

Error Control 프로그래밍 과제

아래와 같은 세 가지 에러제어의 인코더/디코더 프로그램을 C 언어로 작성함. 그리고 프로그램의 소스코드, 실행파일과 함께 프로그램의 동작에 대한 실행 스크린샷(이 파일 뒷부분에 있는 과제 제출용 테스트데이터를 적용한)을 zip 파일 형태로 과제 게시판에 6/3(화)까지 올릴 것. (추가적인 점검을 위한 테스트데이터는 스스로 만들어서 확인하기 바람.)

1) Error detection 을 위한 CRC 인코더/디코더

아래의 G 를 divisor 로 하는 16-bit 의 데이터를 CRC 인코딩과 디코딩하는 프로그램을 C 로 설계 (즉 $n=24, k=16$ 인 CRC 인코더와 디코더)

$$G = X^8 + X^2 + X + 1$$

2) Error correction 을 위한 (7,4) Hamming code 인코더/디코더

(7, 4) Hamming code 의 인코더와 디코더를 C 프로그램으로 설계 ($n=7, k=4$ 인 Hamming code 의 인코더와 디코더)

3) CRC+(7,4) Hamming code 를 이용한 2-layer channel coding 인코더/디코더

위 1)번 CRC 의 인코더 출력을 2)번 (7, 4) Hamming 인코더의 입력으로 사용하여 codeword 를 출력하는 송신단과 수신된 codeword 를 (7, 4) Hamming 디코더와 CRC 의 디코더를 순차적으로 수행하는 수신단으로 구성된 2-layer channel coding 을 설계 (CRC 와 Hamming code 의 순서에 주의할 것 그리고 CRC 의 출력은 24bit, Hamming code 의 입력은 4bit 이므로 각각 data 의 넓이를 고려해야 함)

➤ 인코더/디코더의 설계 상세 설명

1)

CRC 의 경우

- 인코더의 입력은 16-bit 를 키보드로 받고 출력은 24-bit 의 (데이터+FCS)
- 디코더의 입력은 24-bit 를 키보드로 받고 출력은 “No error”와 “ERROR!”로 표기

2)

Hamming code 의 경우

- 인코더의 입력은 4-bit 를 키보드로 받고 출력은 7-bit 의 codeword
- 디코더의 입력은 7-bit 의 codeword 이고 출력은 에러가 수정된 4-bit 의 데이터를 표기

s	t	s	t	s	t	s	t
0000	0000000	0100	0100110	1000	1000101	1100	1100011
0001	0001011	0101	0101101	1001	1001110	1101	1101000
0010	0010111	0110	0110001	1010	1010010	1110	1110100
0011	0011100	0111	0111010	1011	1011001	1111	1111111

$$t = G^T s$$

s : data

$$G = \begin{bmatrix} 1000101 \\ 0100110 \\ 0010111 \\ 0001011 \end{bmatrix}$$

t : codeword

G : generator matrix

r : received codeword

$$H = \begin{bmatrix} 1110100 \\ 0111010 \\ 1011001 \end{bmatrix}$$

z : syndrome

H : parity check matrix

$$z = Hr$$

Syndrome z	000	001	010	011	100	101	110	111
Unflip this bit	<i>none</i>	r_7	r_6	r_4	r_5	r_1	r_2	r_3

위의 syndrome z 를 구한 뒤 위의 표를 이용하여 어느 비트가 오류인지 결정.

3)

2-layer channel coding 의 경우

- 인코더의 입력은 16-bit 를 키보드로 받고 출력은 42-bit 의 codeword
- 디코더의 입력은 42-bit 를 키보드로 받고 출력은 에러가 없거나 수정되었을 때에는 16-bit 데이터를 출력하고 수정되지 않는 에러가 있을 때에는 “ERROR!”로 표기

➤ 점검용 TEST Data

1) CRC-8 인코더

데이터: 1000110101000111

결과: 1000110101000111 10001101

데이터: 0110010100101100

결과: 0110010100101100 01110000

데이터: 0010100101011000

결과: 0010100101011000 10011100

데이터: 1100001000100100

결과: 1100001000100100 00111011

데이터: 1101101100101101

결과: 1101101100101101 11101110

2) CRC-8 디코더

데이터: 100011010100011110001101

결과: No error

데이터: 011001010010110001110000

결과: No error

데이터: 001010010101100010011100

결과: No error

데이터: 110000100110010000111011

결과: ERROR!

데이터: 110110101010110111101110

결과: ERROR!

데이터: 000010110100001000001001

결과: ERROR!

데이터: 101100101010011100011011

결과: ERROR!

3) (7, 4) Hamming 인코더

s	t	s	t	s	t	s	t
0000	0000000	0100	0100110	1000	1000101	1100	1100011
0001	0001011	0101	0101101	1001	1001110	1101	1101000
0010	0010111	0110	0110001	1010	1010010	1110	1110100
0011	0011100	0111	0111010	1011	1011001	1111	1111111

4) (7, 4) Hamming 디코더

데이터: 0101101

결과: z=000 (no error corrected)

데이터: 1010010

결과: z=000 (no error corrected)

데이터: 1001110

결과: z=000 (no error corrected)

데이터: 0011000

결과: z=100 (r5 corrected)

데이터: 1110110

결과: z=010 (r6 corrected)

데이터: 0100001

결과: z=111 (r3 corrected)

과제 제출용 테스트 데이터

1) CRC-8 인코더 (입력 16-bit, 출력 24-bit)

입력: 1111101100101101 (FB2D)

출력: 1111101100101101 01000000 (FB2D 40)

입력: 0000101100110010 (0B32)

출력: 0000101100110010 00001001 (0B32 09)

입력: 1011001010100111 (B2A7)

출력: 1011001010100111 00011001 (B2A7 19)

입력: 0111010110100001 (75A1)

출력: 0111010110100001 10001101 (75A1 8D)

2) CRC-8 디코더 (입력 24-bit, 출력 No error/ ERROR!!)

입력: 111110110010110101000000 (FB2D40)

출력: No error

입력: 000010110011001000001001 (0B3209)

출력: No error

입력: 101000101010011100011001 (A2A719)

출력: ERROR!!

입력: 010101111010000110011101 (57A19D)

출력: ERROR!!

3) Hamming 인코더 (입력 4-bit, 출력 7-bit)

입력: 0001

출력: 0001 011

입력: 0101

출력: 0101 101

입력: 1011

출력: 1011 001

입력: 1110

출력: 1110 100

4) Hamming 디코더 (입력 7-bit, 출력 4-bit, z 값은 출력값이 아님)

입력: 1010010

출력: 1010 (z=000)

입력: 1001110

출력: 1001 (z=000)

입력: 0011000

출력: 0011 (z=100)

입력: 0100001

출력: 0110 (z=111)

5) 2-layer 인코더 (입력 16-bit, 출력 42-bit)

입력: 1111101100101101

출력: 1111111 1011001 0010111 1101000 0100110 0000000

입력: 0000101100110010

출력: 0000000 1011001 0011100 0010111 0000000 1001110

입력: 1011001010100111

출력: 1011001 0010111 1010010 0111010 0001011 1001110

입력: 0111010110100001

출력: 0111010 0101101 1010010 0001011 1000101 1101000

6) 2-layer 디코더 (입력 42-bit, 출력 16-bit/ERROR!!)

입력: 0000000 1011001 0011100 0010111 0000000 1001110

출력: 0000101100110010

입력: 1011001 0010111 1010110 0111010 0001011 1101110

출력: 1011001010100111

입력: 1011001 0010111 1000110 0111010 0001011 1100100

출력: ERROR!!

입력: 0110010 0101101 1001010 0110011 1000101 1101000

출력: ERROR!!