Digital Electronic Circuits
Prof. Goutam Saha
Department of E & E C Engineering
Indian Institute of Technology, Kharagpur

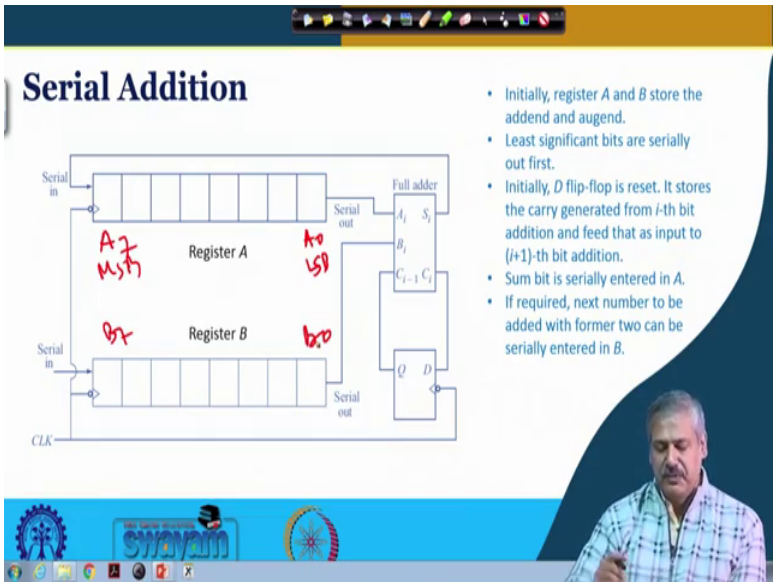
Lecture - 40
Serial Addition, Multiplication and Division

Hello everybody, in this class we shall look at use of shift registers in Serial Addition, Multiplication and Division ok; that means, in arithmetic circuit building. We have seen them this addition, multiplication and division in commutatoral logic related discussion and there we told that there are other ways other way to do it and one is using sequential logic based implementation.

(Refer Slide Time: 00:43)

So, we shall see how it can be done using shift register.

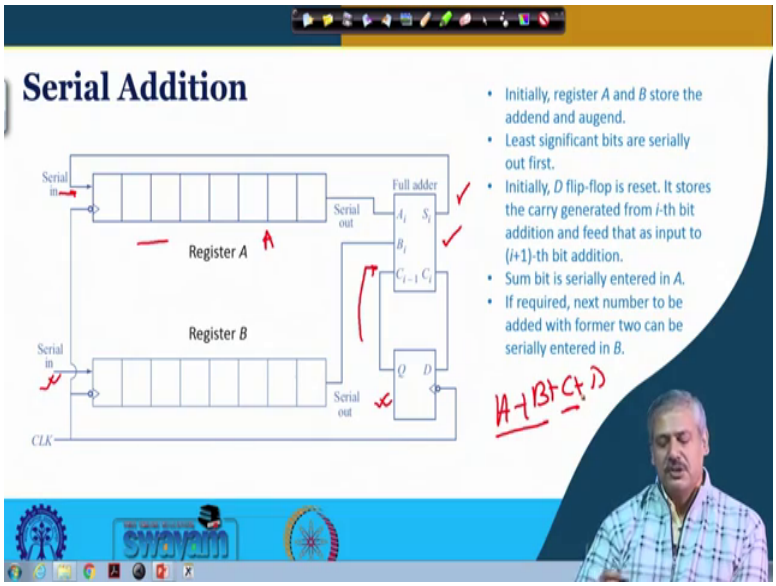
(Refer Slide Time: 00:51)



So, we begin with serial addition so, in serial addition what we have is this shift register two shift register we are putting one is your register A and register B. So, in this the addend and augend they are put and when you are clocking it then this is going serially out coming serially out and the way it is put the most significant bit is this side and least significant is bit is the other side. So, this here so, if it is A 7 then this is A naught so, this is the LSB and this is the MSB. Similarly this is B 7 and this is B naught so, this is the way initially it is put right.

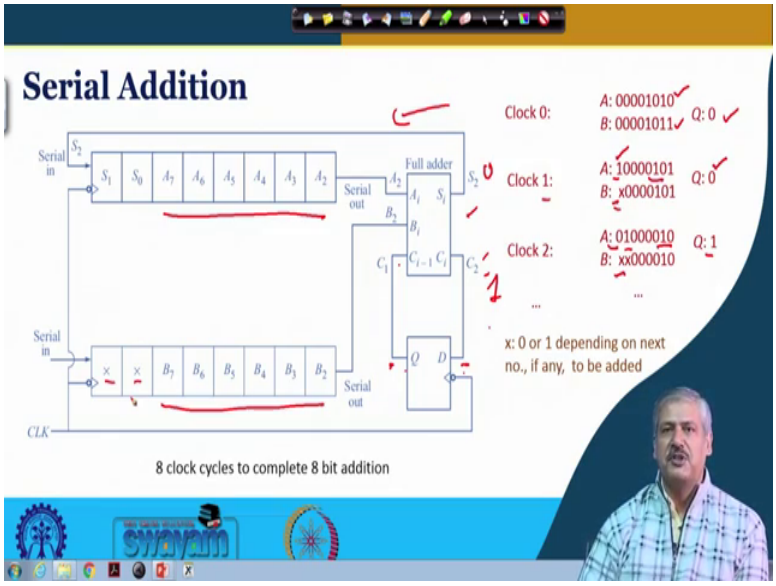
What else, so, instead of if we are talking about 8 bit addition in commutatoral logic context we have seen that eight added units are required. Here we are using only one adder unit the one that you see here right and we are using it successively in every clock cycle it is doing a one bit addition right and the carry output of it that is generated here is stored in a d flip flop and this is feedback here for use it in the next bit addition. And the sum is that is getting generated that is coming as serially in and is stored in register A.

(Refer Slide Time: 02:43)



What happens to study a register B? The serial in that is there we can put another number so, if you are looking for consecutive addition of A plus B plus C plus D more than two numbers so, we can make C enter at that time through this particular serial in ok. So, after 8 clock pulses what will happen? That A plus B result will be stored here with carry and then by that time C is already loaded here. So, with another 8 clock pulse A plus B plus C will be generated so, this is the way it will go on, is it clear.

(Refer Slide Time: 03:37)



So, we can see one example for example, we have two number say 1 40 1010 so, this is A and this is B initially loaded and Q is 0 and when after 1 clock pulse what will happen. So, 1 and 0 it is getting added so, that is 1 right. So, this 1 will be after 1 clock pulse because it is fed back here, it will come over here right since carry generated is 0 10 it is there is no carry.

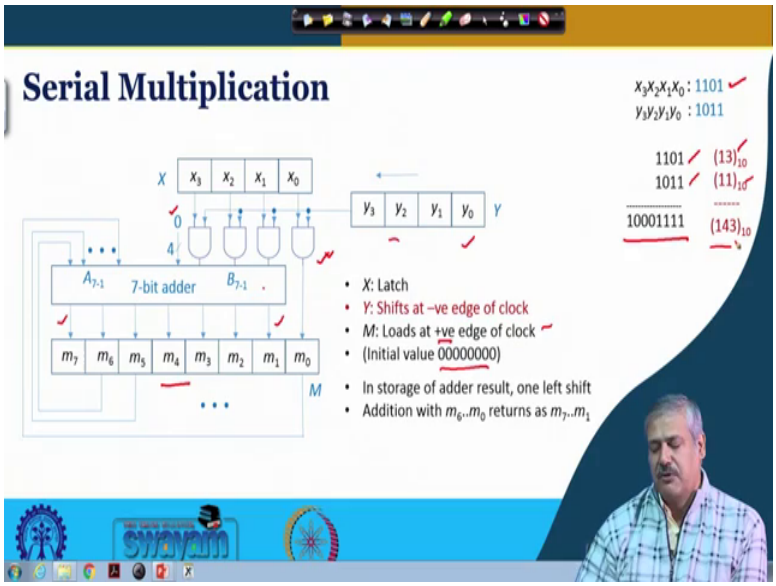
So, Q will be 0 only so, 0 will be coming here right and for B after 1 clock pulse; after 1 clock pulse the next number or some other value you know which is of now significance will come here. So, if it is the next number if it is 0 0 it will be there or 1 it will be 1, otherwise we say it is do not we do not care.

So, next clock pulse what will happen? Within this because it is a commutatoral logic 1 plus 1 which is 0 is generated and carry generated as 1. So, this is 0 and carry generated as 1 with in the same clock cycle, but we need the clock trigger for this 0 to come over here. So, next clock trigger so, this 0 that is generated will come over here this 1 will get pushed right everything will get pushed so, 1 0 is coming over here ok. Now this carry which was at this input with the trigger will come over here and it is there and for B another such input will come ok. So, this way it will keep doing it and after it clock pulses the addition will be completed is it fine.

And here we can see this snapshot of it so, A 7 to A 2, after 2 clock pulses where it is. So, B 7 to B 2 is here so, these are A 2, B 2 and C 1 that is the carry of the previous clock as previous addition, previous bit number one a one and b one addition is over here and S 2 is entering with the clock next clock 3 S 2 will enter here ok. And this two places the next number or some junk value or would not care which value are entering in this particular place right.

So, earlier we are doing it at 1 go here we require 8 clock seconds to complete addition. So, this is the difference between parallel addition using commutatoral logic and serial addition using sequential logic ok. So, there is faster, but requiring more hardware here it is slower it requires more time ok, but it is if it is the clocking is sufficiently fast then you know it will be done in real time ok.

(Refer Slide Time: 06:45)



So, next we look at serial multiplication ok, in serial multiplication what we are having? First let us get familiarized with the hardware that we are using. So, the two factors to generate the product so, one is x 3 x 2 x 1 and x naught right and the other one is y 3 y 2 y 1 and y naught right.

And in addition to that we are using a 7 bit adder, note that it is a 7 bit adder and a register which is 8 bit long. So, we are giving an example of 4 bit and 4 bit multiplication right, so it can be extended for other cases. So, this is a latch right, this is a shift register the arrow direction shows that gradually the it will be shifted ok. So, first will you know multiply with y 3 you will get the partial product which will get added, then again we will multiply with y 2 and this AND gate outputs will be giving you the bitwise multiplication y 2 multiplying these x 3 to x naught.

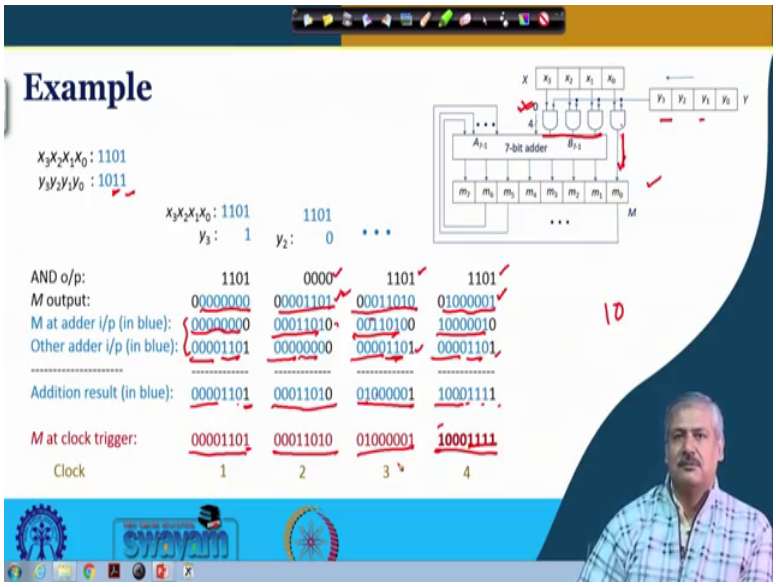
So, we shall see one example of course, right and the other important point to be noted here of course, the m initially is loaded with 0000 is that the m naught to m 6 these values after addition is coming here as m 1 to m 7 please note this part ok. So, inherently in the structure we are having a one left shift left shift occurring, the arrangement is such that the wiring is such that a left shift is occurring when we are a loading the additional result.

Now, this 7 bit addition for which 7 bits are taken from here and this end bank the first is not required we shall see an example the why it is not required. So, this 3 end output will

be going to the adder and the for the rest of the cases we will be having 0's so, that both side we are having 7 bit is it fine. So, this is the hardware part of it and these shifting and loading are not happening in the same clock edge. So, when shifting is happening not that at that time loading is happening.

So, when shift happens at the negative edge right after that this commutatoral circuit the does its operation with whatever propagation delay is required result is available next positive edge the its gets loaded. So, that if the whole thing gets completed in one clock cycle itself ok, but it is in this manner. And the example that we shall take is the one which we have seen before that is 1101 in our commutatoral logic circuit discussion for multiplier and we saw that this was the result and corresponding decimal value was 143 right we shall see that example in the next slide ok.

(Refer Slide Time: 10:21)



So, this circuit in a bit smaller form is placed it here for easier understanding and we start with this x 3 x 2 x 1 and x naught it is always there 1101 and y 3 y 2 this will be coming getting shifted over here. So, first the y 3 will be considered so, y 3 ended with 1101. So, the and output the and output over here, what will be those values? 1101 only is it ok, this end output is 1101 it's clear. M output initially is all 0's right and out of this all 0's 8 bits this one in the blue the 7 1 ok.

So, that will that are getting feedback as the adder input ok. So, this is the this 7 will be there right and other adder input is what? This one is not use this is coming directly as

for this circuit diagram you can see to the m naught and the rest three; that means, 110 ok. So, this 110 and rest four are 0; rest four are 0; rest four are 0 and this is 110 that is this 7 bits for are going for as A and B for the adder, is it clear right.

So, what is the result? So, this one is coming directly as 1 right and the other cases so, this is 0 this is 1 and this is 1 and rest are 0 and after the clock trigger what is happening, what is getting fed here, what is getting loaded here? So, this m naught this 1 is getting directly loaded here and the adder output will be coming at m 1 to m 7. So, 0000110 is it fine understood right.

So, now this is the value of m right this is the value of M it is getting shifted negative edge so, it is becoming is what is the output for the AND gate all 0 right. M output is now here whatever is the output next clock cycle in the beginning this is the output of the M and which of that is going as adder input the m naught to m 6 m naught to m 6 right. So, this is going as adder input in the blue right and to which what is getting added? This 3 bits so, these all 0's in this case and rest all 0 and this 0 is directly coming.

So, what will be the corresponding addition result? This is there addition result ok. So, what effectively you see that it is just getting shifted when you are multiplying with 10 in this particular case, it is getting the first one is getting multiplied properly and then it is just getting shifted by the for the case of the 0 by 1 in it ok. So, that is the inherent shifting that is happening we are not doing any shift register based shifting is not required and we are saving on clock cycles and also the complexity of the circuit is reduced.

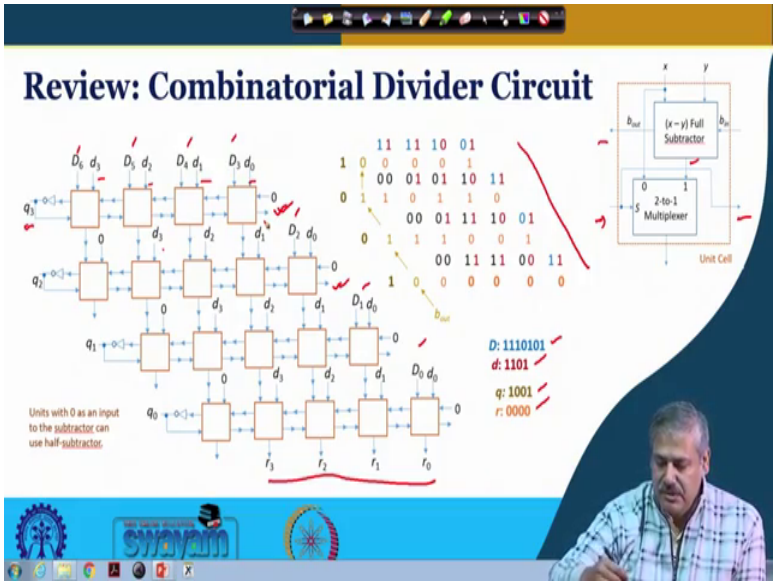
So, after that what we are getting? The value of this shift register becomes 000 this m register becomes 00011010 ok. So, 00011010 and now again it get its getting shifted so, this 1 will come y 1 will come. So, what is the AND gate output? 1101 and out of that this 1 will come directly and 110 appended with four 0's in these 7 bit adder B input and for the A input this m naught to m 6 is coming here m naught to m 6, then if you add this is 1 then 1 plus 1 is 0 1 plus that carry is 1; 0 carry is 1; 0 carry is 1 0 and then we have got a 1.

So, this is your m register which is which will be taking the value after clock 3 right. So, this will be now coming here in the next clock cycle and this is the input this 6 m naught to m 6 7 bits will go as adder input right and here again y naught is 1 so, this is 1101

right, this 1 is coming directly over here and 110 and four 0's that is participating in the addition. So, what we get this is 1; this is 1; this is 1 and this is 1 and this three 0's and 1. So, the after clock trigger this is the adder output this is the value that we get so this 1 is 128 and this four 1's 50 143 ok.

So this is the way we can see that you know we can through this partial product and shift we can get the multiplication term. So, his is the same example we had seen before you can compare the result and you can you perform it by hand also.

(Refer Slide Time: 16:29)



Now, we shall look at division so, for the division we shall just implement the circuit that we had used for commutatoral logic circuit its sequential version ok. So, for which I have brought that particular unit cell based architecture the array we had used ok. So, if you remember fine otherwise please revisit that particular lecture I believe it is lecture number 30 ok. So, week 6 last lecture, then it will be clearer to you how this division takes place. So, we have taken up this example which is there in the right hand side. So, the same example I mean we can see in that particular lecture module also.

So, the this is the dividend, this is the divisor, this is the quotient and this is a reminder and we had seen how it actually getting calculated through this and how we can arrive at this numbers. So, this dividend D 6 D 5 D 4 D 3 this bits are taken first right and then gradually D 2 D 1 D naught are added; added means it comes into the consideration is important because in next circuit sequential logic base circuit we also see that initially, D

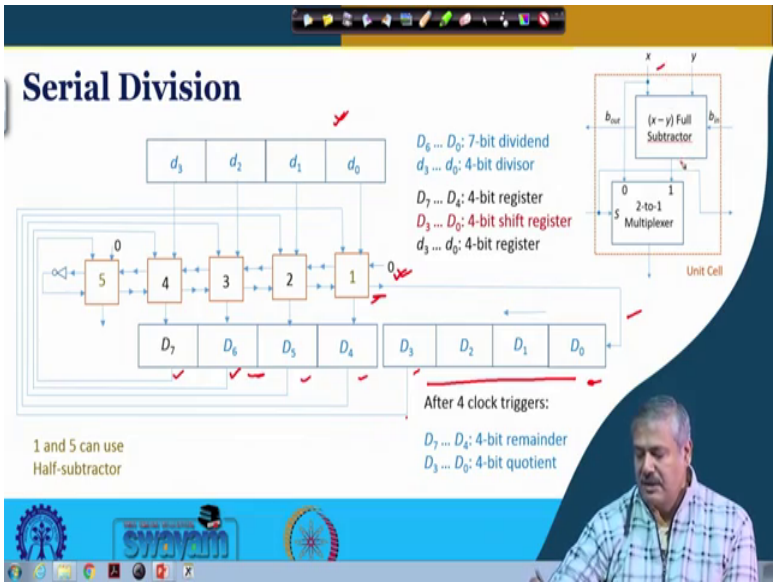
6 D 5 D 4 D 3 are there this 4 bits so, then D 2 D 1 D naught are there. So, this is 7 bit getting divided by 4 bit this is a specific example. So, you can extend it for 8 bit and other cases.

And in every case see this d 3 d 2 d 1 and d naught ok, that is participating in the subtraction process either the subtraction result is going to the output or this x is going to the output because of the multiplexing 2 to 1 multiplexer it is there. So, it is the borrow out which is residing it and borrow out inverted is the quotient that is getting generated and finally, the remainder will come over here is it and this is this are the quotients and this quotient is also available here at this sides.

So, here also you get this quotient in the right hand side also you get the quotient we had in that example we had taken it from the left hand side, but we could have taken from this output also is it fine right. So, this is what we had done in that particular class.

So, now we see how it is getting implemented there. So, D 6 D 5 D 4 D 3 we shall begin with then D2 D 1 D naught will get appended sequentially right and every time d 3 d 2 d 1 d naught will be in consideration q t3 q 2 q 1 q naught will be generated through the borrow and finally, the remainder will be there ok. So, this is the thing that we take note of and this is how the serial division takes place ok.

(Refer Slide Time: 19:51)



So, first of all you can see this D naught to D 6 that was there in addition there is another bit we have put s0, it is 8 bit register ok. So, out of this 8 bit I mean basically we are dividing into a 4 bit register parallel in parallel output kind another is a 4 bit shift register.

So, D naught to D 3 is a 4 bit shift register and this one is a parallel in parallel out register ok. And this is just a latch or a parallel out register which could be serial in parallel in depending on the required, once it is loaded it is not I mean it the output is always being used is it fine right and these are the unit cell one bank of it is only required because it is sequentially used right. So, how it works? How it implements the one that we had seen the circuit the commutatoral circuit base circuit.

So, here in this particular case so, in the beginning D 6 D 5 D 4 D 3 you see this is the output this is taken from here right and D 7 is also there first of all in the in beginning D 7 is not of consequence. So, D 6 D 5 D 4 D 3 that is being used right and it goes through that particular block and the whatever is the generated this q this output it is also generated through here so, this is made available this is this is a commutatoral logic block is not it.

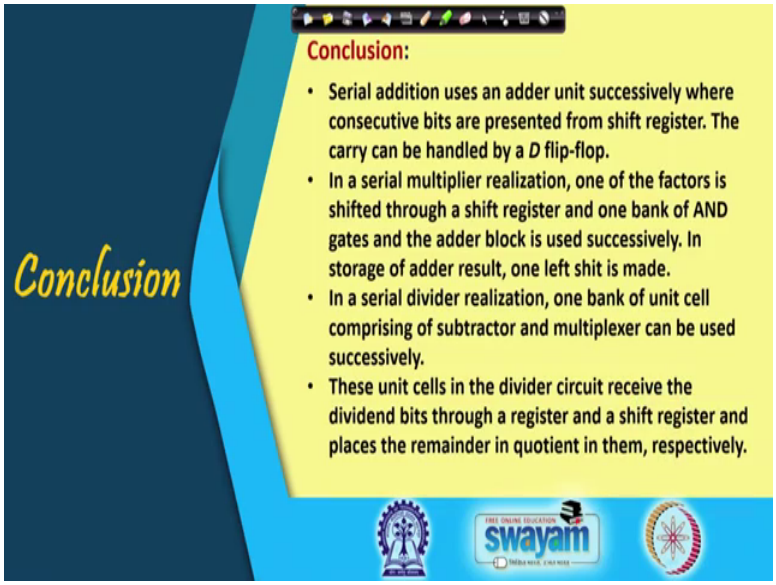
Now with the clock what happens? This quotient q naught comes over here right and D 2 takes place of D 3 right and now this D 6 D 5 D 4 and I mean what is you know the particular value that comes over here from the unit cell either this one or the subtraction value as per the logic, that will be there with that D 3 the D 2 also comes into consideration.

So, this three values 1 2 3 and D 2 comes into consideration and in each case d 3 d 2 d 1 d naught is there next time again this; this; this; this will be coming into consideration and D 1 also comes in to picture. So, this is how it progresses, same thing is implemented in the same manner the example that you had seen you can just put over here and with the clock you can see it moves from the exactly the same manner and the same result will be found at the end right.

And as we are mentioned in the previous case here also this 1 and 5 could use half subtractor instead of full subtractor. Again I say that we are following the division paradigm that you are we had used in the commutatoral logic circuit based divider circuit

discussion. So, please refer to that and then come back, here it will be easier for you to follow.

(Refer Slide Time: 23:43)



So, with this we come to the conclusion of this particular class. So, what you have seen? Serial addition uses an adder circuit adder unit successively where consecutive bits are presented from shift register and the carry part of it can be handled by a D flip flop.

In serial multiplayer realization, one of the factors is shifted through a shift register and one bank of AND gates and the adder block is used successfully. In storage of adder result one left shift is made so, the storage part is made in such a manner intelligent manner that we do not require an any separate you know shifting that we see in the multiplication process ok.

So, in serial divider realization d 1 that we had seen we have a unit cell based approach where one bank of unit cell comprising of subtractor and multiplexer is used in a successive manner ok. And these unit cells in the divider circuit receive the dividend bits through a register and shift register where shift register introduces the lower 4 bits and the remainders are stored in the higher bits that is D 7 to D 6 D 5 D 4 D y you have seen and d 3 to d 1 d 3 to d naught now is the placeholder for the quotient ok. So, with this we conclude the discussion on shift register.

Thank you.