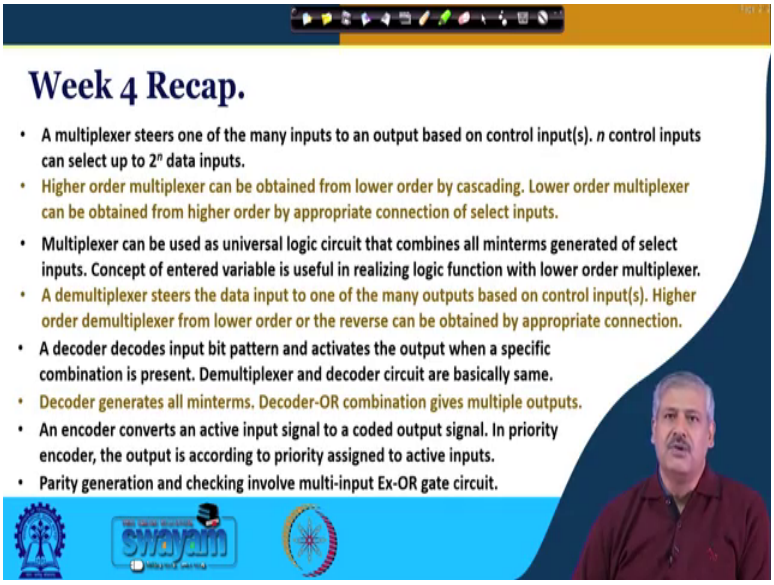
Digital Electronic Circuits
Prof. Goutam Saha
Department of E & EC Engineering
Indian Institute of Technology, Kharagpur

Lecture – 21
Number System

(Refer Slide Time: 00:25)

Hello everybody. We are in week-5 of this particular course. In this week, we shall look at Number System and arithmetic building blocks. Today's class we shall cover some of these concepts, basic number system fundamentals of number system, binary to decimal and decimal to binary conversion, octal and hexadecimal representation, and we shall have a quick look at fixed-point and floating-point representation.

(Refer Slide Time: 00:45)

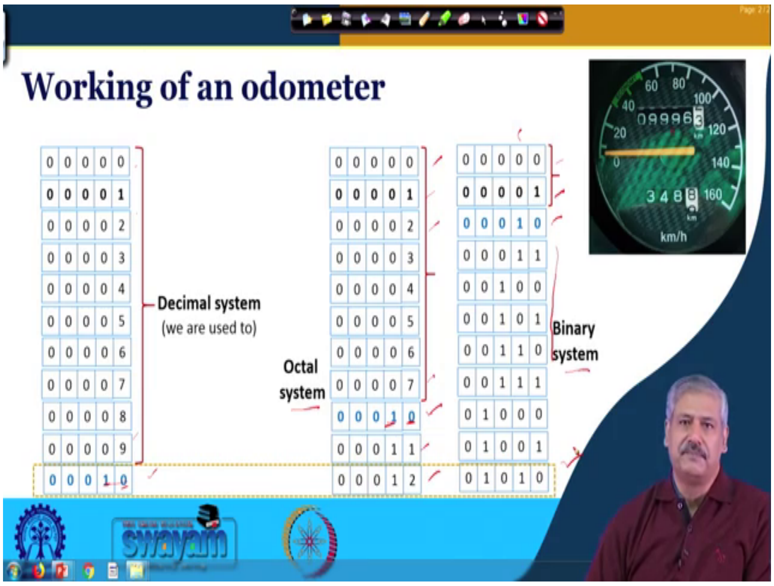


So, before that a quick recap of what we discussed in week-4. So, we looked at logic circuits, important logic circuits. Like a multiplexer, which steers one of the many inputs to the output based on certain control inputs. We looked at de-multiplexer, which is does the rivers operation. So, if the number of outputs and one input need to be steered to towards that based on what is presented at the control input for the selection input, the data input is steered towards one of the data output terminal, so that is what you had seen for the multiplexer operation.

And you also so that de-multiplexer circuit and decoder circuit are essentially same. So, decoder decodes a particular combinations or input bit pattern, when it is present at the input side. And decoder essentially generates all the minterms and this minterms can be combined together with OR gates to produce many different kind of output. So, whenever we talking about we are talking about multiple output generation, then decoder-OR combinations, so multiple OR gates which takes appropriate minterms from the decoder output that can be useful.

And we also so encoder, encoder is reverse of you know decoding operation where in a input signal is encoded in certain number of you know a bits. And in priority encoder, if number of inputs are active simultaneously, then according to the priority assign one of the input gets encoded to the output. And we also looked at parity generation and checking using Ex-OR gate, and we found it is useful for correcting or not connecting detecting one bit error in the transmission or in the bit pattern.

(Refer Slide Time: 02:47)



So, to discuss number system first we look at one example of an odometer, which we have perhaps all of us have seen those who drive vehicle or we have you know seen, how the meter which takes note of the distance in the vehicle is you know showing the increments in the number.

So, you have seen that if 1 unit distance is travels say 1 kilo meter distance is traveled, so the number gets increased by 1 unit over here ok, and then next unit and it gets incremented. So, when all is this in decimal system, so when it you know complete you know 0 to 9, then next digit next digit over here say this one will get incremented ok, so that is what to have already observed.

So, in decimal system if you look at the way the digits are presented, so 0, 1, 2, 3 ok, so these are the valid digits up to 9, after that comes 0 in this unit place and in the decimal place in the 10s place 1 comes into the place ok. So, this is how odometer decimal odometer works.

So, if we considered instead of you know 10 digits decimal system, we had octal systems that means, only 8 digits are available, and we let us number them. Let us consider the representation of them is 0 to 7. So, what would have been the case here? So, up you know 1 unit distance travelled, it goes from 0 to 1 which is fine, then 2 unit goes from 1 to 2, but when it goes up to 7 unit, which is there in the representation 0 to 7.

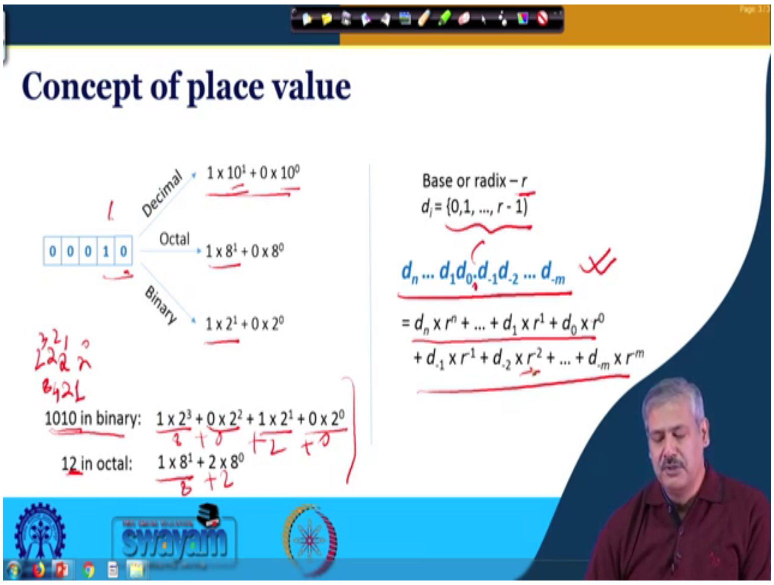
After that we need travels one more unit 1 more kilometer, what will happen? 8 is not available it is octal system, 0 to 7 is only available, so at that time what will happen? The second place over here ok, 8's place, so we will say that it is a eight full if earlier one was 10's place, this is a 8's place that gets incremented by 1 from 0 to 1 and over here it comes you.

So, the meaning here this 1 0, and this 1 0 is different. This 1 0 is 10 9, after that one more unit. Here 7 after that one more unit, so this is 8 is it clear. So, if we took talk about binary system in the digital circuit digital system you are talking about binary 0's and 1's switched on or switched off this is the way we are looking at it, so a binary system basis 2 ok.

So, 0 after that 1 unit travel, so it is 1. So, after one another unit has when it is traveled, what will happen? 2 is not available as a digit, so this 2's placed, now will call it 2's place right 10's place, 8's place, now the second place is 2's place. So, 2's place will get increment by 1, and 0 will come over here ok. So, this 1 0 over here is 1 plus 1, which is 2. And this way, it will continue.

And if you look at you know this incrementing of the next higher position place ok, then at 0 0 1 0 which is decimal 10 for that in octal will be having this is your 8 1 0, then 9 is 1 1, this is 1 2 that is in octal. And in binary system, if you go on you know incrementing the way we have incremented the earlier cases 1 by 1, we will see that it is 1 0 1 0 ok.

(Refer Slide Time: 07:12)



Now, earlier we was talking about 1s place, 10's place, 8s place, 2s place that is a second position. So, the number appearing at units place, and the number appearing at the second place in decimal which is 10s place in octal that is 8s place, and in binary it is 2s place. So, this place provides a different weightage ok, a different place value to that particular number ok.

So, if we look at the number that is 1 0, so in decimal the 1 here is appearing at 10s place. So, 10 to the power 1 multiplied by 1, and this is 0, and it is 1s place value is 1, so 0 into 10 to the power 0. So, this is what actually it means in the amount, if the amount here is this much. Over here if it is in 8's octal system, it is 1 into 8 to the power 1 plus 0, and in binary system it is 1 into 2 to the power 1, so that is what we have already seen.

And by this we see if we go on adding the place value over here 1 0 1 0, so this is units place, this is 2s place, this is 4th place, and this is 8th place ok. So, then this actually goes into like this 2 to the power 0, 2 to the power 1, 2 to the power 2, and 2 to the power 3. So, 1 0 1 0, 1 into 2 to the power 3 0 plus 0 into 2 to the power 2, 1 into 2 to the power 1, and 0 into 2 to the power 0. If you look at the odometer example that we had seen in the previous slide, this is how this if you add them up, this is 8 plus 0 plus 2 plus 0, so 10, so it was decimal 10 equivalent that we had seen in the last row ok.

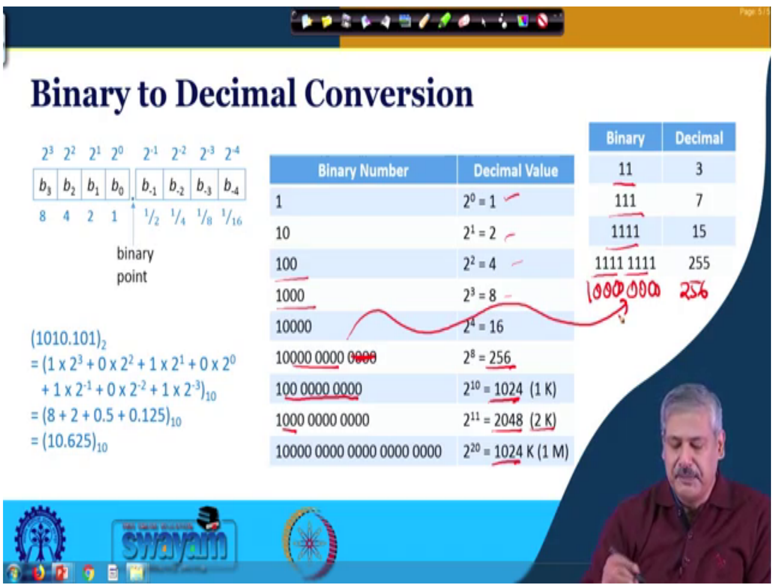
Similarly, in octal it is 1 into 8 to the power 1 that is 8 right and 8 to the power 0 is 1, and 2 is a valid digit, so 2 into 1 is 2, so 8 plus 2 it was 10. So, the last row that is how we had a equivalent of 10 decimal equivalent of 10 for representation 1 2 in octal, and 1 0 1 0 in binary ok. So, this is the concept of place value, when we put it more formally.

So, this is what we would be able to see. So, if we have a number system with a base or radix r by which the allowable digit valid digits are 0, 1, 2, 3 up to r minus 1, for example in octal system 0 to 7 in binary 0 and 1 only ok. So, then the number represented in this manner, where this is the point which divides the integer and fraction part. So, decimal point in decimal system, octal point in octal system, binary point in binary system, so that is the point over here ok.

Then it is corresponding value is represented by from these example that we had seen here, if we continue that this is the way the value is arrived at. So, the there are n plus 1 digit to the left of the decimal place, and that is the integer part the decimal point or binary point or octal point, and then m digit towards right ok.

So, in that case d n into r to the power n r is the base plus d n minus 1 into r n to the power minus 1 that we will continue right up to d naught into r to the power 0 plus. When it goes to the fractional side, what happens? So, d minus 1, this is the first digit which is multiplied by r to the power minus 1, so the plus value here is 1 upon r right. So, this is we have to keep note of the next value plus value is 1 upon r square, so r to the power minus 2. So, this way it continuous and we arrived at the value by this manner.

(Refer Slide Time: 11:57)



So, you are more interested in the binary presentation more most of our work will be using for the binary system. So, we elaborate it little bit more. So, here is a binary number only 4 digits before the decimal binary point and 4 digit after it has been shown. And the corresponding weights, you can see which is written in this places ok. So, these are the corresponding weights ok.

So, if we look at this example 1010 which is 101 0. 101, what will be its value? So, 1010 we have already found out the integer part before right, which is 10 and the fractional part. This is 1 into 2 to the power minus 1 plus 0 into 2 to the power minus 2 plus 1 into 2 to the power minus 3 ok. So, then if you add them up 0.5, and this is 0.125 1 upon 8 is 0.125, so it is 10.625. And this is the way number, so this 2 means base 2 and this 10 means base 10. So, 1010.101 base is 2 is equivalent to 10.625 in base 10. So, this is the way we can convert from binary to decimal ok.

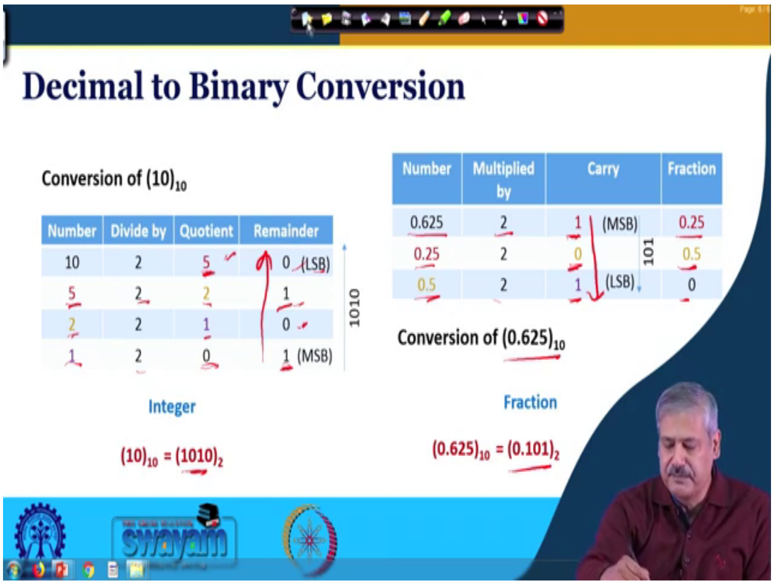
And some important you know this conversion integer part over here it is shown, which we keep in mind. So, only 1 is 2 to the power 0 it is 1, 10 is 2, 100 is 4 ok, 1000 is 8 so this way the sum of the things that we often in encounter.

And here we remember here we remember that this is 2 to the power 8, so this is incorrect. So, when it is 2 to the power 8 ok, it is 256. So, when it is 2 to the power 10, so this is your 2 to the power 10 so it is 1024; so 1024 here also is represented as 1 k. So, when we say 1 kilo bit right, it does it means 1024 number of bits, it is it does not mean 1000 bit that is there in the decimal system.

So, in binary or digital logic digital representation of numbers 1 kilo stands for 2 to the power 10 which is 1024, so 2 kilo 2K represent 2 to the power 11 ok, so that is 2048 if you multiply, if you can you can see that this is that 2048. So, 2 to the power 20 is 1024 kilo that is 1 mega. So, 1 mega is represented in this manner, so that is what we take note of and also that there will be when we talk about 256, after 1 it is 2 to the power I mean 8s 0s are there ok.

And this is another thing we often come across 11 is 3, 111 are 7, 1111 are 15, so if 4 1111 1111 are there, this is 255. So, if you just increase increment by 1 the way we have seen the odometer, so after that will be the next value 10000 four 0 here and four 0000. So, here the play what will get implemented to 256, this is your 256.

(Refer Slide Time: 16:06)



Now, that was binary to decimal. If it is required that we have a decimal number we want its binary equivalent, how do you do ok. So, we do it by let us first look at the integer parts. So, when we look at conversion of say integer parts 10, so we first divide 10 by 2 note the quotient and the reminder. So, this quotient 5 comes over here university colour you know.

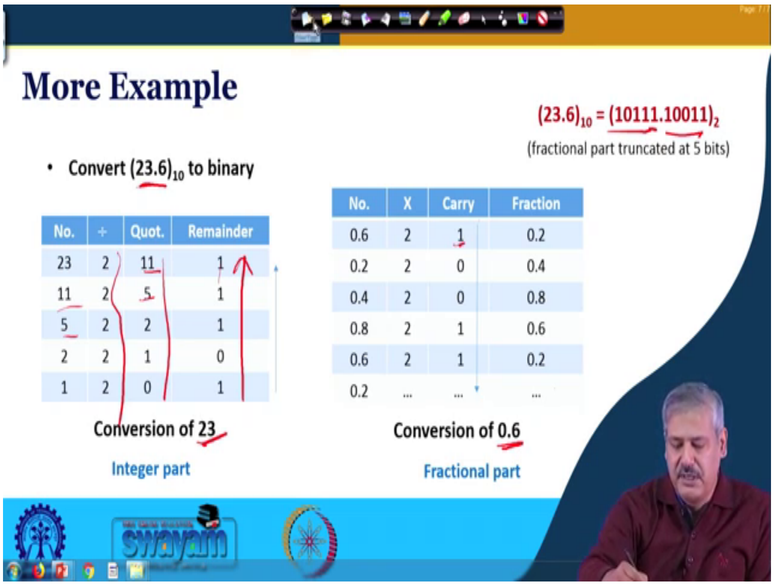
So, again we divide by 2, note the quotient and the reminder. So, earlier when 10 was divided by 2, 5 reminder was 0 this time when it is divided, reminder is 1. So, this 2 again we divide by 2 right, this is 1 and remainder is 0, because it is completely divisible 2 is completely divisible by 2. And then 1 again when we divide by 2, basically we get the quotient you know 0 and the reminder remains 1 ok, so that is because only that the digit is less than 2 right.

So, whenever we reach here there is no further you know scope, we stop this exercise between successive division by 2 of the original number and the quotient that it gets. And the reminders from bottom to top the way that has been shown here, we take note of with the first bottom most 1 is MSB, and top most 1 is the LSB that is 1010 that is the corresponding binary equivalent of the decimal, that is how we look do the conversion for the integer part.

And if we do consider the fractional part earlier, it was division, here it is multiplication right. So, 0.625 if we consider the number we have seen before, so 0.625 first we multiply by 2, so what will be the result? 1.25. So, 1.25 we look at the number which is the integer part of it, we will call it carry we keep it here and the fractional part we keep it here.

So, 0.25, then we multiplied by 2 again, so we get 0.5. So, there is no carry. So, this is 0, and this is 0.5. And then 0.5, we multiply again with 2, we get 1 and this is 0 ok. So, in this is after that there is no further fractional part available. So, this time we will go from top to bottom. So, the topmost one is the MSB MSB in the sense that after comes after the you know the binary point and the bottom most point one is the LSB. So, 0.625, it is corresponding binary equivalent is 0.101. So, this is the way decimal to binary conversion is done ok.

(Refer Slide Time: 19:11)

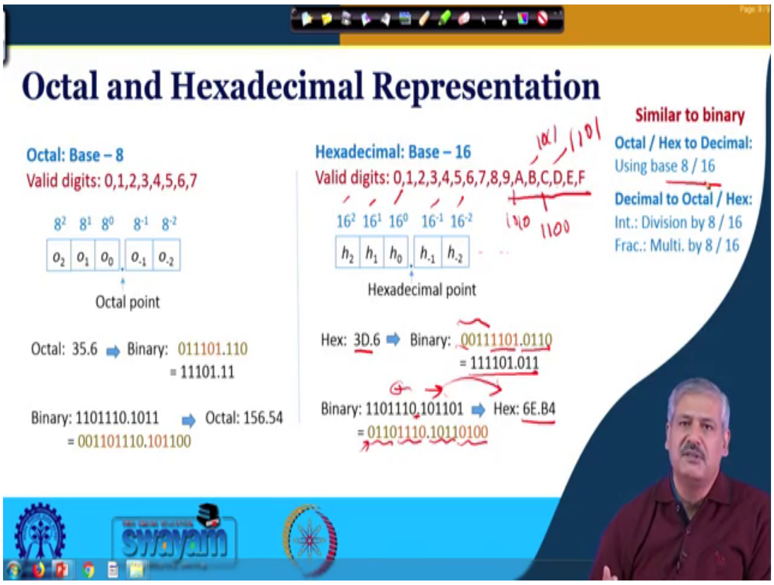


So, we have one more example here 23.6. So, here what we do again we follow the same process, we divide the number into integer part and fractional part. And you keep on you know dividing it by 2 successively, and get the quotients, the quotient again come here, this quotient comes here right. And we note down the reminder, and then we read from top to bottom to top. So, it will come as come as 10111. So, this part this is what will come.

And for the fractional part 0.6, we keep on doing the multiplication right. And whenever there is a carry that is value becomes more than 1 right, we note it here. And then we read it from top to bottom. And we see that when you come over here you it is not actually fractional part is not becoming 0 right. So, it will go on giving some value over here right, so but in practical cases will be needing to truncate it because of some finite number of bits are available to you only in the realization.

So, if it is truncated at 5 bits ok, so from top to bottom if we write, then it is 10011 that is what we see here. So, if there are 6 bits, then another you know if you do the multiplication then what about number come, so that that will right here we will write here 7 bits accordingly between will be presented right.

(Refer Slide Time: 20:56)



So, octal width just seen in the discussion where we have seen the odometer. So, if in some case octal number base representation is required ok, then we know the valid digits are 0 to 7, this is we have already discussed. And these are the corresponding weights for the place value. So, after octal point, it will 8 to the power minus 1, 8 to the power minus 2 the 1 that I have said before. So, octal the valid digits are 0 to 7 only.

So, considered a octal number 35.6 ok, then what is the corresponding binary? It is very easy to convert from octal to binary, and binary to octal. So, what we need to do only, we form a group of 3 group of 3 ok, because in octal this 2 to the power 3, 8 is 2 to the power 3 right binary is base 2 and this base 8. So, 3 we represent it in this manner, 5 represented by this is the octal point which becomes a binary point, and 6 is this one right.

So, first we represent in this manner. And after that the leading 0, before the binary point and the following 0, after the binary point, we that does not carry anything, this is the corresponding binary equivalent. And if you have to convert from binary to octal, so this is the binary number. So, this is binary point is the difference point, so from this binary point we keep forming group of 3 ok. So, this is one group, this is another group, this is another group the integer part.

And when you see that only one member is there, so you can put some leading 0s to make a group of 3. And after the I mean the regarding the fractional part again this is the reference point, so you form the a group of 3. Then again to form a group of 3, you do not have 3 digit is there, only 1 digit is there you put some two succeed in 0's a group of 3 is formed right. And then for each of these group you represent the corresponding octal equivalent. So, 001 is 1, 101 is 5, 110 is 6, and 101 over here is 5, and 100 over here is 4 ok. So, this is way from binary to octal conversion can be done.

And there is another number system you may come across free in the digital electronics discussion, so that is called hexadecimal. So, hexadecimal as got a base 16, and 16 means then 0 to 15, I mean some 15 to 16 digits will be there, but decimal numbers are up to 9. So, after that what is done the capital letters A, B, C, D, E, F are used which is equivalent to 11, 12, 13, 14 and 15 right.

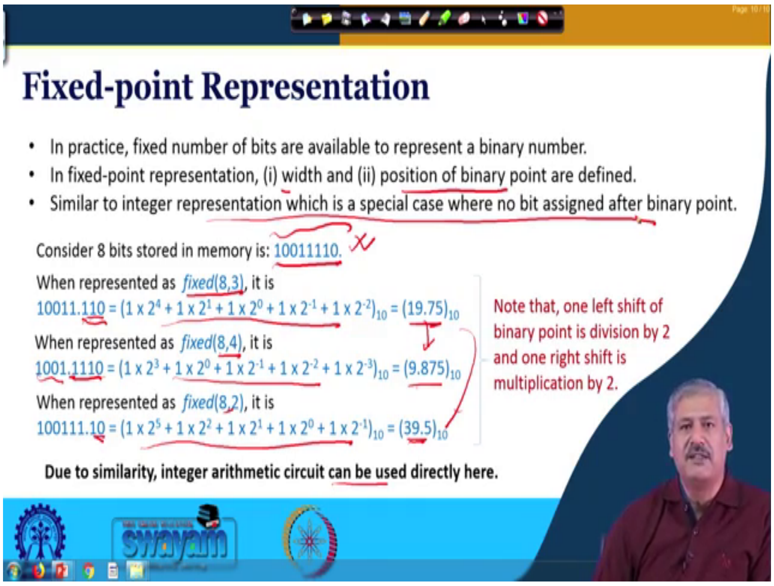
And if you look at the number representation, the weights will be 16 to 2 the power 0 over here units place 16 in the 16th place second place, 16 square in the 3rd place, before this hexadecimal point and 16 to the power minus 1 here, and then to the minus 2 after the hexadecimal point, it will go on right.

And the conversion is similar to octal system, again because it is 2 to the power 6, 2 to the power 4 is 16, so will be forming group of 4 group of 4 binary digit. So, if 3D.6 is to be represented in binary. So, 3 will be represented by 4 binary digits D ok. If you look at the corresponding value over here, it is I mean in the binary representation this is 1010 this is 1 0 1 1, 1100 and D is 1101 ok. So, this is your D right. So, and then 6 is 0110 in 4 binary digit. So, again leading 0 and following 0's before and after the binary point you ignore, and then this is the corresponding number.

And binary to hexadecimal conversion, again will be forming group of 4 with starting from the binary point as reference. For integer part you will go towards left, and for fractional part you will go towards right keep forming groups. And if required you add some leading 0 or succeeding 0s, and then you converted to corresponding hexadecimal, and the job is done.

And to get a decimal to binary conversion, decimal to binary conversion is already by you know integer power to the 8 sorry 8 to the power 1, 2, 3 etcetera or 16 to the power 1, 2, 3, so that kind of what kind of coefficient you add them up, you will get the corresponding decimal value. And decimal to octal earlier you have done for binary, it was division by 2 for the integer part, it will be division by 8 or 16 for the integer part. And for the fractional part it was multiplication by 2, it will be multiplication by 8 or 16 the method the similar.

(Refer Slide Time: 26:34)



Having noted this and also earlier that specially the fractional part, you can do not have you know all the number of bits that is required to get the kind of question that require. So, in actual practice will be having a fixed number of bits available to represent a number. So, with this fix number of bits in a presentation, which is called fixed-point representation of number; the width that is the number of bits total number of bits available to represent the number.

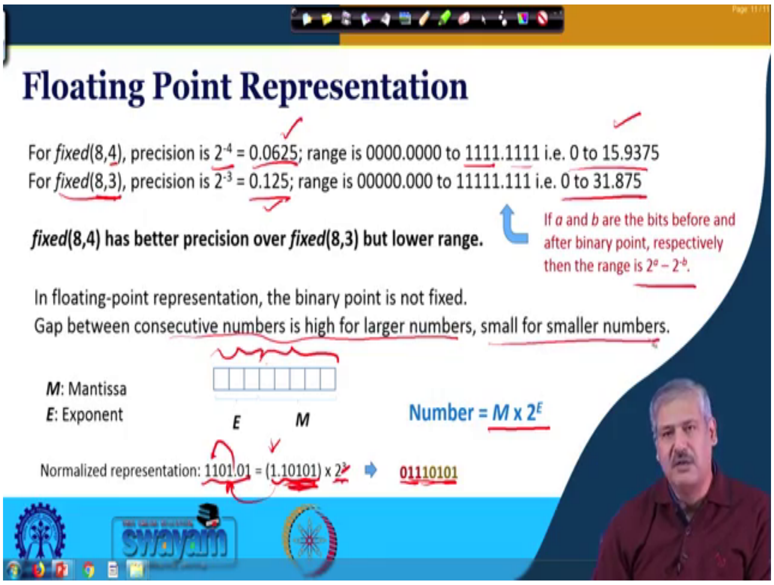
And the position of the binary point are defined ok. So, to give you an example if the 8 bit stored in the memory is this 10011110 ok, and it is represented in fixed 8, 3 representation. So, this is the understanding that this is this number stored in the memory are represented in this manner right. Then of course, the width is 8 ok, so 8 bits to be taken. And then binary point will be fixed at you know after the binary point, there will be 3 digit, so 110. So, this is the corresponding value in the decimal, if you just convert binary to decimal this is what you will get.

See instead if it was 8, 4, so 4 digit should have been there. So, same 8 bits, because with the demons are same right that is what you have mentioned here, the width to be 8, so then what happens. So, this value becomes integer value becomes less, and the fractional part we have got we know more interest. And then if you add them, this is 9.875 that what is you get in decimal equivalent.

And instead if it was 8, 2, it was 8, 3 8, 4, this is 8, 2 so 2 digits will be there after the decimal after the binary point. And again if you convert to binary to decimal, you get 39.5 ok. So, the same number depending on the number system you know the fixed-point representation has different values. And thing you note here that by shifting these binary point ok. If you are shifting it towards left from here your shifting its towards left what has happened the number has become divided by 2. You can see that this number 8 point 9.875 is half of this one.

And if you shift it towards right, this was 0.110, this is 0.10. This is 39.5 it is multiplied by 2. So, if you shift 2 units, it will be 2 to the power 2 that will come into picture you know 2 units to the right. So, it is multiplied by 2. And if it is 2 unit to the left it is divided by multiplied by 4; and if it is 2 units to the left it is multi a divided by 2 to the power 2 that is 4 ok. And one good thing about this fixed-point arithmetic is that integer arithmetic circuit can be used because integer representation is just a special case of this where there is no bit after the binary point ok.

(Refer Slide Time: 30:04)



So, the last point just to conclude the there is another representation called floating-point representation. So, what we have seen in fixed-point representation that is 8, 4. So, there are you know 4 digits after the binary point. So, the precision that is available is 2 to the power minus 4 that is 0.0625. And 8, 3 three digits, so precision is 0.125. So, here the precision is better and here the position is bit less. But in doing that what you see the range over here is you know less compared to the range in the other case. And in fixed-point representation, once you fixed the representation, the precision is always fixed ok. So, this is what we have noted.

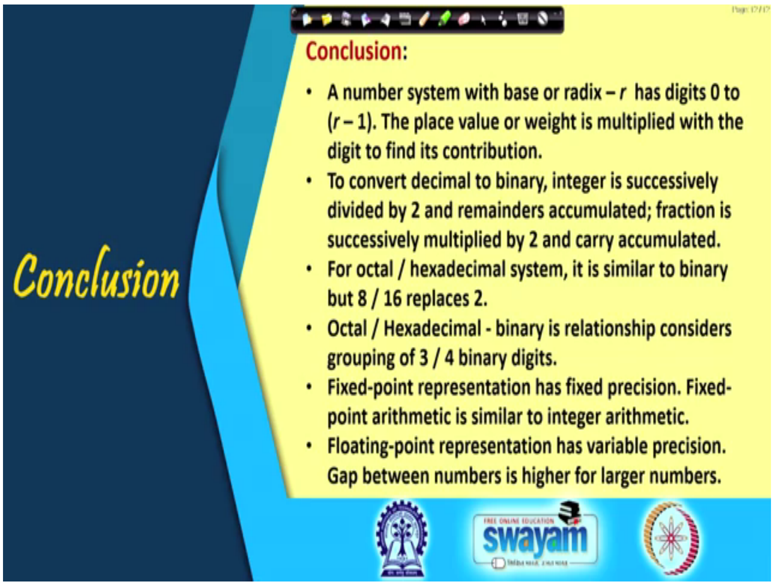
And one simple formula to calculate the range is the number of bits that is before the binary point is A and after the binary point is B, then it is can be calculated in this manner right. So, now in so then came another representation, it is called floating-point representation where the binary point is not fixed, binary point in the number in the bits it will it can move from left or to write.

The idea is always will be having in a normalized version a 1 present here and rest is the number of you know the number of in which is after coming after the fraction. So, if the original number is 1101.01 right, we shall shift the number you have already noted the shift. So, we are we shall shift the number in such a manner the decimal point comes over here. So, normal only 1 will be present before the decimal point ok. So, this is the corresponding number. So, how many shifts are there. So, 3 shifts are there. So, 3 shift means to the left ok, so that means, it is 2 to the power 3. So, this into 2 to the power 3 is same as this number right.

So, in floating-point representation we note this 3 the exponent in the exponent part, and what is after this 1 that is 1 0 1 0 1 this part right, so that is noted in the mantissa part. So, this we mention how many bits are reserved for exponent part, how many bits are reserved for mantissa part also ok. We have not discussed yet the negative number or the sign that part will come later. So, here we are talking about the magnitude part only. So, the number will be represented in this form M into 2 to the power E, M is the mantissa part that is what is coming after the fractional point, when a 1 is present just before the binary point and the number of shift that is required for which the exponent terms gates define defined ok.

So, this number will be stored in the memory in the floating-point representation by this convention is as 011; 011 is the exponent that is 3 over here and the fractional part with 1 in the left hand side 10101, so that is what it will that is how it will be stored in the memory, so that is the floating-point representation. Here the gap between consecutive numbers is high for larger numbers and small for smaller numbers. So, precision is not fixed. So, we shall mostly discuss with deal with fixed-point representation in our subsequent discussion later.

(Refer Slide Time: 34:00)



So, quickly conclude number system with base or a base or radix or has digits 0 to odd minus 1. Place value or weight is multiplied with digit 2 finites contribution ok. So, it is multiplied the 0, then there is no contribution. To convert the decimal to binary, integer is successively divided by 2 and remainders accumulated; and for converting the fraction, it is successfully multiplied by 2 and carry is accumulated. And for octal and hexadecimal system the processes similar for the conversion but instead of 2 we are using 8 or 16 for octal and hexadecimal respectively.

And octal hexadecimal to binary relationship I mean to (Refer Time: 34:49) it can be considers grouping of 3 or 4 binary digits and fixed-point representation as fixed precision. Fixed-point arithmetic is similar to integer arithmetic and we shall discuss elaborately that part later. And floating-point representation as variable precision and gap between number is higher for larger numbers.

Thank you.