[시뮬레이터] Scrambler

2016707044 김영웅

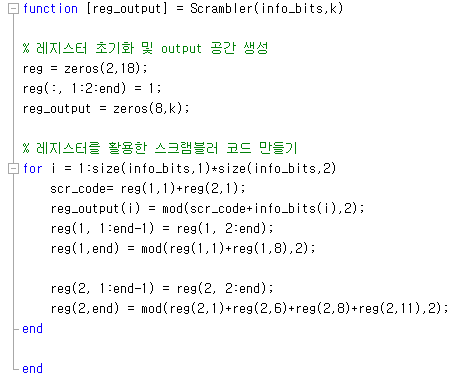
특정 페턴의 information bits나 특정 화일에서 information bits를 읽어,

이를 scrambling하고 기존 코드와 통합하여 descrambling까지 수행하도록 하시오.

사용할 scrambler : 자신만의 초기값을 사용하고, IQ 출력 중 하나만 사용하시오.

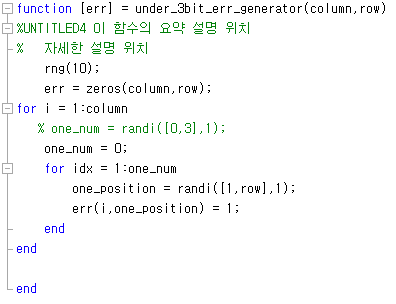
**INDEX**

1. **Common Function**
   1. **Scrambler**
   2. **under\_3bit\_err\_generator**
2. **Linear Block Code**
3. **Cyclic code**
4. **result**
5. Common function
   1. **Scrambler**



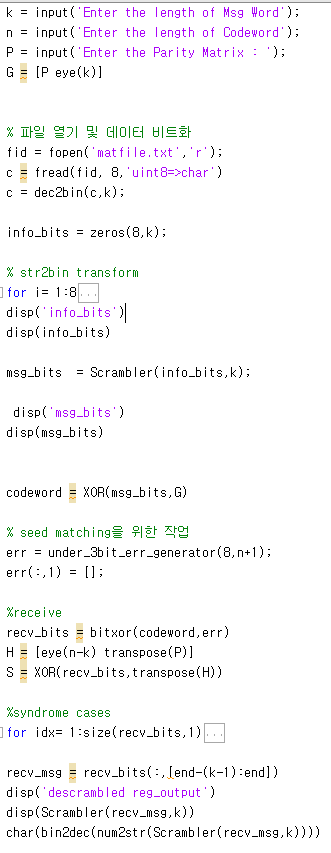
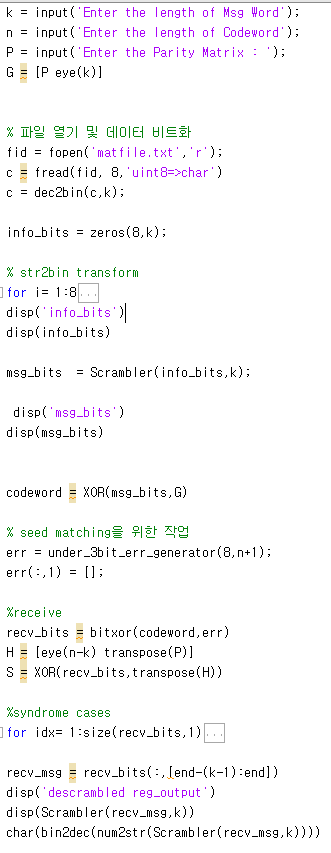
Scrambler를 나타내는 함수이다. 시스템의 I 출력을 사용하였고 입력으로 파일의 bits를 받아 scrambling하여 출력하는 함수이다.

* 1. **under\_3bit\_err\_generator**

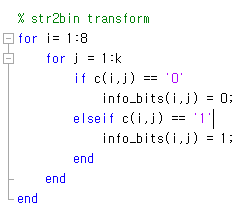


한 행 당, 랜덤 한 위치에 임의의 개수의 1(최대 3개) 생성하는 error generator 함수이다. 한 행당 1의 개수 한도를 3으로 한 것은 error correction을 좀더 효과적으로 보여주기 위해서 지정하였다. Error의 개수를 나타내는 ‘one\_num’ 변수를 0으로 하여 error – free 상황에서 송 수신이 잘 됨을 확인하고, 해당 줄을 주석처리 하고 아래의 주석 처리 코드를 실행하여 랜덤 에러 상황에서 송수신을 확인하였다.

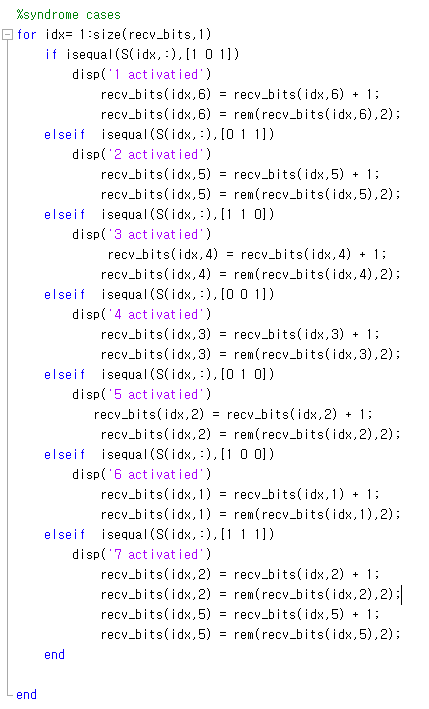
1. Linear block code
   1. **1. Entire code**



* 1. **subset**

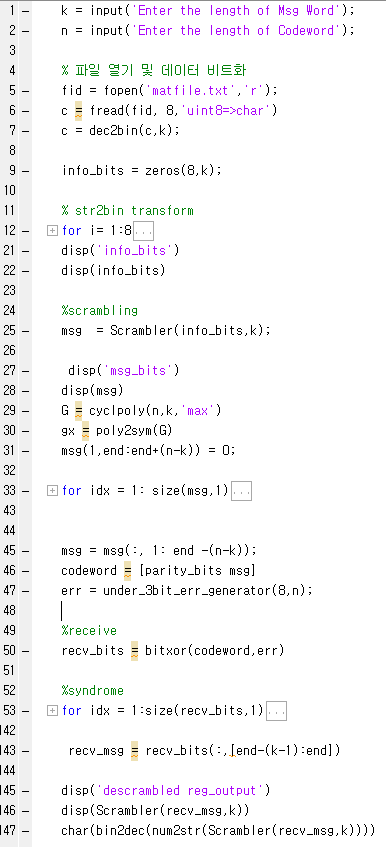


String 형태의 2진수 배열을 binary 형태로 배열에 입력하기 위한 for문이다.



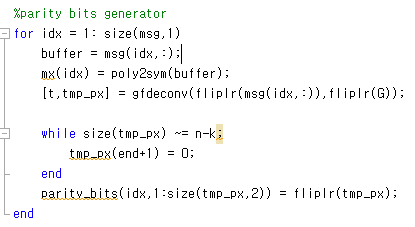
Syndrome case들을 수정 가능한 에러 형태들에 대해 mapping 하기 위한 switch 구문이다.

1. Cyclic code
   1. **Entire code**



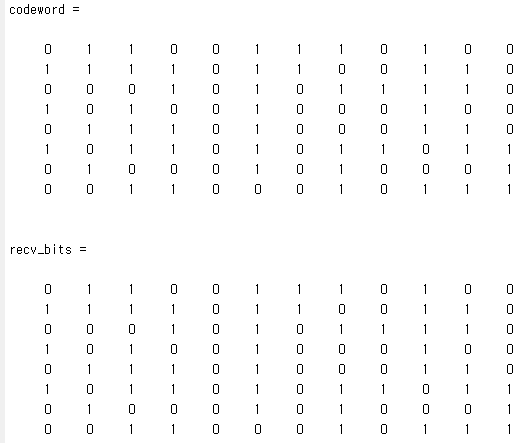
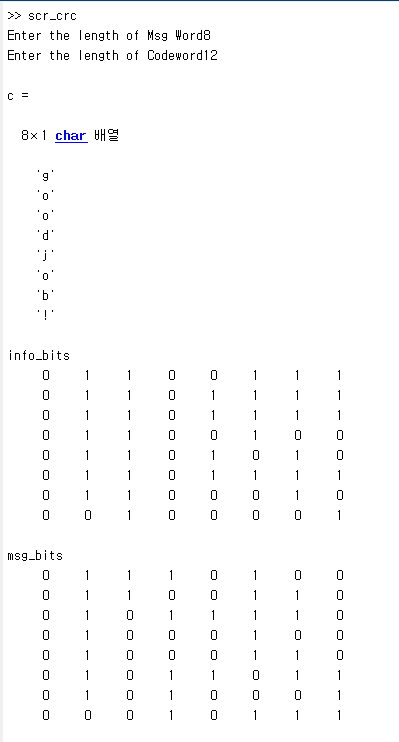
* 1. **Subset**

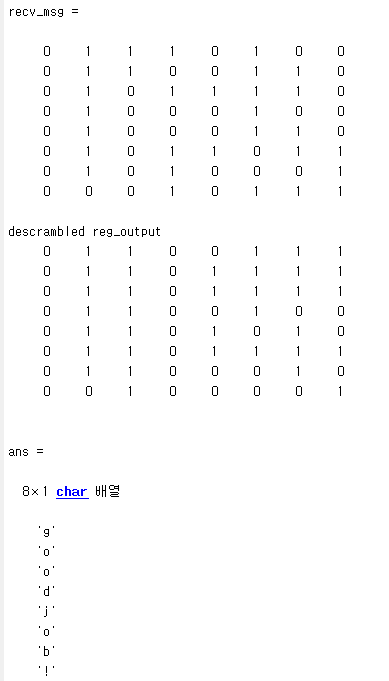
Str2bin과 syndrome 부분은 linear block code와 매우 유사하다.



Parity bits을 생성하기 위해서 generator polynomial로 msg를 나누고 전후 처리를 하기 위한 구문이다.

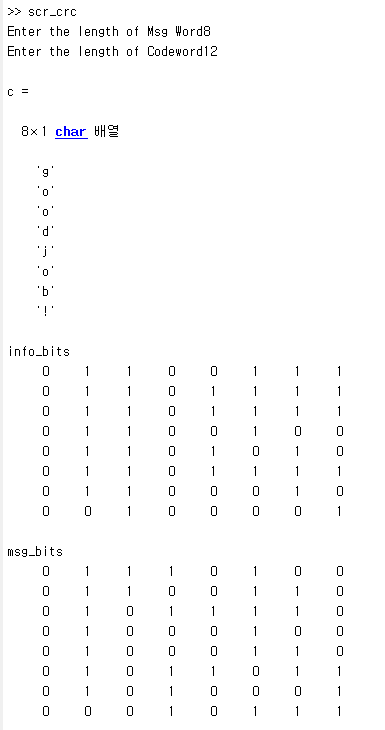
1. Result
   * 1. **Error – free situation**

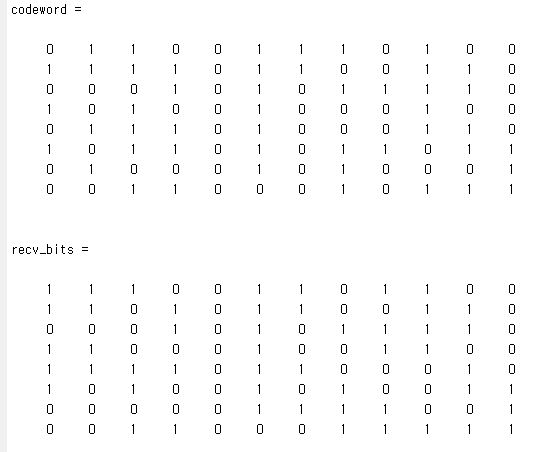
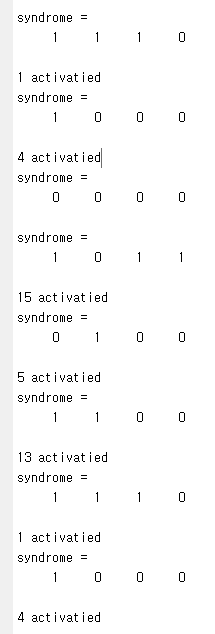




송신 메시지 “goodjob!”이 그대로 수신되는 것을 볼 수 있다.

* + 1. **Random error situation – cyclic code**

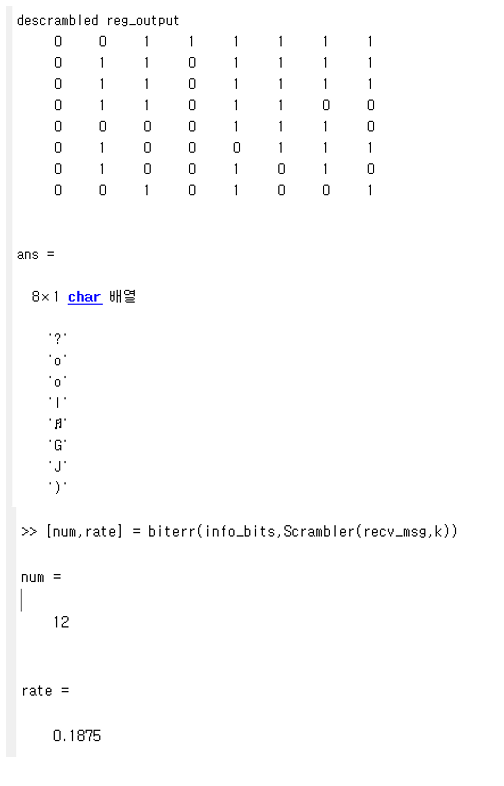


Cyclic code

Block error rate: 6/8 = 75%

Bit error rate: 13/96 = 20.3%



Linear block code

Block error rate: 6/8 = 75%

Bit error rate: 12/96 = 18.75.3%