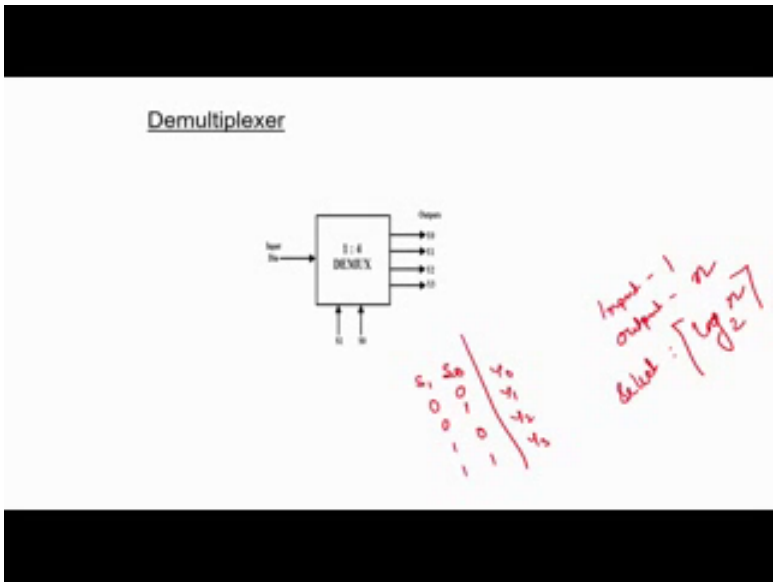
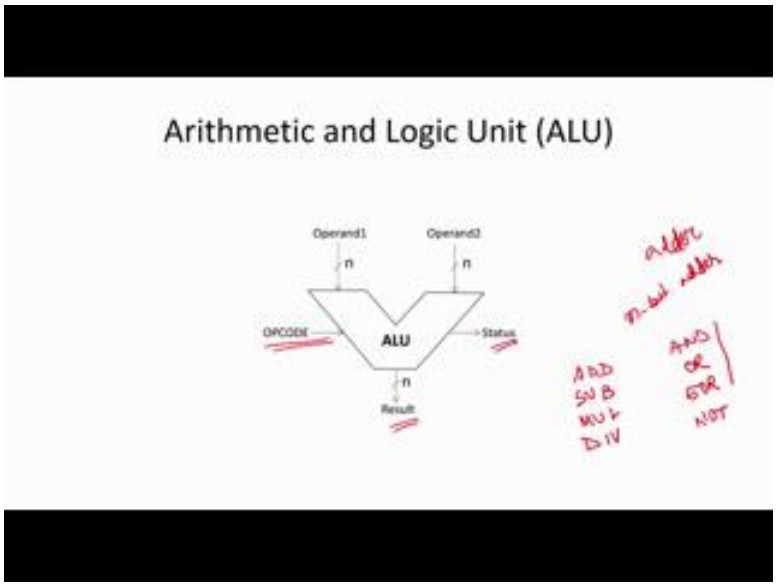
(Refer Slide Time: 40:41)



Another one we are having Demultiplexer which is the reverse of your multiplexer. So, here we are having 1 input line and we are going to transfer it to any one of those particular output line. So, if we are again I can say that input line is 1 output line n then what is how many select line we have? Again this is your $\lceil log_2 n \rceil$. So, we are having 1 input lines now we are having 2 select lines, depending on those particular select line we are going to transfer this input line to any one of those particular output lines. So, again I can say that if it is your S_0 and S_1 that we are having a 4 combination and depending on that what is the output line. So, here input will transfer to Y_0, Y_1, Y_2 and Y_3. So, we are having 1 input signals and now that will be transferred to any one of this particular output lines. So, this is demultiplexer. So, in computer we are extensively going to use those particular building blocks to construct our computers.

(Refer Slide Time: 41:53)



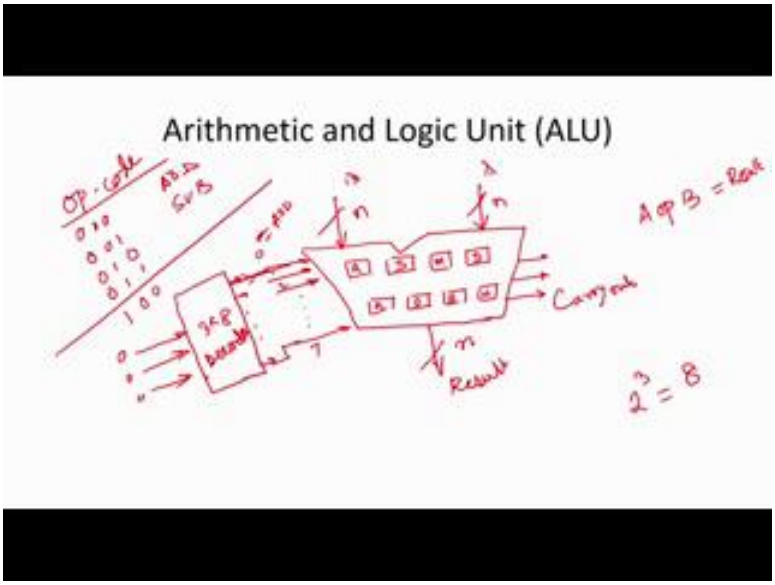
Another unit we are having called arithmetic and logic unit, ALU. This is the basic processing element inside of computer which can perform some arithmetic operation and logic operation. So, this is your block diagram, we are not going to see what is there inside this particular ALU we just say that we are having some circuit like that we having a adder circuit. If it is an n bit ALU that means both the inputs are of size n bits and result is also n bits, that means we are having an n bit header. Already we have seen how to construct an n bit adder.

Similarly what will happen we may have several operation over here. So, I consider that I am having say addition operation, I can have subtraction operation, I can have multiplication operation and say I am having division operation. So, I am having 4 processing element which can perform operation addition, subtraction, multiplication and division. So, these are the say 4 arithmetic operation we have. Along with that we may have some logical operation also logic operation also, I can say that I may have that AND operation, OR operation or maybe say XOR operation or maybe I can say another one say NOT operation. So, in this particular logic operation AND, OR, XOR are your binary operation, but not is an unary operation because it is going to invert one of the inputs and going to get the output as the invert of this particular input.

Now, you just see that like that this is a processing element and I am having 4 addition 4 arithmetic processing element and 4 logic operation at any point of time you are going to perform one of these particular operation. Now, what operation we are going to perform that

will be given by these particular signals. So, depending on that signal we are going to get the result and along with that we are going to get some status also, some of the status bit will be given or some more additional control signals we are going to get and that will be set or reset according to the behaviour of this particular circuit. So, this is ALU is the basic building block or is a basic processing element that we are going to use in our computer to perform some of the arithmetic operation as well as some logic operation.

(Refer Slide Time: 44:30)



So, now just see that how we are going to construct it. Now if I say that this is the ALU I am having these 2 input and input n bit input and, say this is basically I can say that this is A, this is B and here I am going to get n bit result and say that I am having those particular processing element these are the four addition, subtraction, multiplication and division similarly 4 logic operation that say AND, OR, XOR and say NOT.

Now, at any point of time we are going to give 2 inputs over here A and B, and we are going to perform 1 operation and depending on the operation we are going to get our result. Now, how we are going to select this particular operation? Now, we are talking about in the previous slide we have seen that we are going to get output and here we are having some status line one of the status line maybe I can say about carry out like that we can have several status line. Now, select any one of this particular input operation we need the appropriate signals, for that what will happen I can use 8 different signal. So, this is your 0, 1, 2, 3 like that up to 7. So, you can say that when I am giving 0 signal as high than you can say that this is going to perform the

operation addition. So, in that particular case what will happen? This addition circuit will be selected and both the input will be diverted the adder circuit and we are going to get the result; that means, we need 8 control signal to select any one of these circuits.

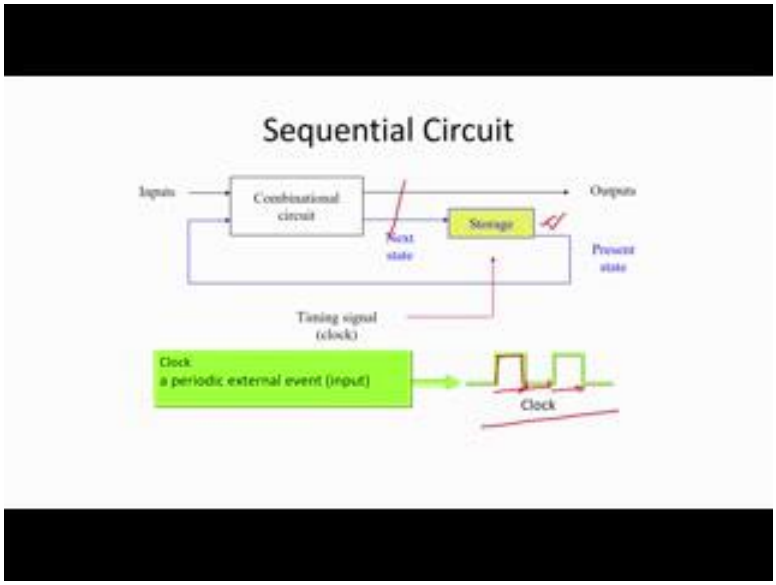
But we can reduce the in signal over here instead of 8 signals what will happen you can control it with the help 3 signals only because we know that $2^3 = 8$, so if I am going to use 3 signals then we can generate this 8 possible combination because this 3 comb signals are going to have 8 different possible combinations. So, in that particular case what we can do, we are going to use a 3 by 8 decoder; that means, we are having 3 inputs and we are having 8 output lines and those output lines will be connected to those particular select signals of this ALU.

Now, when I am giving say all 0 then what will happen? This is the 0 then this signal will be high; that means, you are going to select the adder circuit. So, this is the way I am going to have 3 bit code and this 3 bit codes is known as my opcode which is your operation code. So if it is your 0 I can say that this is your ADD when it is your 001 I can say this is your SUB like that, so 010 and 011. So, these are the arithmetic operation. So, when this third bit I am going to put as 1, then I can say that I am going to use those particular logic operation. So, now, say since with the help of 3 bit I am having 8 different combinations so we can use a 3 by 8 decoder to get select the appropriate operation. So, this is the way we are going use our arithmetic and logic unit in our computer and what operation we are going to perform that will given by the opcode which is a binary code, depending on the combination of the input signal we are going to select one of the operation and this ALU is going to perform that particular operation. So, we are going to use an ALU in our computer. So, we can have 8 bit ALU which can perform operation of 8 bit numbers we may have 16 bit ALU which can perform operation of 16 bit numbers.

Now, I have already mentioned that we are having two types of digital logic circuit, one is your combinational circuit and another one is sequential circuit. I have briefly give idea about the combinational circuit which we are going to use while constructing the digital computer. But we may have many more other circuitry also, but in this particular course we are not going to discuss, but just giving some idea. Now, whatever circuit you are going to have I think you can analyze it and you are going to feel get an feeling how that particular circuit will be constructed or implemented. Now, second class of circuit is your sequential circuit. So, in case of sequential circuit already I have mentioned that the current output depends on some previous output also so that's why it is a combination of your combinational circuit as well as some storage. Say

this is a combinational part of my sequential circuit depending on input we are going to get the output and some of the output is going to act as an input for the next time.

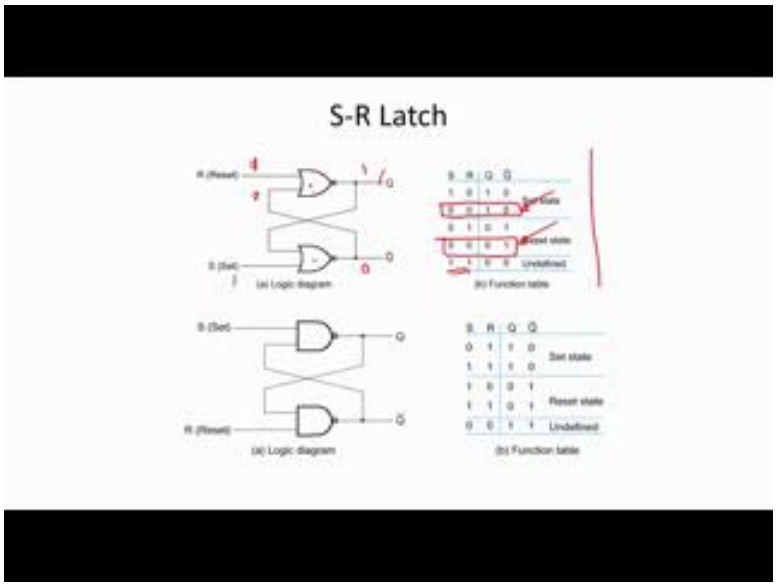
(Refer Slide Time: 49:34)



So, those output we need to retain it and we are going to say this is the some sort of storage element we are having. Now, this storage will come whatever we have stored or retained that will come as an input to the circuit and the next output will depends on the current input as well as the previous output. Now, when we are talking about this thing say current output previous output; that means, it is related to time. So, this time will be maintained by a timing signal which is we call this is the clock signal. So, in every clock signal, say whenever this clock is coming then at that particular point this circuit is going to perform its operation. We are having the input signal, we have the previous output, now this circuit is going to works when this particular clock is high and accordingly it is going to give me output and when this clock is low during this time this circuit is not going to be high.

So, we can just think in that particular way. So, when this is the 1 step this is the second step. Now, this output of this particular present step depends on the output of the previous step. So, it will be controlled by a clock so for that we need the storage. So, when we are going to look for a sequential circuit basically we have to see how we are going to retain the information; that means, the storage element.

(Refer Slide Time: 51:24)



So, for that we are having a storage element called S-R latch, S stands for Set, R stands for Reset set and reset. So, it will be implemented with the help of your NAND gate this is the NAND implementation and this is a NOR implementation.

Now, how it is going to behave just I am going to explain in one of this particular table. Say when S = 0 and R = 0 then I am having two output Q and $\bar{Q}$, one is the complement of the other. So, S means set when S =1 it is going to set the output to 1 Q to 1 and then $\bar{Q}$ = 0 and when 01 combination is here then it is going to reset this particular circuitry and we are going to have this as 0. And now, when 0 and 0 then what will happen what is the output over here. Now, here you just see that I am having 2 combination 00 is your 10 and 00 is your 01. So, see output varies when I am giving both input as 00 in some situation I am going to have 10 and in some situation we are going to have 01. So, it depends on my previous configuration. So if after 10 combination if I am giving I am giving 00 then what will happen it is going to retain the previous output 10 will be retain over here; that means, I set it and after the I keep the signal as 00 then I am going to retain this particular output forever.

Again when I am going to give 01 I am resetting it. So, we are getting 01 output after that if I am giving 01 then what will happen we are going to retain this particular information. So, this is basically the present output depends on my previous output so that is the way we are going to retain our previous information. So, this is the single bit S R latch. So, we can go for more number of bits. So, we are just going to replicate these things. Now, we just see how it is going

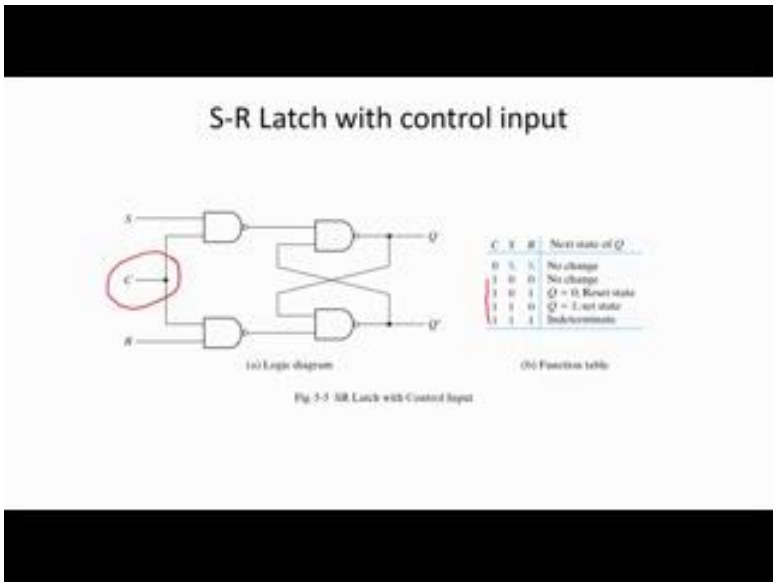
to work. When it is your 10 say I am giving 1 over here. So, as soon as I am giving 1 one of the input is your high then what will happen OR gate is going to give me output high and NOR gate is going to give to me output 0. So, it will be 0. Since this 0 is fed back to this particular point and say reset I am going to put a 0. So, when both are 0 then Q will be one. So, this is your 10.

Similarly you can analyze what will happen in case of your 01 also and in case of 00 it is going to retain the previous input you can see when I put both are 00 it depends on my this output. So, this is S-R latch, but here we are having one problem when we give 11 combination.

So, it is we are saying that one is the complement of other because if we give both as 11 then what will happen, it is going to get 00. So, it is undefined we are saying, but we don't have any problem. But after 11 if we give 10 then we are going to get 10, if you give 01 then we are going to get 01, but what will happen after 11 if you are going to get 0 and 00 because in that particular case 00 may turn up to be a 10 or 00 may turn up to be 01, so what we are going to get?

So, this is the problem and we say this is the race condition because why we will say race condition we are using two NOR gate over here and every gate are having some propagation delay, but it is not possible to fabricate to get with the same propagation delay here will be an fraction of differences, one will be slightly faster than the other. So, after 11 whether we are going to get 10 or 01 it depends on the propagation delay of these 2 gates whichever is faster it is going to advance and accordingly it is going to set the output so that's why we say this is the race condition and we should avoid this particular 11 combination otherwise we don't have any problem.

(Refer Slide Time: 55:55)



So, to avoid these things what will happen? We are going to have a S-R latch with control input, here we are going to put an control input and this circuit is going to work when this control input is 1, when it is 0 that whole circuit is not going to perform there will be no sense; that means, whatever information we have it is going to retain over here if Q is 0 it will remain as 0.

Now, what will happen? Now, in that particular case whenever you want to set it then you are going to give 01 over here, whenever you want to reset it I am going to give 10 over here and if I am going to give 00 then it is going to retain it. But when I am going to apply clock at that particular point we will try to avoid this giving 1 and 1 combination at S and R, that is the way we can control it with the help of this control signal.