

Modular Cell을 이용한 RS 디코더의 집적회로 설계

임충빈, 이광연, 이문기, 김용석, 홍연수, 송동익, 김영용
* 연세대학교 전자공학과 ** 삼성전자 종합연구소

VLSI Design of RS Decoder by Modular Cells

In Chung-Bin, Lee Kwang-Youb, Lee Moon-Key, Kim Yong-Serk,
Hong Heun-Serk, Song Dong-Il, Kim Young-Woong.

*Dept. of Elec., Eng., Yonsei Univ. ** R&D Center Sam Sung Electronics Co.

ABSTRACT

In this paper, Modular cell approach was applied to custom IC design of RS decoder. For the design of RS decoder by modular cells, 3 basic cells and one extra circuit are designed. these are, SYN cell for syndrome calculation, AL cell for error locator polynomial calculation, and REM cell for remaining error transform calculation. RS decoder design by these basic cells is very simple and regular, and naturally suitable for VLSI RS decoder design.

설계가 modular한 셀을 통해 이루어질 수 있도록 하는 설계방식을 제시하였다. 이를 위해 RS 디코더의 내부블록 기능을 따라 5개의 function block으로 나누고, 각 function block을 구성할 기본 단위 셀들을 설계하였다. 이때 RS 디코더의 전체 시스템은 GF(2^m) 상에서의 (n, k) RS 디코더를 가정하였다.

설계된 셀들은 Logic simulation을 통해 그 회로의 설계 및 검증을 수행하였으며 simulation시 회로의 delay 특성은 circuit simulation에 의한 CMOS standard cell의 data를 이용하였다.

2. RS 코드의 정의 및 디코딩 방식

RS 코드는 non-binary 코드로 하나의 심볼을 단위 구성 원소로 한다. 일반적으로 GF(2^m)을 2^m개의 element에 의한 finite field라고 할 때 GF(2^m) 상의 (n, k) RS 코드는 n개의 m비트 심볼로 1개의 codeword가 구성됨을 나타낸다.

한편, k는 1개의 codeword 내에 포함된 원래의 information symbol 수를 의미하며, 한 codeword는 k개의 원 information symbol에 n-k=2t개의 parity symbol이 합쳐져서 이루어진다. 이때 t는 정정의 가능한 심볼의 갯수를 나타낸다.

RS 코드의 디코딩은 보통 5개의 단계로 나누어진다[2].

step 1. 수신된 codeword를 이용하여 syndrome을 계산한다.

S_k = sum_{j=0}^{n-k} r_j alpha^{jk} 1 <= k <= 2t (2-1)

r_i: received codeword

r_k = S_k for 1 <= k <= 2t

step 2. step1에서 얻은 syndrome을 이

1. 서론

RS 코드는 Error-correcting 코드의 일종으로, Linear cyclic 코드인 BCH 코드의 한 부류이다.

RS 코드는 non-binary symbol에 대한 encoding 및 decoding을 다루며, codeword 간의 minimum distance가 가장 큰 에러 정정 코드로 알려져 있다.

RS 코드는 symbol 단위로 에러 정정을 하므로, burst error correction이 효율적으로 이루어질 수 있다는 장점을 갖고며, 우주 통신, 컴퓨터, 가전기기(CD player) 등에 폭넓게 사용되고 있다[1]. 그러나 설계 RS 코드의 인코더와 디코더 설계는 한 심볼을 이루는 비트수가 증가함에 따라, 또 높은 에러 정정 능력의 필요성에 따라 복잡도가 증가한다.

이러한 복잡도는 VLSI 또는 LSI에 의한 encoder 및 decoder의 설계를 불가피하게 한다.

본 논문에서는 이러한 복잡한 RS 디코더의

이용하여, 적절한 알고리즘을 통해 에러위치 다항식 $\Delta(x)$ 를 계산한다.

$$\Delta(x) = (x+\beta_1)(x+\beta_2) \dots (x+\beta_s) \quad (2-2)$$

$$= x^s + \delta_1 x^{s-1} + \dots + \delta_s$$

$\beta_1, \beta_2, \beta_3, \dots$: error location number.

step 3. step2 에서 얻은 에러위치 다항식을 이용하여 syndrome 계산식 이미 계산된 E_k ($1 \leq k \leq 2t$) 값을 제외한 나머지 E_k 값 ($2t+1 \leq k \leq n$) 을 계산한다. 이 관계는 다음과 같다.

$$E_{2t+j} + \sum_{k=1}^t \delta_k E_{2t+k} = 0 \quad \text{for } j \geq 1 \quad (2-3)$$

$$E_{2t+j} = E_{2t+j-n} \quad \text{for } 2t+j \geq n$$

step 4. step1~step3 를 통하여 얻은 E_k 로 부터 에러 값을 계산한다. 이 과정은 다음과 같이 표현한다.

$$e_k = \sum_{j=0}^{n-k} E_k \alpha^{jk} \quad (0 \leq k \leq n-1) \quad (2-4)$$

step 5. 앞의 4 단계를 거쳐 계산된 에러값을 buffer 에 저장되었던 원수신 codeword 로 부터 배서 에러를 지정한다.

이 다섯단계의 RS 코드 디코딩 과정을 block도 로 나타내면 그림 1과 같다.

3. RS 디코더의 각 function block 을 구성할 셀의 기능에 대한 고찰

(1) syndrom 계산을 위한 셀(SYN 셀)의 기능.

syndrom 계산을 위한 셀의 기능은 식(2-1)을 다음과 같이 분해해봄으로 정의 내릴수 있다.

$$S_k = (\dots ((r_{n-k} \alpha^k + r_{n-k-1}) \alpha^k + r_{n-k-2}) \alpha^k + \dots) \alpha^k + r_0) \alpha^k \quad (3-1)$$

식(3-1)을 통해서 볼때, syndrom S_k 의 계산은 $B \leftarrow A + B \alpha^k$ 와 같은 기능을 수행하는 연산 회로를 통해 계산가능함을 볼수있다. 따라서 $GF(2^8)$ 상에서의 (32,28) RS 디코더의 경우, 전체 syndrom 계산 block 은 $B \leftarrow A + B \alpha^k$ 기능의 회로를 k 마다 하나씩 마련하여 줌으로써 설계가능하며, 그 모양은 그림 2 와 같다. (n,k) RS 디코더의 경우 syndrom 의 계산은 $2t$ 개의 SYN 셀을 통해 가능하다.

(2) 에러위치 다항식 계산 block 을 위한 셀의 기능.

에러 위치 다항식 계산 block 은 RS 코드 의 디 코딩시 가장 핵심이 되는 부분으로써, 계산된 syndrom 을 이용하여 식(2-2)와 같은 에러위치 다항식을 계산한다.

본 논문에서는 Modified Euclid 알고리즘 [3]을

이용하여 에러위치 다항식의 계산 block 을 구성할 단위 셀을 설계하였다. Modified Euclid 알고리즘은 Euclid 알고리즘의 수정형으로 Modular 한 특성을 갖고 있으며, 따라서, Modular 한 셀을 사용하여 에러위치 다항식의 계산을 가능케 한다.

Modified Euclid 알고리즘을 이용하여 에러 위치 다항식을 계산할때 하나의 셀은 그림3과 같은 기능을 수행하며, 그림상의 구 변수에 대한 초기값 및 조건식은 다음과 같다.

$$R(x) = x^{2t}$$

$$Q(x) = \sum_{k=1}^t \delta_k x^{2t-k}$$

$$\lambda_0(x) = 0 \quad (3-2)$$

$$\mu_0(x) = 1$$

$$SAME = 1 \quad \text{if } d(Q) = a(R) \quad a(k): \text{degree of } k$$

$$COM = 1 \quad \text{if } d(Q) > d(R)$$

이 flow-chart 의 기능을 수행하는 AL 셀의 내부 block 도는 그림4와 같다. 한편, AL 셀 array 를 통하여 얻는 $\lambda(x)$ 는

$$\lambda(x) = \lambda_0 x^t + \lambda_1 x^{t-1} + \dots + \lambda_t \quad (3-3)$$

의 형태로 식(2-2)와 같은 $\Delta(x)$ 의 기본형으로 바꾸어 주어야 한다. 이를 위해 $\lambda(x)$ 를 λ_0 (0 이 아닌 첫 상수항)으로 나눈다. Hardware 로는 이러한 기능의 회로를 그림5와 같은 block 도로 표현해 줄수 있는데, 이 중 inverse multiplier 는 Massey

Omura inverse multiplier 를 사용하였다.

AL 셀을 사용하는 경우 에러위치 다항식의 계산은

$2t$ 개의 셀과 $\lambda(x) - \Delta(x)$ 회로에 의해 가능하다.

(3) Remaining error transform E_k 계산을

위한 셀(REM 셀)의 기능

2번째 단계까지의 의해 $\Delta(x)$ 가 계산되면, 이를 이용하여, 나머지 E_k 성분 ($2t+1 \leq k \leq n$) 을 계산한다. E_k 는 에러위치에 에러값을 곱해준 것의 의미를 갖는다. 나머지 E_k 계산부분은 식(2-3)의 전계를 통하여 그 회로를 얻어낼 수 있으며 $t=2$ 인 경우 회로의 구성은 그림6과 같다. 일반적인 경우 E_k ($2t+1 \leq k \leq n$)의 계산은 REM 셀 t 개를 통해 가능하다.

(4) 에러값의 계산

입단 E_k ($1 \leq k \leq n$) 가 계산되면, 이 값으로 부터 에러값을 계산해 낸다. 이때의 기본 수식은(2-4)로, 전개해보면 (3-4)식과 같다.

$$e_k = (E_n + \alpha^{(n-k)} ((E_n \alpha^k + E_{n-1}) \alpha^k + E_{n-2}) \alpha^k + \dots + E_{n-k}) \alpha^k \quad (3-4)$$

식(3-4)의 계산에 이용되는 셀은 syndrom 계산을 위해 설계된 셀에 약간의 수정을 가하여 설계가능하며 (n,k) RS 디코더의 경우 에러값의 계산은 n 개의

이러한 셀에 의해 가능하다.

(5) error 의 정정

계산된 에러값은 buffer 에 저장되어 있던 원 codeword 의 에러 정정이 사용되며 이는 2-입력 XOR 회로에 의해 가능하다.

4. 각 function block 을 구성할 셀의 집적회로 설계 및 simulation

3절에서 언급된 각 function block 을 구성할 3개의 기본적인 셀, 즉 syndrome 계산을 위한 SYN 셀, 이력위치 다항식의 계산을 위한 AL 셀 및 ER 계산을 위한 REM 셀과 $\lambda(x) \rightarrow \delta(x)$ 회로를 설계하였다. 이 셀들의 설계는 logic simulation 을 사용하여 검증하였으며, simulation 시 사용한 동작 주파수는 10 MHz 였다. 로직 시뮬레이션에 사용된 각 기본 기어블들의 delay 는, 미리 설계되어있던 cmos standard cell library 의 자료를 이용하였다. 그림 7~10 은 설계된 회로의 내부도 및 타이밍 다이어그램이다.

5. 결론

본 논문에서는 복잡한 RS 디코더의 설계를 Modular 한 셀을 이용하여 설계하는 방식에 대하여 논하였다. 이 방식에 의해 복잡한 RS 디코더의 설계는 규칙적인 셀의 array 를 통해 설계 가능하며, VLSI 및 LSI 수준의 RS 디코더 설계에 매우 편리한 설계양식으로 사용될 수 있을 것이다.

6. 참고문헌

1. I. Matull, "ICs for compact Disc Decoders," Electronix and Components and Applications. Vol.4, No.3, May 1983.
2. Kuang Yung Liu, "Architecture for VLSI design of Reed-Solomon Decoders," IEEE Trans. comput, Vol. C-33 No.2, eb.1984
3. Howard M. Sho, T.K. Troung, "A VLSI Design of a pipeline Reed-Solomon Decoder," IEEE Trans. on Comput. Vol C-34 No.5, May 1985
4. 이 문기, 신 경옥, 임 충빈, 유 종근, 최 병순, "Reed-Solomon Encoder 와 Decoder 설계". 다목적 공동 설계 개발에 관한 최종 연구 보고서
5. Elwyn R. Berlekamp, Algebraic Coding Theory, New York, McGraw-Hill. 1968

6. SHL Lin Daniel J. Costello Jr., Error Correcting Code, Fundamentals and Applications. Prentice-Hall 1983

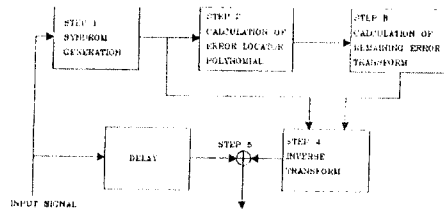


그림 1. RS 디코더의 전체 블록 다이어그램

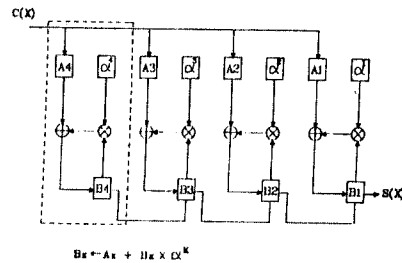


그림 2. GF(2) 상에서 (32,28) RS 디코더의 syndrome 계산 블록도

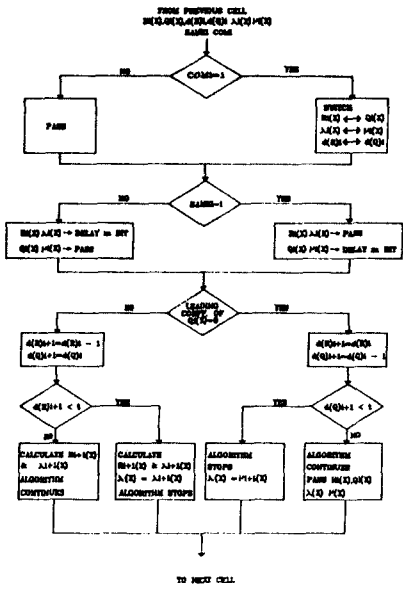


그림3. AL Cell 의 기능 흐름도

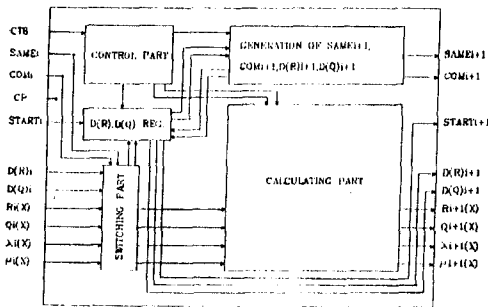


그림4. AL Cell 의 내부 블록도

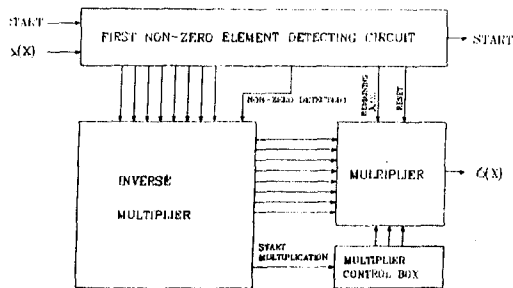
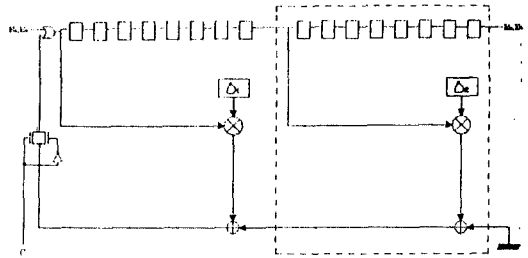
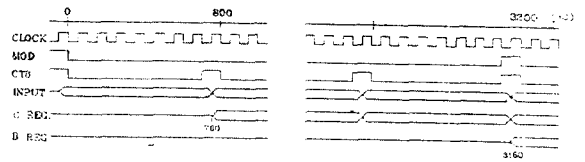
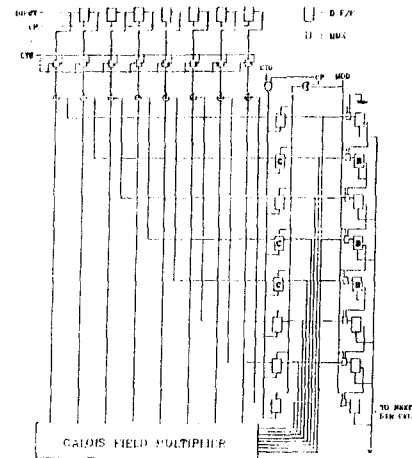


그림5. $\lambda(x) \rightarrow \delta(x)$ 회로의 블록도



C = 0 ON AND BEFORE E₀-ENTER
C = 1 AFTER E₀-ENTER

그림6. n=2 인 경우 2K 계산회로의 구조



TIMING DIAGRAM OF SYNDROME CALCULATING CELL

그림7. SYN Cell 의 내부 회로도 및 타이밍 다이어그램

