

POLAR CODES (BITSYNC)

- Done By: Nithin,Taarun,Hemanth,Ananth



Polar codes

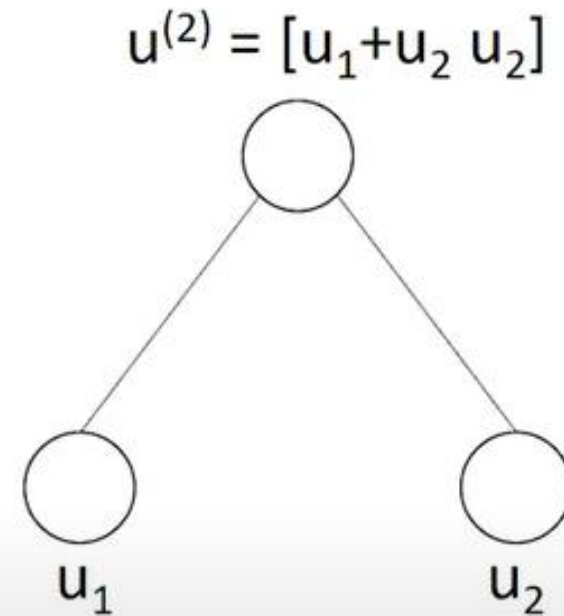
- Invented by Erdal Arikan 2008
- Based on the idea of channel polarization
- First codes to have explicit proof for approaching capacity
- Included as codes for the control channels in the 5G standard
- Sequential in nature
- Defined using generator matrix in a recursive (Kronecker product) definition

Polar transform: G_2

$$G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

- $[u_1 \ u_2] G_2 = [u_1+u_2 \ u_2]$

Binary tree representation



$u^{(2)}$: length-2 vector

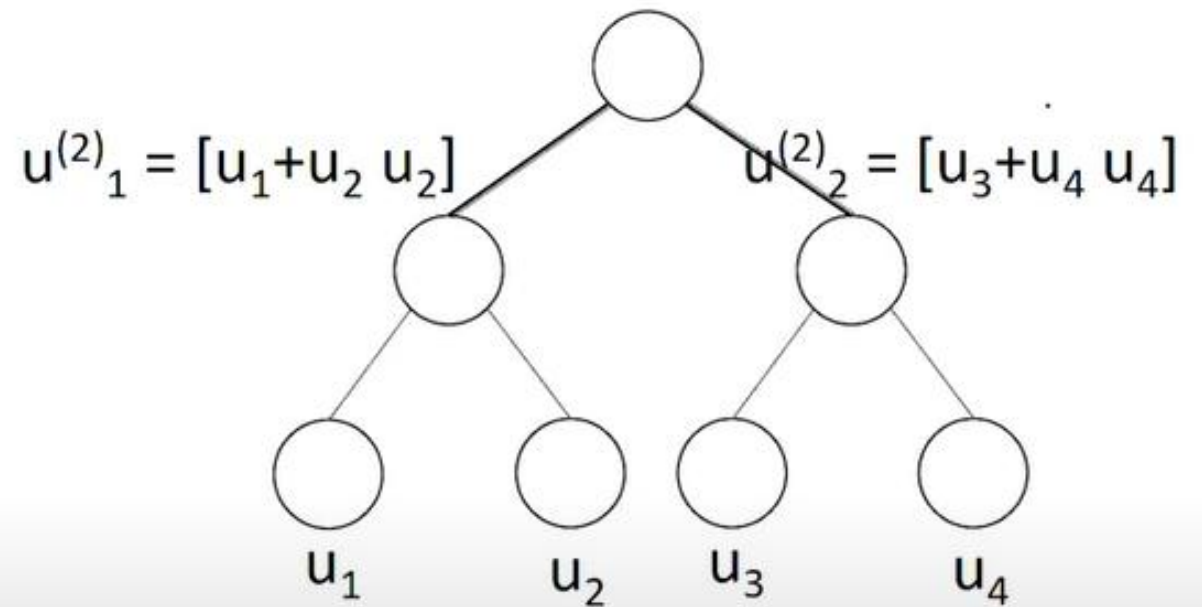
Polar transform: G_4

$$G_4 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

- $[u_1 \ u_2 \ u_3 \ u_4] G_4 = [u_1+u_2+u_3+u_4 \ u_2+u_4 \ u_3+u_4 \ u_4]$

Binary tree representation

$$u^{(4)} = [u^{(2)}_1 + u^{(2)}_2 \quad u^{(2)}_2]$$



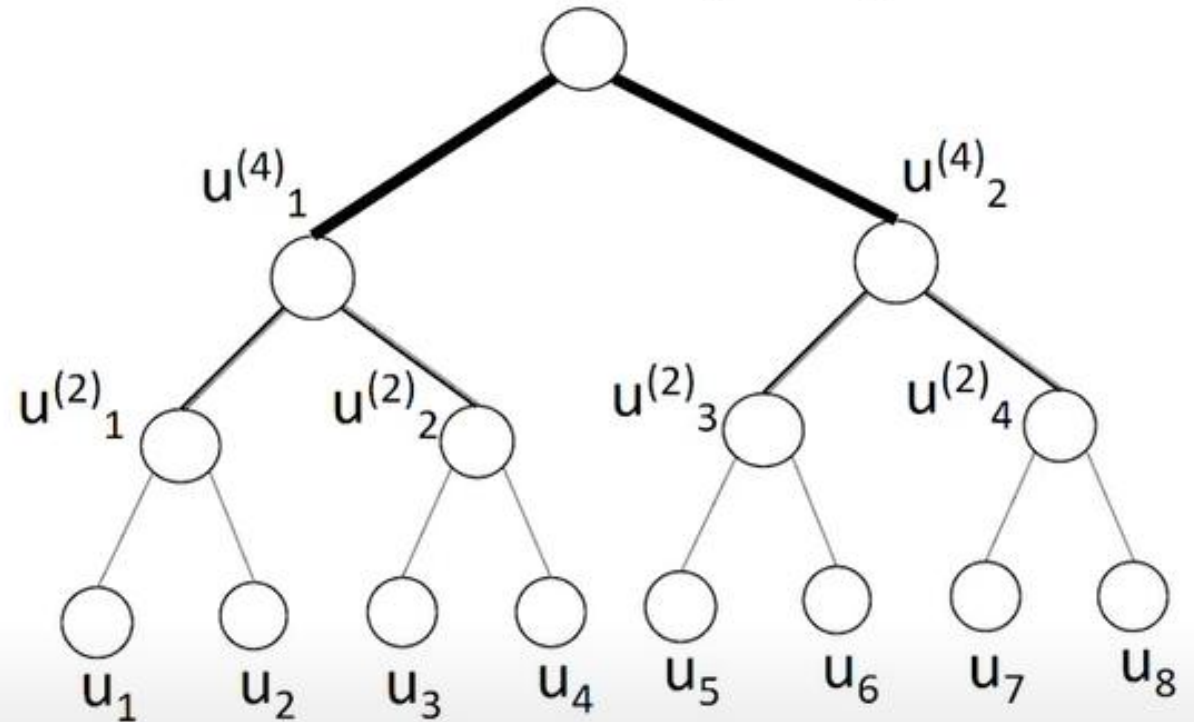
Polar transform: G_8

$$G_8 = \begin{bmatrix} 1 & 0 & \textcircled{\otimes 3} \\ 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Binary tree representation

$$u^{(8)} = [u^{(4)}_1 + u^{(4)}_2 \quad u^{(4)}_2]$$



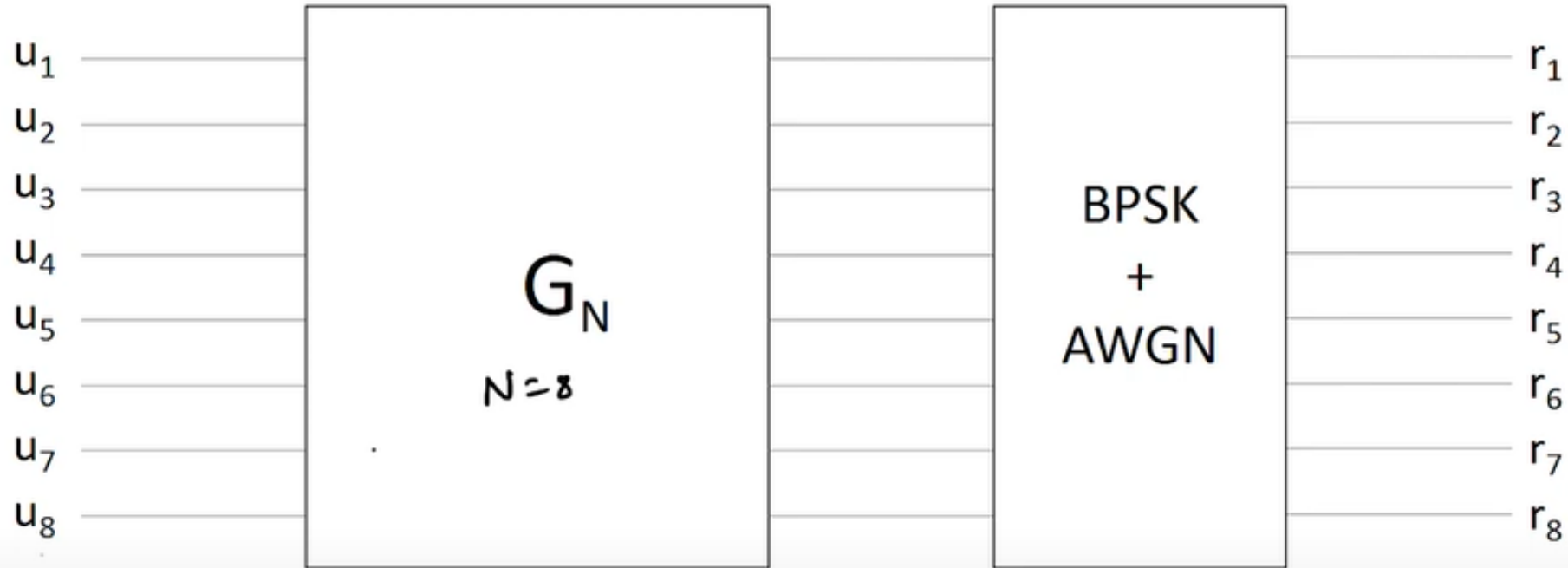
$$[u_1 \ u_2 \ u_3 \ u_4 \ u_5 \ u_6 \ u_7 \ u_8] G_8 = \begin{bmatrix} u_1 + u_2 + u_3 + u_4 + u_5 + u_6 + u_7 + u_8 & u_2 + u_4 + u_6 + u_8 & u_3 + u_4 + u_7 + u_8 & u_4 + u_8 \\ & u_5 + u_6 + u_7 + u_8 & u_6 + u_8 & u_7 + u_8 \end{bmatrix}$$

Polar Transform: General

$$G_{2^n} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{\otimes n}$$

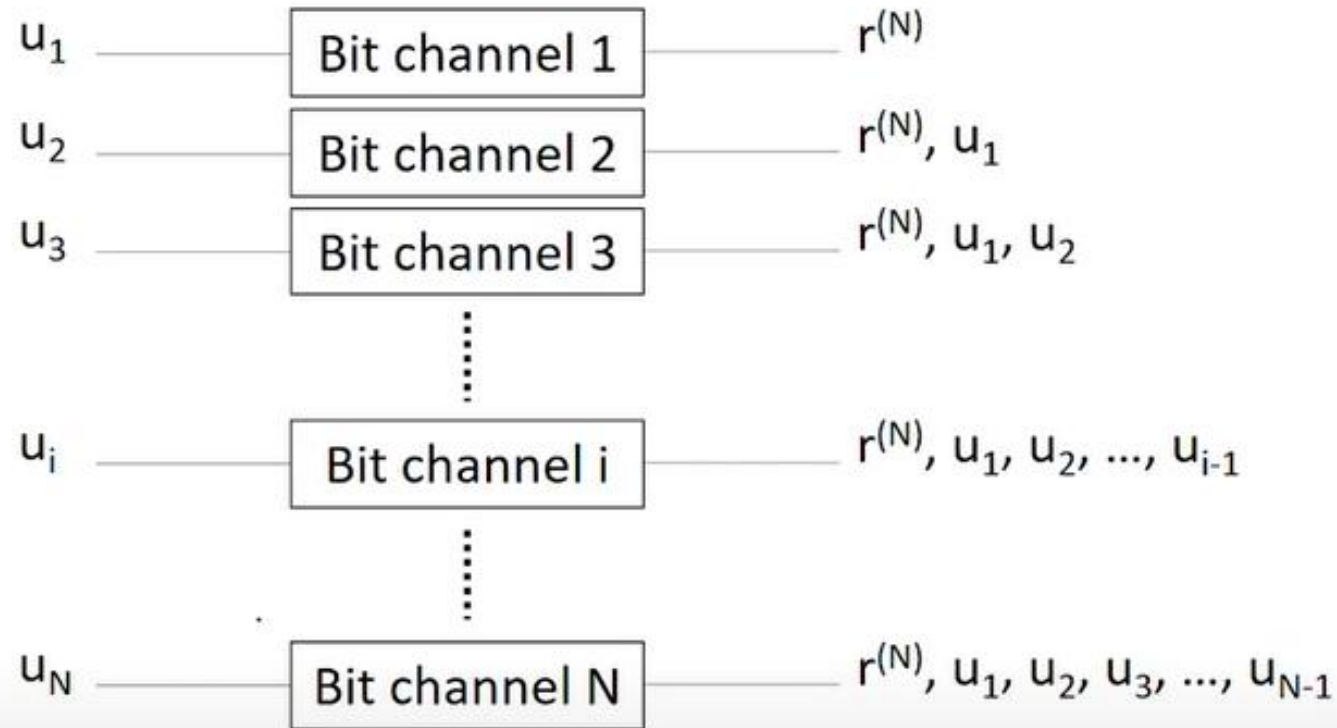
- $N = 2^n$
- G_N : $N \times N$ matrix, Kronecker product of 2×2 kernel
- Binary tree representation
 - Depth n
 - $u^{(N)} = u G_N$: evaluated on tree with u at bottom and $u^{(N)}$ at top
- 5G: uses up to $n = 10$

Polar transform and BPSK-AWGN



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

Bit channels and polarization



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

Reliability sequence: worst to best (5G)

- $N = 8$

- 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 $N = 1024$

Reliability sequence: N = 1024

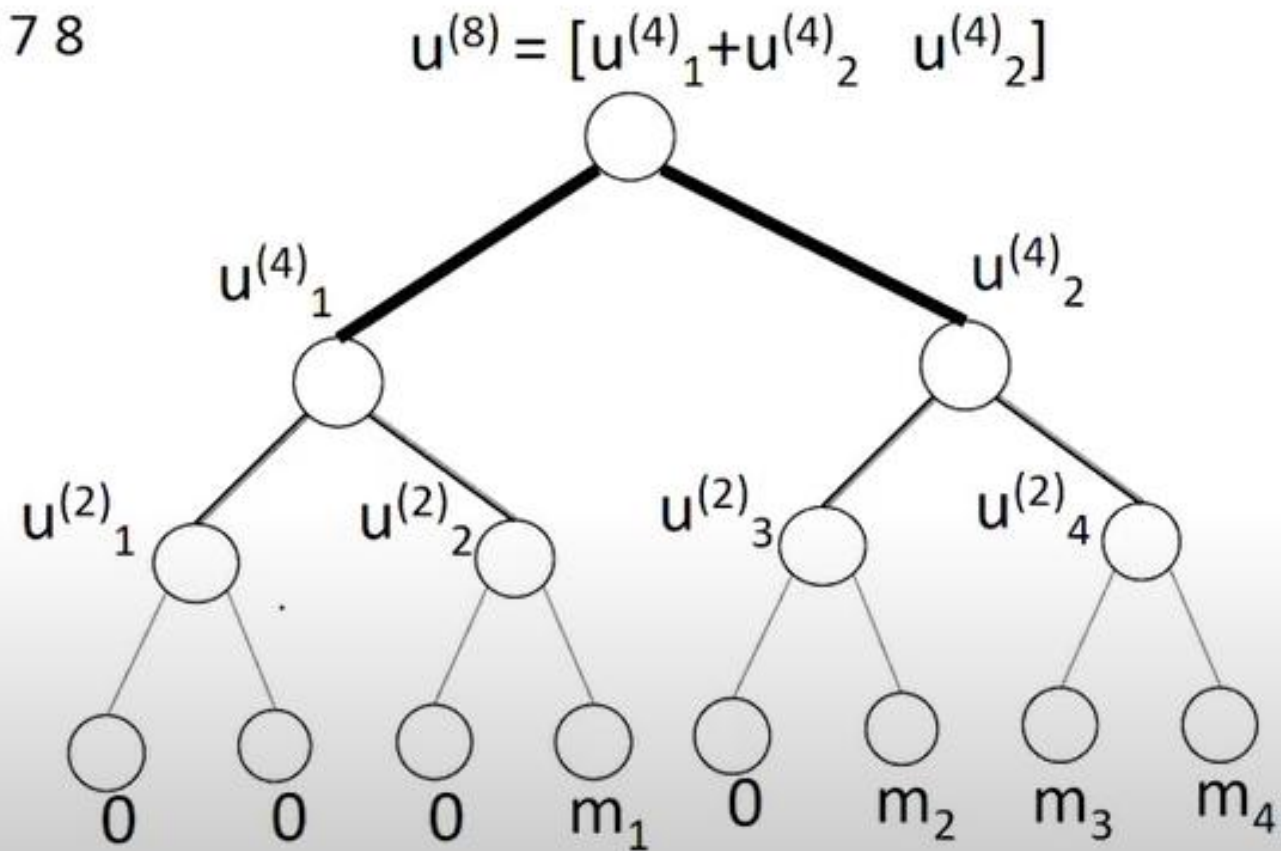
1 2 3 5 9 17 33 4 6 65 10 7 18 11 19 129 13 34 66 21 257 35 25 37 8 130 67 513 12 41 69 131 20 14 49 15 73 258 22 133 36 259 27 514 81 38 26 23
137 261 265 39 515 97 68 42 145 29 70 43 517 50 75 273 161 521 289 529 193 545 71 45 132 82 51 74 16 321 134 53 24 135 385 77 138 83 57 28 98
40 260 85 139 146 262 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
174 122 554 200 785 180 229 339 313 705 391 175 555 582 394 284 123 449 354 562 204 64 341 395 528 583 557 182 296 286 233 125 206 183 644
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
419 303 650 772 361 540 112 332 215 310 189 450 218 409 610 597 552 651 230 160 421 311 542 774 611 658 334 120 601 340 219 369 653 231
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
365 655 660 336 481 316 222 371 614 423 426 452 615 544 236 413 344 373 776 318 223 427 454 238 560 834 805 713 835 662 809 780 618 605
434 721 817 837 348 898 244 663 455 319 676 619 899 782 377 429 666 737 568 841 626 239 360 458 400 788 592 679 435 678 350 246 459 667
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
698 432 608 490 867 724 487 909 719 814 477 857 840 726 699 915 753 869 820 815 440 930 491 624 672 740 917 464 844 382 498 931 822 727
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
973 762 878 953 496 704 936 979 884 763 504 926 879 736 994 886 940 995 981 927 765 942 968 887 832 948 508 890 985 752 943 997 972 891
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

(N,K) Polar Code

- $N = 2^n$
- 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: $u G_N$

Polar code example: (8,4)

- Reliability sequence: 1 2 3 5 4 6 7 8
 - Frozen: 1 2 3 5
 - Message: 4 6 7 8



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

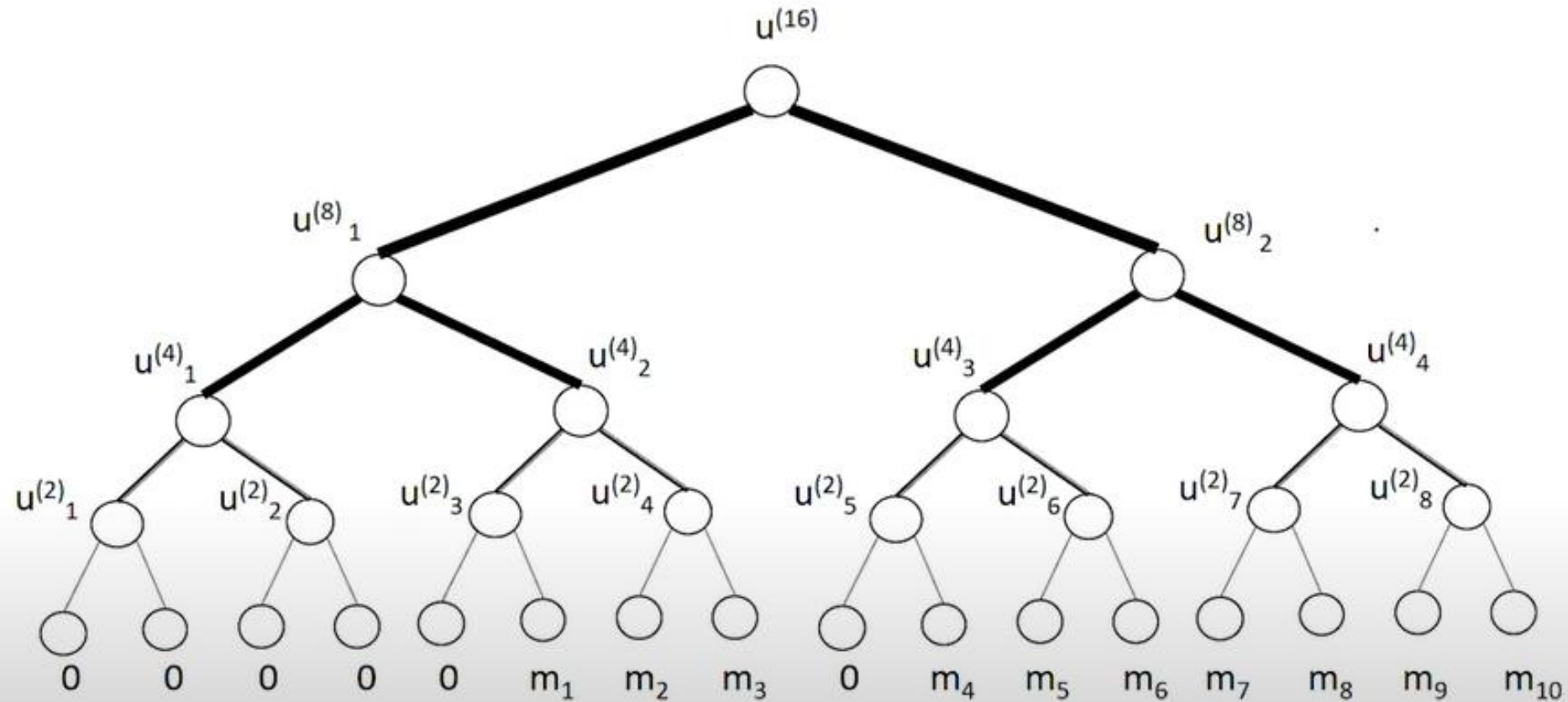
(16,10)

- 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

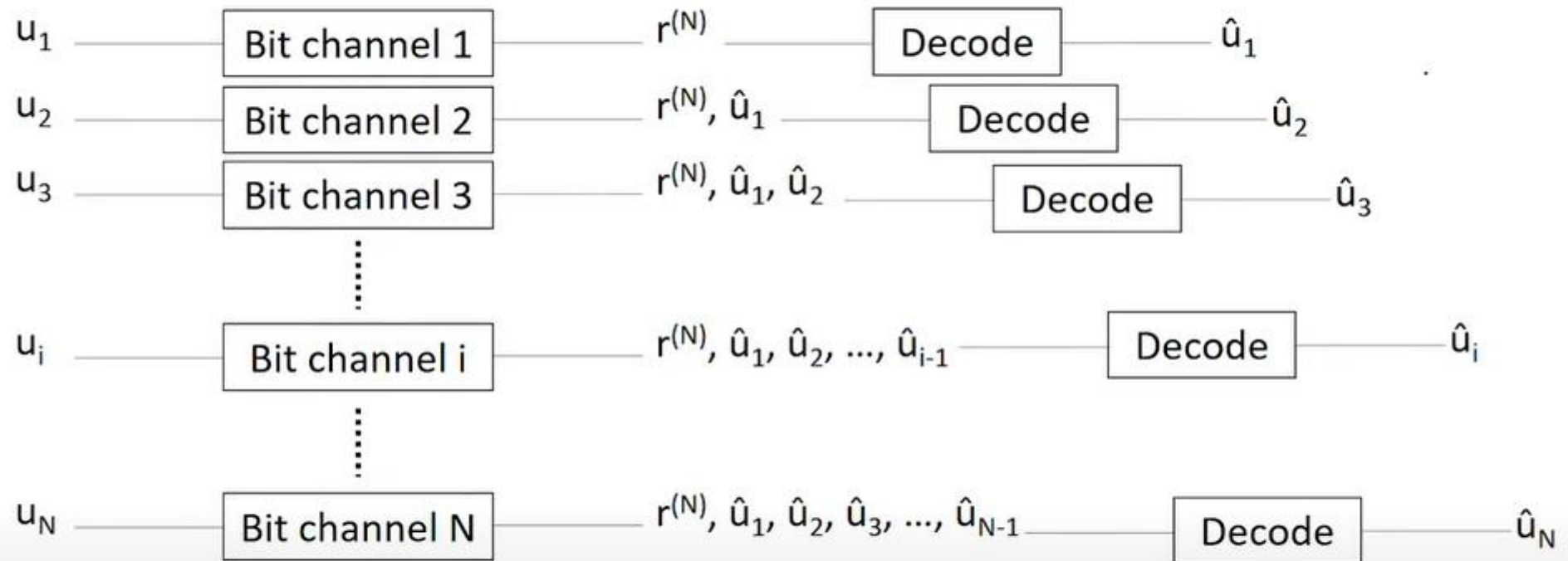
(32,20)

- 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

Binary tree representation: (16,10)



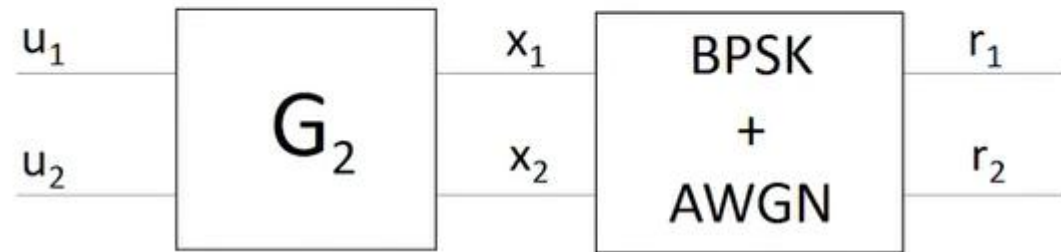
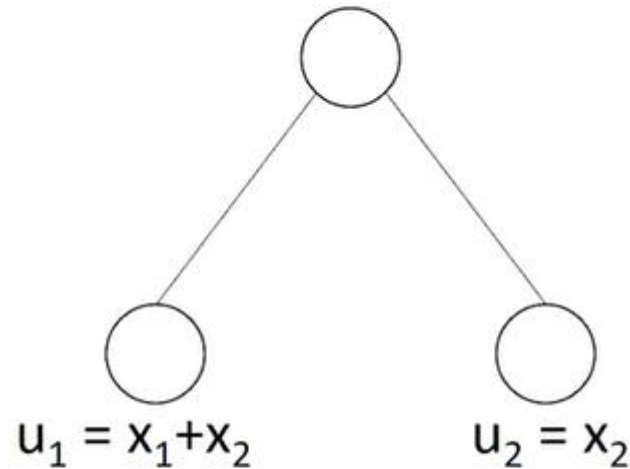
Successive Cancellation (SC) Decoding



- If bit i is frozen, $\hat{u}_i = 0$
- If bit i is a message bit, \hat{u}_i is found using a SISO decoder

Basic building block of SC decoder: N = 2

$$\mathbf{x} = [x_1 \ x_2] = [u_1 + u_2 \ u_2]$$



- SISO decode u_1 first (SPC)

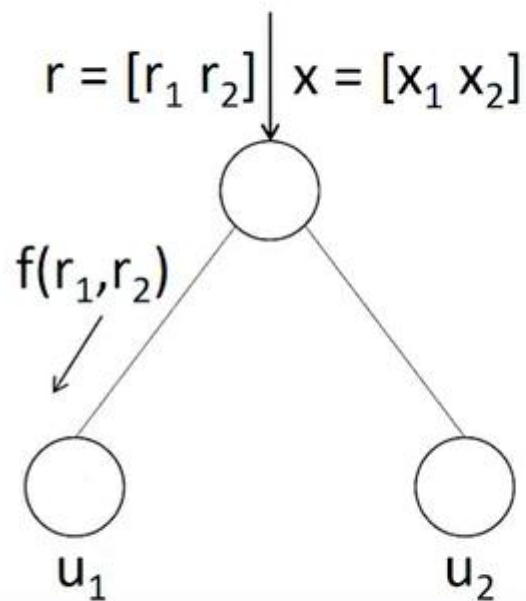
$$L(u_1) = f(r_1, r_2) = \text{sgn}(r_1) \text{sgn}(r_2) \min(|r_1|, |r_2|)$$
$$\hat{u}_1 = 0, \text{ if } L(u_1) \geq 0; \hat{u}_1 = 1, \text{ if } L(u_1) < 0$$

- Given \hat{u}_1 , decode u_2 (Rep)

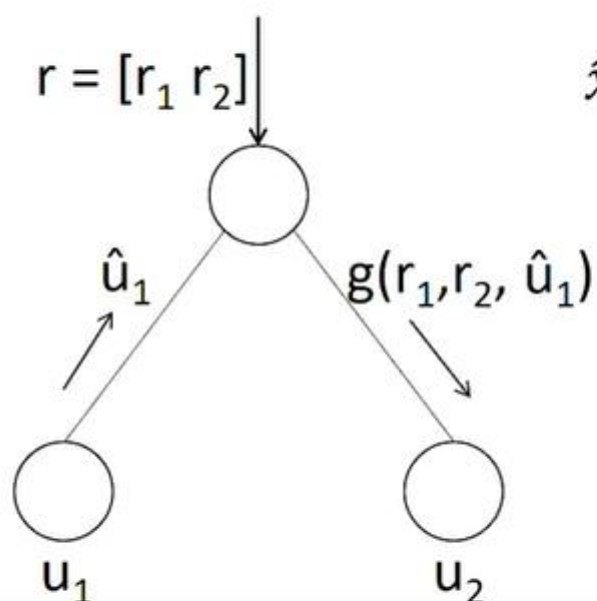
$$\text{If } \hat{u}_1 = 0, L(u_2) = r_2 + r_1 \quad (\mathbf{x} = [u_2 \ u_2])$$

$$\text{If } \hat{u}_1 = 1, L(u_2) = r_2 - r_1 \quad (\mathbf{x} = [\bar{u}_2 \ u_2])$$

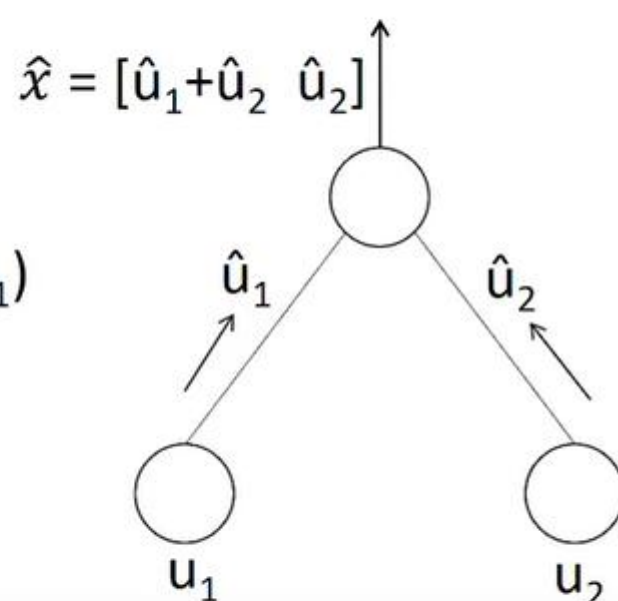
Basic building block: message-passing on tree



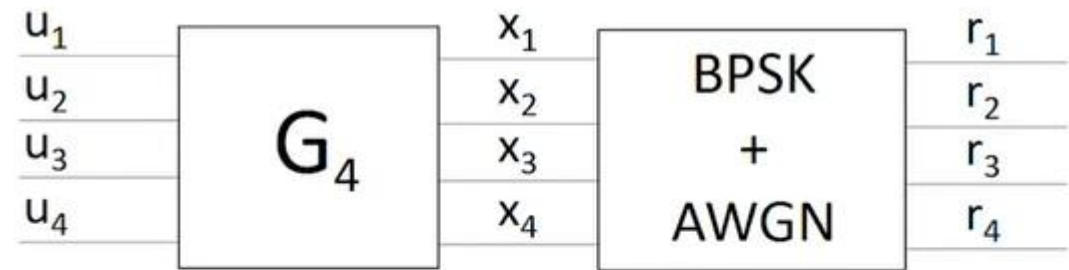
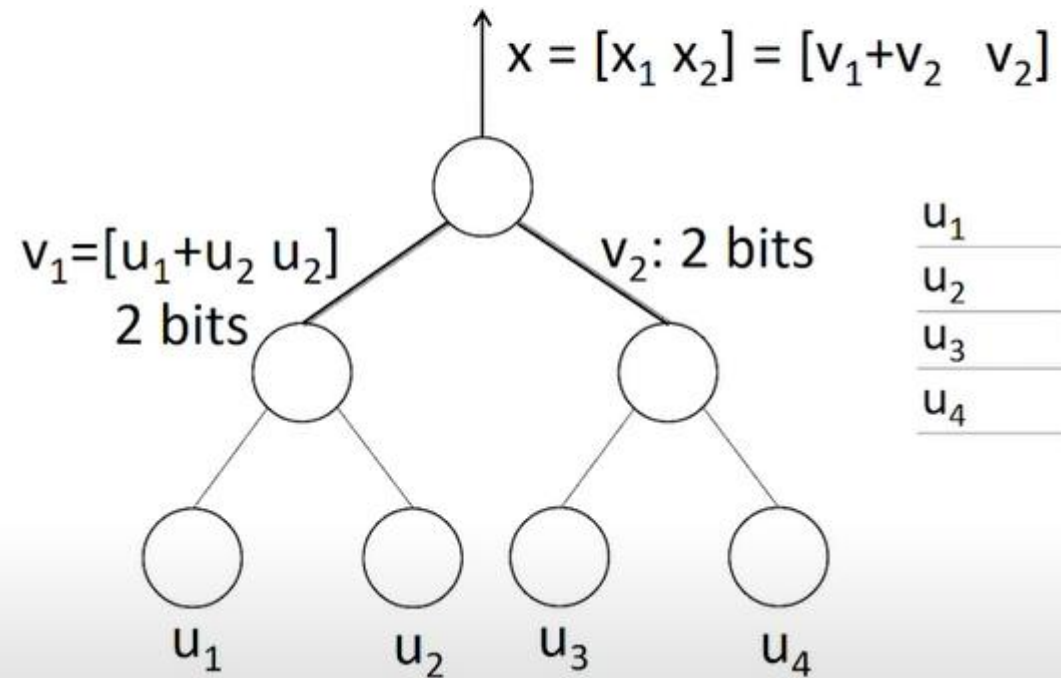
$$f(r_1, r_2) = \text{sgn}(r_1) \text{sgn}(r_2) \min(|r_1|, |r_2|)$$



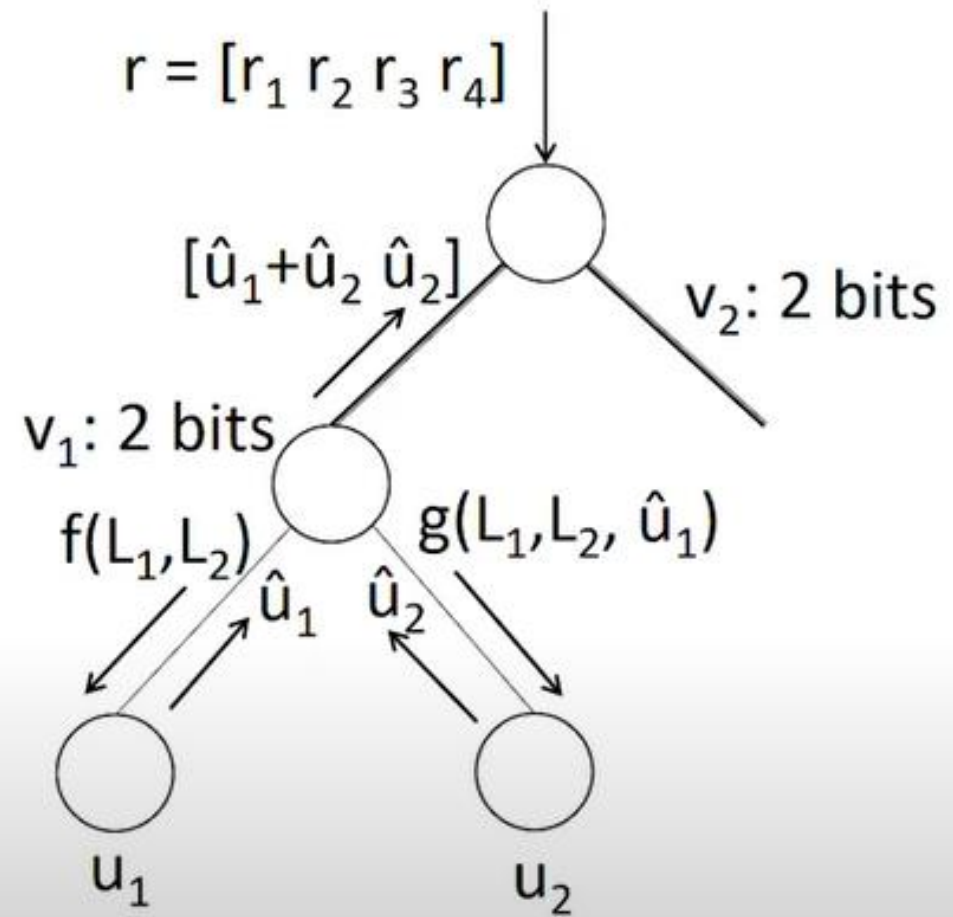
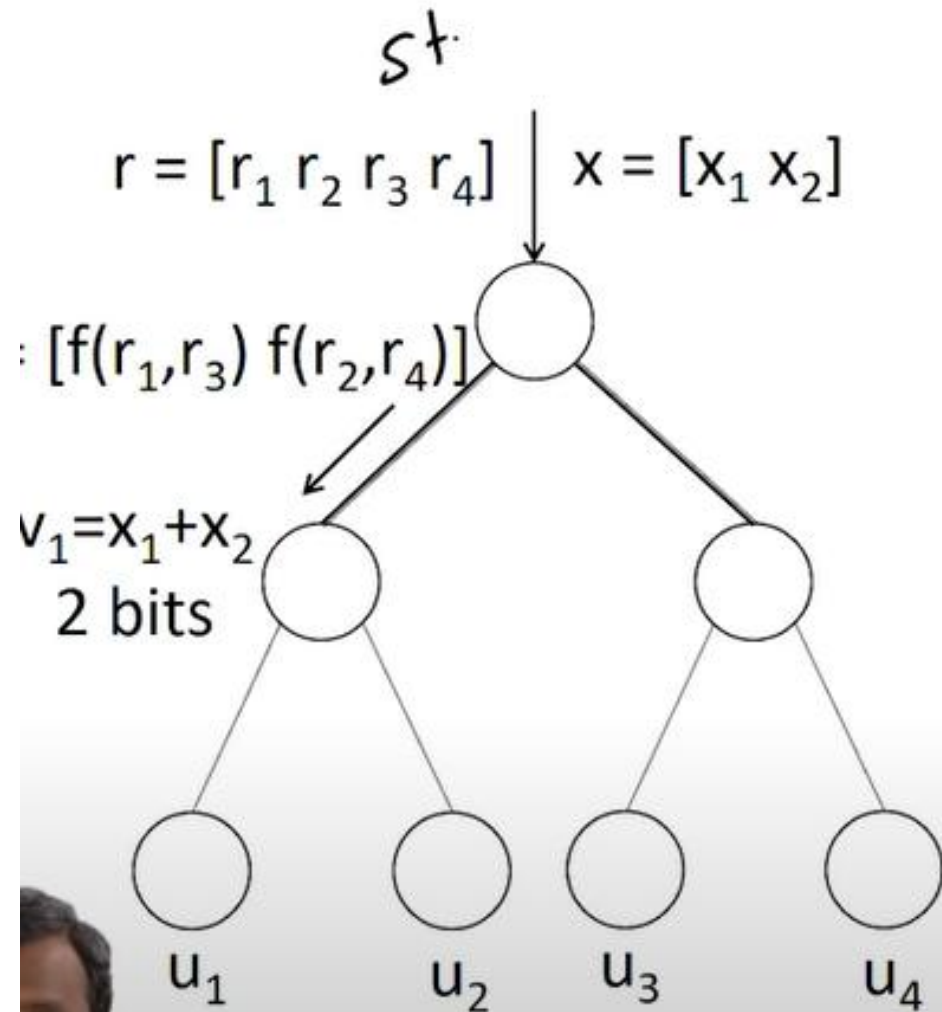
$$g(r_1, r_2, b) = r_2 + (1 - 2b) r_1$$



SC decoder: $N = 4$

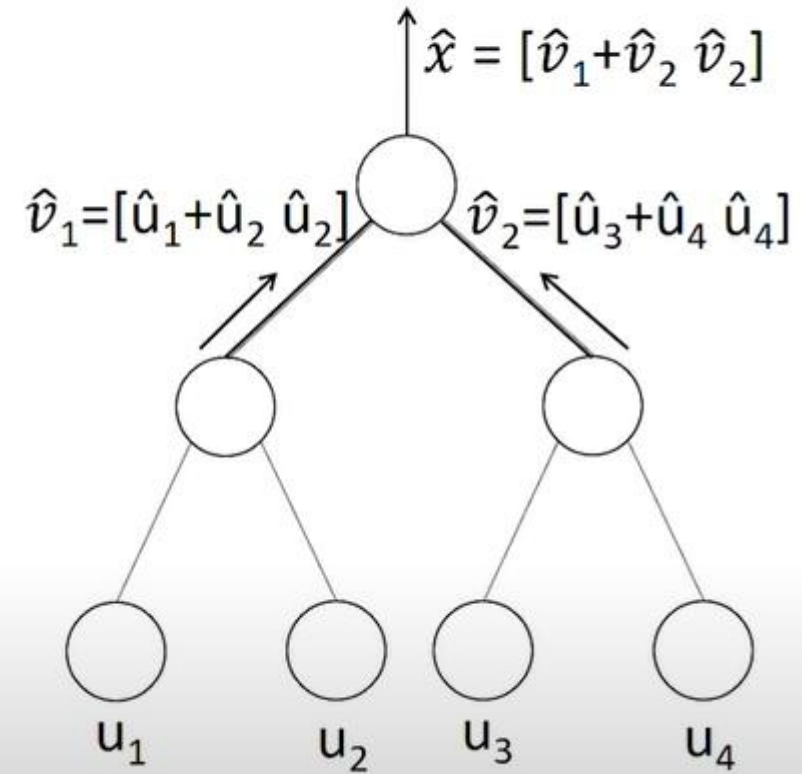
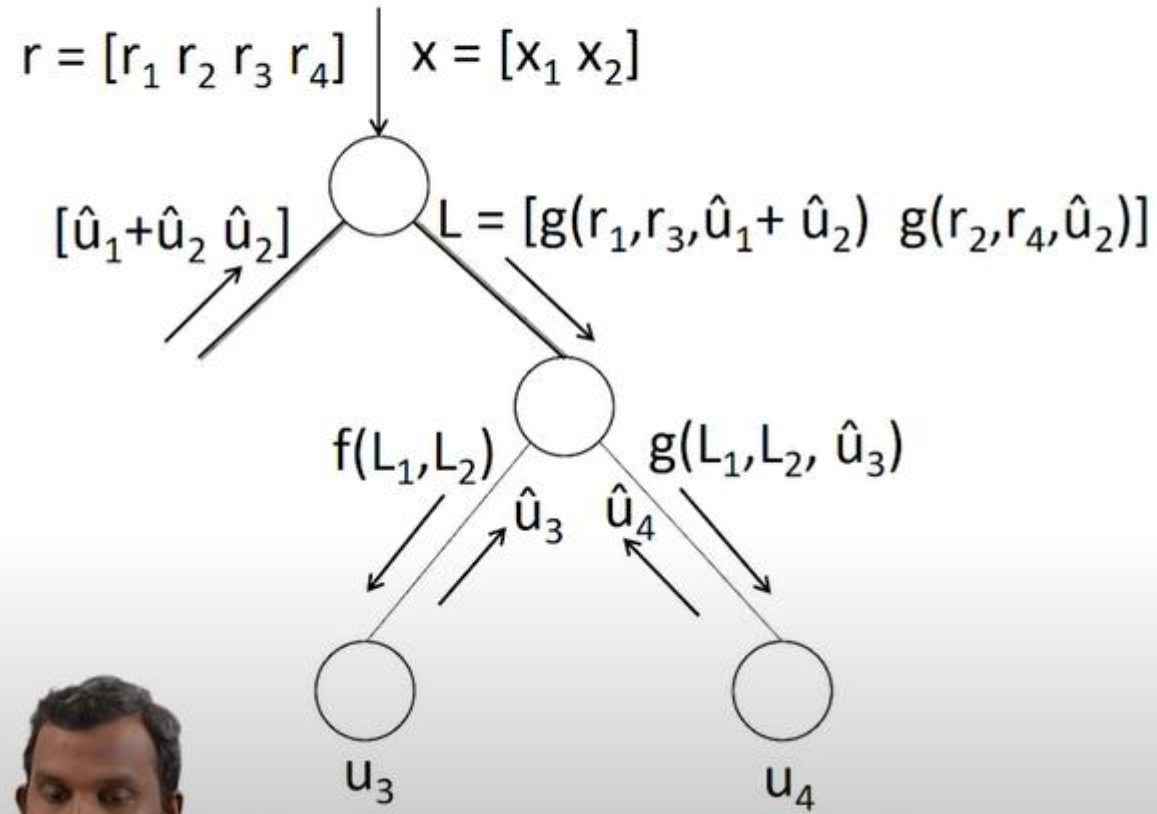


SC decoder: $N = 4$

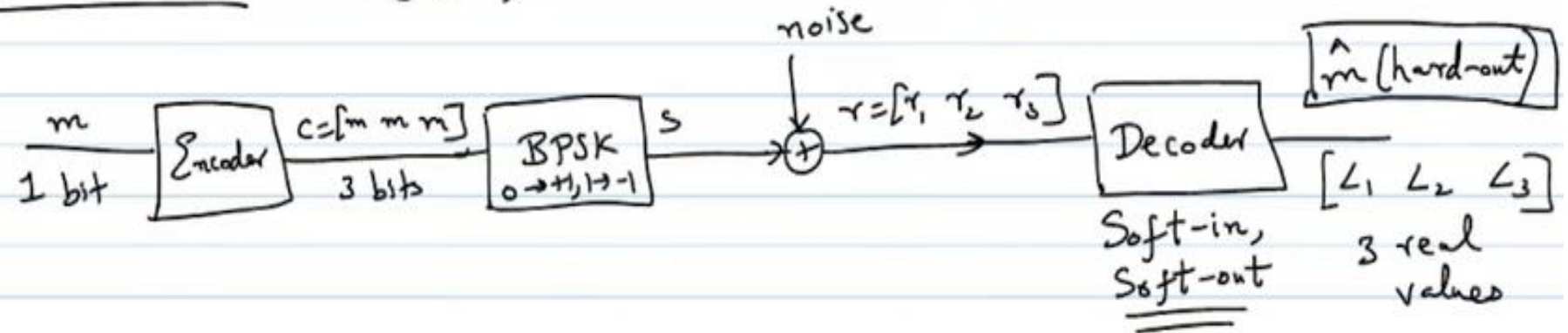


SC decoder: $N = 4$

s_t



Repetition code (3,1)



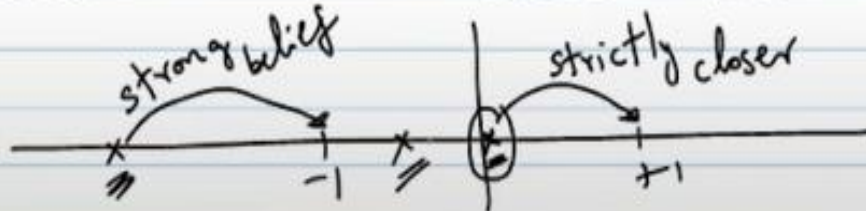
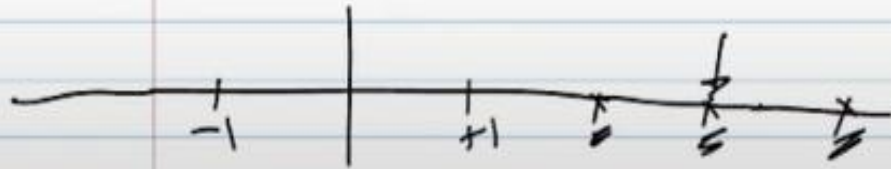
L_i : "belief" that bit c_i is 0.

$\sum x_i$

$$r = [3.1 \ 2.4 \ 4.3]$$

"belief" about first bit based on r_1 alone

$$r = [0.01 \ -2.2 \ -0.5]$$



$$L_1 = \left(\underbrace{r_1}_{\text{intrinsic}} + \underbrace{r_2 + r_3}_{\text{extrinsic}} \right) \times \frac{2}{\sigma^2}$$

$$L_i = (r_1 + r_2 + r_3) \times \left(\begin{array}{c} \text{+ve} \\ \text{factor} \end{array} \right)$$

↑ set it $\propto 1$.

Single parity check code (SPC code)

$(n, n-1)$ linear binary code
↑
 k

$$\underline{m} = [m_1, m_2, \dots, m_{n-1}] \xrightarrow{\text{Encoder}} \underline{c} = [m_1, m_2, \dots, m_{n-1}, p]$$

↓ parity

$$p = m_1 \oplus m_2 \oplus \dots \oplus m_{n-1}$$

xor of all message bits

$(3, 2)$ SPC code :

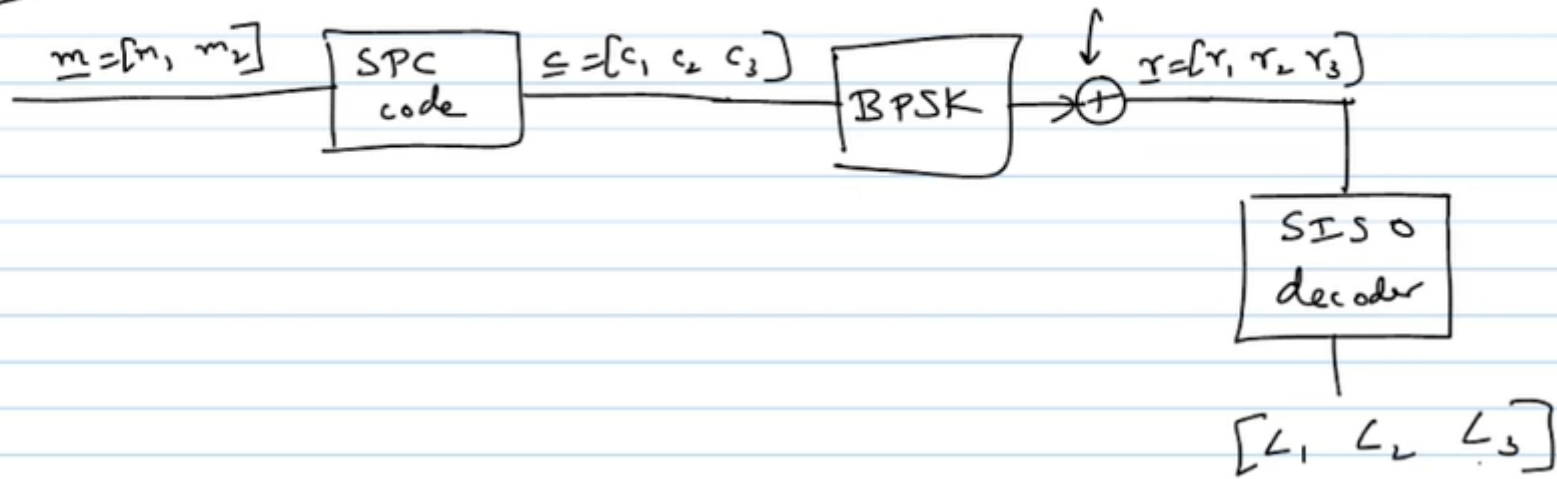
\underline{m}	\underline{c}
0 0	0 0 0
0 1	0 1 1
1 0	1 0 1
1 1	1 1 0

of 1s in \underline{c} : even

SPC code: consists of all even-weight vectors of length n

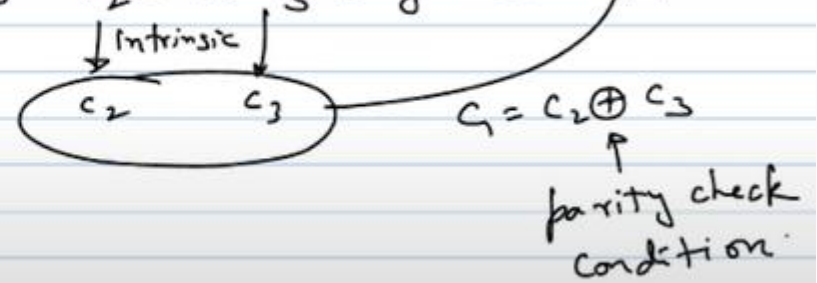
SISO decoder for SPC code

(3,2) code



$L_1 \leftarrow$ intrinsic of r_1

extrinsic: what do r_2 and r_3 say about c_1 ?



$$c_1 = c_2 \oplus c_3$$

$$l_2 = \log \frac{\Pr(c_2=0|r_2)}{\Pr(c_2=1|r_2)}$$

$$l_3 = \log \frac{\Pr(c_3=0|r_3)}{\Pr(c_3=1|r_3)}$$

Given p_2 & p_3 : what is $p_1 = \Pr(c_1=0|r_2, r_3)$?

c_1	$\leftarrow c_2$	c_3	
0	0	0	$p_1 = p_2 p_3 + (1-p_2)(1-p_3)$
0	1	1	
1	0	1	$1-p_1 = p_2(1-p_3) + (1-p_2)p_3$
1	1	0	

$$p_1 - (1-p_1) = p_2(p_3 - (1-p_3)) + (1-p_2)((1-p_3) - p_3)$$

$$p_1 - (1-p_1) = (p_2 - (1-p_2))(p_3 - (1-p_3))$$

$$\log \frac{p_1}{1-p_1} : \text{extrinsic LLR}$$

$$\tanh = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$\frac{p_1 - (1-p_1)}{p_1 + (1-p_1)} = \frac{(p_2 - (1-p_2))}{p_2 + (1-p_2)} \frac{(p_3 - (1-p_3))}{p_3 + (1-p_3)}$$

$$\frac{1 - \left(\frac{1-p_1}{p_1}\right)}{1 + \left(\frac{1-p_1}{p_1}\right)} = \frac{1 - \frac{1-p_2}{p_2}}{1 + \frac{1-p_2}{p_2}} \cdot \frac{1 - \frac{1-p_3}{p_3}}{1 + \frac{1-p_3}{p_3}}$$

$$\frac{1 - e^{-l_{ext,1}}}{1 + e^{-l_{ext,1}}} = \frac{1 - e^{-l_2}}{1 + e^{-l_2}} \cdot \frac{1 - e^{-l_3}}{1 + e^{-l_3}}$$

$$c_1 = c_2 \oplus c_3 \xRightarrow{\text{tanh rule}} \boxed{\tanh\left(\frac{l_{ext,1}}{2}\right) = \tanh\left(\frac{l_2}{2}\right) \cdot \tanh\left(\frac{l_3}{2}\right)}$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

odd function:
 $x < 0 \quad \tanh(x) < 0$
 $x > 0 \quad \tanh(x) > 0$

Signs : $\text{Sgn}(l_{\text{ext},1}) = \text{Sgn}(l_2) \cdot \text{Sgn}(l_3)$

Absolute value : $\tanh\left(\frac{|l_{\text{ext},1}|}{2}\right) = \tanh\left(\frac{|l_2|}{2}\right) \tanh\left(\frac{|l_3|}{2}\right)$

$$\log \tanh\left(\frac{|l_{\text{ext},1}|}{2}\right) = \log \tanh \frac{|l_2|}{2} + \log \tanh \frac{|l_3|}{2}$$

$$f(|l_{\text{ext},1}|) = f(|l_2|) + f(|l_3|)$$

$$|l_{\text{ext},1}| = f\left(f(|l_2|) + f(|l_3|)\right)$$

$$\text{Sgn}(l_{\text{ext},1}) = \text{Sgn}(l_2) \text{Sgn}(l_3)$$

$x > 0$
 $f(x) = \left| \log \tanh \frac{x}{2} \right|$
 $f^{-1}(x) = f(x)$

Signs : $\text{Sgn}(l_{\text{ext},1}) = \text{Sgn}(l_2) \cdot \text{Sgn}(l_3)$

Absolute value : $\tanh\left(\frac{|l_{\text{ext},1}|}{2}\right) = \tanh\left(\frac{|l_2|}{2}\right) \tanh\left(\frac{|l_3|}{2}\right)$

$$\log \tanh\left(\frac{|l_{\text{ext},1}|}{2}\right) = \log \tanh \frac{|l_2|}{2} + \log \tanh \frac{|l_3|}{2}$$

$$f(|l_{\text{ext},1}|) = f(|l_2|) + f(|l_3|)$$

$$|l_{\text{ext},1}| = f\left(f(|l_2|) + f(|l_3|)\right)$$

$$\text{Sgn}(l_{\text{ext},1}) = \text{Sgn}(l_2) \text{Sgn}(l_3)$$

$x > 0$
 $f(x) = \left| \log \tanh \frac{x}{2} \right|$
 $f^{-1}(x) = f(x)$

$$C_2 = C_1 \oplus C_3 \quad \begin{aligned} |l_{ext,2}| &= f(f(|l_1|) + f(|l_3|)) \\ \text{sgn}(l_{ext,2}) &= \text{sgn}(l_1) \text{sgn}(l_3) \end{aligned}$$

$$C_3 = C_1 \oplus C_2 \quad \begin{aligned} |l_{ext,3}| &= f(f(|l_1|) + f(|l_2|)) \\ \text{sgn}(l_{ext,3}) &= \text{sgn}(l_1) \text{sgn}(l_2) \end{aligned}$$

$$S = f(|l_1|) + f(|l_2|) + f(|l_3|) \quad P = \text{sgn}(l_1) \text{sgn}(l_2) \text{sgn}(l_3)$$

$$l_{ext,1} = f(\underbrace{S - f(|l_1|)}_{\text{absolute value}}) \cdot \underbrace{P \cdot \text{sgn}(l_1)}_{\text{sign}}$$

$$C_2 = C_1 \oplus C_3 \quad \begin{aligned} |l_{ext,2}| &= f(f(|l_1|) + f(|l_3|)) \\ \text{sgn}(l_{ext,2}) &= \text{sgn}(l_1) \text{sgn}(l_3) \end{aligned}$$

$$C_3 = C_1 \oplus C_2 \quad \begin{aligned} |l_{ext,3}| &= f(f(|l_1|) + f(|l_2|)) \\ \text{sgn}(l_{ext,3}) &= \text{sgn}(l_1) \text{sgn}(l_2) \end{aligned}$$

$$S = f(|l_1|) + f(|l_2|) + f(|l_3|) \quad P = \text{sgn}(l_1) \text{sgn}(l_2) \text{sgn}(l_3)$$

$$l_{ext,1} = \underbrace{f(S - f(|l_1|))}_{\text{absolute value}} \cdot \underbrace{P \cdot \text{sgn}(l_1)}_{\text{sign}}$$