Digital Circuits and Systems
Prof. S. Srinivasan
Department of Electrical Engineering
Indian Institute of Technology, Madras
Lecture - 28
MSI and LSI Based Design

(Refer Slide Time: 1:40)

Digital Circuits and Systems

- * Introduction to MSI and LSI Based Design
- * Multiplexers
- * Combinational Design using Multiplexers

Lecture #27
Prof. S. Srinivasan
Department of Electrical Engineering

Starting today we will see some designs using Medium Scale and Large Scale Integrated circuits. First we will look at the combinational circuits and then include the sequential part of it. I have mentioned this several times earlier in the course that ultimately the success of the design depends on how efficiently we are able to make a design for a given set of specifications in terms of the size and in terms of the power consumption and in terms of the cost. So if you want to have a design, of course the types of design, the size of the designs we have seen so far in the class are small so a gate based approach is justifiable. But when the circuits become larger and larger the system becomes larger and larger, it becomes extremely very large, extensive in order to have large number of gates. So with large number of gates come things like reliability problems, interconnectivity between gates, external interconnection is unreliable compared to the integrated circuits.

Of course you can make very sturdy and rugged printer circuit boards but still a circuit which is fabricated as an integrated circuit will have much more reliability in terms of failure compared to a circuit which you put together outside. Then the size of an integrated circuit package as you see it there has to be minimum size of package for you to handle it so that you can put it in a board and you can wire it. Even though we are talking of micron technology and submicron technology that means we are talking about dimensions of transition being in the order of microns and submicrons you cannot handle them. If I give you a small piece of transistor fabricated using a micron technology how will you handle it, you cannot so you need to put it in a package and the package has to be hermetically sealed for protection from environment but you need to connect it properly.

So in IC the cost of the extra circuit that you put in a chip is not proportional to the final cost of the circuit. That means there is a minimum cost in the fabrication process in terms of the infrastructure, in terms of the technology, clean rooms, lithographic techniques, packaging, in terms of reliability, in terms of testing this is a very high component of the total cost of an IC. So to give you an example a circuit with about ten gates and as I said earlier a ten gates circuit does not mean there are ten gates inside but a circuit whose performance is equivalent to the performance of a circuit with ten gates is sold let us say for ten rupees it does not mean that I make a circuit with hundred gates using the same technology same processing same packaging it does not mean it will cost hundred rupees. That is the cost of the circuit is not proportional to the complexity of the circuit.

There is an incremental cost increase a small increase in the cost because of the extra design effort but I don't think you will have to account for chemicals that we use as extra in the process, the wafer size is as I said is the same, you cannot handle less than that, the package size may be same, the packaging technology may be same so instead of ((trans-)) may be you can sell it at 12 rupees so this is dramatic, it is not an insignificant factor we are talking about here. So, as the circuit becomes larger and larger the cost per function, if you want to define the cost per function it drastically reduces.

Take up my memory chips. Why we are able to get memory chips in computers with a huge capacity for a much smaller price because it’s 1 IC we are able to get it. Suppose a memory of particular size let us say one kilo bytes cost a certain amount, earlier if you wanted to make a 64 Kb you need to buy 64 of those memory chips but today we can make a 64 Kb memory the cost will not be sixty-four times what it was but of course may be may be twice or even let us say five times I don't know depends on the technology, design effort, the testing effort and extra changes you have to make in the processing steps. But you get the point it is not sixty-four times it maybe at the most three or four times that accounts for a very drastic reduction in the cost of electronic circuits today. We have computers which are always costing less than what it was earlier for a given performance.

Is it probably only one of the very few areas of human activity where the cost goes down any other area be it agriculture, be it textiles, be it manufacturing or whatever the cost will be proportional to the effort and also you have to take into account inflation and everything. But here today I can buy an electronics hardware which cost less from last year for the same function or for a given cost of the circuit this year I can buy a circuit with much better performance compared to what I could have bought for the same price last year that is because of the advancement in technology.

Why I am saying all this is because now the time has come for us to study from gate approach. Even though the gate approach is required for us to design, to understand the problem, to minimize and then see how it functions, functional understanding of the problem is very important. But I would rather use ICs which are larger because there are a very few of them compared to a large number of gates for a given performance, given set of specifications. If I can design using five gates five ICs instead of hundred ICs I would definitely prefer it for the reasons I mentioned earlier like the cost, size that is going to occupy in the PCB.

Finally when you make a chip it has to go as a board which gets into your computer and other electronic devices and appliances. Especially when it is space application and things like that the size is of major importance so I can put in a smaller area the performance of the circuit, function system compared to if I had to do the whole thing using functions of less complexity so size reduction, cost reduction is what we already talked about.

And in power reduction because power is not again proportional to function there is lots of power which is wasted in switching activity in distribution of the power rails along the ICs power of course is a proportional function to a certain extent and the switching activity to certain extent but there is also some saving in power consumption when the number of ICs are small. For example, with large number of ICs they all get switched on and switched off whether they are used or not so there may be wastage of power dissipation. And finally and most importantly it is the reliability.

As I said earlier if I can connect something together with a reliable connection a connection which is very very tight where the practical probability of failure is very very small, remote, almost nil that is assured only in the processing because it’s a chemical processing. All other processing such as lithography and implantation techniques is not a mechanical technology where you put and fasten things together where sometimes the coupling goes or the screw gets weakened or even in electronic circuitry where we put external circuits ICs and connect them together by either wires or by printer circuit board channels. Here it is all bonded together as the circuit is made, as the circuit is fabricated.

That means if once the IC is tested and functioned there is a very very remote possibility that it will fail. It fails occasionally because of a condition which is not tested as occurred in the operation of the circuit or it could be a radiation of some other aspect which has not been considered earlier. Otherwise you connect a wrong power supply and then pump extra current much more than what it can handle. So radiation and then driving with voltages much higher than what you have designed the circuit for are not considered as failure because of the circuit.

Occasionally there may be a circuit failure because of a condition which has not been tested earlier for that condition you have not tested. So assume that you work for that condition when the condition really happened.

Few years ago you might have been in high school at that time there was a major problem on Pentium. The Pentium processors, microprocessor by Intel which is used in all these computers now go through a very elaborate design procedure and testing procedure and they suddenly reported problems of failure where the circuit didn’t work it was found that at a rare combination of certain functions which was not tested had occurred by and it appeared that it is not a very infrequent thing it could occur very frequently in some computation so what Intel did was to recall all those chips and then they made the corrections in the designs and reissued those. These things are very rare.

Otherwise what I am trying to say is reliability is very very high when you do integrated circuit rather than connecting circuits discretely outside the chip. Because of these reasons we have to go in for ICs which can do more than a simple gate function but what about the price you pay for it, the price you pay for it you can never optimize. For example if I give you a circuit problem a design problem using combinational or even sequential logic you will always optimize for the minimum number of gates using Karnaugh Maps and then say this is best possible solution but then here I don't have gates, I may have come up with a gate solution but where are those gates, I am asking you to connect something which is not gate, there are components these components will have equivalent gate functions but they are not the gates.

For example, suppose I require about half a dozen AND gates and few OR gates and inverters I may have an IC which may not exactly have those combinations so I should be able to map on the given hardware my function. My function that I want to implement I should be able to map. Of course when I do the mapping I will do it the most efficient way possible but it will not be the efficient way in terms of the gate count. I may be spending more functional equivalence of gates so I will call it loosely more gates. I may end up using more gates to implement a function compared to what I could have done I would have done if I had all the gates available.

But that is the price you pay but that doesn’t matter because I am now using smaller number of ICs which is going to be cost effective, which is going to be of speed of high speed and is also going to be of less power and of small area but what does it matter.

In fact if I am going to use all of them all the parameters I am interested in monitoring are going to be better in my design how does it matter if you are going to use minimum number of gates, it doesn’t matter. So the mapping of the hardware mapping of the given problem when hardware is available may not be the most efficient way of doing it but you will have to do it by experience. Sometimes there are some tools or procedures available for this but not always, but for most of them it is experience or they call it heuristic, intuitive, it’s an intuitive design, this is what is required, this is what is the available component so this is best way to connect it.

As I said sometimes there is a procedure or there may not be procedures but still you can. Once you have experience of designing you will have a feel for it. If you connect it this way you get a better performance compared to if you connect it this way or sometimes you may have to do trial and error to find the best possible solution to that, it’s also fine with us so this approach is called intuitive approach or people even call it heuristic I don't know whether you have heard of this term heuristic approach rather than a fixed procedure. Of course procedure may be there but this may not be the efficient procedure and always procedure may not be there.

For a simpler IC there may be procedures. Again as you go higher and higher into ICs if I have a gate count of thousand and above one thousand gate function my complexity of the circuit design is also about one thousand gates I am trying to map this functional requirement into the hardware available, I have a hardware which is thousand gate equivalent, I have a function which is thousand gate equivalent I have to map it, I have to do a best possible way, of course I can't do it manually I have to depend on the software designers limitation whatever he or she knows about it and that person may not be the person who is knowledgeable in the particular application you are doing so you may get a generalized efficient way but when you do it yourself by hand you may probably do it better.

But these problems are very very insignificant compared to the advantages in terms of the cost saving, power saving, improvement of performance and size saving. So with this little introduction we will talk about combinational logic first and later on we will do sequential.

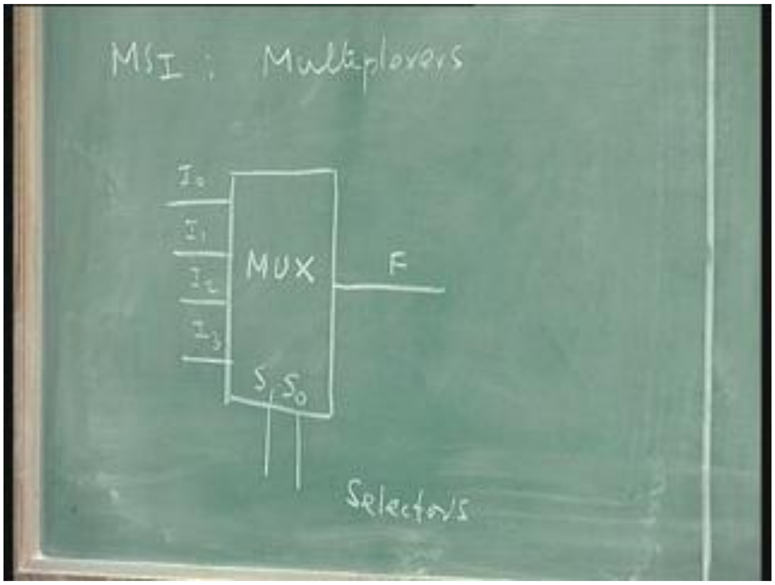
Combinational Logic Design using MSI and LSI circuits, Medium Scale Integrated and Large Scale Integrated Circuits so it's always possible to map everything is gates when it is a full adder, a 4-bit adder, whether it's a decoder, whether it's a priority encoder, the parity checker, code converter is all gates and what we have to optimize also gates.

So this is sort of a common sense it tells us that these gates are required and these gates are available in this hardware piece I am giving you. I am giving you hardware piece in which there are gate functional equivalents and I am trying to design a circuit with gate functional equivalents so how best can I map it and it's possible the best possible way that may not be available to you because inputs and outputs should be available you should be able to feed in things and you must be able to take them out so you may come up with a best scheme but it may not be possible because there is no input pin for that you cannot break open an IC and put a signal inside that so you will have to work with the inputs and outputs which are available to you.

So first we will talk about MSIs and the first MSI we will talk about will be multiplexers. And then we will also talk about decoders and then we will go to LSI and we will talk about proms and programmable logic arrays and then probably go to sequential circuit with counters and shift registers with which we all are familiar but how to use them in a design.

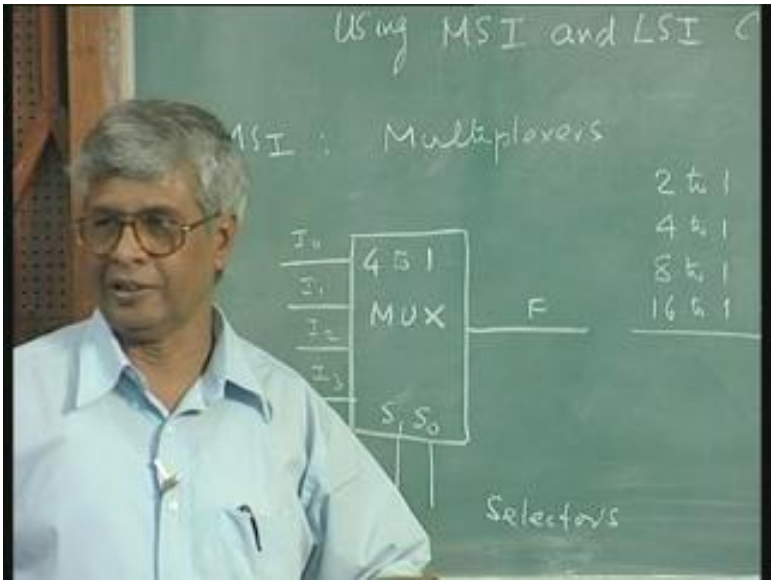
What is a multiplexer basically? Multiplexer, as the name suggests it multiplexes. That means I have several inputs and you have to select one of them so it is a selector, a multiplexer is nothing but a selector. So, if I have hardware which IC all multiplexer MUX in short, now we have several inputs let us say I_0 I_1 I_2 I_3 are the three inputs and at any given time I want one of these outputs I want F to be either I_0 or I_1 or I_2 or I_3 based on my requirement so I need to connect one of these inputs to the output as required. I need some control to know when do I connect I_3, when do I connect I_2 and so on. That means I need signals which are called control signals or selectors and number of selectors will depend on the number of inputs.

(Refer Slide Time: 22:04)



We have four inputs and one of them has to be output so how many selectors we need, we need two selectors because two selectors can have four combinations and each one of those combinations can connect the given input to the output. So I will call these selectors S_1 and S_0 so S_0 and S_1 the selector signals and I_0 I_1 I_2 I_3 are the input signals and F is the output signal so the description of the multiplexer is F is equal to S_1 bar S_0 bar I_0 plus S_1 bar S_0 I_1 plus S_1 S_0 bar I_2 plus S_1 S_3 I_3. That means if I make both these inputs 0 0 both these selector inputs 0 0 F gets connected to I_0. So it's a selection process that's all. I have several signals coming into the input and one of them as to be connected to the output and which one should be connected depends on the combination of the S_1 and S_0 so if it is 0 0 I_0 will go through, for 0 1 I_1 will go through, for 1_0 I_2 will go through, and for 1_1 I_3 will go through.

(Refer Slide Time: 24:30)



Can you think of an application for this type of circuit?

Telephone exchange is a right example.

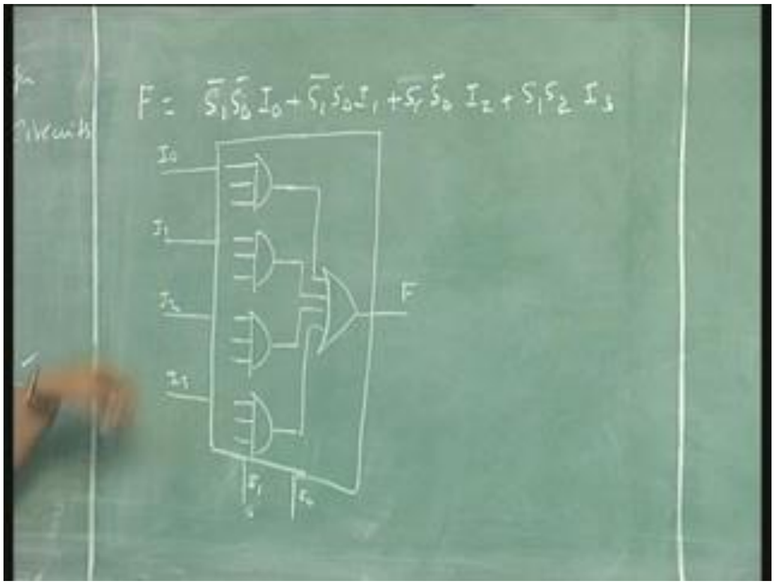
As an example I have only one of the four subscribers and only one line is connected. This may be your local exchange and this may be my local exchange, there are two villages, of course I am making the problem too trivial but just to give you an idea. So I have four subscribers in this town and four of the subscribers in this town and only one person can talk at a time so I have only one line coming from this town to this town so depending on which person is dialing that person will have a code and that signal will go through and on the other side we have what is known as the reverse of this operation naturally it will be called a de-multiplexer when we have a multiplexer.

So I will have one input and four outputs and which one is connected depends on the selection you make. It’s an example of a selection. There are several occasions in digital signals. In digital system design where there may be several signals I may have to select one out of those signals to be applied to a particular circuit. So there are several signals and only one of those you have to select to be applied to a particular circuit you use a multiplexer. Now this F is a combinational logic.

So, if I don't have this IC I can go to the electronic store and buy this, ask for a multiplexer. Of course you can’t say give me a multiplexer you have to tell him how many inputs and how many outputs you require. So it’s called 4 to 1 multiplexer, you don't have to give how many selectors because it is fixed and since this is a 4 to 1 so naturally its selector is two. Or I can have two to one, I can have two to one MUX, I can have 4 to 1 Mux, I can have 8 to 1 MUX and have sixteen to one MUX and beyond that you have to make your own Muxes that is possible having a smaller MUX you can build a larger MUX. You can connect a tree of multiplexers to realize larger Muxes.

Suppose you are a die hard gate user I don't want anything other than gates that’s all I can use so please don't give me a multiplexer then can we build one? Of course because this is the function and you know how to implement this function so how many gates do we need for this? We need four AND gates with three inputs each and one OR gate with four inputs and then I need to have two inverters for S_1 and S_0. So I can have for example:

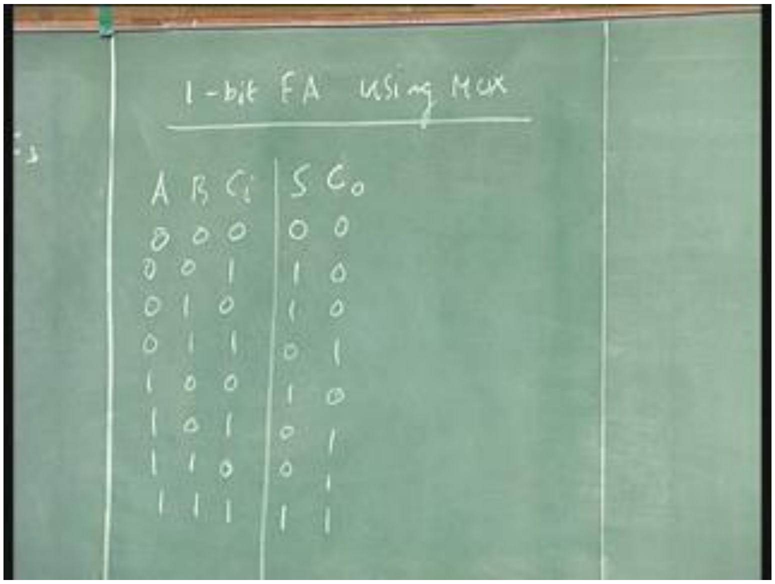
(Refer Slide Time: 27:04)



So, for this gate for example I need to feed this, this S_1 becomes an inverter, this S_2 is an inverter so like that we have to do, you can complete the drawing. But instead of buying these four AND gates with three inputs each two inverters and one OR gate which will be how many ICs? Because AND gate and OR gate will not exist in the same IC so naturally I need to buy one IC for this, and more than one AND gate can exist in one IC but then generally it is 4 two input gates or 3 three input gates so I may have to buy two of those, 2 three input AND gates I need to buy it’s a package usually like an inverter doesn’t come in single, I can’t buy a single inverter instead I have to buy six inverters, one package of inverter will consist of six inverters, two input NAND gates will contain four of them, quad package.

Similarly, AND gates will come with two inputs and one output and four of them in one package but then there are three inputs so I need to buy two of them so two packages of three input AND gates, one package of OR gates four input OR gates and one package of inverters I need so how many ICs so total four here it is 1, this is MSI, gate function wise it is 1 2 3 4 5 but then you don't count like this, the two inputs and one output is considered as a gate function so I may have to count as 1 2 3 4 5 and add some more numbers (29:00) for more inputs and inverters everything so it is about ten gate equivalent approximately, don't worry too much about these calculations, I am not going to ask you to find out the calculation equivalent AND gates for a gate function for a given function and anything like that but about ten gates put together in this one package it does the same job with all the advantages listed earlier in this lecture about the size being small and cost being small so naturally I will go for this rather than this. This is the concept of medium scale circuit integrated design. So now having decided to go in for this Muxes than gates how do I use it in a design procedure? The same combinational logic design which I know how to do exactly in an optimized way, most efficient way saving the number of gates using Karnaugh Maps provided you are asked to use only gates now how am I going to modify my design procedure to do the same problem using multiplexers. So let us do a simple example of a full adder.

(Refer Slide Time: 31:10)



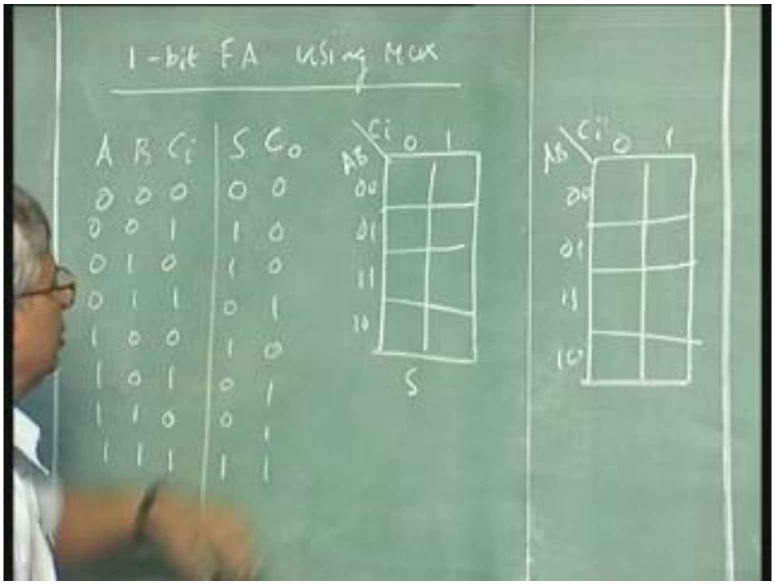
1-bit FA using MUX

A	B	C_i	S	C_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

1-bit full adder using MUX:

We are going to design using MUX one bit full adder and all of you know the truth table. AB are inputs and C_i is the carry then S sum and C_o carry out so it is (refer Slide Time: 31:07) and sum is 0 1 1 0 1 0 0 0 1 0 0 1and the carry would be 31:30). Now let me draw the Karnaugh Map for this just to show you how it is going to be placed or I will draw it this way.

(Refer Slide Time: 33:02)



1-bit FA using MUX

A	B	C_i	S	C_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

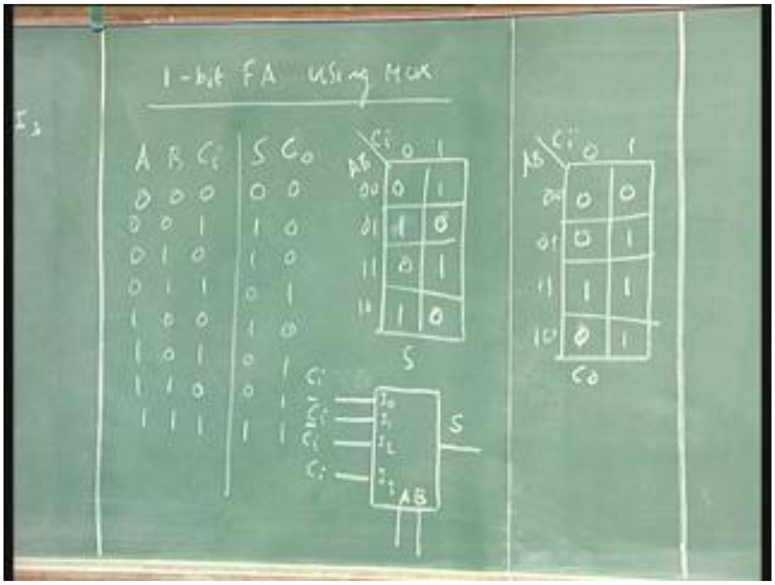
So quickly mapping this and this it is 1 0 1 0 0 1 and on this it is 0 0 0 1 1 1. This is the carry output and this is the sum. Now as I said it's not an intuitive design as I said a heuristic design of course it is all that it is a very straight forward problem but later as the complexity of the circuit becomes bigger and bigger and also it is not this symmetrical then you will have more of thinking to do. But in this case there is a thinking step but that thinking step is very very obvious and straight forward. So I would like to use a 4 to 1 multiplexer I need to select two of these variables as selectors. I have three variables AB and C_i and out of these I have to select two of them as selector variables and when these selector variables are 0 0 what should be given as C_i so that the output is what is required according to the table.

So what I am going to do is I am going to use a 4 to 1 multiplexer I_0 I_1 I_2 I_3 and I am going to get a sum out of this multiplexer and the two selectors I am going to use, my two bits A and B. Now when AB is 0 0 you know that S is equal to I_0. We know from this equation or from whatever that is here that when AB is 0 0 which is S_{10} S_{00} the output S is connected to I_0. So if I make I_0 what is required according to this Karnaugh Map then that will be implemented. Thus when AB is 0 0 the sum should be same as C_i, the sum should be 0 if C_i is 0 sum should be 1 if C_i is 1. So if I connect C_i to this if C_i is 1 when AB is 0 0 the output will be 1. When C_i is 0 output will be 0 and sum will be 0 when AB is 0 0 and when C_i is 1 when AB is 0 0 and sum will be 1. That's what it is mapping the hardware. It is mapping on the given hardware our requirement which is here. But what is so intuitive about it is I chose this A and B as the selector variables and found out that all these can be done this way.

I could also do in some other way. Now when AB is 0 1 output will be 1 when C_i is 0 and 0 when C_i is 1 so I need to connect C_i bar. Likewise for this 1 0 C_i bar (37:02) and when AB is 1 1 sum is 0 if C_i is 0 and sum is 1 if C_i is 1 so I need to connect C_i.

So with a simple 4 to 1 Mux and only one gate for inverting C_i I can get my sum whereas if I had used this I would use 2 two input exclusive OR gates, that is what is the efficiency about this.

(Refer Slide Time: 37:25)



1-bit FA using MUX

A	B	C_i	S	C_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

AB \ C_i	0	1
00	0	1
01	1	0
11	0	1
10	1	0

S

AB \ C_i	0	1
00	0	0
01	0	1
11	1	1
10	0	1

C_o

Coming to the carry again there are four inputs so I will use AB as the selectors and this MUX is a 4 to 1 Mux, this and this are 4 to 1 Muxes. Though in this output I will get carry and from this map I will do this. So I am now mapping onto this given hardware namely a 4 to 1 multiplexer, it is a hardware mapping and not mapping like drawing, a hardware mapping and a given 4 to 1 Mux the given Karnaugh Map to get the function implemented. Now in AB 0 0 output is 0, there is no carry irrespective of whether C_i is 0 or C_i is 1 so I put a 0 here carry is always 0 if AB is 0 0.

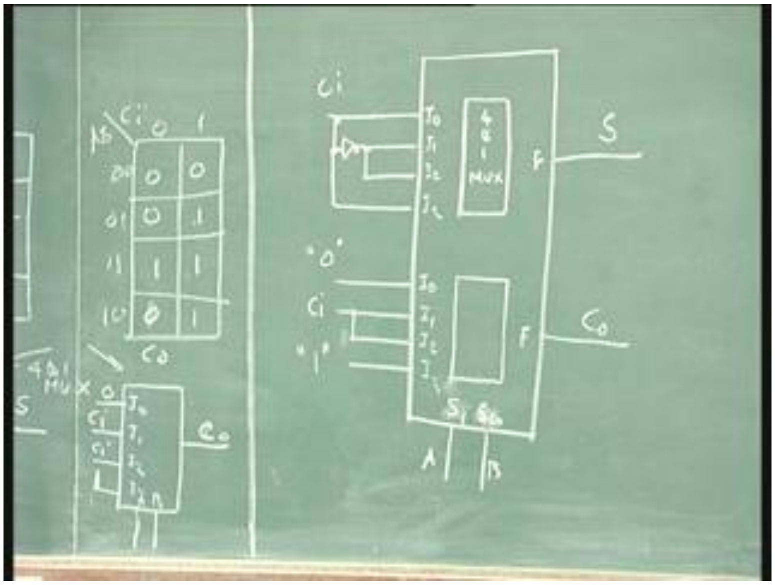
If AB is 0 1 carry will be 1 if C_i is 1 so I put C_i here, if it is 1 0 again C_i and if AB is 1 1 whether C_i is 0 or C_i is 1 and we have carry out. So, carry out is independent of C_i and it always has to be 1 so I need to connect a 1 here. So with two 4 to 1 Muxes I got my full adder and the beauty about the whole thing is I was just now telling you that these things come as packages, 4 two input AND gates come as a package, 4 two input NAND gates come as a package, 6 inverters come as a package likewise 2 four input multiplexers come as a single package. I can nowhere go and find a four input MUX individually. When I go and buy a 4 to 1 Mux I will always get a 2 in 1 package.

That means in one IC package I am getting a full adder as against the number of gates we would have used in the other approach so that is what is called MSI based design. So I will have this full adder which is going to look like this two of them in one is a dual MUX. In first MUX I will connect C_i and for second and third I will connect Ci bar. This is I0 I1 I2 I3 corresponding to the first MUX sum. This is 4 to 1 Mux (Refer Slide Time: 41:09).

Now we have another 4 to 1 Mux here. This will be connected to 0 that means ground and we get the ground potential a 0 potential and this is a 1 means it is the V_{cc} whether it is five volts or three point three volts whatever the operating voltage is that voltage will be given here and these two are connected to C_i. And only in the second gate or second MUX these are the inputs and these are the outputs and this is my carry out. And to the two selectors we connect two bits A and B which would be connected internally.

I am not showing the internal connections.

(Refer Slide Time: 42:20)



We are having only these things available to us so I will connect A and B here. So internally this would be taken here, this would be taken here, this would be taken here, this would be taken here, this will go here and this will go here (42:40). Internal connections I cannot make but it is there. So in a single 4 to 1 piece I mean single dual 4 to 1 Mux I am getting a full adder so it is a single chip solution of a full adder.

So this tells you there are ICs available which are larger in function, functionally larger compared to the simple gates and these will be used in design in combinational logic. Even in sequential logic the combinational logic is the steering logic, the steering logic is the combinational logic so we need combinational logic. Whether it is a combinational logic design or a sequential logic design it doesn't matter. So we use these ICs which are medium scale ICs which have several gate functions in it the design becomes simpler but a sort of extra effort is required. It is an effort in terms of properly identifying the requirements of the chip.

For example, how will you first of all decide even, if I didn't tell you use MUX they say use an MSI and get a 1-bit full adder so you should know how to do it? Of course with other ICs also you can do. I can get decoder using other ICs also. So first thing is make an intelligent choice of the MSI, look at the specifications of the requirement, look at the capabilities of the hardware available try to make a match to the extent possible and this is very symmetric and straight forward case. Sometimes you may have to do little extra sort of a manipulation and in that process we may end up with a less than efficient ef…..44:40) but it is worth it because we are going to reduce the total number of ICs.

This only an example but you can try several other examples because anything that you can do with a four variable Karnaugh Map ABCD. We have done this earlier using gates and now you can go ahead and do it using multiplexers. This is a three input Karnaugh Map I mean the total of AB Ci even though I split it's a three input Karnaugh Map. Suppose I give a four input Karnaugh Map how you would do this using a 4 to 1 multiplexer. Or sometimes I may have to go for eight input multiplexer 8 to 1 multiplexer. So there is no need for me to give you several home works. What you can do is you can take any of the three variable Karnaugh Map reduction or four variable Karnaugh Map reduction and try to map them into Muxes first using 4 to 1 Muxes and if necessary sometimes you can use 8 to 1 Muxes 8 to 1 Muxes and 4 to 1 Muxes.

Of course we will do some more examples in the class but there is no need for you to wait for me to give you a formal home work assignment sheet. Whatever you have done earlier with gates in principle you must be able to do it with multiplexers. So what we will do is next see some more circuits using Muxes and then also see what are the other ICs other than MUX which are Medium Scale Integrated circuit ICs which can be used for combinational logic as well.