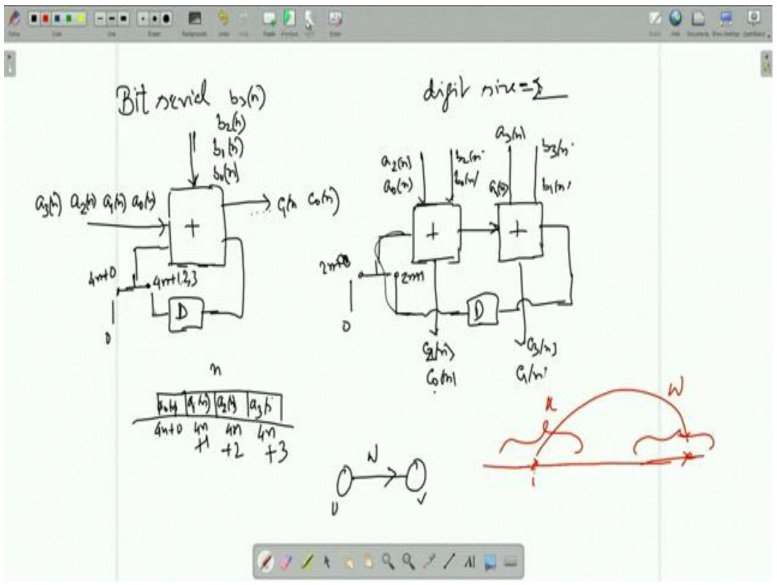
VLSI Signal Processing
Professor Mrityunjoy Chakraborty
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur
Lecture 19
Unfolding a Switch

(Refer Slide Time: 00:30)



In the previous class we are considering bit serial, digit serial at word serial structures. As an example I took four bit full adder, which was working on fully word serial basis that is, four for one input a, full length for one input b, you are coming in parallel. All the bits, four bits of a, four bits b, and output four bits were going in parallel that is a fully word serial or b parallel structure.

They may said that I am explain that the structures are fast, but they consume more hardware because whatever operation are not processing you do like an addition on the LSB, same we have to do for next two LSB, again next to, next to LSB so it just gets replicated. So, pay in power, pay in hardware but gain in speed, but the other extreme is that a bit serial where may be the n th cycle.

Suppose the every word cycle say n th cycle clock this will be divided into four, because I could say a four bit case. So, four bit cycles, so their index will be 4n plus 0, 4n plus 1, n plus 2, 4n plus 3. So, if 4n plus 0 I will send in the LSB say, not necessary could have been LSB also. But follow a convention that of LSB first. So, a0n comes here.

Then next bit cycle a1n, like this a0n followed by a1n, followed by a2n, they enter the system, same for B. So, system is working at a faster rate. So, output will be c0n, c1n, dot,

dot, dot, dot. Suppose it start with bit serial structured and this of course the carry, is the LSB cycle whatever carry was, I mean it required input carry 0. So, that timing 4n plus 0 this is a switch, the switch switches to 0.

So, the carry input of the adder, it takes output from a switch, switch switches to a fixed binary 0 at the cycle 4n plus 0. Because that is the when where is a time when a0 and b0 they are getting added. So, initial carry should be 0. Then the new carry generated will be stored in a flip flop.

Then I moves to the next cycle, next bit cycle 4n plus 1 that time a1n and b1n will get added the previous carry should come up here next cycle. So, switch moves to the side at 4n plus 1. Then again a2n, b2n get added you get c2n, new carry comes that gets stored in the flip flop. Then the next cycle it comes up and so on and so forth. So, there is a switch.

Therefore, from the adder output to its input, carry input there is a path which has a delay, but the path has a switch. So, path is discontinuous that is, it is not active and it is not valid or it is not existing at all the clock cycles for his steps at 4n plus 0. This path does not exist. I mean this gets disconnected cut, what is the path from a fix binary 0 to the carry input is a path, its get connected only at 4n plus 0 otherwise it is not connected the situation.

Then we can, then I considered a digit serial structured, where instead of just sending in two bits, a one bit at a time, we can take two bit at a time say. So, the two bits will be called a digit. So, a0 and a1 a0n and a1n together they will come in the same cycle. So, n th cycle it will be divided into two parts 2n plus 0 and 2n plus 1. In 2n plus 0 who comes? a0n and a1n together and 2n plus 1 who comes? a2n, a3n.

Same for B, so digit side is two. So, at 2n plus 0, initial carry is 0, so the carry inputs switch moves to 0, a0n, b0n get added, real comes up c0n, new result, new carry goes in. Then a1n, b1n get added, result comes in c1n carry gets, carry output get stored in a flip flop. Then we move to 2n plus 1 cycle, switch moves to this side. So, this carry comes here a2n and b2n get added with that carry new result comes, new carry comes. New result c2n, new carry comes. Then it adds a3n and b3n, result new result comes c3n and carry generated which will be overflow.

And then, we already know what was the original fully bit parallel or fully what serial structure that we started with? Now, my claim is giving the bit serial structured if I unfold it by two, unfold. Then I can get back this digit serial structure. Because that time every two bits, you know will be parallel in unfolding we assume the bits are parallel to a buffer.

So, here buffer size will be two. So, a0 and a1n and they will come out of a buffer. So, the adder will be copied twice like that. Then a2n, a3n will come. So, the digits will be formed a0n and a1n coming together, then a2n, a3n coming together like that through the those two same lines.

Anyway I elaborate on this today is my only claim. And then if I unfolded it by a factor four then all the four bits will be fully parallel, adder will be copied four times. And similarly the four bits of b all fully parallel that will get us back the original fully word serial or bit parallel structure, where four bits come in parallel. Now, how to do the unfolding? Here unfolding the problem is, in unfolding the problem is there is a path or there are paths which are not existing in all the clock cycles.

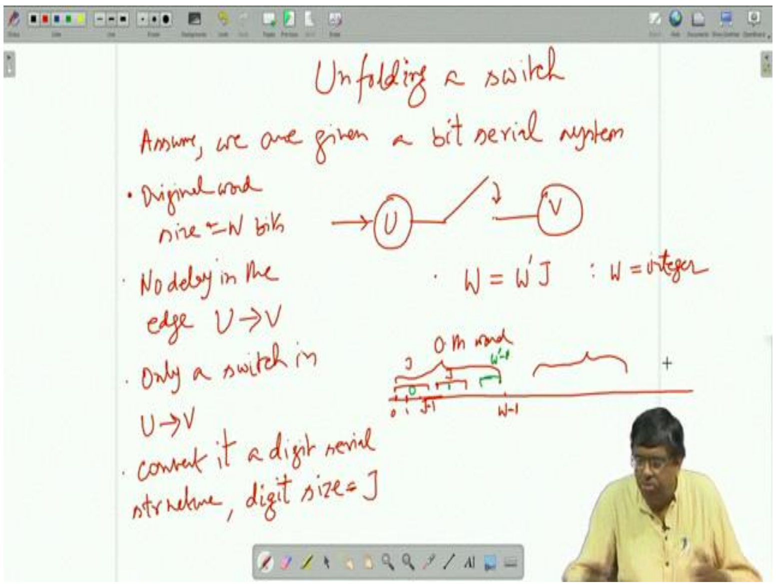
If I say U and V and W delay, then I have seen how to unfold. If I have to unfold by J, U will be copied J times parallel, V will be copied J times. There will be from a V i th node of U, Ui, it will go to some node of the V, say Vj with a particular delay system cycle delay. And particular for a particular i it will go to particular j, we already have that cossent formula reminder for our remainder formula.

But here the basic assumption was that this path is existing. This age is existing, get all the clocks, in all the clock period whatever comes here that gets delayed by W cycle and comes here. It is not that is certain clock cycles it is this corrected. I did not consider that, my basic assumption in the timing diagram was that, you know the path is existing in all the cycles.

That is why consider the timing diagram that time. I said that suppose I am in the k th and I am somewhere at the i, i th point I delayed by W. I get you some other slot, but my assumption is for every i the path that is for whether I am at this input clock or this input clock or this input clock or this input clock, this edge exist. So, data will always be delayed by W input cycle to reach here, if I start from i and likewise for other points.

But now I will be considering cases, where for certain indices say I, maybe L, maybe odd, there is no path. So, there will be no movement of data as is the case here. So, now I will first consider unfolding a switch.

(Refer Slide Time: 08:14)



Unfolding a switch

Assume, we are given a bit serial system

- Original word size = W bits
- No delay in the edge $U \to V$
- Only a switch in $U \to V$
- convert it a digit serial structure, digit size = J

We assume, we are given a bits serial system, original word size, W bits. That is originally W bits formed a word. Now, all the bits, all these W bits per word cycle will be sent down a line, will be sent down a line serially. All right, will be send out a line serially, one after the other because it was a bit serial system. I must also considering no delay in the edge U to V, I will not consider, but I will consider a switch.

How the switch works and unfolding factor? That is, I will convert it to digit serial structured, digit size J, W. W has some I mean, one word cycle has W the number of bits, I am assuming. So, I am taking J bits at a time out of W bits and making them parallel, so at one digit, then next J bits again taken made them parallel, next digit like that.

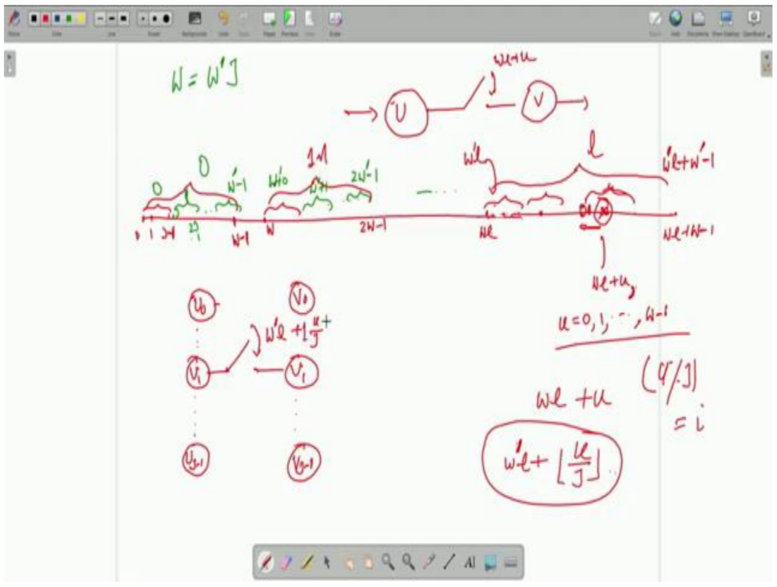
So, I am assuming W has an integral number of sub J blocks, one block of J bits made parallel that is one digit within that word, next block of J bits now make again parallel, there is a next digit within the same word block, so on and so forth. So, in one word block I will have W prime number of digits, W prime and integer,

If W prime is one that is if J is W that means all the W bits together will be made parallel by one go. So, it will be fully word serial because all the W bits within the W block, within the block of W bits will be parallely to available together. So, from fully bit serial it will be fully word serial. But what we are doing here is these that suppose I have got 0 1 dot, dot, dot up to W minus 1, this bits they form one word originally this bits were parallely going into this system.

But that is not given to us. What is given to us is a bit serial system where all the bits at bit at G is, bit at G is, bit at G is, they go into the system serially, go through the system serially one after another with a faster clock. There is clock period is much that one dumb. What I will do. I will take J of them. This may be called the 0 th word. So, I take J of them. That is 0 to J minus 1, I will J minus 1. So, I will make the parallel that will be by 0 th digit.

Then I will take next J, again make them parallel that will be by digit number 1 dot, dot, dot, all right indicate this way, this a 0 th digit, this the first digit, this will be word. W prime minus 1, W prime minus 1. Why? Because W is W prime J. So, how many J within one W, W prime. So, 0 th first dot, dot, dot, so W prime minus 1. So, this will be called 0th digit, first digit, second digit, W prime minus 1 th digit.

(Refer Slide Time: 14:10)



Next go to the next word. Let me go to the next page at draw it more elaborately, our standing point is the switch, this also I will redraw that is I am redrawing the diagram U, V, no delay here. And let me draw this timing diagram here. I have got 0, 1, dot, dot, dot up to J minus 1. This is by 0 th digit cycle, this was my 0 th word cycle, how many bits? W bits, but W equal to you know an integer W prime times J. So, within this I will have how many J blocks of J bits W prime number.

So, first one I called 0 th then next what I call there is turning from J, J plus word up to 2J minus 1 I call like J to 2J minus 1, I call it first dot, dot, dot, dot. This is W prime minus 1. So, 0 th digit cycle, first digit cycle dot, dot, dot. W prime minus 1 is digit cycle. How many digit cycle 0 to W prime minus 1 is W prime each sign J that is why W prime into J I have got W bits.

Next I have another, this is first word clock again here I will have J bits all right, that will be called W prime th because I ended with, just a minute. I ended with W prime minus 1 th. So, next will be W prime th, then will be W prime plus 1 dot, dot, dot. W prime means W prime plus 0 you can say. So, W prime plus 0, W prime plus word, total this one will be W prime plus W prime minus 1, this because total is W prime, that many number of blocks of J.

So, that will be twice W prime minus 1 that is W prime plus 0, one block of J bits. W prime plus 1 another block of J bits, how many blocks again of J bits W prime number? So, W prime plus 0, W prime plus 1 dot, dot, dot, W prime plus W prime minus 1 that is why total W prime. So, part word cycle have got W prime number of digits, they are coming parallelly.

Which bits I am unfolding? I will be unfolding by a factor J. So, in these bits first J bits will be bit parallel as though they are coming out of a buffer of sign J, makes J bits will come out of the buffer in that next word digit cycle which is the system cycle, so on and so forth. So, in general now let me considered, a general cycle, general word cycle. Maybe I draw little bigger here, what was this point? This point is 0 to total how many bits? W so, this will be W minus 1, this is W to another W minus 1.

So, 2W minus 1. 0, 1, 2, 3 J minus 1 J, J plus 1 dot, dot, dot, if you count total number of bits is W, so 0 to W minus 1. Next will be bitwise it will be W th bit to two W minus 1, if it is second, it will be 2 W th, to 3 W th minus 1. So, if it is L th, it will be Wl, it will go up to Wl plus another W minus 1. Wl means Wl plus 0. So, Wl plus 0, Wl plus 1, Wl plus 2 dot, dot, dot, dot, Wl plus W minus 1. So, how many bits again W bits 0 to W minus 1 W bits that is a, in this index will be Wl this Wl minus, Wl plus W minus 1.

Here again I will considered the same way, sorry. Here again I will take J bits, so Wl plus 0 Wl plus 1 up to Wl plus J minus 1, (())(18:49) parallel. Then again another J bits like that. What is given is this in every word cycle, every word cycle like l th word cycle, the switch will close to this sign, which means the path will exist only once.

So, within this range from Wl to Wl plus W minus 1 maybe at some point only, which you can say WL plus some U. U can be 0 then you are here, U can be 1, you are here, U can be W minus 1 you are here. So, U from. This closes only at Wl plus U, it closes.

In fact will consider multiple closing cases also that is it closes not only at this point, it closes at multiple points within the same clock, same word clock. But that is later, to start with make line very simple, no delay in this edge and only closing of the switch only once, per word cycle.

So, the edge exists only at one point, one midpoint. When you are doing bits signal form for the bits are coming serially in 1 cycle 1 full word cycle that is that is the W bits coming seriously of one word. Only at one of these bit clock that data out of U, will go to data the input of V with no delay. Because I did not save any delay at other locations it is dot.

Now, under these, if I want to unfold it by J, unfold it by J, what will happen? This J bits will be coming parallely. So, will have then next J bits coming parallely, that is you will have. J unfolded system, so start with these 0 to J minus 1. 0 th will be doing U0. First will go to V1 dot, dot, dot. i th will go to ViN, Ui, sorry not. In the next digit cycle these bit will.

So, one word cycle will consist of how many such digit cycles? W prime number of digit cycles. First is 0 th, then first, then this first, second up to W prime minus 1 th, in the first when I moved to the first word block and then take digit cycles, digit cycles within these. First digit cycles where the bits will be now parallel, J bits that will be called W prime because previously I had W prime minus 1 th digit cycle. Now, will be W prime th, then next this J will be parallel that will be called W prime plus 1 th and like that.

So, what will be this one? When it is 0, 0 th why is it first? W prime th, the starting one. When it is second? It will be next to these, so two term to prime th. Why it is l th? You can count it will be W prime l th, W prime l th. Because every W has how many W prime and it starts from 0 goes up to W prime minus 1.

So, I move to l th, how many times I have got such W prime number of digit cycles? I have got W prime number of digit cycle here, here, here. So, how many times I move into l? l times that is why W prime plus l, counting starts from 0 goes up to W prime minus 1.

Then again, W prime to 2 W prime minus 1, then 2 W prime to 3 W prime minus 1. So, for l th, for l th it starts at W prime l and goes up to W prime l plus W prime minus 1. So, W prime l th, digit cycle. W prime l th plus 1 th digit cycle dot, dot, dot up to W prime l plus W prime minus 1. Because every word cycle has got W prime number of digit cycles. First word will be W prime l. Now, so they are parallel.

Then next new cycle again this bits are parallel, but you see, if you end up these bits are coming here and processed by U0 and all later coming out here, they are not moving to V because a switch closes only at one point not at other points. Earlier this point, data from here U side will move to V side. What is this point? That is under which digit cycle? I have this point under which digit cycle, I have this point.

And then within the digit cycle, how much is this? If it is 0, it will U0. If it is 1, it will U1. So, how much is this offset? So, you can see that I already have W prime l, then W prime l plus 1. So, this much is W prime l plus U, this much, this is W l plus U up to W l I have W prime l. And now U within the U, within the EU, how many blocks of J are involved? So, I divide U by J take the cossent. If cossent is 0, U is here.

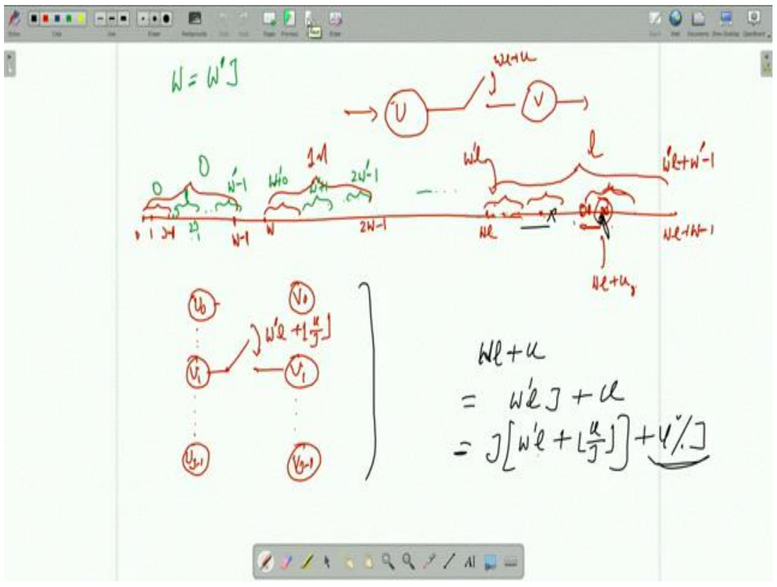
So, I am in the same digit cycle. If U is here, cossent is 1. I have digit cycle 1 after W prime l, W prime l plus 1. If U was here, if U was here, U if U is here, This U point is here, you take this cossent U by j, it will be 0, because U is here. So, I will read the W prime l digit cycle only, if cossent is 1 that will happen if is here that is W prime L plus 1, cossent is 1 W prime l plus 1. If cosset is 2 I will be the next W prime l plus 2. So, whatever is the cosset that will be W prime l plus that? That will be the digit cycle. That will be the index of this digit cycle.

So, in that digit cycle, this point exits, so in that digit cycle, there will be movement, but then within that digit cycle only at this point. Now, what is this point? This will be given to U0 in the same digit cycle, next will be given to U1 dot, dot, dot this cross, this point will be given to one of them, whom? That will see that is the remainder. If you have U percentage J that is if you divide and take the remainder, this is the remainder. If remainder is 0 you are here. So, that it will be U0.

So, within that digit cycle U0 output will go to V0 no, not for anybody else. If reminder is 1 then U1 will go to V1 not anybody else. So, I take the remainder, if remainder is UI. Suppose remainder is smaller i. than you I will go to Ui, will you have Vi, others will never go only one point Ui, if the remainder is I.

And it is not going in all the digit cycles, is going only a particular digit cycle. So, there will be switch, it will close at that digit cycle, which one? This one that is for the l th word, the corresponding index of the digit cycle, while this closing occurs, there is point remains under this umbrella that will be this one alright, others will never get connected.

(Refer Slide Time: 28:28)



Now, these I have explaining physically again again, but mathematically you can easily show this by closing index is, this is the time where it closes. Wl plus U, W you write mathematically I am proceeding W write as W prime J. So, W prime l into j plus U, take out j, how many j? So, W prime l plus you divide U by j if the cossent is these. So, that cossent times j and then the remainder. So, J into this integer J into 0, J into 1, J into this integer.

So, that digit cycle, that is why it is coming here. And within the digit cycle, where all the bits are parallel which node will take the responsibility of this, this that node, which is the remainder. So, only this it will close and it will go to i, Ui to Vi because there is no delay here. U0 should go to V0, U1 as I told you during unfolding, if there is an edge from U to V. And there is no delay, then whatever be the unfolding factor U0 will always go to V0, U1 will always go to V1, U2... So, that is why UI is going to VI. So, remember this formula?

These I will now, will now extend to the case of multiple closing, not only here at many places, in that case I will treat one at a time. What is the position of this in terms of all these you know I mean in terms of digit cycle, how many digit cycle like this? Which index, which is the remainder that is which processor index for this. Similarly if there is another one, then I will handle it separately the same way which digit cycle and what is that processor index and like that?

Next I will consider some examples, in the next step where this multiple closing case will be considered. Now, from now onwards there will be routine application of this formula, I will just take an example, I will consider multiple closing cases, not only once which we close at multiple points within the same block, each has to be handled separately like this.

Then next I will be introducing delay in this edge that is, there will be not be just switch, there will be delay, how to unfold that? And then here I have only one path U to V with a switch, now there can be another one U prime that also goes to V with another switch, how to handle those cases?

And then I will apply all these to that bit serial adder for bit which you took up and on unfolding by 4. I will show that we get back the original 4 bit full adder, unfold it by 2 will show that we get the original two digit adder that is all for this all. Thank you very much.