

Architectural Design of Digital Integrated Circuits
Prof. Indranil Hatai
School of VLSI Technology
Indian Institute of Engineering Science and Technology, Shibpur, Howrah

Lecture – 39
Fixed Point Number Representation

Welcome to the course on Architectural Design of IC's. So, till now we have seen different basic building blocks; that means, how to design basic building block like this 2^x to the power x then \log_2 base x . So, this kind of structure we have seen. So, after that different kinds of adder architecture we have discussed. So, again after that we have discussed there are several types of multiplier, how we can design that, how we can design and how we can implement them for different different application? So, that we have seen ok.

So, whenever we are designing this hardware at that time the primary thing is that we are basically working in the Fixed Point Number Representation. So, what is there in that; that means difference between floating point and fixed point? Floating point means, actually there I can have; that means, the number I can have 2 parts; that means, the integer part, as well as the fractional part ok, but whenever we are talking about hardware so, at that time hardware design.

So, at that time mainly we are basically focusing on the number, which is mainly dependent on the integer part. Unless and until I am because there is no such; that means, proper method to; that means, to represent that decimal point, where is your decimal point that you have to notify. So, how you can notify in hardware design? So, but whenever we are doing that in for implementing in the DSP systems, so, at that time I need to know or I have to work on the in the fixed point domain not in the floating point domain.

So, for that reason whatever is the algorithm that we have developed mainly we work on the algorithm on the floating point domain, because we do the algorithm verification that is mostly on the; that means, programming environment using MATLAB or C. So, this kind of language we use, but whenever we are designing one hardware for that particular algorithm.

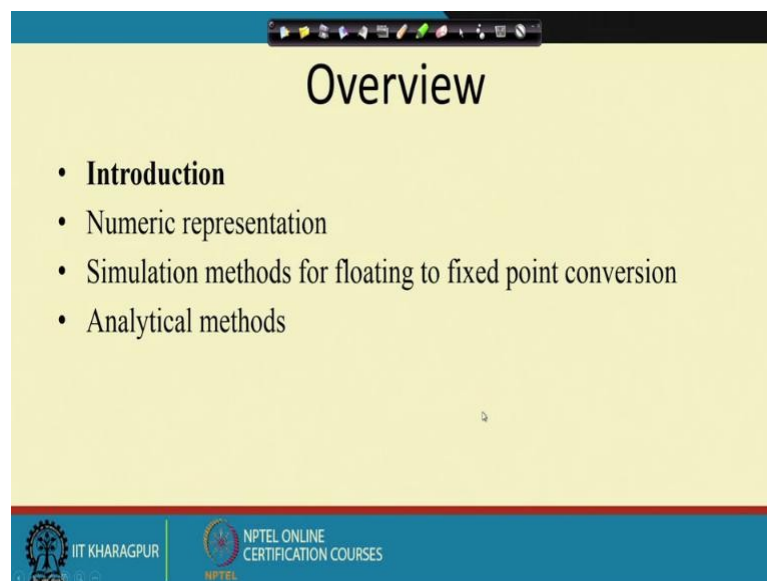
So, at that time we never use not never actually we do not use prefer to use C language at that time. Because, we have to describe the hardware by using any hardware description language that may be system Verilog, Verilog or VHDL log systems that kind of structure I need at that time.

So; that means, I need to know what will be the effect of doing this; that means, changing from the; that means, changing from the means the algorithm in the algorithm we are working on the floating point number representation, but whenever we will come to the hardware; that means, hardware design. So, at that time we are in the fixed point number representation. So, at that time what is the effect of that, as we are doing this conversion from floating point to the fixed point.

So, at that time what will be my effect to that particular circuit or to that particular algorithm on the hardware implementation chart? So, that we will see; that means, in today's class which is this fixed point number representation. So, what is that, what are associated with that, whenever we are working on this digital this hardware design.

So, at that time it should be always remember. So, there is an effect for this conversion ok. So, what is that effect, how it will effect? So, that we will see in today's class ok; so, this is the topic name is fixed point design.

(Refer Slide Time: 04:12)



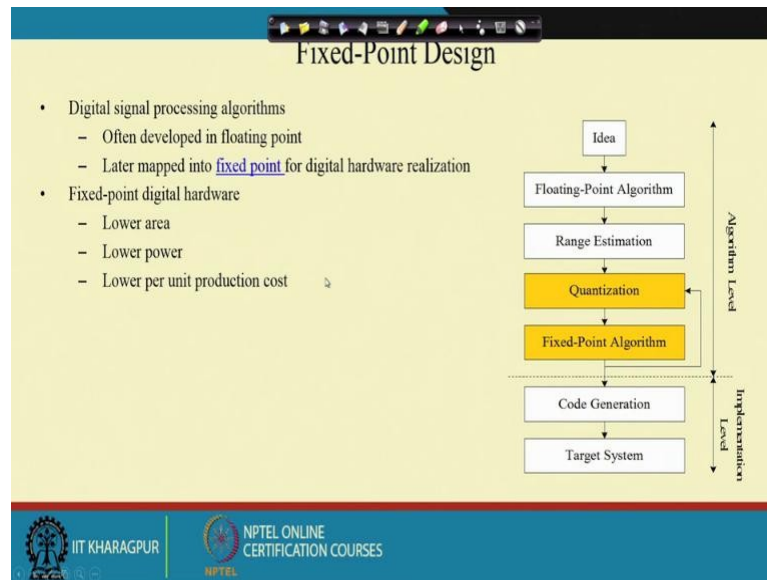
Overview

- **Introduction**
- Numeric representation
- Simulation methods for floating to fixed point conversion
- Analytical methods

IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES
NPTEL

(Refer Slide Time: 04:14)



So, the first thing is that suppose mostly we use these for DSP algorithms ok. So, in DSP algorithms mostly, they are being developed in the floating point domain ok. So, if you just see this flow chart. Suppose this is my idea, idea about designing any system ok. So, in a system means they are mostly; that means, related to this digital signal processing part; that means, there may be several kind of digital signal processing algorithms ok. So, now, I have to implement that idea or I have to realize that idea, in the real hardware using the real hardware ok. So, whenever we will do that at the time initially what we have to do, that idea is basically implemented using any algorithms, which is basically work on the floating point domain ok.

So, after that we calculate the range estimation; that means, any algorithm that may be iterative; that means, in any algorithm we basically perform some of the operations or we implement the function and then for the function basically dependent on the operators. So, using the operators we basically do the operations ok. So, in any algorithms whenever we are doing at that time that may be iterative or that may be parallel.

So, any kind of algorithm we can implement. Mostly as they are working on the floating point side. So, at that time; that means, due to; that means, that particular algorithm, what will be the maximum value I will gather or that will be collected for that particular algorithm? So, that initially we have to estimate ok. So, what will the maximum value I can get for that particular algorithm? So, that we have to calculate. So, after that it is known that infinitely we will run that particular algorithms ok.

So, there may be certain limitations that how much or what is; that means how much time I will run that particular algorithm or run that particular system ok. So, based on that now we have to do this whenever we will come from floating point to the fixed point we have to do this quantization ok. So, after that whenever we are in the fixed point that algorithm suppose we have changed from floating point to the fixed point then it ready for the code generation; that means, at that time you can actually now a days this EDA tools they are. So, much; that means, advanced that from that particular if you just convert it into the fixed point ready; that means, algorithm.

So, from that particular point now you can develop the, or you can auto generate the HDL for that particular circuit, sorry that particular algorithm ok. So, after that so, the next step whenever you are having your fixed point algorithm ready. So, at that time; that means, your algorithm is now ready for code generation, code generation in terms of HDL ok. And, then if you have any targeted system as we have seen that we can go for this full custom or semi-custom FPGA based design. So, any of this target system is that only so, based on that now you can go for the implementation part.

So, here if you just see mostly the digital signal processing algorithms, they are being developed in the floating point domain, later they have been mapped into the fixed point for digital hardware realization ok. This fixed point digital hardware, why I need this fixed point digital hardware because, whenever we are designing one hardware for; that means, any system or any algorithms.

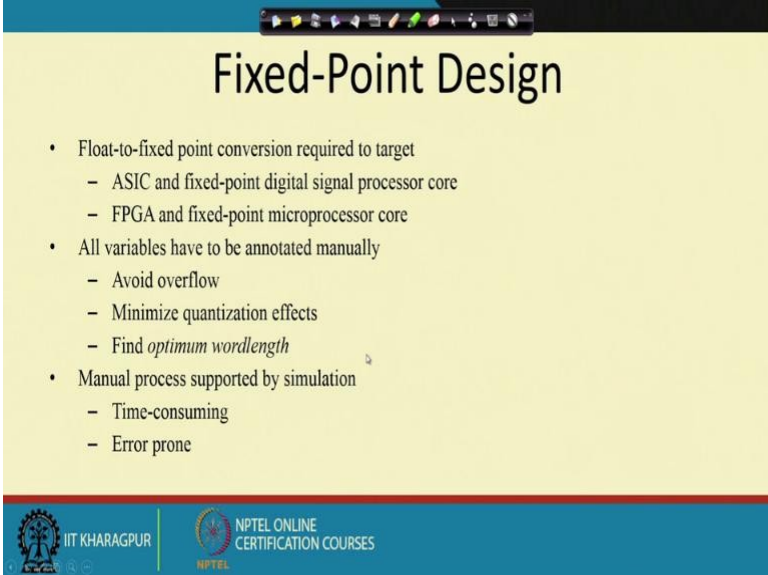
So, at that time my primary target is to achieve these 3 things, which is that this low identity area, low power as well as the maximum frequency of operation and along with that there this lower; that means, in production cost per unit that also I need. That means, these are the things which I have to be remember whenever we are designing the corresponding hardware for that particular system or the algorithms.

So, that is why the in floating point what happens? The corresponding range that is very much wide ok, but in fixed point whenever we are generating the hardware at that time the range of the data should be within the limit. Otherwise, what will happen if I just go for the; that means, wide range of data so; that means, that the area requirement or the

that to hold the value or to do the operation with that particular length, it requires more number of hardware or more number of logical component to do that particular operation.

So, more number of logical components if I need; that means, the area as well as the power along with the speed of operation I will perform; that means, the performance of that system that will be degraded ok. So; that means, I have to from this due to this floating point to the fixed point conversion there is an effect which basically degrades the algorithmic performance, but it enhance the; that means, this hardware performance ok. So, what is that algorithmic performance, what is this hardware performance that we will see later of this lecture?

(Refer Slide Time: 10:46)



The slide is titled "Fixed-Point Design" and contains the following bulleted list:

- Float-to-fixed point conversion required to target
 - ASIC and fixed-point digital signal processor core
 - FPGA and fixed-point microprocessor core
- All variables have to be annotated manually
 - Avoid overflow
 - Minimize quantization effects
 - Find *optimum wordlength*
- Manual process supported by simulation
 - Time-consuming
 - Error prone

At the bottom of the slide, there are logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES.

So, this float to fixed point conversion required to target, whether the target may be one ASIC or it may be one FPGA based ok. So, here whenever we are doing that all the variables have to be annotated manually, why I need to be; that means, annotate that manually to avoid the overflow or to minimize the; that means, quantization effects and to find the optimum word length for implementation ok.

So, what is the meaning of these 3 lines means; suppose let us consider this to one simple example ok. Suppose I have to add; that means 2 numbers that is A plus B ok. So, now, if I do; that means, let us consider 2.35 plus 3.5. So, the result will be 5.85 right, but in hardware, if I just say that this is only fixed point no; that means, fractional point it consider. So, at that time what it will be it will take only the integer part which is 2 and the other integer part which is 3. So, the results will be 5 at that time.

But, in the algorithm in actually I need the data to be represented as 5.85, but here due to the fixed point conversion I am getting the data as 5. Now, let us consider this particular addition operation I run in a consecutive loop for let us consider 100 times; that means, for one iteration I am getting the error for this float to fixed point conversion, I am getting the error of 0.85 right. If I run that same loop for 100 times; that means, 85 is the number of error, which I will accumulate due to this fixed point sorry floating point to fixed point conversion. So, the number 85 is a huge means what? So, at that time the result will be 585, but I will get instead of 585 I will get 500 ok.

So; that means, if you just calculate the percentage of error at that time it will be near about 17 percent so; that means, 17 percent error I am accumulating because of this floating point to fixed point conversion ok. So, that is why; that means, now whenever we are implementing the hardware. So, at that time I have to do this, I have to find out what will be the optimum word length. So, that this quantization error that will be minimized one just one thing that is one thing another thing is that to avoid the overflow.

So, what is the overflow let us consider another example. Another example means, what I said whenever we run that algorithms for let us say 100 times. So, at that time what I need I need to represent the data within a; that means, within a fixed word length ok. So, if I run that particular suppose that that original result is 585 ok, but due to the this not considering the fractional part I am getting the result as 500.

So, 500 means I can basically represent the data by 512 is the I think there is 9 bit I require to hold the proper 500, but in actual if I do something; that means, if; that means, need more of the data; that means, if I if my to hold the output value if I take 9 bit, but if my data value or the result value, if it is greater than the maximum value of; that means, for 9 bit what can happen 511 is the maximum number which I can represent or I can hold if the data or if the result value is more than 511.

So, at that time what will happen I will lose the data. So, 512 that will be represented as 0 at that time, though the original value of the data is 512 so; that means, to reduce this I have to initially, I have to calculate, what will be the maximum range of the data. So, that

I can properly; that means, define what will be the maximum word length of the results? Otherwise what will happen? If, I miss any of the bits so; that means, the whole result what I will get that will be error ok.

So, that is why we have to do this whenever we will convert this floating point to fixed point. So, at that time we have to be remembering this kind of things ok. So, this if I say this all are variables have to be annotate manually for doing this for avoiding this overflow or to find out the optimum word length, it is basically do it manually it is basically time consuming and while doing it manually at that time you can make the error too ok.

So, that is why you can write the algorithms for that particular to check, what will be the maximum; that means, data range and then what will the what is the quantization noise or the; that means, quantization error I am getting? What will be the corresponding that to overflow to avoid the overflow? What will be the maximum data range? That, we can calculate basically calculate using in a automated way ok.

(Refer Slide Time: 17:32)

Fixed-Point Representation

- Fixed point type
 - Wordlength
 - Integer wordlength
- Quantization modes
 - Round
 - Truncation
- Overflow modes
 - Saturation
 - Saturation to zero
 - Wrap-around

SystemC format
www.systemc.org

Integer wordlength = -2

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, then this fixed point type that may be 2 kind of things that is this word length another one is this integer word length ok. So, this integer word length means, the suppose this fractional portion is basically I have denoted here and then this portion is basically denoting the fractional part, this portion is the integer part and this S is the sign part ok.

So; that means, MSB represent the sign the later of this represent the integer portion, the magnitude of the integer portion, and this particular point represent the fractional portion magnitude ok. So, we have already seen, what it is the weight of this particular position that we have already seen.

So, in the quantization mode whenever we are we do this quantization. So, at that time there may be 2 kind of this quantization; that means, error that can be due to rounding or due to truncation ok. So, in the overflow whenever we will; that means, consider this overflow. Overflow method it can have 3 modes; one of this is saturation mode, another one is saturation which is 0 another one is wrap around ok. So, these are the 3; that means modes of the overflow.

(Refer Slide Time: 19:07)

The slide is titled "Tools for Fixed-Point Simulation". It lists four tools:

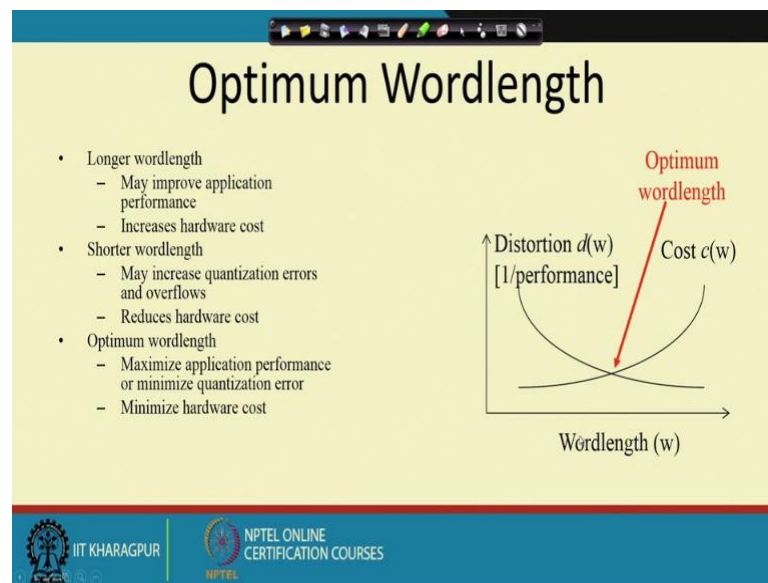
- gFix (Seoul National University)
 - Using C++, operator overloading
- Simulink (Mathworks)
 - Fixed-point block set 4.0
- SPW (Cadence)
 - Hardware design system
- CoCentric (Synopsys)
 - Fixed-point designer

Below the list, there is a diagram showing the conversion of floating-point variables to fixed-point. On the left, a yellow box contains the code: `float a;`, `float b;`, `float c;`, and `c = a + b;`. An arrow points to a green box on the right containing the fixed-point equivalent: `gFix a(12,1);`, `gFix b(12,1);`, `gFix c(13,2);`, and `c = a + b;`. Below this, a screenshot of a simulation tool window titled "fixed_sum" is shown. The window displays a block diagram with "To Fixed Point" and "From Fixed Point" blocks, and a "Scope" block. A status bar at the bottom of the window says "Ready 100%". At the bottom of the slide, there is a yellow box with the text: "Wordlengths determined manually" and "Wordlength optimization tool needed". The slide footer includes the logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES.

So, what I said that manually to do this calculate this floating point to fixed point; that means, to calculate the corresponding quantization noise, then calculate the optimum word length or to avoid the overflow or calculate the data range. So, we need the tools. So, that there are several tools are which are available in the market.

So, they are something like this gFix, then Simulink, which is from Math works, then Signal Processing Workshop, which is from Cadence and then CoCentric which is from Synopsys. So, in this particular platform, you can check or you can do it in an automated way from floating point to the fixed point conversion.

(Refer Slide Time: 19:51)



So, why I need this optimum word length? Why I need this optimum word length? Because, this longer word length what I said which what we consider for the floating point number representation, it may improve the application performance; that means, application performance means as there is no algorithmic error. So; that means, the performance at that particular; that means, calculation wise, it will be the performance of that particular system that will be good enough ok.

But, as I am considering more of the word length, it consumes or it increase the hardware cost ok. Not, only the hardware cost it increase the; that means, power to or it also degrades the speed of operation ok. Then, again for that if I choose shorter word length it may increase the quantization error and over flows. Means, what if I choose longer word length. So, at that time algorithmic error will be minimized right that is very much ok.

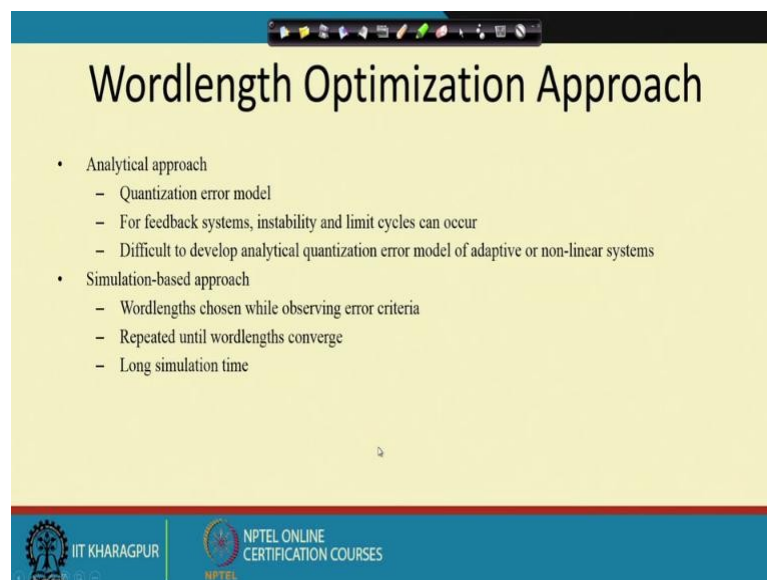
But, if I use shorter word length that may reduce the or that not may that will reduce; obviously, it will reduce the hardware cost, but it may increase the quantization error. As we in the example we have seen that if we do not; that means, for adding these 2.5 with 3.35. So, at that time I will get the error of 17 percent if I run that for 100 times ok. So; that means, for shorter word length you will accumulate the quantization error more and the overflow will also occur ok.

So, that is why I need one optimum choice for the word length, by which I can minimize the hardware cost, as well as I can maximize the application performance or minimize

the quantization error ok. So; that means, all the benefit if I have to get then I have to find out the optimum length for that ok. So; that means, if I just; that means, put the curves. So, if I just go in this particular direction if I put the word length. So, increasing with the word length the performance is basically degraded ok, but if I just; that means, sorry if I just increase the word length; that means, the corresponding cost is inversely proportional; that means, the cost will basically reduce. If, I just increase the word length sorry if I increase the word length. So, the cost is also increasing ok.

But, the performance basically if I just increase the word length the performance will increase, but this is one by performance. So, that is why this is basically decreasing ok. So, edge this is basically inversely proportional. So, if you just see then, where these 2 curves are basically meeting ok. Or intersecting to each other this point is the optimum word length, where the cost along with the performance is the maximum 1 ok. So, this point we have to find out, we have to find out for getting the and for at that particular point, what is the word length. So, that is the optimum word length for your system design ok.

(Refer Slide Time: 23:53)



The slide is titled "Wordlength Optimization Approach" and is presented in a yellow-themed interface. It lists two main approaches:

- Analytical approach
 - Quantization error model
 - For feedback systems, instability and limit cycles can occur
 - Difficult to develop analytical quantization error model of adaptive or non-linear systems
- Simulation-based approach
 - Wordlengths chosen while observing error criteria
 - Repeated until wordlengths converge
 - Long simulation time

The slide footer includes the IIT Kharagpur logo and the text "NPTEL ONLINE CERTIFICATION COURSES".

So, to finding out the that means optimum word length. So, we can have two approach; one is that analytical approach, another one is simulation based approach ok. So, in analytical approach this we have to develop this quantization error model for feedback system, instability and limit cycle can occurs, difficult it is very difficult to develop

quantization error model of adaptive or non-linear system ok. But, simulation based approach the word length chosen while observing the error criteria, repeated until word length converges, and long simulation time ok. So, there are 2 approaches for finding out the word optimum word length for that.




(Refer Slide Time: 24:39)

Fi type

- Integer arithmetic with a fixed number of fractional digits

```
>> a=fi(pi, true, 8, 5);
>> bin(a)
0 1 1 . 0 0 1 0 1
s 2 1 . 1/2 1/4 1/8 1/16 1/32

>> double(a)
3.15625
```




So, in MATLAB this basically fi indicates the fixed point tool box.

(Refer Slide Time: 24:52)

Fi object

```
>> a = fi(pi)
a =
    3.1416015625
```

		data
DataTypeMode: Fixed-point: binary	}	point scaling
Signed: true		numeric type
WordLength: 16		
FractionLength: 13		
RoundMode: nearest		fimath
OverflowMode: saturate		
ProductMode: FullPrecision		
MaxProductWordLength: 128		
SumMode: FullPrecision		
MaxSumWordLength: 128		
CastBeforeSum: true		

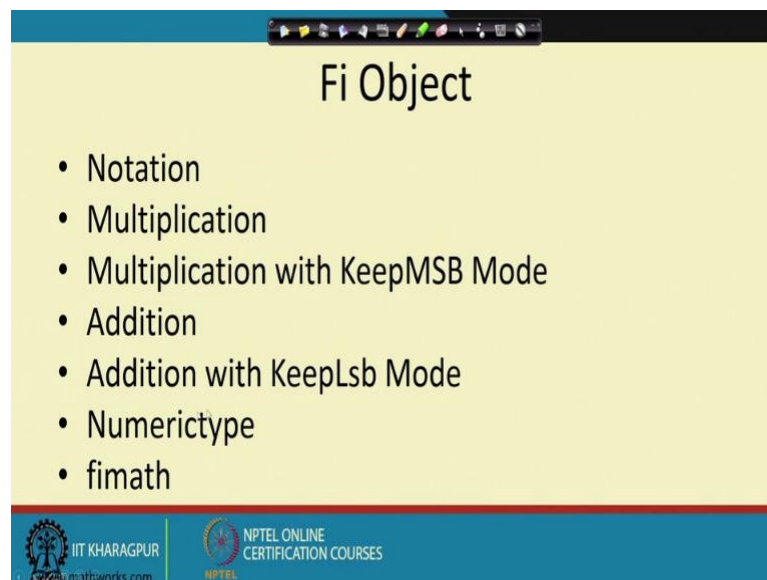




So, if you just write this comment in MATLAB ok. So, a equals to fi of pi pi means this is this pi function. So, the value will be something like 3.1416015625 ok. So, here the

data type mode is fixed point binary point scaling signed is true word length; that means, word length is 16, fractional length is 13.

So; that means, among them fractional length 13 means, it represent 13 bit for doing this particular representing this particular terms. And, another 3 bit to represent this, why because this is 3 means only 2 bit is sufficient enough and then I need one sign bit for that. So, sign and then 2 another 2 bit so, these 3 bits is basically left for the integer part. And, another 3 and another 13 bit which is basically representing the decimal part, and this then this is the rounding mode, it has chose to nearest, overflow mode is saturate. So, if you just type it this on the MATLAB you will get this information ok.

(Refer Slide Time: 26:04)



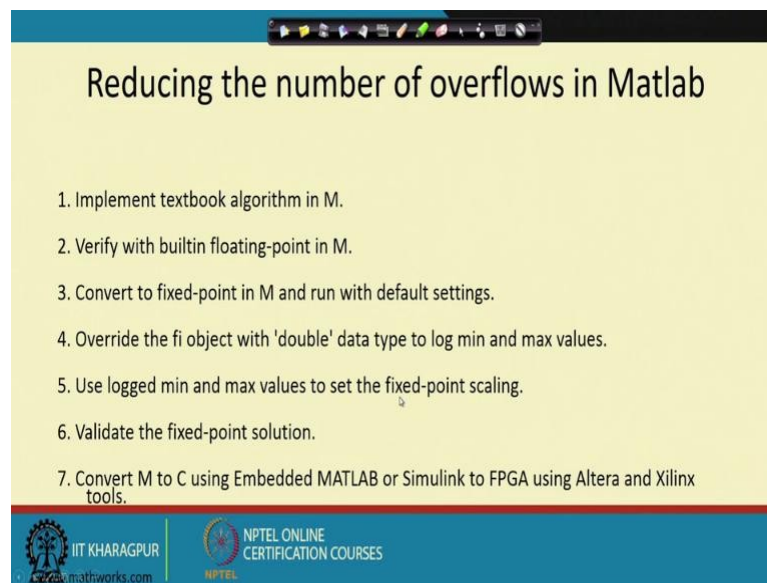
The slide is titled "Fi Object" and lists the following topics:

- Notation
- Multiplication
- Multiplication with KeepMSB Mode
- Addition
- Addition with KeepLsb Mode
- Numericity
- fimath

At the bottom of the slide, there are logos for IIT Kharagpur and NPTEL Online Certification Courses.

So, this Fi object that can be you can; that means, implement or you can that do it for multiplication. Multiplication keeping the; that means, KeepMSB mode, then addition, then addition with KeepLsb mode, then for Numericity; so, these are the things which you can do in MATLAB using this fixed point tool box ok.

(Refer Slide Time: 26:32)



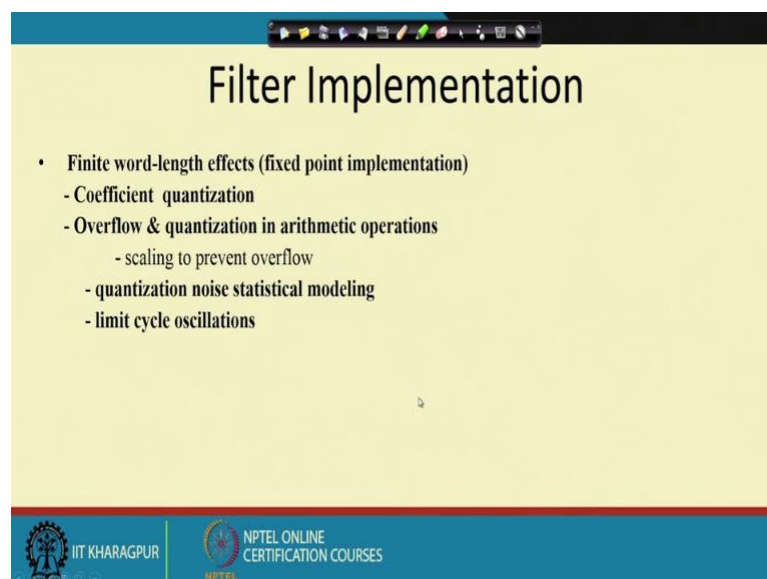
Reducing the number of overflows in Matlab

1. Implement textbook algorithm in M.
2. Verify with builtin floating-point in M.
3. Convert to fixed-point in M and run with default settings.
4. Override the fi object with 'double' data type to log min and max values.
5. Use logged min and max values to set the fixed-point scaling.
6. Validate the fixed-point solution.
7. Convert M to C using Embedded MATLAB or Simulink to FPGA using Altera and Xilinx tools.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, for finding out the overflows ok, there is one this text book algorithm ok, that I think in MATLAB that is inbuilt if you just; that means, you want to know more, you please go through the corresponding reference of this fixed point tool box in MATLAB ok. I think we will get so many help as well as the demos are also available on MATLAB. So, you can just follow for that ok. So, if you need more information about this fixed point tool box, then you can just; that means, inform me via discussion forum that you can do ok.

(Refer Slide Time: 27:18)



Filter Implementation

- Finite word-length effects (fixed point implementation)
 - Coefficient quantization
 - Overflow & quantization in arithmetic operations
 - scaling to prevent overflow
 - quantization noise statistical modeling
 - limit cycle oscillations

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, we will take one example of that, how basically this floating point to fixed point that basically effects the implementation ok. So, for that if I choose one filter implementation as an example. If, we choose so, in filter basically what happens we have the coefficient. And the coefficient values they are basically less than 1 ok. So, less than 1 means they are basically represented through floating point number, but whenever we will built the filter implementation in using the hardware. So, at that time I have to convert it in fixed point. So, for that I need to do the quantization of the coefficient. Quantization of the coefficient means; if I just represent the data in floating point so, at time the fractional part that may be 32 bit long ok.

But, whenever we will do that fixed point or the hardware design. So, at that time I cannot take 32 bit long data, because if I take 32 bit long data at that time my cost of the corresponding hardware that will be more. So, to get one optimized performance of the hardware, we have to take one decision. That ok this is the optimal word length, which we will consider for implementation.

So, that the accuracy is not; that means, I am not doing any; that means, the accuracy within the satisfying limit along with I am increasing the or I am getting the satisfying performance too ok. So, that we will see so, and then this in fir filter, not only the quantization happens with the coefficient, the quantization and the overflow also happens to the arithmetic operation ok. So, as in the filter we have to do this multiplication and then we have to do this addition, where addition is basically happening in a chain. So; that means, we have to in the arithmetic operation also, we have to do this quantization and then overflowing ok.

And, for that we have to do this MATLAB tool so, for today this is it. In the next day we will see with this example of filter design in MATLAB basically we have, you can or we can do that considering different; that means, word length for the filter or the for arithmetic operation. And, then what will be the effect on that or what will be the effect for the transfer function or the corresponding this performance of the overall filter design so, that we will see in the next class.

Thank you for today.