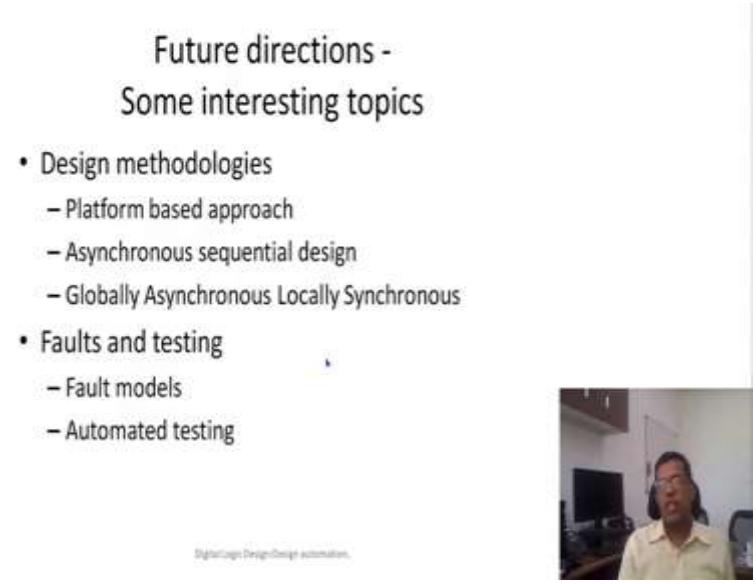
Digital System Design
Professor Neeraj Goel
Department of Computer Science Engineering
Indian Institute of Technology, Ropar
Future directions

(Refer Slide Time: 0:18)



Hello everyone. So, this is the last lecture or last part of the lecture for this whole course, in this part of the lecture, we will talk about what are the interesting topics or what are the future directions from where, so you can go further to learn about this particular topic. So, essentially during this course, we have ignited.

So, this is only the vetting of the appetite, we, this is just the starting of the digital design, we have covered most of the basic and fundamental aspects of the digital design, but if you have to design your own chips or you want to design your own ASIC, then there are a lot of lot more things which you can learn, which would be helpful for the understanding of your whole chip design process as well as designing your chips, designing your own digital circuits.

So, in this part of the lecture, I would like to suggest some of the topics which can be useful if you would like to go for a specific kind of design. So, one thing, the one important topic or one important thing could be like you see, we talked about a chip, which could be a billion transistors chip.

So, a billion transistors chip cannot be designed from a bottom up or top down approach, it would take quite a good amount of time. So, the popular approach which has been adopted by most of the industries is right now, they do not design everything, they will design some of the specific parts and we will buy rest of the part which is off the shelf components, these off the shelf components are called IPS integrated sorry, intellectual property blocks.

So, these IPS could be processors, it could be accelerators, it could be buses, it could be interconnects. So, and also it could be interfaces like USB interface, HDMI interface. So, when you connect all these components, which are off the shelf and you design your own SOC system on chip, this approach of platform based design is the one using which you can design hugely sophisticated SOCs within a short period of time.

So, you may be sometime thinking that your design process is so complicated how people come up or how companies come up with a new Smartphone every six months or every year, because of this particular platform based approach where they will get the component off the shelf and then finally fabricate or here they have to mostly focus on which components to be used, how they would be verified, and they are to design glue logic or logic that would connect to different components.

So, that is an important design methodology which could be, which could serve as a starting point if you would like to design your own SOC. We have, during the course we talked about sequential design, we have talked about combinational design and in the sequential design we have talked about only synchronous sequential design. So synchronous sequential design was which was bound with your clock.

So, what would happen if there are two different circuits which are working at very one working at high speed another working at low speed, they cannot if they will work even the same clock period, then either it would make circuit very fast or will make circuit very, it will make eventually circuit very slow because the slowest circuit will determine the clock period.

So to avoid that, the other interesting design methodology is asynchronous design when we are saying that these two designs although it would be sequential in nature, it would be step by step but asynchronous in nature. So, this asynchronous is a approach which can help bind different units which could have different speed or different latencies.

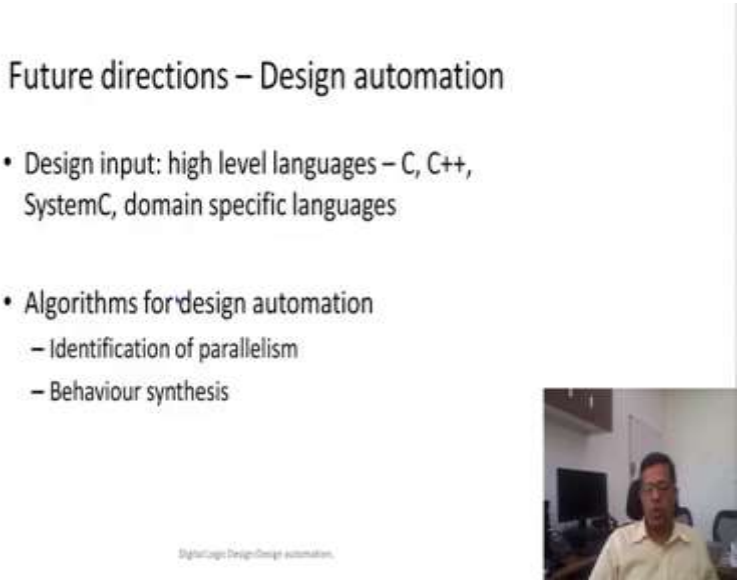
Now, particularly to reduce the power then globally asynchronous, but locally synchronous that kind of methodology is also there that locally each of the unit would be synchronous, but globally it would be asynchronous. So, during the whole design process, we have talked only about the design and we mentioned about the verification part, but we have never gone into too much of detail about how verification is done, how testing is done.

So, for verification, in this course we have limited ourselves to basic test bench design or basic stimulus, so that we can activate couple of patterns and see whether it works or not in practically what we have to do is we have to see that if there is a possibility of a fault, we have to say what kind of faults, what kind of errors are possible in a digital design, let us say a wire is not connected or wire is if it was not supposed to be connected, but it is connected.

So, those kinds of faults are there. So, different kinds of fault models are there. And given those fault models, then we have to instantiate, we have to give this stimulus accordingly so that we can check whether any of those fold exists or not. So, doing that instancing all the force or initializing all these or checking all the faults is also a systematic process is called verification engineering.

And now, in the verification there is further scope of doing the automation. So, that is called automated verification or automated testing. So, those are also some interesting topics, which one can look for, and these topics are also good for industrial applications, because most of the like in the design engineering a good amount of effort goes for the verification. So, these are a couple of the interesting topics which could go, which could be seen.

(Refer Slide Time: 7:03)



Another future direction for the course is doing the design automation. So, we talked about a platform based design. So, that is one way of designing very large scale integrated circuits. The other way is that instead of specifying in the hardware description language like verilog we specify in higher level abstraction languages like C, C++, system C.

So, the design specification if it is given at high level of abstraction, then it could make the design process faster. So, when you are doing design using verilog you would have observed that verilog takes time first of all to write all those details and also it is much more verbose. So, many things, which may be obvious, you have to write all those details.

So, if we write in C, C++, then we are leaving all the task to the automated, automation tools and these automation tools would be able to extract the parallelism and it will try to identify that which blocks could run in parallel and how those could be scheduled, or those could be the RTL could be generated.

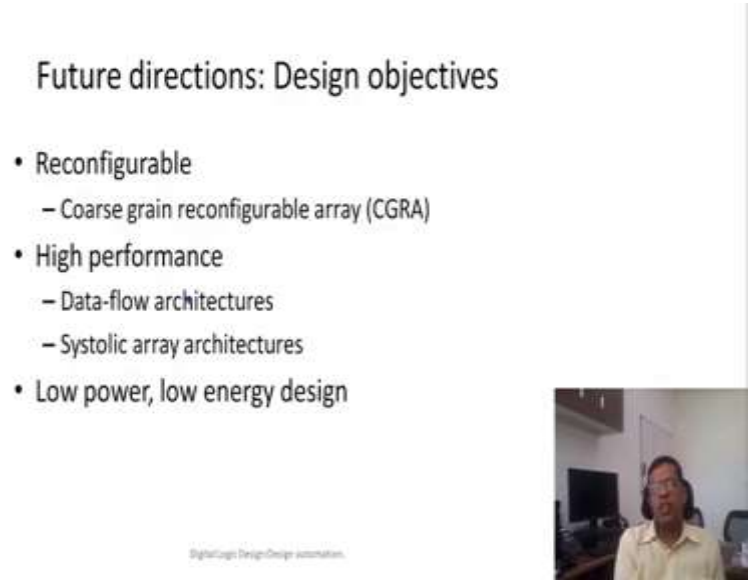
So, but C, C++ has certain limitations in terms of specifying parallelism, etc. So, that is why some domain specific languages are also there, which many universities keep on proposing and yeah. So, when any of these languages like C, C++, system C, system C is also a kind of a C++ library it is a C based language, C++ based language, but it is sort of only a library. It is a modeling style essentially.

So, all these domain specific languages, we design our specification in that if we input our specification in these languages, then the other field of study could be the algorithms. So, that algorithm, so that the whole process of design could be automated. So, given these C, C++ specification, system C specification, then how the whole design process could be automated, so algorithms to design those process is also an interesting field of study.

So, particularly, there are two important topics there. So first of all, that how we can identify the parallelism. So, one way is that we leave it to the user, he while writing the, specifying the application in form of C, C++ or system C, they would specify the parallel hardware or parallel blocks, or the design tool itself could find the parallel blocks.

And once those parallel blocks has been identified, the whole behavioral synthesis could kick in and it can do scheduling, resource allocation, binding, etc to finally generate an RTL circuit, which has, which knows that what would be the area, how many cycles it will take and in each cycle what operations would happen. So, this behavior automation of this is also an interesting topic.

(Refer Slide Time: 10:32)



Another direction which can be from here is like the design objectives could be different for different cases. So, it could be a low power design or a low energy design. So, final design could be we would like to have low power or extremely low power. It could be very high performance,

high performance means the latency of overall could be latency of overall application could be very less or throughput could be very less.

So, this high, when we are talking about high performance different kinds of architectures are possible, like our data flow architecture or a systolic array architecture in systolic array architecture, you will have multiple instances placed in a 2D, two dimensional grid and then input is flowing from one to another.

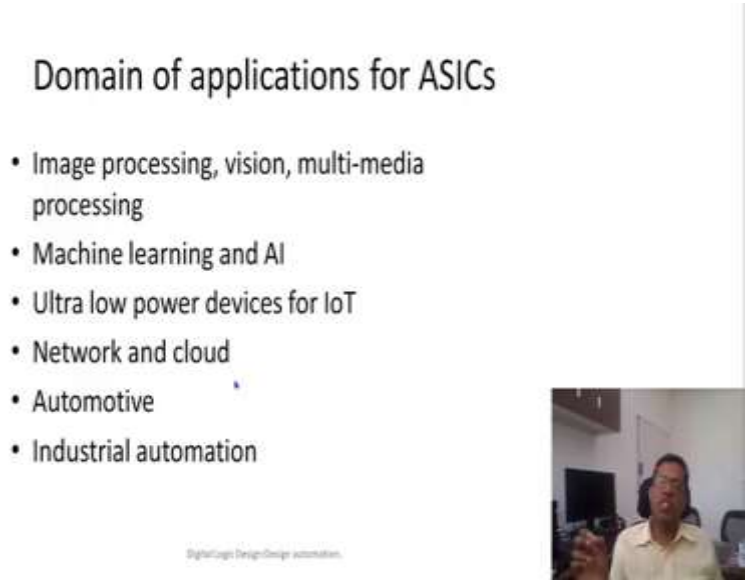
So, there are a couple of these architectures. So, particularly this architecture for example, for machine learning applications, even graphics processors are not able to give that much performance which is required, so tensor processing units or machine learning processors, so all of them are using this kind of architecture, so that the type performance which is required for executing or doing deep neural networks or these, those machine learning architectures could be done within the short span of time.

So, these architectures could be another direction where you can learn about different kinds of architectures, high performance architectures, and it could be some application specific like tensor processing units or something similar. The other objective could be reconfiguration. So, basically, there are two kinds of reconfiguration which is possible one is fine grain reconfigurable circuit or course grain reconfigurable circuits.

So, in fine grain configurable, fine grain reconfigurable circuits are essentially FPGAs where the level of reconfiguration is at the level of CLB configuration logic unit or at the level of LUT. So, basically each of the unit could be reconfigured, but at a course level, it could be like you can design a couple of logic units together. So, it would be you can visualize it like reconfiguring a circuit at the level of ALUs or at the component level.

So, where multiple such components are there and then you can reconfigure each one into to perform a certain task. So, that could be also one of the objective. So, objectives is one dimension the other dimension is the application. So, there could be various applications which are possible.

(Refer Slide Time: 13:33)



Now, as the word itself says application specific integrated circuit, so what kind of replication where integrated circuits need to be designed. So, applications could be image processing, vision, multimedia processing, so this has been popular applications for ASIC since last 20, 30 years.

So, because image processing, vision, multimedia, they take their, they take huge amount of time, significant amount of time when they are done, when they are executed on a processor or microcontroller. So that is why a specific IC need to be designed so that they can be done in a lower power as well as a high performance or in a less they can execute in less time.

Machine learning and artificial intelligence is also an upcoming application area for ASIC. Because the machine learning has taken prominence in last couple of years just because it assumed lot of performance, lot of computing power. But all of this computing power was coming from the cloud.

Now, if you want to do those machine learning applications own to your small devices, it is not possible because you do not have that much of computation power. That is where you would require an high performance ASIC which will take less power but still can be done on to your mobile devices.

The other important application domain is ultra-low power designs for IoT. So, Internet of Things, basically your, you would like to have those devices placed everywhere in your home, your body or your industry everywhere those small devices are there and you do not want to recharge them every time.

So, the notion has to be that once they have been installed, they can work for a couple of years. So, that means we had to design an ASIC so that they can live for a couple of hours without even changing the batteries.

Networking and cloud is also another domain of application for ASIC. So, because of the high throughput, so we were, we have started from 2G network to 3G network, then 4G network, 5G networks. So, the speed of expectation or bandwidth requirement has been increasing day by day. So, that is why we would require some good, a new different integrated circuits for to meet these demands.

And network is also closely related to cloud because we can, we are able to do computation on cloud because of the possibility of network. So, you also require some application specific design for clouds. Automotive and industrial automation are another areas where we would require different application specific integrated circuits, because the applications are vary.

In industrial automation, every industry would require different kind of automation, they have different automation requirement, and to meet those automation requirement different kind of ASIC would be required. So, all of this thing suggests that there is a huge potential in this particular area or this particular sites of learning, where we if we design ASIC, then it could immensely benefit in almost different all areas of our interaction.

So, this course should be meant or should be taken as a starting point and then you can go ahead in learning different topics or different kinds of ASIC or different methodologies. And then final goal could be you can design your own ASIC. So, with this, I would like to close this course, as well as this lecture and wish you all the best. Thank you very much.