Error Control Coding - Low-Density Parity-Check Codes

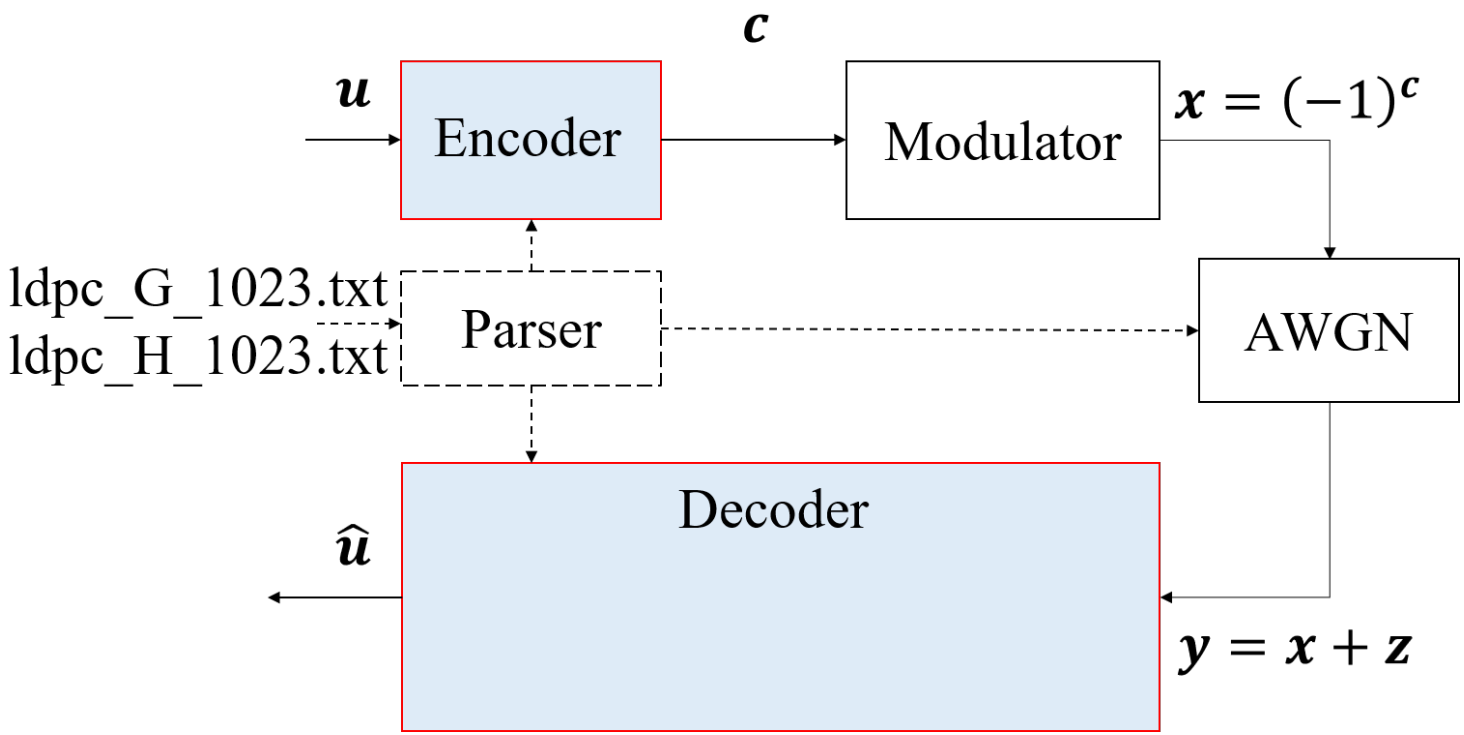
Chao-Yu Chen (陳昭羽)

國立成功大學
電機工程學系/電腦與通信工程研究所

May 30, 2023

Low-Density Parity-Check Codes

- Consider the $(1023, 781)$ low-density parity-check (LDPC) code with block length $n = 1023$, dimension $k = 781$, row weight $\rho = 32$, and column weight $\gamma = 32$.
- Block Diagram:



Encoding of Low-Density Parity-Check Codes

- Use the recursion

$$u_{l+6} = u_{l+1} \oplus u_l, \qquad \text{for } l \geq 0$$

with the initial conditions

$$u_0 = 1, u_1 = u_2 = u_3 = u_4 = u_5 = 0$$

to generate $k = 781$ information bits.

- The generated sequence is $100000100001\ldots$ with period 63.

Low-Density Parity-Check Codes

Additive white Gaussian noise (AWGN) channel:

- For a binary code of rate R with BPSK modulation, the noise variance σ^2 is given by

$$\sigma^2 = \left(2R\frac{E_b}{N_0}\right)^{-1}$$

  where E_b/N_0 is the bit signal-to-noise ratio (SNR).
- The code rate is $R = 781/1023$ here.

Low-Density Parity-Check Codes

- Please use the following pseudo code:

```
#define IA 16807
#define IM 2147483647
#define AM (1.0/IM)
#define IQ 127773
#define IR 2836
#define NTAB 32
#define NDIV (1+(IM-1)/NTAB)
#define EPS 1.2e-7
#define RNMX (1.0-EPS)

main()
{
    ...
    long *idum;
    idum = (long *)malloc(sizeof(long));
    *idum = SEED;  //SEED must be a negative integer.
    ...
    ...
}
```

Low-Density Parity-Check Codes

- Please use **normal()** to output two independent normal random variables, n_1 and n_2.

```
normal(n1, n2, σ)
{
    do{
        x1 = ran1(idum);
        x2 = ran1(idum);
        x1 = 2x1 − 1;
        x2 = 2x2 − 1;
        s = x1^2 + x2^2;
    } while (s ≥ 1.0)
    n1 = σx1 √(−2 ln s/s);
    n2 = σx2 √(−2 ln s/s);
}
```

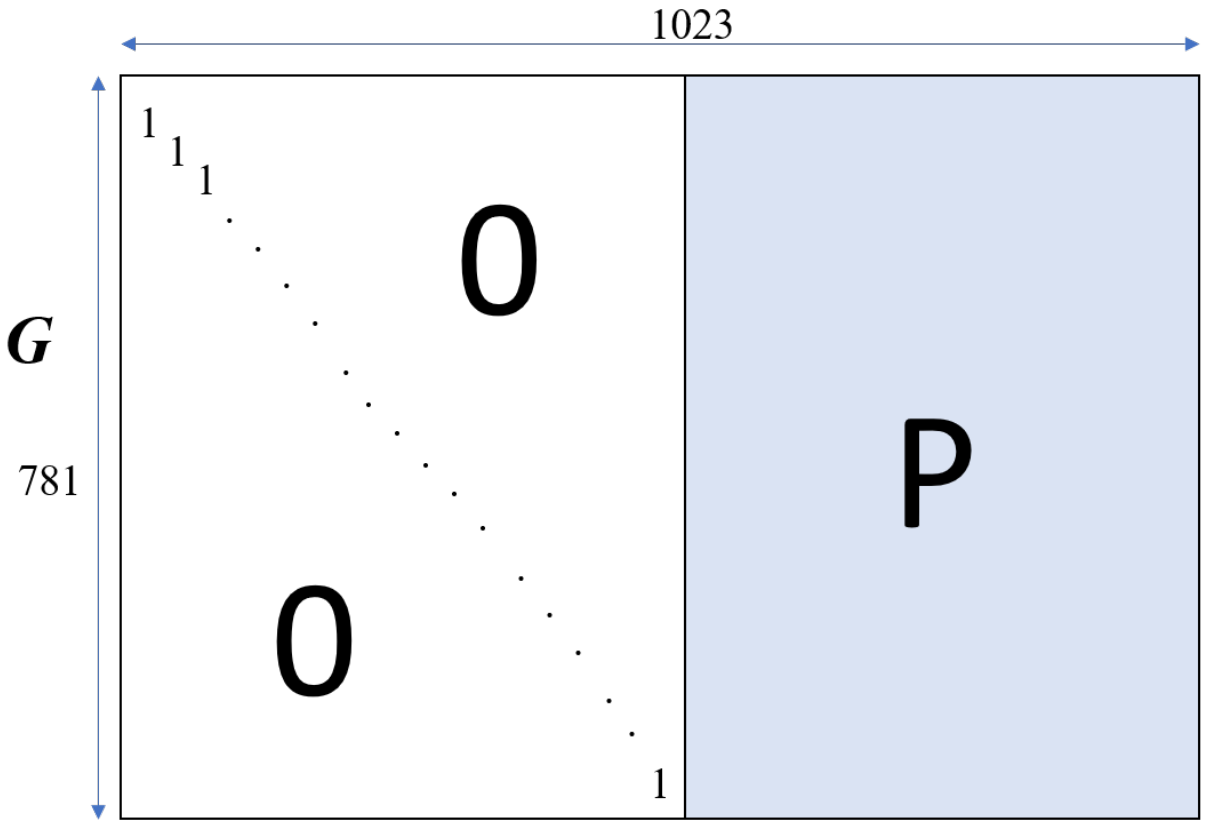
Low-Density Parity-Check Codes

- Please use **ran1()** to generate a random variable uniformly distributed in the interval $(0, 1)$.

```
ran1(long *idum)
{
    int j;
    long k;
    static long iy=0;
    static long iv[NTAB];
    double temp;
    if (*idum <= 0 || !iy){
       if (-(*idum) < 1) *idum=1;
       else *idum = -(*idum);
       for (j=NTAB+7;j>=0;j--){
           k=(*idum)/IQ;
           *idum=IA*(*idum-k*IQ)-IR*k;
           if (*idum < 0) *idum += IM;
           if (j < NTAB) iv[j] = *idum;
       }
       iy=iv[0];
    }
    k=(*idum)/IQ;
    *idum=IA*(*idum-k*IQ)-IR*k;
    if (*idum < 0) *idum += IM;
    j=iy/NDIV;
    iy=iv[j];
   iv[j] = *idum;
   if ((temp=AM*iy) > RNMX) return RNMX;
   else return temp;
}
```

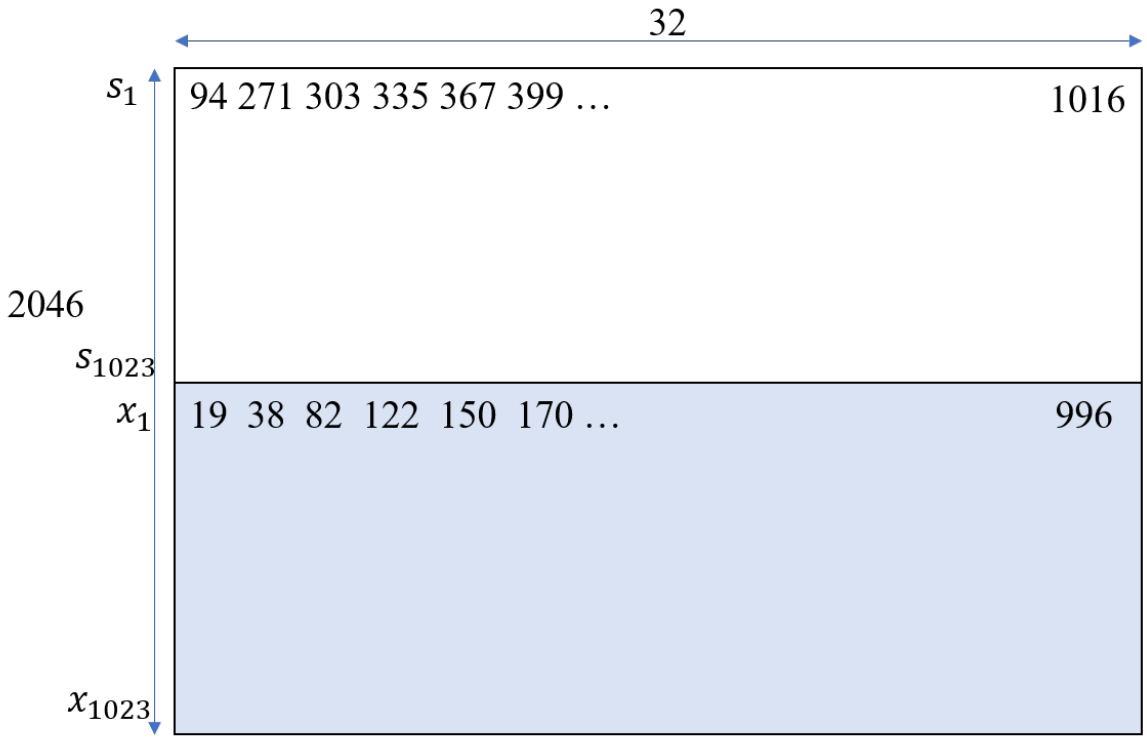
Low-Density Parity-Check Codes

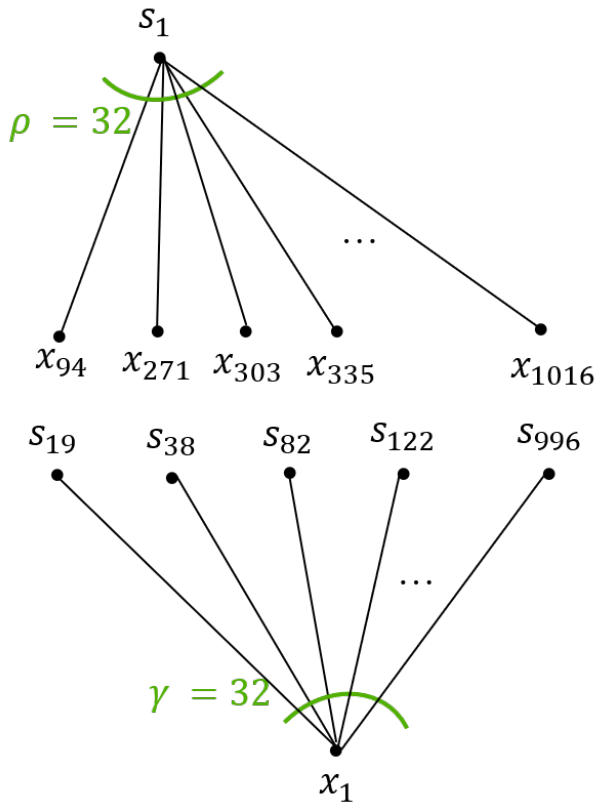
- Input ldpc_G_1023.txt:



Low-Density Parity-Check Codes

- Input ldpc_H_1023.txt:





Decoding of Low-Density Parity-Check Codes

程式DEMO方式

- 沒有DEMO (不需跟助教比對)。
- 5/31開始進行期末Project。

上台報告

- Bonus
- 已經完成的同學做簡報分享，需要先跟我報名登記 (限量)。
- 6/7 上課時間。

報告繳交

- 6/14 12:00前私訊檔案給“助教_高子傑”。
- 檔案名稱格式: 學號_姓名.zip 例如: E94081042_許博士.zip
- 壓縮檔內包含所有的.cpp (必須要有註釋) 以及 一個報告.pdf。
- 報告檔名格式: Project2_學號_姓名.pdf 例如: Project2_E94081042_許博士.pdf

Decoding of Low-Density Parity-Check Codes

報告繳交內容

- 報告必要內容 (Baseline):
  - 解碼採用Sum-Product Algorithm;
  - 系統架構圖 (細部blcok diagram);
  - 程式流程解釋 (流程或是Pseudo code);
  - 參數設定: number of maximum iterations = 100 等其他參數;
  - 模擬數據 (表格): SNR (dB)、No. decoded bits、No. decoded bit errors、bit error rate (BER);
  - 模擬圖: 至少到 BER $= 10^{-4}$.
- 報告加分內容 (Bonus):
  - 不同 iteration個數的比較，收斂的狀況與分析;
  - 其他演算法: bit-flipping algorithm、min-sum algorithm、或其他文獻上的做法。
  - 錯誤率能夠越低越好，但是要數值穩定的曲線。
  - BER之外，提供Block error rate (BLER)的模擬效能圖。
  - 其他觀察或比較。

Decoding of Low-Density Parity-Check Codes

參考數值：

- Sum-Product Algorithm (Maximum 100 iterations):
  - BER = 4.0×10^{-2} at $E_b/N_0 = 2.2$ dB;
  - BER = 1.0×10^{-3} at $E_b/N_0 = 3.0$ dB.