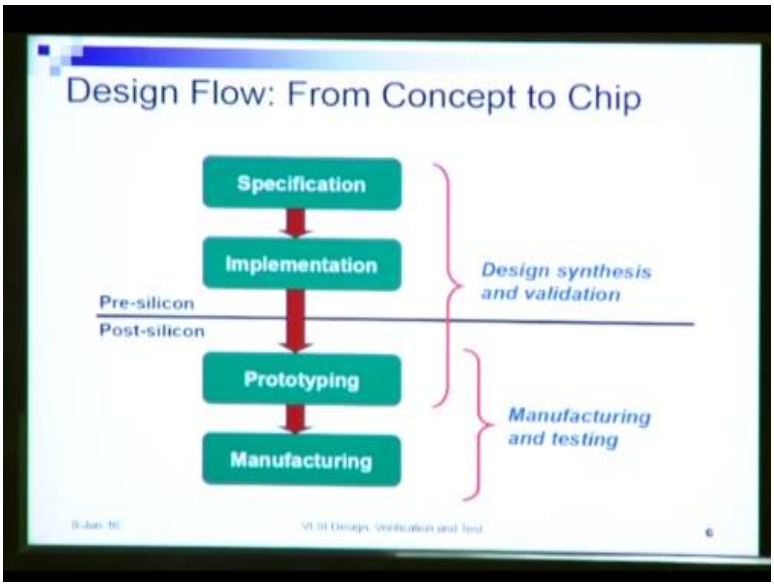
In module 1 we presented the need for computer aided design.

(Refer Slide Time: 00:13)



In module 2 we will take an overview of how the design flows from the specification to the final packaged chip. If you have the words are you taking you through all the sets? So first we want to be have, so what word are we trying to do, we have a function or an idea or an application in mind and that application that we have in mind have to be finally hardware into a chip, and that chip or then hardware will realize the functionality that we have in mind.

So the design flow basically composes of the sets that they will design from the initial concept to the finally packaged chip. So firstly what do we have that initial informal engage the idea that we have in mind that could be finally as that is to be formally jotted down and specified. So that shape get by the first module here specification.

Specifications are done by let us say hardware description language, languages like say Verilog, PhD, system C etc., or you will at a high level architecture description languages like say Lisa for example anadem can be also be done through graphical tools like say space charts or specksy etc., and then this such a specification then will be in steps taken to the final implementation.

So firstly we do not have any idea about what the structure of the chip will be, where and how what will be the gates, what will be the architecture, we do not have that idea initially. Initially we have again idea of the behavior or the functionality. What function should this chip or my application perform? And we jotted down in the form of say, high level C like program as processors as processors which communicate among themselves.

So process could be a module and the modules communicate among themselves right. So we have basically a program of what we have in mind, from that we take steps that slowly and purposively takes the design to more detailed levels of hardware structure. Finally until we obtain the final layout or the implementation, then layout is the final software output or the design output that will be edged or in the subscript.

So that is the final implementation of the design, that implementation is then is edged on to the subscript for is put on to as SPJ and period of prototype. The prototype is a design which has not all the full fledged features of the application, but has enough so as to be able to test the design particulars as well as the implementation connectors.

So we understand that the design versus what is the connectors of the design, what we intended to design has been produced as the final implementation. So that has to be verified and validated against all firstly the logical output as well as also the performance booths is it is the output being produced within the correct time does it have the required reliability is it producing hotspots are we being able to prevent hotspots.

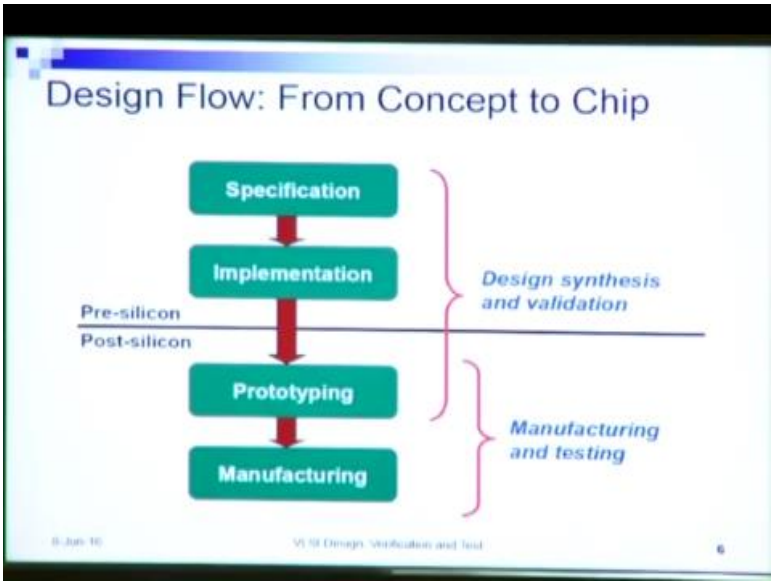
And is it would efficient is it consuming power within the bounds that are specified or not all these have to be verified on the implementation and we do prototyping as well and after the prototype has been found to correct we go for mass manufacture the reason as we stated in the last module that hardware's is almost irreparable after we manufactured.

So we need to have 0 defect designs and manufacturing process will produce the shape lacks and cores of possibly of these chips so we do not afforded to have errors in the finally manufactured

chips whatever errors are done as to be found out debug and the correct chip should already manufactured so that is the manufacturing process and then as I said up to from the specification to the implementation to the prototyping the design goes through a syntheses process so what is this synthesis it is taking the design to a more detail one from a less detailed specification.

So we are at the higher levels of design we have an idea of the functionality but have very less idea about the structure of the chip as we go down to the details that is synthesized the design progressively from the higher to the lower levels we progress from behavior to the augment structural implementation and each step it has to be validated against the performance required with at the correctness and also now after we have prototype we have to see that the prototype chip is error free that means the prototyping process.

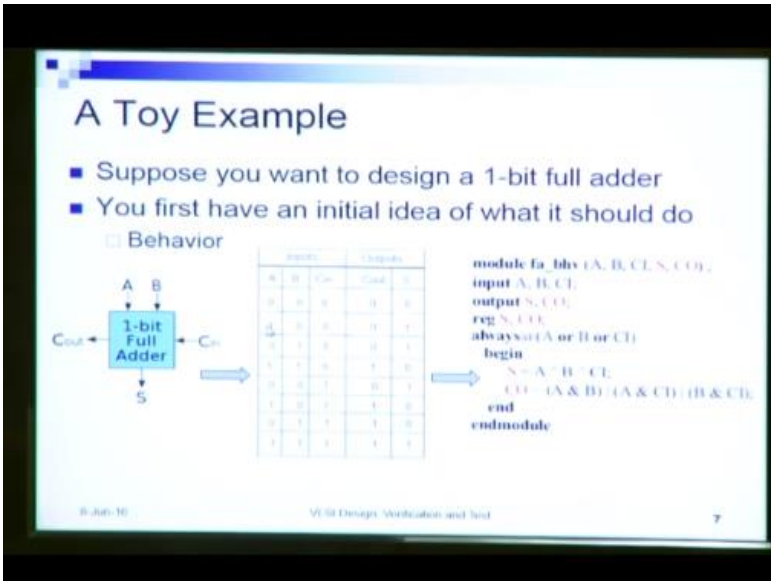
(Refer Slide Time: 06:22)



As implementation take care of all issues of the issues we have to test it so testing means on the implemented chip we try certain test vectors to find out whether there are falls in the chip right so that is called prototype testing and we also manufacturing testing which is after the final tape down chip although they have been produced in cores possibly but they also have to go through minimal set of test where certain minimal important test vectors are driven to find out whether

the finally manufactured chip also is correct functionally and in terms of performance right so this in an action the whole design flow from concept to chip. So if we just take but five example of what we have been taking.

(Refer Slide Time: 07:02)



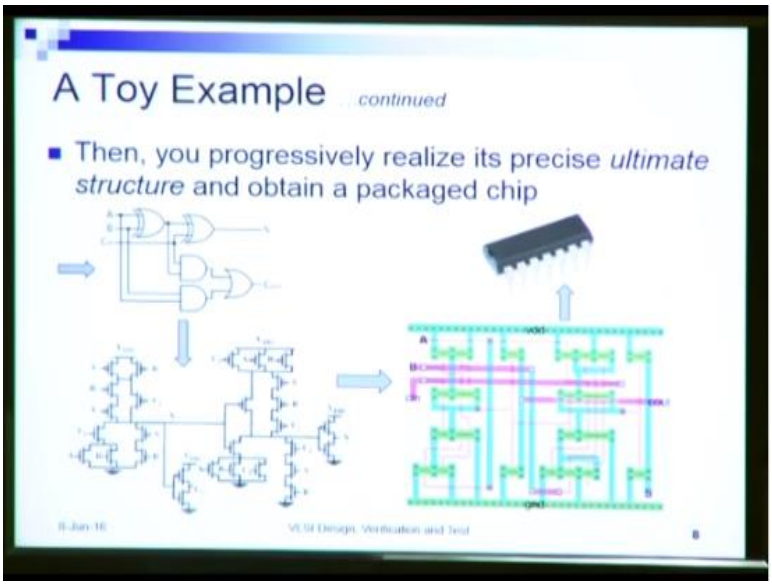
Suppose we want to design a one bit full adder so first what we have an initial idea of what we want to do we have two inputs A and B carry input and we produce two outputs one is the sum bit and 1 is the carry bit it is a very simple one bit adder so here then what we do when we have an idea of the functionality we represent the function here directly in the form of the truth table because this is a very simple application and we know how the outputs will be given so we know that whenever there be an odd number of these inputs as one that sum it with you, either all the three term one that submit all the input that is A, B and C if they are one they submit with you one or if only one of the input is one then the summit will be one.

And when does that carry be equal to one whenever atleast two of the bits of input bits atre one that the carry will be equal to one and correspondingly we can write a program to formally represents this on the computer all the truth table is also a formal representation we can obtain

that ca n be two individual rises tastes this truth table as input and process this program as output but the program as is a behavioral Verilog code which says the same thing.

So the sum bit is A XOR-B XOR C or carrying and the output is can be or AC, CAN or BCN right so we said whatever we said because this sum is an odd function means is an XOR old history so any ODD number of base in the input is one that always produce a one and the carry will produce one whenever atleast two of the bits are equal to one so this is the representation of the behavior.

(Refer Slide Time: 09:35)

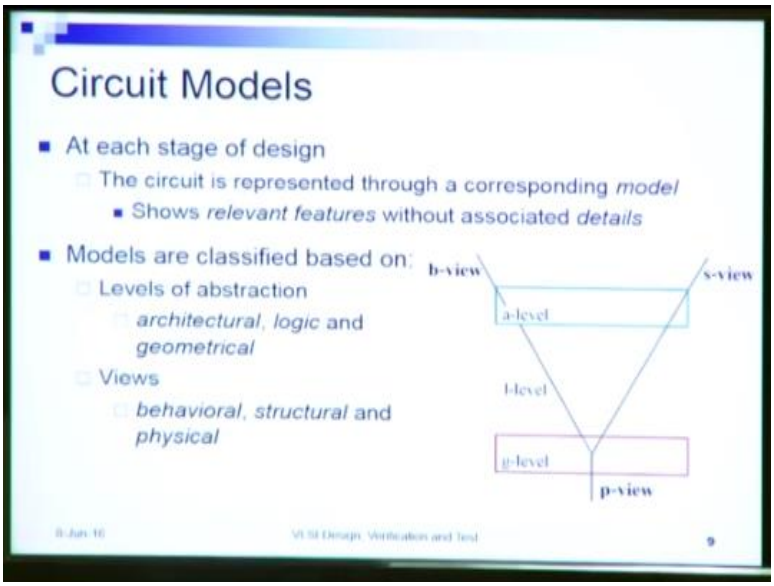


From this behavior of representation we have to what we have, we slowly progress through the steps of the design process and we will study in detail how the design process flows and we will have what is known as this gate level design all the logic after the logic level through the all the logic, so this we will get a gate level design this can be grown by 9 this is very simple but we can assume therefore complex circuits such as logic structure such a logic circuit is difficult to design at once by hand or a complex functionality.

However the thing remains that from the behavioral descriptions we progress one step towards structure here towards a logic level description of the application that we have from that logic level application we will have a transistor level application, transistor level design the transistor level design is a seem of design of the same one bit full error, so from the from this one from this design will progress again more towards structure.

This is so this is the most structural representation how should be the structure of this application full adder and this is the more detail specification how should this gate been implemented in the form of transistors and then how should this transistors be made out of the subscripts so we have rigorous layout design from this layout design it will be H on to the H on one two as substrate and we will get a finite package chip like this right. So this is a toy example but provides you but shows you in an action as to how the design progresses from the initial specification that we have in might of its functionality to the finally package chip.

(Refer Slide Time: 11:24)



Now at any level of the design we said that the design progresses through different stages one levels at any stage of design we will have what is called the circuit will be represented in terms of a model okay, so and in stage of design the circuit is represented through a corresponding

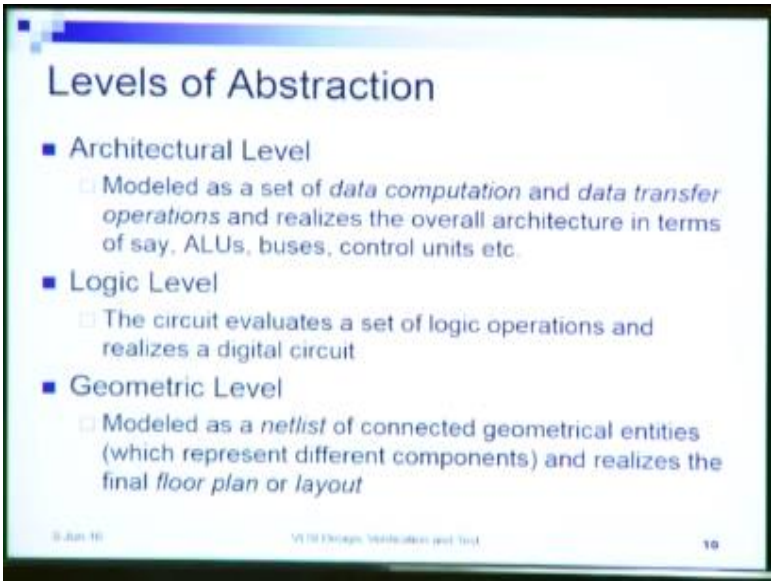
model, a model shows very well features of the design that we require with the associated details, so we understand that well when we are at the architectural level we had very little idea of the structure. So we will have a model that allow us to specify that allows us to specify what to specify the functionality. For example, the truth table can be taken as model the program that we had the very long code is the model and then the model progresses to the logic level and then to the gate level and then finally the physical design. So models are classified based on two things, two aspects.

So any model at any level will be classified based on two aspects, one is the level of abstraction so I can have three major levels of abstraction one is the architectural abstraction on model, then logic level abstraction level on model or and the geometrical level abstraction on model, this figure shows this three levels so one is this is the architectural level model, this is the logic level model and this is the geometric or physical level model the geometric level one.

And then each of this levels there can be multiple views for example, the architectural level as the behavioral view and the structural view on this side so views we said are of three types here the behavioral view the structural view and the physical view. So the architectural level at the architectural level we will have the behavioral view and the structural view at the logic level we will have the again we will have behavioral view as well as the structural view.

And then when we know more and more about the actual physical structure of the device we will have at the geometric level finally a physical view as well and the behavioral view and the structural view coincides together at the physical level and we get a physical view of the circuit.

(Refer Slide Time: 14:04)



So let us quickly now take a look at what this basic levels of fundamental levels of abstraction are, so if the first is the architectural level of abstraction. So at the architectural level the circuit is modeled as a set of data computation and data transfer operations, okay and so this is the behavior or the function the function is realized as a set of data computation because we see we have been able to represented it as a program so we have a set of data computations and what does it do whatever computation we do in a program basically the data the input data is transformed progressively to the output.

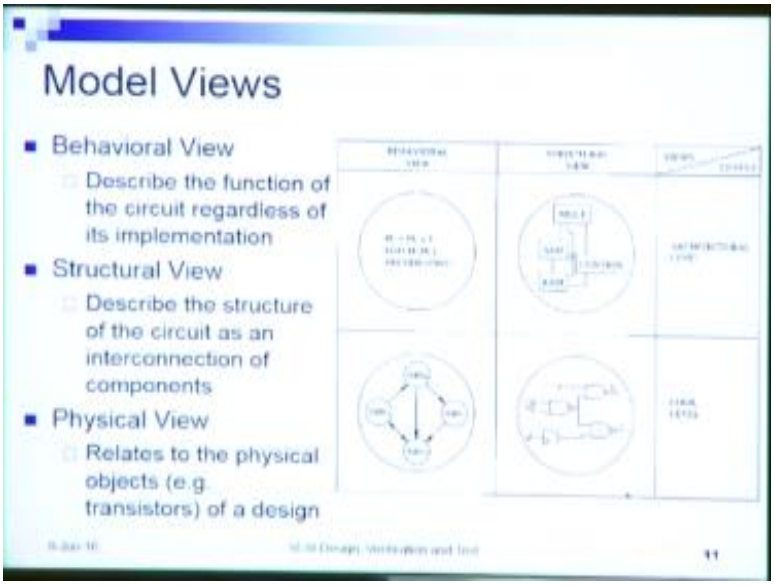
So there is a data transfer logic within the program, and this particular this functionality in terms of data computation and data transfer is then realized in terms of the architecture in term of ALU, buses, control units so these operations as well as data transfer this operations in terms of data computation and data transfer will then we realize in terms of an architecture of the design. So from the operations we will first get an architecture of the design at this level.

So the architecture will be in terms of the ALU units, buses how are they connected how are the they will be control units that control operations at each times it, okay then we come to the logic level at the logic level the circuit is represented basically as a set of logic operations or let us say a state machine with respect to a controller and a set of Boolean equations that represented the

register transfers that happen. So these Boolean equations and state machines together will fall the behavior description at the logic level and what will we do in the logic level we will realize the circuit in terms of a digital circuits in terms of gats and flip plots okay and form this one we will go through a strange of technology mapping and finally we will have the physical component the physical design where we will have completely structural view of the design.

And there the circuit will be modeled as a net list of connected geometrical entities these geometrical entities where represent different component of the design and then from these met list or connected geometrical entities we will realize the lay out and the other floor plan based on which I will finally in get an image of what we have to edge on the subtract and finally obtained the package stitching.

(Refer Slide Time: 17:17)



Then we also said that at these level of design we also have views we have abstraction levels and we also have views so first we have the behavior of the view so behavior of view as we have already said it distract the function of the circuit regardless what the implementation of the structure is and w also had the structural view it describes the structure or the circuit as an inter connection over components.

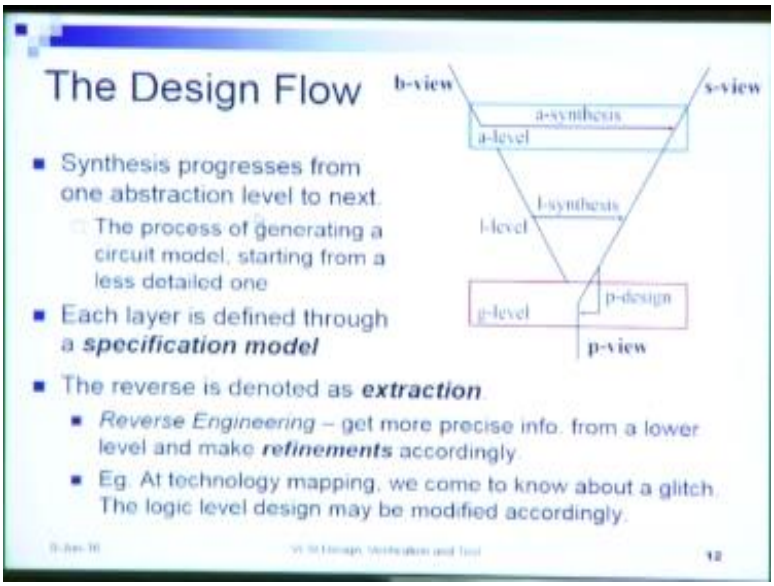
And there is a physical view that will edge to the physical objects example transistor of a designing as we said that at arc structural level and the logic level we know very little about the exact physical transistors or the physical level design right you do not have the physical view essentially however at the geometric physical level the structure and the behavior con inside then we get a physical view.

So her is it diagram that shows the behavioral and structural views at the arc structural level and the logic level so we said that at the behavioral view at the structural at the arc structural level we will be composed of a program so there is a pc pc + 1 f something decodes something we are dieting in terms of high level program that high level program will be synthesized in to a corresponding structure which is composed of will should composed of say multipliers adders rams control units and the inter connection among them.

This structure will basically realized the function that is shown in this behavioral view at the logic level on the other hand we will have let us say a state machine corresponding to the controller that we have here so the controller will basically control bit operations should be executed at which times step and hence this controller will essentially will represented in the form of an state machine.

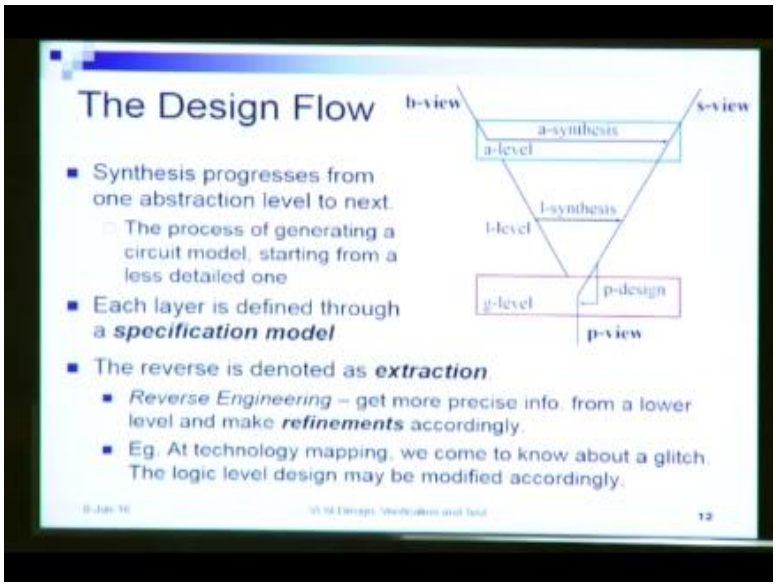
And that stat machine again will be realized in terms of the k^{th} level description which will correspond to the structure so the FSM at this FSM here at the logic level will represent its behavioral view and the corresponding structural view will be this k^{th} level description.

(Refer Slide Time: 19:48)



So now if we look at entire design flow as we said we looked at what we said the important levels of abstraction as well as the views at each level that the possible views at each level. Now how does the design flow the design flows through a set of synthesis processes so what is a synthesis.

(Refer Slide Time: 20:16)



It takes a design from one abstraction level to the next from level abstraction the process of generating a circuit model from a less detailed one is called synthesis and at each level the design first starts to specification so the behavioral description describes formally interns of the program is a specification that specification is synthesized that the behavioral at the architectural level here and from this abstraction we get into another abstraction of the circuit here through the synthesis process here and we obtain as fractional view okay.

So similarly at the logic level we have the bullion equation and the plus state machine and we go through the logic synthesis process to obtain the gate level description and the minimized bullion gate level description digital circuit after the synthesis process at the logic level so however the synthesis not a one directional thing there can be mistakes in the synthesis at a specification level at a given level at any level at the specification mistakes which cannot be understood at that level until a synthesis is done.

So after the synthesis is done we may we may find at the design has certain errors and needs to be corrected so then therefore the design has to be taken back to the previous step and correct it again and then again proceed so this is a integrative process which will go until the design at a

particular level is correct so therefore what we said we need to reverse the design so this reveres is denoted as extraction so if synthesis takes suppose the architectural level from this behavior to this structure the extraction process will take place we will take it from this structure to the back of the behavior.

At the logic level if you have this gate level description the synthesis will take back to this one and also will also between level is we need to get back for example foe example until we go to the circuit level let us say until we go to the transistor level we do not understand that there is a clinch There is, there is an error that is called a clinch so clinch says that because the two inputs to these gates do not come at the same time the output could be enormous so therefore that can be found when the design and the transistor level has been done now if we have found this how can we correct this we have to go back to the logic level.

We have to correct the design at a logic level and come back to the transistor level design again I will find out whether it is correct whether still features or not right so this is why this extraction process is also very important along with the which means that Let us say I have a gate and this gate has two inputs and the gate would not come through a set of prior previous gates and the inputs to this gate do not come at the same time the come at different time so for a very short period of time.

synthesis process. Therefore with the over view of the entire design flow with an action we come to the end of module 02 of the lecture 01.