 3.

First question the edge A2 dum, there is to switch is a normal usual unfolding, you know A0 will go to I believe you know it by already. A0 will now go to this one under the same system cycle node delay, A1 will go to this place with one systems cycle delay. A2 will go to these with word systems cycle delay. Because 2D, A to D dum.

So, 3 bits is unfolded by 3. So, if this just to remind you, we had this situation of unfolding. Remember this thing in unfolding KJ plus 0, KJ plus 1, KJ plus 2, this if you go to the right by 2, 1, 2. So, from 0 you go to 2 that is why A0 going to dum 2 with but no system delay because under the same umbrella. But KJ plus 1 will go to the starting index here. So, one system cycle delay and from 1 it will go to index 0 that is word and this will go to index 1, this is pretty simple, this we already know. I did that just to get, just to remind you about what you did.

Now, consider dum to C, C also will be copied three times. So, C0, C1, C2 alright. Now, consider 6l plus 1, this between dum and C this edge it is closing at 6l plus 1, 6l plus 5. So, started 6l plus 1, 6l plus 1, take J common 3, 12 plus, divide 1 by 3 that is 0 and 1. So, 12 plus 0 th digit cycle dum 1 will go to C1, 1 and there is no delay here. So, it gets corrected dum 1 goes to C1. So, dum 1 will go to C1, C1 actually I am not connecting I am putting a switch here.

Well very soon you will see that this is a switch. This is getting connected at 12 plus 0. So, 12 plus 0 is going connected to C1, dum 1 getting connected to C1 at 12 plus 0. Take the next one that is 6l plus 5. Take 3 common that is J, so 12 plus 1 and then 2. So, dum 2 will get connected to C2 at 12 plus 1, so this will get connected to C2 at 12 plus 1.

Now, those are the two switch indices between dum site and C side, other are from, from B side to C side. So, now next we consider the other ones from B side to C side 6l plus 0, 6lplus 0 take 3 out 12 plus 0 plus 0. So, 12 plus 0, B0 goes to C0. So, B0 will go to C0 at 12 plus 0. So, C0 has a switch, this side is at 12 plus 0.

Next one is 6l plus 2, take 3 out plus 2. So, B2 goes to C2 at 12 plus 0, so B2 goes to C2, B2 goes to C2 at 12 plus 0, still some more are left. 6l plus 3. So, at 12 plus 1 B0 goes to C0. So, B0 goes to C0 at 12 plus 1 and last one is 6l plus 4 take 3 out. So, 12 plus 1 plus 1, so 12 plus 1, B1 will go to C1. So, B1 will go to C1, ohh (()) (20:22) correction required sorry.

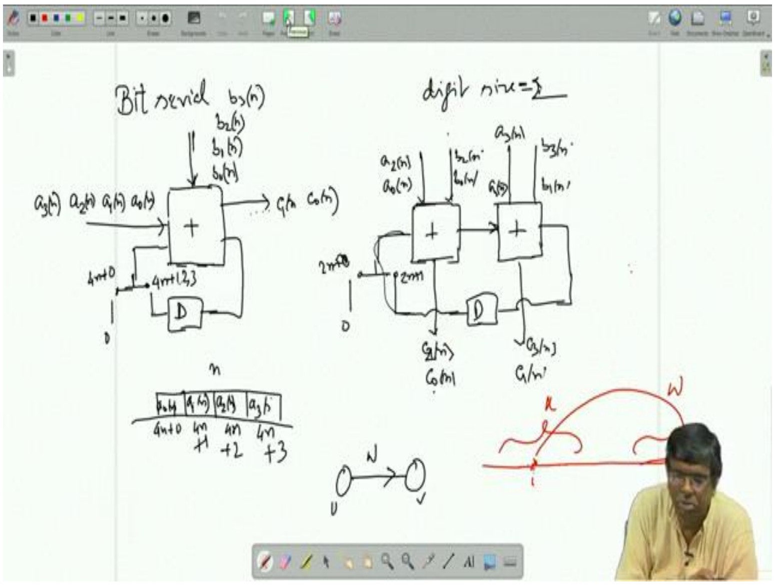
This is showing 12 plus 0 and 1, so downward should go to C1 at 12 plus 0. So, my convention is upper side is 12 plus 0, lower side is 12 plus 1, upper side 12 plus 0, lower side 12 plus 1. If I were to stick to that convention, let me just interchange the two.

So, dum 1 going to C1 at 12 plus 0. So, 12 plus 0, this my convention upper side to the switch 12 plus 0, north side it connects at 12 plus 1. But to stick to that 12 plus 0 and now I am getting the 12 plus 1, again B1 goes to C1 that is why B1 goes to C1 at 12 plus 1.

Now, you can simplify it. Firstly you see this dum 0, dum 1, they do not do anywhere, they are not of any use. So, these edges, you can remove because they are not taking you anywhere. So therefore, I am sorry, dum 1, sorry that that edge, this edge. And other dums is short circuit.

Because what does it dum do? What for it takes something and gives the same thing. It does not do any processing. I just created the dummy node, so this will be again you just sort it, another interesting thing you see B0 it is connected to C0 in both 12 plus 0 th cycle and 12 plus 1th cycle, in both the cycles, 12 plus 0 and then 12 plus 1 th that means it is always connected. So, I will have a direct connection between B0 and C0. So, that is all you have, these unfolded structure.

(Refer Slide Time: 23:13)

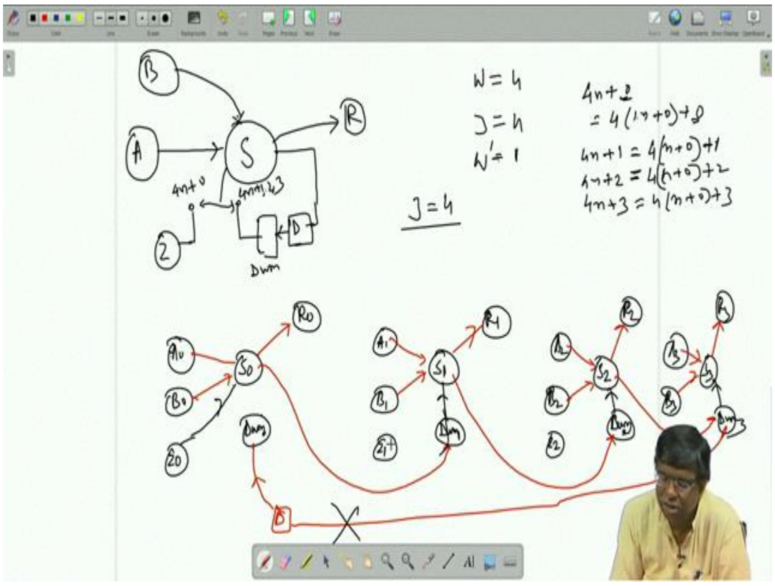


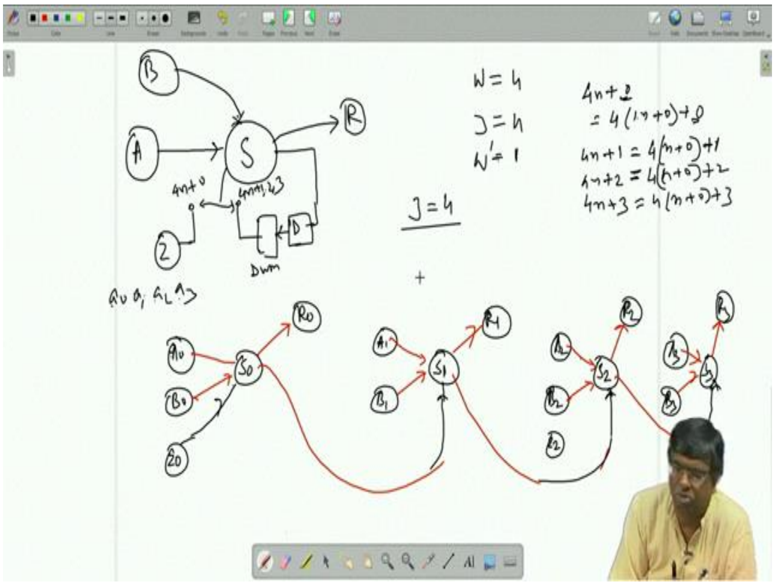
Now, you should consider, how to convert this into? How to apply this to the bit serial stuff? The bit serial adder and unfold it, remember we had this bit serial adder, maybe I can go back to that page and it says my drawing. This was a bit serial adder right. So, if you want to draw a DFG, this will be one processor node, it is coming from a source node called A, and these bits are coming from another source called B, A, B. These is another source which is permanently holding by the value 0.

So, from A there is a edge going to source node, adder node, adder is summer, you can call it S. So, from a source node A which is delivering the A bits, it will go that edge will go to S node, from a source node B another edge will go to this S node, from our source node Z, Z stands for 0 which holds binary 0.

There is an edge which goes to the summer but through a switch, switch closes only once per n th what clock that is at 4n plus 0. And from summer S output and edge comes. So, there is a loop with a delay, but this edge has a delay plus this has a switch, it does not exist always and this goes to this output, output node, may be R, capital R, capital R for result these are situation.

(Refer Slide Time: 24:38)





So, you can then draw DFG for it, S for that adder, A for the source node of the bits, A0, A1, A2 that bit, those bits, B is generating the B bits, this is producing the result in R, going to the sync node, there is a Z node. But there is a switch, Z node is connected at 4N plus 0. And this is connected the carry by a delay at 4N plus 1, 2, 3. So, there is a delay I will put a dum here, this is dum so between dum and this S there is no delay, there is only switch.

Now, unfold by 4 and unfold by 2, if I unfold by 4 we will see, we get back the original structure, original bit parallel or structure, if you unfold by 4. Then what is happening? A will be copied four times, B will be copied four times and like that. So, we have A0, instead of making them vertical A0, A1, A2 like we do in U0, U1, U2 I am making them horizontal that is all or maybe I erase this, here A0.

Then here we A1, A2, A3, but let me also do for B0. Here is S0 every node will be copied four times, R0 and W also will be copied four times, Z will be copied four times. So, Z0 dum 0. I am unfolding by J equal to 4, to start with unfolding with J equal to 4. Similarly will have A1.

Now, and lastly running sort of space here A0, A1, A2 done, I will have A3, B3, S3, R2, dum 3. Now, first S2 dum, this is one delay? So, if I unfold by 4, these are the four points, you are delaying by one. So, if you start from here you are going here only, under the same cycle. So, no delay, which means S0 will go to dum 1.

So, this will go to dum 1 without any delay. S1 will go to dum 2 without a delay. S2 will go to dum 3 without a delay. S3 will go to, last one will go to dum 0 but with a delay, I am moving to next cycle, from here I am going to here. So, this will go to dum 3 with one delay. It was very simple. A, A0 because these edges A2 to S for example I have no delay. So, A0 will go to S0, A1 will go S1 and so and so forth. So, A0 goes to S0, A1 goes to S1, A2 goes to S3.

Similarly, S to R also no delay, so S0 goes to R0. This brings very trivial alright. Then started this switching things W equal to 4, J equal to 4, meaning W prime equal to 1. So, started this indices 4n plus 0, 4n plus 0 you take 4 out Wj, j into W prime l, W prime is 1, so 1 into n plus 0 by 4 0 plus 0. So, at n th, see now n th word cycle is same as n th digit cycle because all the four bits are made into parallel.

So, n th word cycle, whole word cycle is a digit cycle? So, n th word cycle has got only one digit cycle which is the n th. So, n th digit cycle, who will get connected? 4n plus 0. So, Z0 will get connected to, it will get connected to whom? S0, so this will get connected to S0 n th word cycle. And n th word cycle means, the whole of the word cycle, no switching is required.

Whereas if you see this edges 4n plus 1, so it is a dum output, dum 1, dum 1 will go to whom? In the n th word cycle… and therefore equivalently added digit cycle, dum 1 will go to S1. So, dum 1will go to S1, no switch required because the whole word cycle is the digit cycle now. I do not have to switch, dum 1 goes to S1. Similarly dum 2 goes to S2, dum 3 goes to S3 and here this goes here with 4n plus 0 that is for this side, Z side.

So, Z0 goes to index is 0. Z0 goes to S0, and you see this does not go anywhere. So, no need to have this point, this at all. So, this is your full adder because you see A0, B0 these two bits, what does this node do? It takes the LSB only, four bits are packed and made parallel. So, LSB 0 th bit is given to this source, 0 th bit is given to. So LSB, LSB come a binary 0 is always giving the carry initial carry, result goes out new carry goes.

Now, dum you short circuit. So, it goes to S1, this Z1 has no meaning, so I can erase them now, all the dums I can short circuit. So, A0, B0, initial 0 carry result, new carry goes in, this carry goes in, carry input A1, B1. There is a next two bits because A0, A1, A2, A3. In any cycle, this will go here A1 will be small A1 will come here, small A2 will come here, small A3 will be coming here, same for B.

So, now the same cycle then, because they are parallel A1, B1 get added here alright A2, B2, get added here, they all get initial carry, they carry from the previous addition and you get the result, this is nothing but your four bit full adder. Interesting thing will be J equal to two case, when from bit serial, we convert it into a digit serial adder with digit side equal to 2. So, that will be done in the next class. Thank you very much.