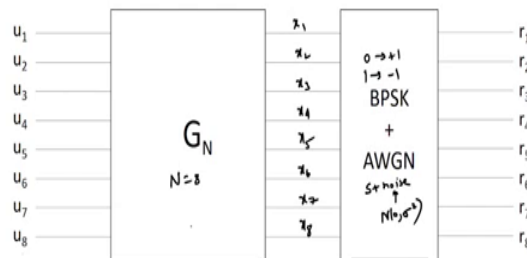


LDPC and Polar Codes in 5G Standard
Professor Andrew Thangaraj
Department of Electrical Engineering
Indian Institute of Technology, Madras
Channel Polarization, Definition of (N,K) Polar Code and Encoding

(Refer Slide Time: 00:17)



Polar transform and BPSK-AWGN



$\underline{r}^{(N)} = [r_1 r_2 r_3 \dots r_N]$: received vector
received vector



PROF. ANDREW THANGARAJ
 IIT MADRAS

Channel Polarization, Definition of (N,K) Polar Code and Encoding


So let's look at what happens if you use the polar transform along with a channel ok. So here is an example I have shown 8 bits here, 8 could be any n I have used n equals 8 for illustration but it could be any other n ok. So you have 8 bits coming in you use the polar transform which is 8 you get 8 outputs ok so this outputs can be anything and usually you can denote it as $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8$ so this are 8 bits and the you can transmit them through BPSK-AWGN, what is BPSK? Once again 0 goes to plus 1, 1 goes to minus 1 and what is AWGN? It is S plus noise right, so noise is normal with mean zero and variance sigma square, some sigma square.

So you create this noise you get received values ok so this is how a polar transform could potentially be used in the BPSK-AWGN channel, AWGN channel and then this r_N is the received vector this is just notation for the received vector ok it is fine we used it, it is just an invertible transform of why would you use it, what is the point of using a polar transform before

a channel? So turns out if you don't use the polar transform every bit sort of goes through the same channel in some way ok so there is nothing much maybe you can do.

The polar transform combine this bits and gives us one some sort of reordering or changes the channel in some way you can view it in a different way ok. So this notion of looking at individual bit channels after the polar transform is the key innovation polar codes and it turns out those channels vary la lot in terms of quality ok so that is the next picture. So this picture is easy enough there is nothing much going on you have so many bits coming in you do the polar transform you can do it for any n I have shown it for any n equal 8 you can do it for n equals 1024 also, 1024 bits coming in you hit it with a polar transform you get outputs, do the BPSK-AWGN you get received bit ok. Now what is the effect that this has? How do you view this in a polarization point of view? Here is the picture.

(Refer Slide Time: 02:29)



Bit channels and polarization

input u_1

u_2

u_3

...

u_i

...

u_N

Bit channel 1

Bit channel 2

Bit channel 3

...

Bit channel i

...

Bit channel N

$r^{(N)}$ *entire received vector*

$r^{(N)}, u_1$ *not a "real" output*

$r^{(N)}, u_1, u_2$


...

$r^{(N)}, u_1, u_2, \dots, u_{i-1}$

...

$r^{(N)}, u_1, u_2, u_3, \dots, u_{N-1}$

- Bit channels polarize and can be ordered based on "quality": quality varies from very good to very bad



PROF. ANDREW THANGARA
IIT MADRAS

Channel Polarization, Definition of (N,K) Polar Code and Encoding

Ok so this picture is the central idea behind polarization. Once again you will not go deep into the theory of this picture except that they will present that from a high level to understand what is going on ok. So it turns out after polarization you can sort of decompose or view what has happened in terms of what is going on to the individual bits in a sequential way ok so you can convert the n channels that you had into n combined channels that you had in the polarization after when you did polarization and split them up sort of into individual bit channels and the first bit channel is defined as follows.

You have u_1 going in, this is the input to the first bit channel and the output is the entire received vector r_N ok remember once again the entire received vector. This is the first bit channel split bit channel this is just a definition at this point ok I am defining the first bit channel to have u_1 as the input and the entire received vector as the output ok. So I am reordering or viewing the previous channels so in the previous picture we had this kind of picture I am seeing this in a different way ok remember I have got all the bits all the received values at the output ok and remember the first bit would have effected all this bits in some way.

So all the received values will have some information about the first bit ok. So it is not wrong to view it this way ok. So we will take input as u_1 and the bit channel I will imagine has a huge output has all the outputs together the entire received vector collected together that is the output ok. Now I will define sort of a second bit channel which has input as u_2 which is the second bit but look at the output here pay attention particularly the r_N is fine ok so you might say the entire received vector is there but to this I will add u_1 as one of the outputs of this bit channels ok.

So now at this point this is not a real output ok, why is that? Because the channel is not giving me u_1 ok the channel only gives me r_N ok I will imagine that somehow I have this u_1 ok and we will see how to create this u_1 in the actual physical channel it turns out you can create a good estimate for this u_1 in some way and cleverly use it and get this effect of channel polarization but at this point I am defining the second bit channel as input u_2 and having output as the entire received vector and the previous bit u_1 as one of the outputs ok just a definition.

Next comes the third bit channel so here I will go little bit further ok so I have the entire r_N as the output u_3 is the input again ok entire r_N is the output along with that I will say both u_1 and u_2 I already have the output ok once again this are not real outputs at this point but mean just carry along with me, you will see later on how to get some substitutes for this u_1 , u_2 later on. For now u_1 , u_2 will imagine somebody is giving me at the decoder some benevolent person is giving me what u_1 and u_2 are and imagining this (second) third bit channel.

So likewise you can proceed, what do I do for the i th bit? For the bit channel i , I will imagine that the entire received vector is there that s not very hard to imagine I have the entire received vector then I will also imagine I have the all the previous bits that were transmitted so u_1 to u_{i-1} I already have at the output this will be what I will imagine ok so likewise I will go all

the way upto the n th bit channel which has all the received values that is ok and then all the previous bits ok. So now what special about this split channel? What is special about this bit channels is? This bit channels are not similar at all ok, they are very-very-very different from each other over $(0,1)$ picture they are all very different in quality.

In the previous case all the channels here all the channels where which you received r_1 are the same. Here this n channels will have very different behavior ok. So you can imagine why, right so first of all this channels are very different from the previous channels ok they have one bit input but the outputs are huge ok and also you are assuming the previous bits are available so they can be very different in quality and that is true ok so it turns out you can study this bit channels very carefully and you can sort of order them based on quality ok and the quality you can show where is a lot from very good to very bad.

Ok and it turns out as you increase N they either become very-very-very good or very-very-very bad there are literally no channels which are sort of mediocre neither very good nor very bad ok nothing in the middle no middle path all bit channels either become very-very good or very-very bad so this is the essence of polarization. What does polarization mean? Go either go to the north pole or the south pole right polarization (polar) one of two things nothing in the middle ok.

So that way this bit channels get polarized ok they either became very-very-very good or very-very-very bad ok, so this are called this is called polarization.

There is a wonderful information theoretic proof for this polarization for what type of channels it happens and all that and once again we will now study that at least in this course will be later on will add some additional content for this but for now we will just look at how to use this polarization to define polar codes and how to do use this polarization to design decode as for polar codes ok and achieve good performance that will be the focus of the lectures in this course but this idea is very central but keep note of this not having the real output and what do you do for that, we will postponed this discussion when you come to the decoder.

So now we will use this idea of polarization and start talking about how to define polar codes ok you come to this motion of how to recreate this u_1 at the output side later on ok.

(Refer Slide Time: 08:37)



Reliability sequence: worst to best (5G)

↑
ordering of
bit channels

- N = 8
 - 1 2 3 5 4 6 7 8
- N = 16
 - 1 2 3 5 (9) 4 6 10 7 11 13 8 12 14 15 16
- N = 32
 - 1 2 3 5 9 17 4 6 10 7 18 11 19 13 21 25 8
 - 12 20 14 15 22 27 26 23 29 16 24 28 30 31 32
- Specified for N = 1024 in the 5G standard
 - Sequence for smaller N derived from the sequence for N = 1024



PROF. ANDREW THANGARAJ
@ IIT MADRAS

Channel Polarization, Definition of (N,K) Polar Code and Encoding

Ok so now the next important idea is that of a reliability sequence ok so I mentioned how after polarization if you take the bits you polarize them and transmit them one after the other end over the BPSK-AWGN channel there is a lot of polarization ok. Some bits become some bit channels and then you can define this clever bit channels which have previous inputs.

If you do that then some channels become very-very good some channels become very-very bad. Now which channels are good which channels are bad? Ok that is given by the reliability sequence this tells you the ordering of channels, ordering of bit channels from worst to best ok. Which is the worst among this N bit channels!? And which is the best among this first bit channels? I am giving you the reliability sequence as per the 5G standard ok. Now there are multiple ways to create this reliability sequence and a lots of research going on and how accurately you can find the reliability sequence, how realistic it is etc-etc-etc all of that we are not going to go into detail but we will simply provide the final answer ok.

So it turns out for N equals 8 according to the 5G standard where reliability sequence is this ok. One the bit channel 1 is the worst channel, next comes 2 ok after that is 3, after that is 5, after that is 4 then 6, 7, 8 ok. So it is not the same order as 1, 2, 3, 4, 5 so you see this two got switched for N equals 8 ok. So the 4th bit channel is better and the 5th bit channel according to this reliability sequence ok. So like I said the way in which you do this is not very precise it is sort of noise and simulations and all that.

Maybe I will point you to some other references and if you are interested in reading about how to find this reliability sequence? But in this lecture we won't focus on how to find this reliability sequence? We will just think about how to use them for designing for encoding and decoding polar codes ok. So here is the reliability sequence for where n equals 16 so you see this 1, 2, 3, 5 comes again and then suddenly 9 comes ok, 9 becomes bad channel then you have 4, 6, 10 this is the sequence ok. So for instance 14th channel, 14th bit channel is better than the 12th bit channel.

The 8th bit channel is worst than the 12th bit channel like that, that is how you read it, this is the sequence in which the quality of the channel is (\cdot) (11:02) 16 bit channels they are ordered like this. This is for N equals 32, remember all this numbers like I said I am pulling from the 5G standard, the 5G standards specifies this for N equals 1 0 2 4 ok and sequence for smaller n is derived from the sequence for m_0 , m equals 1024 ok.

So I am just picking it up from there ok so you can also do this if you have access to the 5G standard you can go pick up their reliability sequence for 1024 and then from that find the subsequences which are $5n$ equals 8, 16, 32 and all that ok so that is the final summary. Once you do the polar transform and transmit on the BPSK-AWGN channel for instance or any other channel for that matter you can take a global so the reliability sequence the bit and then create this bit channels. The bit channels become polarized and they can be ordered from the worst to the best and the sequencing a sort of given here.

Ok you can take one sequence, how do I find the sequencing? When I am going into details that but the final answer is given to you ok so this is the sequencing alright, so this is the notion of reliability sequence it is very crucial for designing and defining polar codes very important ok.

(Refer Slide Time: 12:15)



Reliability sequence: $N = 1024$

```
1 2 3 5 9 17 33 4 6 65 10 7 18 11 19 129 13 34 66 21 257 35 25 37 8 130 67 513 12 41 69 131 20 14 49 15 73 258 22 133 36 259 27 514 81 38 26 23
137 261 265 39 515 97 68 42 145 29 70 43 517 50 75 273 161 521 289 529 193 545 71 45 132 82 51 74 16 321 134 53 24 135 385 77 138 83 57 28 98
40 260 85 139 140 262 30 44 99 516 89 141 31 147 72 263 266 162 577 46 101 641 52 149 47 76 267 274 518 105 163 54 194 153 78 165 769 269
275 519 55 84 58 522 113 136 79 290 195 86 277 523 59 169 140 100 87 61 281 90 291 530 525 197 142 102 148 177 143 531 322 32 201 91 546 293
323 533 264 150 103 106 305 297 164 93 48 268 386 547 325 209 387 151 154 166 107 56 329 537 578 540 114 155 80 270 109 579 225 167 520 553
196 271 642 524 276 581 292 60 170 561 115 278 157 88 198 117 171 62 532 526 643 282 279 527 178 294 389 92 585 770 199 173 121 202 337 63
283 144 104 179 295 94 645 203 593 324 393 298 771 108 181 152 210 285 649 95 205 299 401 609 353 326 534 156 211 306 548 301 110 185 535
538 116 168 226 327 307 773 158 657 330 111 118 213 172 777 331 227 550 539 388 309 217 417 272 280 159 338 551 673 119 333 580 541 390
174 122 554 200 783 180 229 339 313 705 391 175 555 582 394 284 123 449 354 562 204 64 341 395 528 383 557 182 296 286 233 125 206 183 644
563 287 586 300 355 212 402 186 397 345 587 646 594 536 241 207 96 328 565 801 403 357 308 302 418 214 569 833 589 187 647 405 228 897 595
419 303 650 772 361 540 112 332 215 310 189 450 218 409 610 597 552 231 160 421 311 542 774 611 658 334 120 601 340 219 369 653 231
392 314 451 543 335 234 556 775 176 124 659 613 342 778 221 315 425 396 674 584 356 288 184 235 126 558 661 617 343 317 242 779 564 346
453 398 404 208 675 559 786 433 358 188 237 665 625 588 781 706 127 243 566 399 347 457 359 406 304 570 245 596 190 567 677 362 707 590
216 787 648 349 420 407 465 681 802 363 591 410 571 789 598 573 220 312 709 599 602 652 422 793 803 612 603 411 232 689 654 249 370 191
365 655 660 336 481 316 223 371 614 423 426 453 615 544 236 413 344 373 776 318 223 427 454 238 560 834 805 713 835 662 809 780 618 665
434 721 817 837 348 898 244 663 455 319 676 619 899 782 377 429 666 737 568 841 626 239 360 458 400 788 592 679 435 678 530 246 459 667
621 364 128 192 783 408 437 627 572 466 682 247 708 351 600 669 791 461 250 683 574 412 804 790 710 366 441 629 690 375 424 467 794 251
372 482 575 414 604 367 469 656 901 806 616 685 711 430 795 253 374 606 849 691 714 633 483 807 428 905 415 224 664 693 836 620 473 456
797 810 715 722 838 717 865 811 607 913 723 697 378 436 818 320 622 813 485 431 839 668 489 240 379 460 623 628 438 381 819 462 497 670
680 725 842 630 352 468 439 738 252 463 443 442 470 248 684 843 739 900 671 784 850 821 729 929 792 368 902 631 686 845 634 712 254 692
825 903 687 741 851 376 455 471 484 416 486 906 796 474 635 745 853 961 866 694 798 907 716 808 475 637 695 255 718 576 914 799 812 380
698 432 608 490 867 724 487 909 719 814 477 857 840 726 699 915 753 869 820 815 440 930 491 624 672 740 917 484 844 382 488 931 822 727
962 873 493 632 730 701 444 742 846 921 383 823 852 731 499 881 743 446 472 636 933 688 904 826 501 847 746 827 733 447 963 937 476 854
868 638 908 488 696 747 829 754 855 858 505 800 256 965 910 720 478 916 639 749 945 870 492 700 755 859 479 969 384 911 816 977 817 918
728 494 874 702 932 757 861 500 732 824 923 875 919 930 934 744 761 882 495 703 922 502 877 848 993 448 734 828 935 883 938 964 748 506
856 925 735 830 966 939 885 507 750 946 967 756 860 941 831 912 872 640 889 480 947 751 970 509 862 758 971 920 876 863 759 949 978 924
973 762 878 953 496 704 939 979 884 763 504 926 879 736 994 886 940 995 981 927 765 942 968 887 832 948 508 890 985 752 943 997 972 891
510 950 974 1001 893 951 864 760 1009 511 980 954 764 975 955 880 982 983 928 996 766 957 888 986 998 987 944 892 999 767 512 989 1002 952
1003 894 976 895 1010 956 1005 1011 958 984 959 988 1013 1000 1017 768 990 1004 991 1006 960 1012 1014 896 1007 1015 1018 1019 992 1021
1008 1016 1020 1022 1023 1024
```



PROF. ANDREW THANGARA
@ IIT MADRAS

Channel Polarization, Definition of (N,K) Polar Code and Encoding

So here is the full reliability sequence for N equals 1024 I will not expecting you to learn this by heart you don't have to do this you don't need to mug up this sequence but this is the sequence. It is given in the standard and you can see how it goes there is no pattern here right. So it is sort of random in some sense but not all that random sort of increases from 1 to 1024 in some generic way but suddenly you will have some big numbers showing up in early on right so it sort of difficult to predict there is no real pattern here as far as at least I know there is no pattern here it is hard to come up with this estimates and also you may if you read more will see if the noise variance changes in the AWGN channel this reliability sequence actually changes. Ok so it actually changes so it is hard to pick up one reliability sequence for a wide range of noise variances but people use some close simulation studies and carefully study it and come up with something like this.

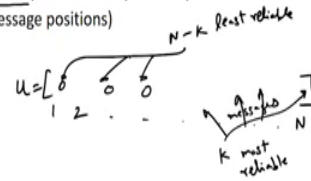
Ok so the reliability sequence is a difficult thing to find ok so not very easy to find. There are good methods available today in the literature but like I said even if you find it strictly it will vary from noise variance to noise variance and it is not sort of contain so many cases it is not contained some cases it is contain, so you can use some approximations and that is the structure but it is a very hotly research topic to, so reliability sequence is important but for us, we are not going to focus too much on the difficulties around reliability sequence. We will simply take the one that is given in the standard and be happy about it ok so if you are doing that there is no problem.

(Refer Slide Time: 13:56)



(N,K) Polar Code

- $N = 2^n$ $n = 2, 3, \dots$ $K \leq N$
- Message: m of length K bits
- Form a vector u of length N bits as follows
 - Find $N - K$ least reliable (worst) channels from reliability sequence
 - Set u_i for those $N - K$ channels to zero (called frozen positions)
 - m : remaining K bits of u (called message positions)
- Codeword: $u G_{N,N}$



PROF. ANDREW THANGARAJ
IIT MADRAS

Channel Polarization, Definition of (N,K) Polar Code and Encoding

Ok so having done all this work I will study the polarization having studied this bit channels having understood what the definition or this reliability sequence is we are ready to define the polar code ok. So N, K polar code is defined for N equals 2 power n ok and the small n could be when usually you want to start at 4 (sorry) 2 at least $2-3$ so on ok. The message m of length k bits and this k is going to be a less than or equal to n ok. Message m is of length k bits that is ok what do you do to define the polar code I will define the encoder for polar code, ok this is the encoder for the N, K polar code.

I will form a vector u of length n bits ok, how do I form that? I will first find the n minus k least reliable or worst channels from the reliability sequence ok there is a reliability sequence. What is the first n minus k least reliable, the first n minus k numbers in the sequence is the least reliable right so you take the n minus k least reliable and I will set the u_i for this N minus K channel to zero ok so I am saying channels here remember this are like N minus K positions ok. The first N minus K least reliable positions as per the least reliability sequence use set those u_i to be equal to 0 , remember u_i has length n ok the first N minus K positions whatever they are you set them to 0 ok, the u_i corresponding to the first N minus K positions in the reliability sequence remember it is not in the same order, the order sometimes changes.

You set those to zero now there will be K remaining positions for u ok, the K remaining positions you set them to be equal to the message ok alright so you have this vector u ok it has n positions

1, 2, so on till n ok the N minus K least reliable positions you put 0 ok so you put 0 here, you put 0 here wherever they are least reliable you put 0 ok and then the remaining guys you put message K most reliable so least and most with respect to the reliability sequence ok, so there you put m messages, messages go there.

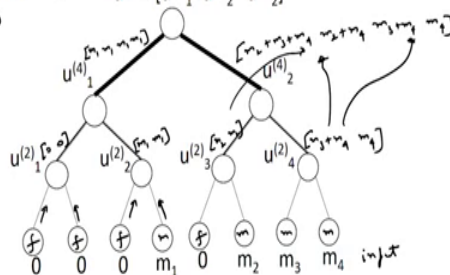
Ok so you form a vector u like this and then what is the code word? Code word is simply the polar transform of u ok u times G N polar transform of u is it ok, that is the definition for the N K polar code for nay n you can do this once you have reliability sequence you have a polar code right so as simple as that right. So hopefully this is clear we started with a 2 by 2 polar panel we created G N polar panel then we looked up that polarization property and then we define the reliability sequence. Once you define the reliability sequence you have a polar code ok, so that is it.

(Refer Slide Time: 17:26)



Polar code example: (8,4) $N=8, n=3$

- Reliability sequence: 1 2 3 5 4 6 7 8
 - Frozen: 1 2 3 5
 - Message: 4 6 7 8
- $U^{(8)} = [U^{(4)}_1 + U^{(4)}_2 \quad U^{(4)}_1 \quad U^{(4)}_2]$



PROF. ANDREW THANGARA
IIT MADRAS

Here are a few examples I am starting with N equals 8 small n equals 3 ok and defining the 8, 4 polar code this is the reliability sequence so the frozen positions the once for which you choose them to be zero this are called the frozen positions ok so once again I think I missed defining that the least reliable N minus K least reliable channels which you set to zero are called the frozen positions and then the remaining are message positions ok. So if you do 8, 4 code reliability sequence is 1 2 3 5 4 6 7 8 so the frozen positions are 1 2 3 5 ok, the message is 4 6 7 8 ok.

So how do you visualize the polar code construction? You can go back to the binary tree representation of the polar transform ok and 1 2 3 and 5 you set as 0, so this is all frozen ok I will write f here to indicate that this are frozen and then write m here to indicate this are message ok. So the first, second, third and fifth leaf nodes are frozen to 0, the remaining are set to m_1 , m_2 , m_3 , m_4 . Now will this encoding happen? You will do 0 here, 0 here and you will get a 0, 0 here right 0 0 goes to 0 0 you get 0 here m_1 here and you will get m_1 (m_1) here right.

So 0 is coming here m_1 is coming here so 0 plus m_1 is m_1 then m_1 itself is retain here likewise you have 0, m_2 so this will be m_2 m_2 ok this is m_3 , m_4 so will have m_3 plus m_4 , (m_4) ok then what will, happen in the next step? This will this is 0 0 m_1 m_1 so this 4 bit vector will be m_1 , m_1 , m_1 , m_1 this will be m_2 plus m_3 plus m_4 , m_2 plus m_4 , m_3 plus m_4 and then m_4 right so that is what happens at this node you combine m_2 m_2 and m_3 plus m_4 m_4 you take the ex-or of this two keep it here and then this is retained as it is ok so remember this is retain as it is. The ex-or of this two goes here.

Ok that is what happen in this and then finally the code word 8 bit code word is going to be again ex-or of this two and then this retained as it is right so let me write it down laboriously for you m_1 plus m_2 plus m_3 plus m_4 , m_1 plus m_2 plus m_4 , m_1 plus m_3 plus m_4 , m_1 plus m_4 and the remaining 4 are the exact same things as this m_2 plus m_3 plus m_4 , m_2 plus m_4 , m_3 plus m_4 , m_4 this is the encoded version so this is the input to the transform and this is the output of the transform and that is the code word so this is the code word ok.

So you have a message m which is m_1 m_2 m_3 m_4 , m is m_1 , m_2 , m_3 , m_4 and the code word is this ok remember plus again as ex-or or modulo 2 addition you do all this things ok.

So if you want you can create or generate a matrix out of this you will get when this easily specifies the generator matrix ok. You can construct that if you like ok you will get a 4 by 8 generator matrix and you notice this is not a systematic encoding, isn't it? m_1 , m_2 , m_3 , m_4 does not appear by itself in the code word it is not systematic but this is the encoding that you have ok. So once you fix length and the number of message bits and once you have reliability sequence you can always do polar encoding and this tree representation gives a very simple way in which you can implement the encoding aspect.

(Refer Slide Time: 21:40)



Polar code example: (16,10) and (32,20)

(16,10) $N=16$
• 1 2 3 5 9 4 6 10 7 11 13 8 12 14 15 16
• Frozen: 1 2 3 5 9 4
• Message: 6 10 7 11 13 8 12 14 15 16



(32,20)
• 1 2 3 5 9 17 4 6 10 7 18 11 19 13 21 25 8 12
20 14 15 22 27 26 23 29 16 24 28 30 31 32
• Frozen: 1 2 3 5 9 17 4 6 10 7 18 11
• Message: 19 13 21 25 8 12 20 14 15 22 27 26 23 29 16 24 28
30 31 32



PROF. ANDREW THANGARAJ
@ IIT MADRAS

Channel Polarization, Definition of (N,K) Polar Code and Encoding

I am going to give a few more examples here is a 16, 10 example ok so for n equal 16 the reliability sequence is this and you have to freeze the first 6 positions right, so first 1, 2, 3, 5, 9 and 4 will get frozen in your tree you have 16 leaf nodes the 1st nodes, 2nd node, 3rd node, 5th, 4th and then 9th are going to be set as zero the message will go into the remaining place and then will start encoding as before you will get the code word at the output ok a code word each bit of the code word will be some ex-or of some m_1, m_2, m_3 like that.

If you want you can write out and generate a matrix for that and you will have a definition for 16, 10 polar code ok. the same thing happens for 32 ok there is nothing different about it n equals 32 so and 20 is the K so you have to freeze the first 12 positions you go all the way up to 11, so what are the frozen positions? 1, 2, 3, 4, 5, 6, 7 then you have 9, 10, 11, 17 and 18 ok some sort of numbers that they get frozen to zero remaining at the message bits once again you can combine and create a generate a matrix if you like ok. So in fact viewing the freezing as $(())$ (22:54) times $G N$ is also quite easy ok.

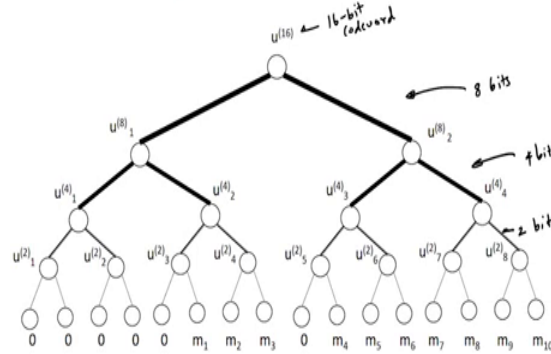
So this is the polar transform you look at the non-frozen positions in U and the rows corresponding to $G N$ in that, that will give you the generator matrix ok so that is a quick way to generate come up with a generator matrix for the polar codes ok. So this is example.

(Refer Slide Time: 23:15)



Binary tree representation: (16,10)

$N=1024$
 $K=512$



PROF. ANDREW THANGARAJ
IIT MADRAS

Channel Polarization, Definition of (N,K) Polar Code and Encoding

And here is for 16,10 the full blown binary tree representation I am not going to go through and show you the computation but this is what happens here frozen bits zero and then the message bits 10 of them and they combine and you create the 16 bit code word. Remember once again these are all 1 bit here, these will be 2 bits right and then you have 4 bits, then you have 8 bits here ok and then finally you have 16 bit code word ok. So that's it that is the polar code and even if you have so you can imagine if you have $N=1024$ and $K=512$ what do you do?

(Refer Slide Time: 24:07)



Reliability sequence: $N = 1024$

```
1 2 3 5 9 17 33 4 6 65 10 7 18 11 19 129 13 34 66 21 257 35 25 37 8 130 67 513 12 41 69 131 20 14 49 15 73 258 22 133 36 259 27 514 81 38 26 23
137 261 265 39 515 97 68 42 145 29 70 43 517 50 75 273 161 521 289 529 193 545 71 45 132 82 51 74 16 321 134 53 24 135 385 77 138 83 57 28 98
40 260 85 139 146 262 30 44 99 516 89 141 31 147 72 263 266 162 577 46 101 641 52 149 47 76 267 274 518 105 163 54 194 153 78 165 769 269
275 519 55 84 58 522 113 136 79 290 195 86 277 523 59 169 140 100 87 61 281 90 291 530 525 197 142 102 148 177 143 531 322 32 201 91 546 293
323 533 264 150 103 106 305 297 164 93 48 268 386 547 325 209 387 151 154 166 107 56 329 537 578 549 114 155 80 270 109 579 225 167 520 553
196 271 642 524 276 581 292 60 170 561 115 278 157 88 198 117 171 62 532 526 643 282 279 527 178 294 389 92 585 770 199 173 121 202 337 63
283 144 104 179 295 94 645 203 593 324 393 298 771 108 181 152 210 285 649 95 205 299 401 609 353 326 534 156 211 306 548 301 110 185 535
538 116 168 226 327 307 773 158 657 330 111 118 213 172 777 331 227 550 539 388 309 217 417 272 280 159 338 551 673 119 333 580 541 390
174 122 554 200 783 180 229 339 113 705 391 175 555 582 394 284 123 449 354 562 204 64 341 395 528 383 557 182 296 286 231 125 206 183 644
563 287 586 300 355 212 402 186 397 345 587 646 594 536 241 207 96 328 565 801 403 357 308 302 418 214 569 833 589 187 647 405 228 897 595
419 303 650 772 361 540 112 332 215 310 189 450 218 409 610 597 552 230 160 421 311 542 774 611 658 334 120 601 340 219 369 653 231
392 314 451 543 335 234 556 775 176 124 659 613 342 778 221 315 425 396 674 584 356 288 184 235 126 558 661 617 343 317 242 779 564 346
453 398 404 208 675 559 786 433 358 188 237 665 625 588 781 706 127 243 566 399 347 457 359 406 304 570 245 596 190 567 677 362 707 590
216 787 648 349 420 407 465 681 802 363 591 410 571 789 598 573 220 312 709 599 602 652 422 793 803 612 603 411 232 689 654 249 370 191
365 655 660 336 481 316 223 371 614 423 426 452 615 544 236 413 344 373 776 318 223 427 454 238 560 834 805 713 835 662 809 780 618 665
434 721 817 837 348 898 244 663 455 319 676 619 899 782 377 429 666 737 568 841 626 239 360 458 400 788 592 679 435 678 50 246 459 667
621 364 128 192 783 408 437 627 572 466 682 247 708 351 600 669 791 461 250 683 574 412 804 790 710 366 441 629 690 375 424 467 794 251
372 482 575 414 604 367 469 656 901 806 616 685 711 430 795 253 374 606 849 691 714 633 483 807 428 905 415 224 664 693 836 620 473 456
797 810 715 722 838 717 865 811 607 913 723 697 378 436 818 320 622 813 485 431 839 668 489 240 379 460 623 628 438 381 819 462 497 670
680 725 842 630 352 468 439 738 252 463 443 442 470 248 684 843 739 900 671 784 850 821 729 929 792 368 902 631 686 845 634 712 254 692
825 903 687 741 851 376 445 471 484 416 486 906 796 474 635 745 853 961 866 694 798 907 716 808 475 637 695 255 718 576 914 799 812 380
698 432 608 490 867 724 487 909 719 814 477 857 840 726 699 915 753 869 820 815 440 930 491 624 672 740 917 484 844 382 498 931 822 727
962 873 493 632 730 701 444 742 846 921 383 823 852 731 499 881 743 446 472 636 933 688 904 826 501 847 746 827 733 447 963 937 476 854
868 638 908 488 696 747 829 754 855 858 505 800 256 965 910 720 478 916 639 749 945 870 492 700 755 859 479 969 384 911 816 977 871 918
728 494 874 702 932 757 861 500 732 824 923 875 919 503 934 744 761 882 495 703 922 502 877 848 993 448 734 828 935 883 938 964 748 506
856 925 735 830 966 939 885 507 750 946 967 756 860 941 831 912 872 640 889 480 947 751 970 509 862 758 971 920 876 863 759 949 978 924
973 762 878 953 496 704 936 979 884 763 504 926 879 736 994 886 940 995 981 927 765 942 968 887 832 948 508 890 885 752 943 997 972 891
510 950 974 1001 893 951 864 760 1009 511 980 954 764 975 955 880 982 983 928 996 766 957 888 986 998 987 944 892 999 767 512 989 1002 852
1003 894 976 895 1010 956 1005 1011 958 984 959 988 1013 1000 1017 768 990 1004 991 1006 960 1012 1014 896 1007 1015 1018 1019 992 1021
1008 1016 1020 1022 1023 1024
```



PROF. ANDREW THANGARAJ
@ IIT MADRAS

Channel Polarization, Definition of (N,K) Polar Code and Encoding

Ok you go back to the big slide that I showed you ok this is N equals 1024 and K equals 512, what will you do? You take this big vector and then mark out the first 512 positions and make them frozen and imagine this tree and on the leaf you freeze all of them set them to zero put the message bits in the remaining places and then do the encoding ok I mean it is hard to do it by hand but you can write a quick program and that will work quite fast ok.

So that is the end of description of the polar codes hopefully this was clear to you once again the important things to focus on at the polar transform is notion of bit channels, how we define them and the reliability sequence is defined that is very-very crucial and once you have a reliability sequence you have a polar code ok once you have it for any N it is the polar code ok so N , K polar code your frozen positions and message positions you freeze the first N minus K and you put your messages in the remaining K ok, that is the end of this lecture on polar codes. In the next lecture we will see MATLAB programming for writing a quick encoder once again we will write a simple encoder that works for doing the polar encoder ok, thank you very much.