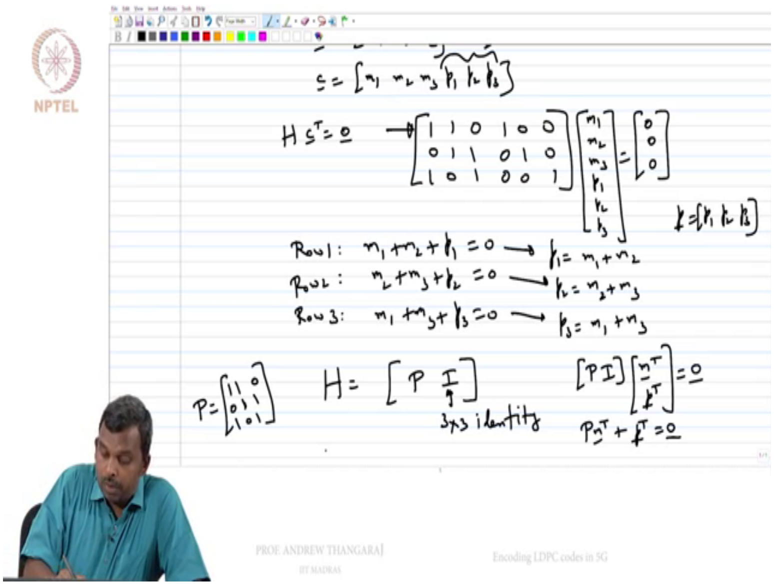
LDPC & Polar codes

Prof. Andrew Thangaraj
Department of Electrical Engineering
Indian Institute of Technology Madras

Encoding LDPC codes in 5G

And will to this lecture, I am going to talk about encoding LDPC codes in the 5G standard, we will begin by looking at couple of simple ways of encoding using the Parity check matrix and then show small toy example and then I quickly generalise to the encoding of LDPC codes. Okay, so we saw encoding of Hamming codes before in the previous week and the first week of this course, the encoding that we will use here will be slightly different, we will not be using the Generator matrix, will be use the parity check matrix for the encoding, it is largely similar, except for some minor changes. Okay, so let me show you how that works first for a very, very small example and then we will move on to the codes which are in the 5G standard.

(Refer Slide Time: 1:03)



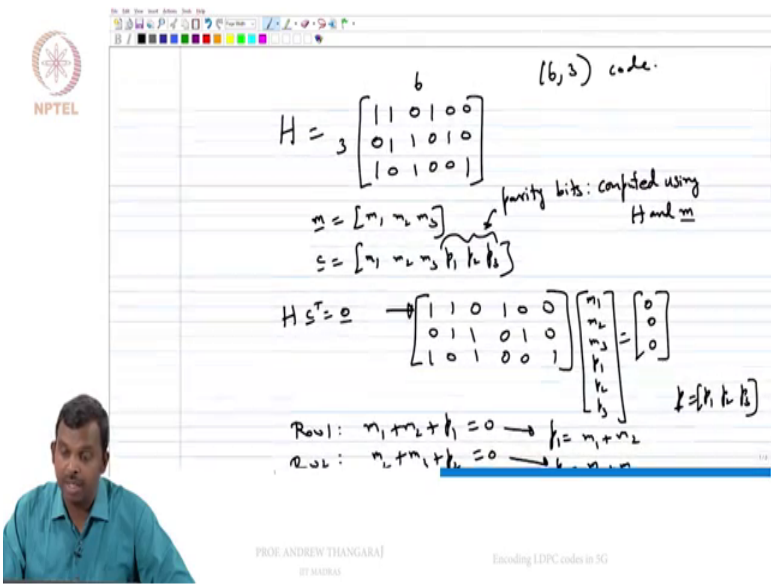
Okay, so heres the small example I am going to take, let us say we have a code with parity check matrix, which is very, very simple 110, 011, 101, 100, 010, 000, I am sorry 001, this is the parity check matrix you can see very quickly that it is a 6, 3 code. Okay, it is also in systematic form, so if you have the message vector M, which is M1, M2, M3 the code word C will be M1, M2, M3 and P1, P2, P3. Okay, so these are the three message bits M1, M2, M3, these are the three parity bits and these are computed, we will show how this can be easily computed using H and M.

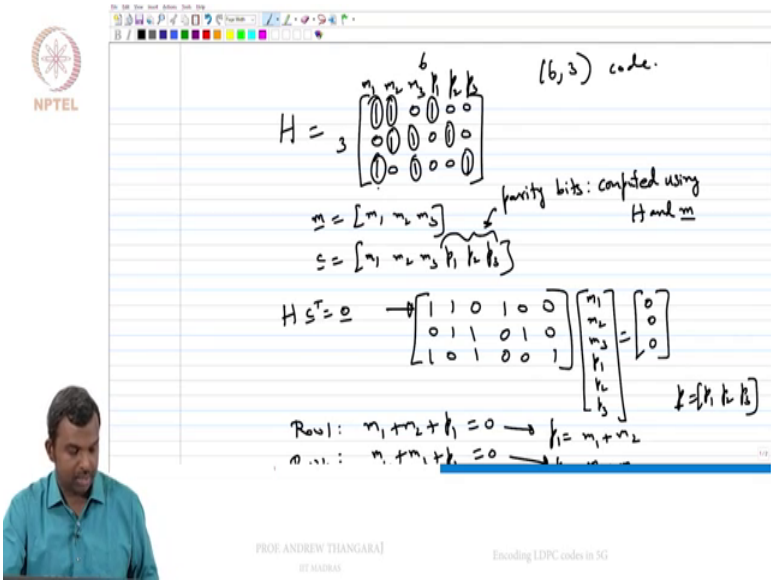
Okay, now what us H tell you, H tells you that, H times C transpores equals 0. Okay, which another words, it is this equation, right, so 110, 100, 011, 010, 101, 001, multiplied on the right with and M1, M2, M3, P1, P2, P3 goes to 0, right 000. Okay, so now if you look at what this means, this actually means three equations, what is the first equation, you multiply the first row with this guys, so the first row gives you M1 plus M2 plus P1 equals 0, the second row gives you M2 plus M3 plus P2 equals 0, and the third row gives you M1 plus M3 plus P3 equals 0.

Okay, so now, this is all a binary exhaust or modular two equations, so this is the same as P1 equals M1 plus M2, this is the same as P2 equals M2 plus M3 and P3 equals M1 plus M3, so this is the way to do encoding with the parity check matrix, if you well, it is just a small rephrasing of the encoding with the generator matrix, you look at each row of the parity check matrix and if it is in systematic form, you know that one parity bit is directly specified by that.

Okay, so there is a various ways of writing this things down, so for instance one more way to write it as you can write H as P and then I, right, so this is 3 cross 3 identity and P is just this matrix, right, so P would be 110, 011, 101. Okay, and then you can look at the multiplication here in a slightly different way PI times M transpores, P transpores equals 0. Okay, so what is P? P is P1, P2, P3 the vector of parity bits and then you rewrite this, this simplicate that P transpores P times M transpores plus P transpores is 0.

(Refer Slide Time: 4:49)





And you can quickly see that the parity bits, P transpores is obtained as P times M transpores, so another words, if you want to write it differently, P1, P2, P3 is P times M1, M2, M3. Okay, so this is just another way of writing, it is all the same thing written in different ways. Okay, so this is the way in which one uses the parity check matrix, in fact encoding of LDPC codes in the 5G standard also uses the same principle, the parity check matrix is specified, the code word is specify, the message bits are given and then you use the parity check matrix to compute the parity bits from the message bits.

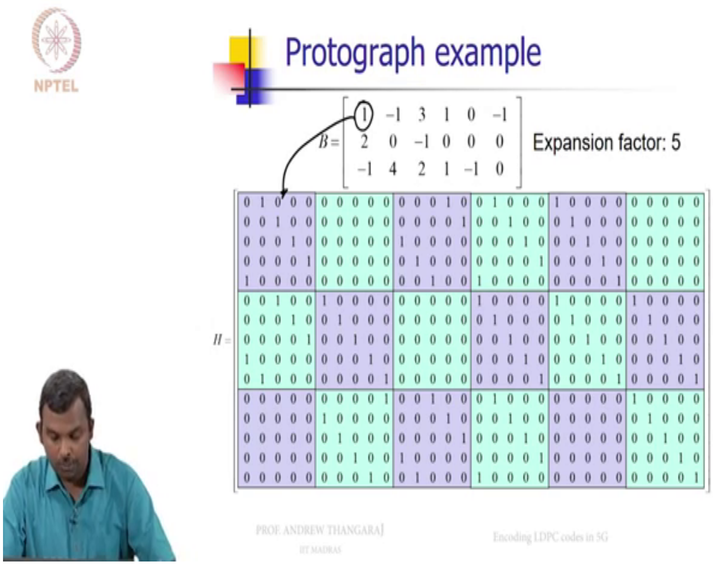
Okay, except that the parity check matrix is not directly in systematic form, you have to do some small manipulation, which is the manipulation, I will show you in this lecture. Okay, so hopefully this is clear, so one important thing to keep in mind is so like I said this is 3, 6 code usually when people put messages and the code parity bits on the top of the parity check matrix, you imagine that the first column corresponds to M1, second column corresponds to M2, third column corresponds to M3, the fourth column is P1, P2, P3 etc.

Okay, so this kind of visualisation of the code word with respect to the parity check matrix is useful. Okay, so you write down the parity check matrix, the columns of the parity check matrix corresponds to the bits of the code word H times C transpores is 0, so that is the multiplication, right the bits of the code word multiplied the columns of the parity check matrix and you can associate each bit of the code word with each column of the parity check matrix, this is the same thing we did in the canagraph as well.

Okay, this is a very natural association, so you imagine putting the messages on the systematic part of the code word and then computing the parities using the parity check matrix, so you see here the first thing is your P1 as M1 plus M2. Okay, so P2 is M3 plus M2, okay, P3 is M3 plus M1. Okay, so directly gives that. Okay, so this is how, this is just a basics of encoding.

So now let see how to use this, the parity check matrix specified in the 5G standard, there is a one little thing to take care of, I will point that out alone and then you will see the rest of it follows very naturally in a normally easy manner.

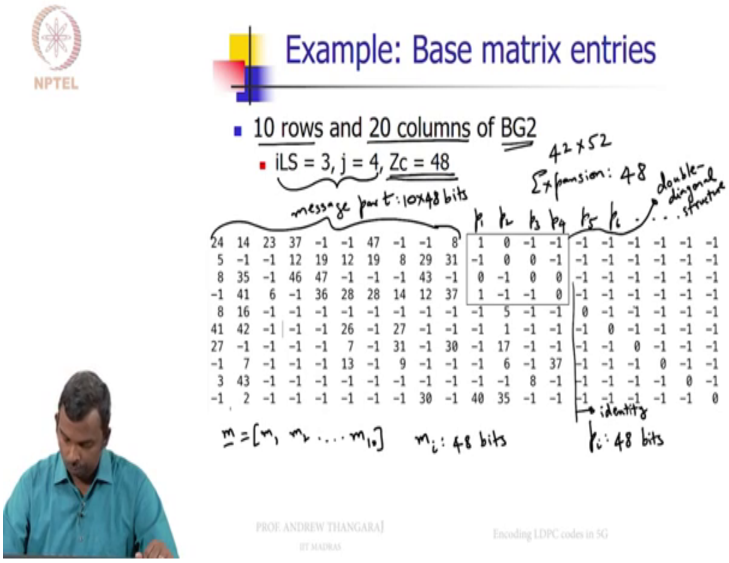
(Refer Slide Time: 7:11)



Okay, so let us get started with that, so in the previous lecture I gave you this example of a base matrix, a toy base matrix with an expansion factor 5 to illustrate how the parity check matrix is specified, this is a 3 by 6 based matrix, each entry is from minus 1 to plus 4, expansion factor is 5 and each entry is replaced by a 5 cross 5 binary matrix, minus 1 is replaced by the all 0 matrix and any other number, say zero or one or two or three or four, this replaces by the 5 cross 5 identity matrix, column shifted, column rotate shifted that many times to the right.

Okay, so one would correspond to the matrix, identity matrix, column shifted by one position like for instance, this, so this one for instance goes out as this matrix, all the other matrixs are like that. I showed that in the previous lecture.

(Refer Slide Time: 8:10)



Okay, so what will do in this class is, we will start looking at the base matrix for the actual LDPC codes in the standards, so this is another slide I showed in the previous lecture, in the, this is part of a base matrix for the second BG2 which is 42 by 52 matrix, it is quite large, it is difficult to capture on the screen, so I am showing just 10 rows and 20 columns of that matrix, the expansion factor is 48. Okay, so is expansion factor is 48, so this two things are just indices, which tell you how to look it upon the standard etc, expansion is 48, so you will see the entries go from minus 1, 0, all the way to 47. Okay, and these are the first 10 rows, so if you notice this is the message part. Okay, this is the message part, there will be 10 message blocks. Okay, so 1, 2,3,4,5,6,7,8,9,10. Okay, each message block actually will expand to a 48 by 48 binary matrix.

So overall after expansion the message part will be 4 under an 80 bits lot. Okay, 10 into 440. Okay, so that is the way 48, I am sorry 10 into 48 bits. Okay, so usually will consider this in some blocks fashion as an first message the itself will be taken as M1, M2 till M10 each MI will be 48 bits. Okay, so I think of it as a vector just concatenate to put together the message part. Okay, and then you will have the parities, this is the first parity, second parity, third parity, four parity, fifth parity so on.

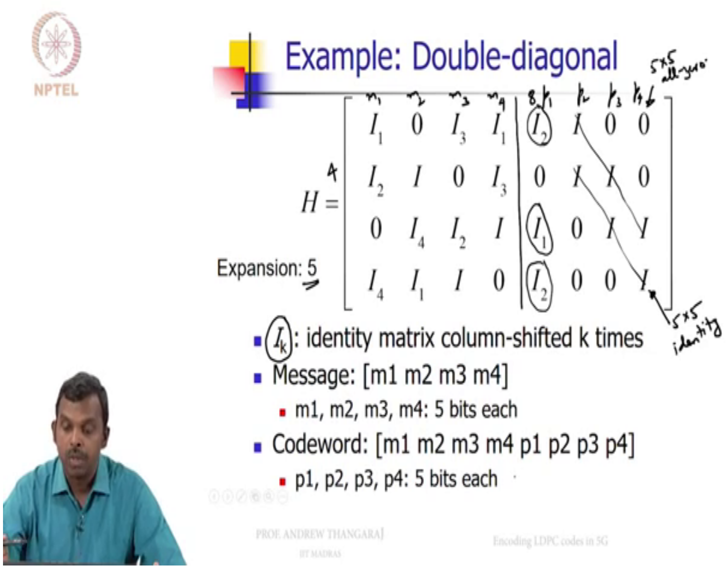
Okay, so each PI is also 48 bits. Okay, so it is all expanded in the base matrix you have to think of each of this things as a block of bits. Okay, if you look at it, except for this 4 by 4 part which is shown in the box here, this, except for this 4 by 4 part, I actually have an identity matrix. Okay, so from this part now on, it is just an identity. Okay, so it is similar to the small little example we saw the beginning of this lectures, this identity and P.

Okay, but this first part this not clearly identity, know you have this minus 1, this minus 1 gets replace by zero, but you have a zero which is identity, another right below that, it does another row of zeros. Okay, so this is a double diagonal structure. Okay, so this structure is called a double diagonal structure. Okay, so this is important for the, form a performance point of view, so you need some sort of double diagonal structure like this, for the code to have good performance, particularly at high rates you need this structure gives you good performance, so but nevertheless this makes encoding a little bit more complicated but then is out there is a certain structure here which you can exploit and encoding is actually quite simple.

The double diagonal structure will work out very nicely, I will show you that, so that is the only complexity in encoding of the LDPC code in the 5G standard, that how to deal with the double diagonal structure in this part here and at is also small part, so once you deal with that and find out P1, P2, P3, P4 the rest of it straightforward. Okay, so once you find out P1, P2, P3, P4 the rest of it is just identity you can find P5, P6, P7 so on as much as you need. Okay, so this is the basic idea behind the encoding.

So to illustrate this double diagonal we will take a toy example, a much smaller example as suppose to a direct example from the standard, we will take a smaller example, illustrate how does work and then come back and tell you how to apply it in this case as well. Okay, so this is how it is goings to work, of course we will write a Matlab program to do this encoding, so then a but this is how it works.

(Refer Slide Time: 12:23)

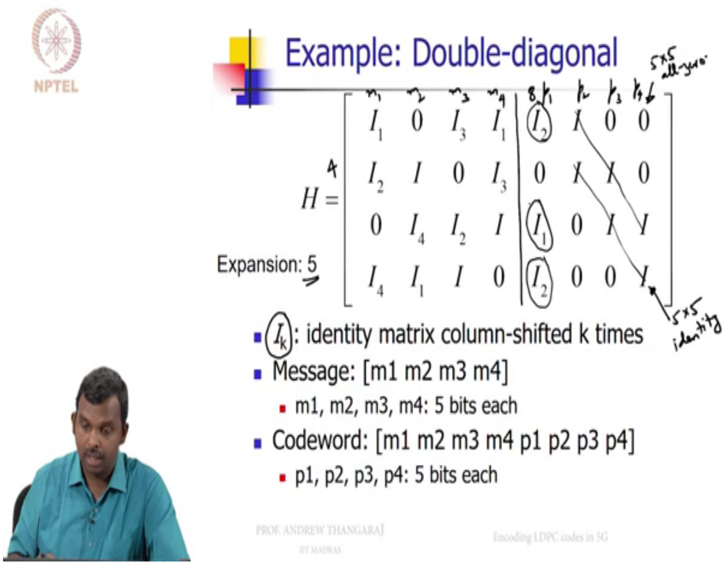


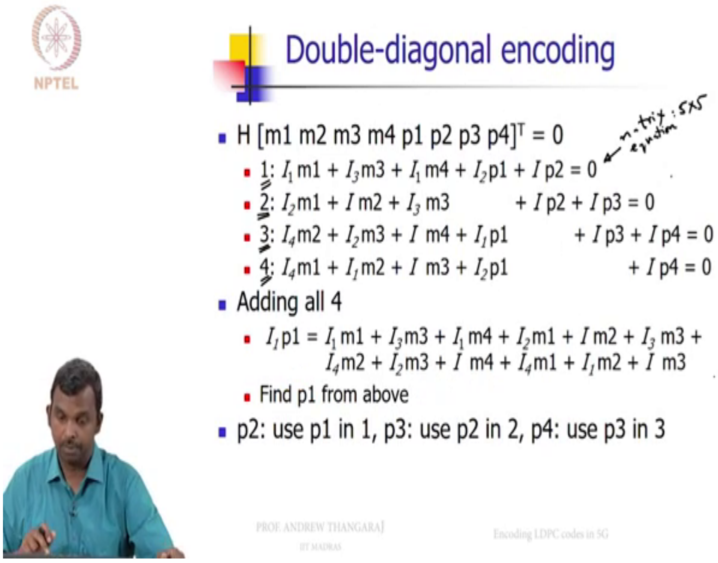
Okay, so here is my example to illustrate the double diagonal structure, so you see, this is a 4 by 8 a block base matrix and the expansion is 5. Okay, so I am using a little slightly different notation, compact notation to capture this, I am going to show, use this I K to indicate the identity matrix column shifted K times. Okay, so instead of using entries like 1, 2, 3, I am using I1, I2, I3, just to show it is the 5 by 5 identity matrix shift at like that and the zero represents 5 by 5 all zero matrix. Okay, so I will be 5 by 5 identity. Okay, and then I1 will be 5 by 5 shift at right by one position, I2 will be 5 by 5 shifted at right by two positions so on. Okay, so this is a matrix, its, you can see, this is a 4 by 8 matrix, this is the message part M1, M2, M3, M4 and this is the parity part P1, P2, P3, P4.

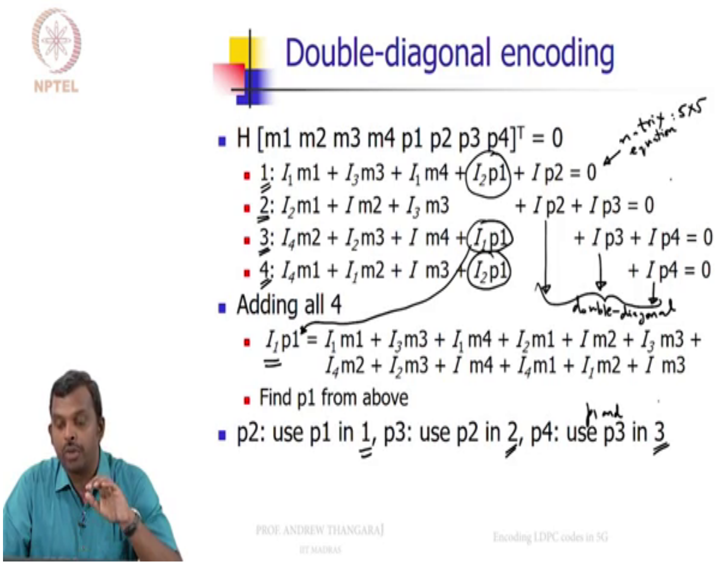
So you will see the diagonal structure here, there is a diagonal structure here, double diagonal structure and then the zeros are all here and then you have this I2 being the same as this I2 here and this I1 being the front. Okay, so this is sort similar 2 every base matrix in the 5G standard as well. If you look at the first four rows, the first four parity blocks, they will always have the structure. Okay, and the structure is very interesting and its design so that encoding can be ease.

Okay, so that is what I am going to show you next. Okay, so like I said, messages M1, M2, M3 M4 am not using subscript in the notation here, it is just for easy typing hopefully there is no confusion. Okay, the M1 is the same as M subscript one. Okay, so M1, M2, M3 M4 is 5 bits each, code word is M1, M2, M3 M4, P1, P2, P3, P4, this case are also five bits each.

(Refer Slide Time: 14:25)







Okay, so now how does the encoding work here, so we have H times, the code word transpores is equal to 0, I am going to write each block row as one matrix equation. Okay, so this is a matrix equation 5 x 5. Okay, so there is a 5 x 5 matrix as here, so each M1, M2, each one M1, M3, M4, P1, P2, are all length 5 vectors. Once again there is this transpores here which I am sort of not showing in the notation. Hopefully it is clear to you, so I put I1 and then put M1 as a column vector can multiply and add up.

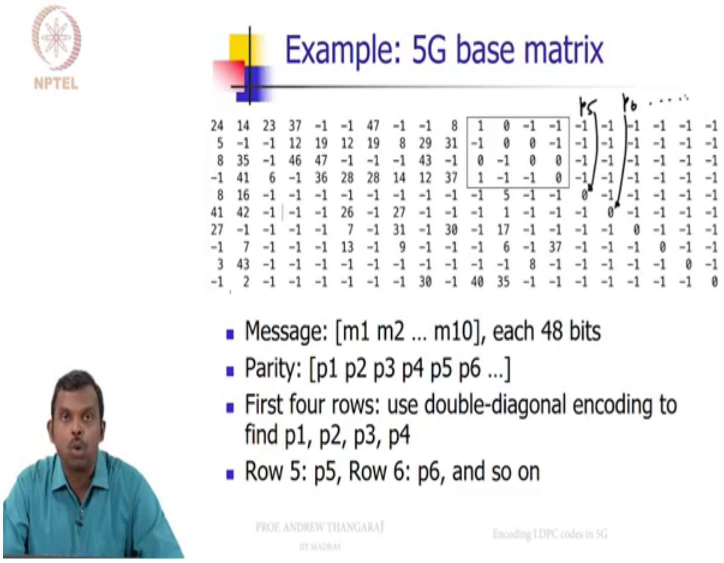
Okay, so I get this equation and then, this is the second block row, this is the third block row and this is the fourth row. Okay, second, so it is indicated here the second block row, third block row, fourth block row. Okay, the first block row, so hopefully it is clear to you how the first block row came, if you want I can go back here and write it down, so if you put M1, M2, M3, M4 on the right side, the row one here corresponds to I1 times M1 plus I3 times M3 right, you can say I1 times M1, I3 times M3 plus I1 times M4 plus I2 times P1 equals 0, ohh sorry, plus I times P2 equals 0.

Okay, so you can see that is what the first who corresponds to, and that is what every time here. Okay, each is a matrix equation, alright, so know if you notice, this two are the same this to are the same, these two are the same, so this comes from the double diagonal structure. Okay, so the double diagonal structure gives you this, the same things here and then here this two are the same and then you just have this, I1, P1 left. Okay, so this is suggesting is, you have to add the four equations, this four block equations you have to add, if you add them you will see that they all sorts of cancellation here, this two will cancel, whatever is equal will cancel, once it cancels, you will be just left with whatever is on the left side, right, I1 times M1 plus I3 times M3 plus I1 times M4 and then the second row I2, I3 extra, then the third row and the fourth row and here you will be just left with I1 times P1, that is the only thing you will be left with, so of course you can bring it to this side and write equal to 0, so I bought the I1 times P1 alone to that side and kept it as an unknown.

Okay so you can find P1 from above. Okay, so from this equation, you can ready-made just, this is just a direct equation for P1, of course you have the I1 here that just shifts the P1 a little bit, so you can reorder and find out P1. Okay, hopefully this is clear to you, so the basic trick is the double diagonal structure is structured such that, when you add all the four equations everything cancels except for the first parity block P1. Okay, so that is the secret to this double diagonal structure which tarated for a while and it is been done, so that when you add up the first four rows, first four block rows only P1 remains and you can find P1.

Okay, once you find P1 you are done right. The reason is, if you have P1 you go to the row one. Okay, so you use P1 in the first equation you get P2, P2 is only unknown them and then you use P2 and then you get P3 from second equation, then you get P4 by using P3 in the third equation. Okay, so remember the messages are given to you M1, M2, M3, M4, you have to compute the parities P1, P2, P3, P4, you add the four block rows, you find P1 and then you use P1 in the first row right, first block row equation you get P2, likewise you use P2 in the second block row equation P3, then you use I guess P1 and P3 in the third block row to get P4. Okay, that is the way this whole thing proceeds, hopefully this is clear to you, now this same structure and exist in the 5G standards LDPC codes as well. Okay, so it is a just a simple case of picking up the structure and using the same thing in your encoding, you find P1, P2, P3, P4.

(Refer Slide Time: 18:46)



Okay, once you find P1, P2, P3, P4. Okay, this is again the same 5G matrix illustration, part of the 5G matrix 42 by 52. So I am showing you only a part, you have 10 message blocks, each 48 bits, the parity blocks goes as P1, P2, P3, P4, P5, P6 etc. And for the first four rows you can recognise the double diagonal structure here by, so this zeros are all I, minus 1 are zeros than 1 is I1 and then you have I itself here, so when you added all up , you will get P1. Okay, so you can find P1, P2, P3, P4 using this double diagonal encoding idea and then once you find that, once you know P1, P2, P3, P4, this P5 is direct from this equation. Okay, see because if you look at this equation P5 is the only unknown, so you can find the P5. Okay, it is just a diagonal and then P6 can find from this equation and so on. Okay, so that is the simple encoding method, hopefully you agree with me, it is designed so that it can be implemented quite easily. The matrix itself is sparks so will not be doing too many exhort competitions in the encoding and you do not have to remember too much as well, except for maybe the base matrix and how it is generated. Okay, so this is the end of this lecture, and the next lecture we will write a small Matlab code for implementing such encoders. Thank you very much.