論文 98-23-6-19

수정된 유클리드 알고리듬을 적용한 리드솔로몬 부호기 및 복호기의 설계 및 합성

正會員 이 상 설*, 송 문 규**

Design and Synthesis of Reed-Solomon Encoder and Decoder Using Modified Euclid's Algorithm

Sang-Seol Lee*, Moon Kyou Song** *Regular Members*

※본 연구는 96년도 교육부 반도체분야 학술연구조성비(ISRC96-E-2018)에 의하여 연구되었음.

요 약

본 논문에서는 연집에러에 대한 대처방안으로 효과적인 RS(Reed-Solomon) 부호를 이용한 FEC(forward error correcting) 기법에 대한 연구가 이루어졌다. RS 부호화기 및 복호화기의 ASIC 구현을 위한 회로를 수정된 유클리드 알고리듬을 사용하여 설계 및 제안하였다. 제안된 회로의 동작을 흉내내는 방법으로 C 프로그램을 작성하여, 여러 가지의 에러 및 삭제 오류가 발생한 통신 선로를 가장하여 동작을 확인하였다. 이를 바탕으로 RS 부호화기 및 복호화기의 단일칩 구현을 위한 회로를 VHDL을 사용하여 시스톨릭 어레이 형태를 사용한 파이프라인 구조로 VLSI 설계하고 로직 시뮬레이션을 통해 검증하였으며 최종적으로 회로 합성에 성공하였다.

ABSTRACT

Reed-Solomon(RS) code which is especially effective against burst error is studied as a forward error correction technique in this paper. The circuits of RS encoder and decoder for ASIC implementation are designed and presented employing modified Euclid's algorithm. The functionalities of the designed circuits are verified through C programs which simulates the circuits over the various errors and erasures. The pipelined circuits using systolic arrays are designed for ASIC realization in VHDL, and verified through the logic simulations. Finally the circuit synthesis of RS encoder and decoder can be achieved.

I. 서 론

임의의 전송선로에서 발생하는 에러는 주로 진폭이 크고 연속시간이 짧은 임펄스성 잡음에 의하여 발생된다. 하지만 영상 서비스와 같은 고속 전송에서는 짧은 잡음일지라도 다수의 비트에 영향을 미치게 되어 집중적 형태의 연집 에러를 발생케 한다. 자연적

* 원광대학교 전기공학과
** 원광대학교 제어계측공학과
論文番號:97393-1029
接受日字:1997年 10月 29日

으로 발생하는 번개나 회로망의 열에 의해 발생하는 잡음도 여러 개의 전송 비트에서 발생하게 되는데 이러한 것이 연집 에러의 대표적인 예가 된다. 이에 대한 대처방안으로서 RS 부호가 가장 효과적인 것으로 알려져 있다[1].

자기 테이프/디스크 드라이브, 위성통신/VSAT, HDTV/CATV, 고성능 모뎀, LAN/WAN 등 다양한 분야에서 RS 부호가 사용되고 있으며, 디지틀 오디오 디스크에서 에러 정정을 위해서 상호인터리빙된 RS 부호의 단축쌍이 사용되고 있다. 특히 B-ISDN과 초고속 통신망은 PCS 무선망과 연동을 통한 확장 기로에 있으며, 무선의 경우에는 특히 다경로(multipath), 에코우(echo) 혹은 낙뢰와 같은 순간적인 연집 에러 등에 대한 잡음 대책이 더욱 절실히 요구되고 있다[2].

한편 VLSI 집적도의 가속화로 모든 부품들은 초소형화 추세로 가고 있다. FEC에 대한 연구 결과를 VLSI화 함으로써 FEC에 대한 대외 경쟁력은 물론 향후 유사한 FEC의 개발에 신속한 대응력을 갖추게 된다.

본 연구는 비트 에러 및 셀 손실(삭제)에 대한 정정을 위해 Reed-Solomon 부호 기법을 적용한 것으로 128 심볼(octets)의 블록에서 최대 2 개의 심볼(octet) 에러나 최대 4 개의 삭제(erasure)를 정정할 수 있는 (128, 124) RS 부호를 설계하고 VHDL로 구현하여 논리 시뮬레이션을 통해 회로를 검증하고, 회로를 합성하고자 한다.

II. RS 부호 및 복호기의 알고리듬과 설계

본 논문에서는 GF(256) 상에서 최대 2개의 에러 심볼 또는 최대 4 개의 삭제 심볼을 복원할 수 있는 (128, 124) RS 조직 부호화기를 설계하였고 이는 GF(256) 상에서 구현되며, 생성 다항식 $g(x)$는 다음과 같다.

$$g(x)=\prod_{i=0}^{3}(x-\alpha^{i+k}) \tag{1}$$

여기서 α는 이진 원시 다항식 $x^8+x^7+x^2+1$의 근이고, k는 생성다항식의 기저 지수로서 120이다. RS 부호화기에서는 상위 계층으로부터 입력된 124 옥텟에 4 옥텟의 검사심볼이 붙여진다고 한다[3].

초기에 수학적 견지의 호기심에 만족해야 했던 RS 부호가 오늘날 가장 강력하고 인기있는 에러정정 시스템의 하나로 전환될 수 있었던 것은 이 부호에 대한 효율적인 알고리듬의 발견에서 비롯된다. Reed-Solomon 부호의 복호화 알고리듬에는 Peterson-Grenstein-Zierler 알고리듬, Berlekamp-Massey 알고리듬, Euclid 알고리듬, 수정된 Euclid 알고리듬 등이 있다[2]. 본 연구에서는 가장 효율적인 수정 Euclid 알고리듬을 적용하여 회로를 설계하였다. 각 설계 부분에 대한 설명은 다음과 같다.

1. 부호기의 설계

본 논문에서 사용한 생성다항식은 다음과 같다. 여기서 밑(base)은 120으로 선택하여

$$\begin{aligned} g(x) &= (x+\alpha^{120})(x+\alpha^{121})(x+\alpha^{122})(x+\alpha^{123}) \\ &= x^4+\alpha^{162}x^3+\alpha^{35}x^2+\alpha^{150}x+\alpha^{231} \end{aligned} \tag{2}$$

이 된다. 메시지 다항식이

$$\begin{aligned} m(x) = \; & m_{123}x^{123}+m_{122}x^{122}+\ldots \\ & +m_2x^2+m_1x+m_0 \end{aligned} \tag{3}$$

일 때, 생성된 코드의 다항식은

$$\begin{aligned} c(x) &= c_{127}x^{127}+c_{126}x^{126}+\ldots+c_2x^2+c_1x+c_0 \\ &= x^4m(x)+d(x) \end{aligned} \tag{4}$$

이 된다. 여기서 검사심볼은

$$d(x)=x^4m(x) \mod g(x) \tag{5}$$

이다.

그림 1에서 클럭 신호가 124 개 동안 i=0에서 123까지는 $x^4m(x)$를 g(x)로 나누는 회로를 구성하고 있다. 그 몫은 최우측의 덧셈기의 출력으로 생성되나 스위치가 최종출력과 연결되어 있지 않으므로 이용되지는 않는다. 반면 레지스터 d_3, d_2, d_1, d_0에는 나

누어진 나머지 다항식 $d(x)=d_3x^3+d_2x^2+d_1x+d_0$의 계수들이 남게 된다. 최초 124개의 신호 동안에 최종 출력 단자에는 단순히 메시지신호 $m(x)$의 계수에 해당되는 심볼이 코드신호로 생성된다. 124 개의 클락 신호가 지나고 나면 입력 신호는 더이상 들어오지 못하도록 스위치가 바뀌어지고 $x^4m(x)$를 $g(x)$로 나누는 회로도 정지되어 더이상 동작하지 않는다. 그리고 나머지 다항식 $d(x)$의 계수에 해당되는 심벌이 출력되어 마지막 4개의 심벌이 합해져서 128개의 코드 단어를 형성시킨다.

부호기의 회로는 복호기의 설계와 비교하여 보면 상대적으로 대단히 적은 크기로 구현될 수 있다[4]. 4개의 GF(256) 덧셈기, 4개의 GF(256) 단일 입력 곱셈기[5]와 3개의 스위치가 주된 요소들이다.

2. 복호기의 설계

복호기의 전반적인 블록도는 그림 2와 같다. 입력

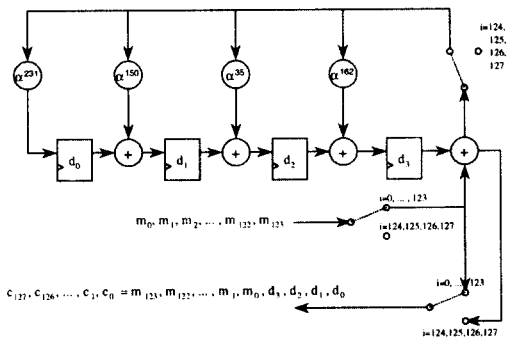

그림 1. RS 부호기의 구성도
Fig. 1 Block diagram of the RS encoder

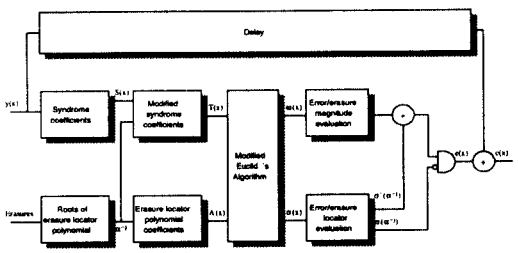

그림 2. RS 복호기의 구성도
Fig. 2 Block diagram of the RS decoder

으로는 수신단어 $r(x)$와 이에 대응하여 안쪽 복호기(inner decoder)에서 소실 심벌에 대한 정보가 같이 전달되어지게 된다. 수신된 단어는 파이프라인 형태의 구조[6][7]를 갖고 정해진 절차에 의거하여 동작한다. 출력으로는 정정된 코드단어인 c(x)를 갖게 된다. 각 부분 블록의 회로 및 동작은 다음과 같다.

① 신드롬의 계산

신드롬의 계산은 수신단어에 해당되는 다항식에 α^{120}, α^{121}, α^{122}, α^{123}을 대입하여 각각

$$\begin{aligned}
S_{120}&=r(\alpha^{120})=\sum_{i=0}^{127}r_ix^i\Big|_{x=\alpha^{120}}=\sum_{i=0}^{127}r_i(\alpha^{120})^i\\
S_{121}&=r(\alpha^{121})=\sum_{i=0}^{127}r_ix^i\Big|_{x=\alpha^{121}}=\sum_{i=0}^{127}r_i(\alpha^{121})^i\\
S_{122}&=r(\alpha^{122})=\sum_{i=0}^{127}r_ix^i\Big|_{x=\alpha^{122}}=\sum_{i=0}^{127}r_i(\alpha^{122})^i\\
S_{123}&=r(\alpha^{123})=\sum_{i=0}^{127}r_ix^i\Big|_{x=\alpha^{123}}=\sum_{i=0}^{127}r_i(\alpha^{123})^i
\end{aligned}\tag{6}$$

을 얻는다. 이를 구현하기 위해서

$$\begin{aligned}
r(x)=&\ r_{127}x^{127}+r_{126}x^{126}+r_{125}x^{125}+\ldots\\
&+r_2x^2+r_1x+r_0\\
=&\ ((\ldots(((r_{127}x+r_{126})x+r_{125})x+r_{124})x\\
&+\ldots+r_2)\cdot x+r_1)x+r_0
\end{aligned}\tag{7}$$

로 표현될 수 있다. S_{120}인 $r(\alpha^{120})$을 얻기 위해서는 $S_{120_t}=S_{120_{t-1}}\alpha^{120}+r_{127-t}$와 같이 재귀적으로 표현될 수 있으며 신드롬을 계산하는 회로의 설계가 가능하다(그림 3).

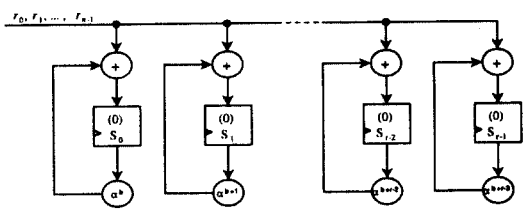

그림 3. 신드롬을 계산하는 시스톨릭 어레이
Fig. 3 A systolic array to compute syndrome

② 삭제위치자다항식의 근 생성 구조

삭제위치자 다항식을 생성하기 위해서는 그 근에 대한 정보로부터 시작된다. 그 근은 안쪽 복호기의 결과로 얻어지는 삭제 정보로부터 그 위치의 형태로 알아 낼 수가 있다. 이 위치 정보는 수신 심볼과 함께 그 진위의 값이 전기적 신호로 입력됨으로써 복호기에 알려진다. 만일 j번째의 수신 심볼이 삭제되었다 가정하면 j번째의 수신 심볼이 입력됨과 동시에 삭제을 알리는 신호가 동시에 들어온다. 이 때의 삭제위치자의 근으로써 α^{-j}를 취하게 된다. 이를 이용하여 회로를 구성할 수 있으며(그림 4), (128, 124) 코드에서 삭제위치자다항식에 대한 근의 개수는 최대 4개이다.

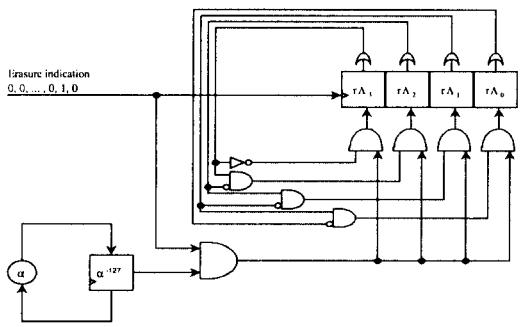

그림 4. 삭제위치자다항식의 근을 구하는 회로도
Fig. 4 Circuit diagram to find the roots of the erasure locator polynomial

③ 삭제위치자다항식 생성 구조

앞에서 추출한 삭제위치자 다항식의 근으로부터 이제 삭제위치자다항식을 추출한다. 삭제위치자 다항식의 추출은 그 계수의 표현으로 대변된다. 그 계수들은 레지스터열 Λ에 저장된다. 근이 최대 4개일 수 있으므로 삭제위치자 다항식의 계수의 수는 최고차항의 계수를 1로 하여 총 5개가 된다. 따라서 레지스터열 Λ의 갯수는 5개가 된다. 삭제위치자 다항식의 계수를 결정하기 위하여 다른 회로가 이용된다. 입력으로 근이 $r\Lambda_3, r\Lambda_2, r\Lambda_1, r\Lambda_0$의 순으로 들어온다. 다항식 $\Lambda_{i-1}(x)$가 형성 중에 근 $r\Lambda_i$이 들어오면 이는 $x+r\Lambda_i$이 곱하여지는 경우로서

$$(x+r\Lambda_i)\Lambda_{i-1}(x)=x\cdot\Lambda_{i-1}(x)+r\Lambda_i\cdot\Lambda_{i-1}(x) \quad (8)$$

이 된다. 이 식의 우변은 $\Lambda_{i-1}(x)$를 좌로 천이시킴과 동시에 근 $r\Lambda_i$와 $\Lambda_{i-1}(x)$가 곱하여 더해지는 형상이다. 이는 그림 5와 같은 형상을 지니며 다만 레지스터열의 Λ_0의 초기 계수 α_0를 갖는다.

④ 수정 신드롬 다항식 생성 구조

수정신드롬다항식은

$$\begin{aligned} T(x) &= S(x)\Lambda(x) \bmod x^4 \\ &= S(x)\prod_{i=3-v+1}^{3}(x+r\Lambda_i) \end{aligned} \quad (9)$$

로 주어지고 이 때 v는 근의 개수이다. 한 개 항만의 곱으로 나타내면 $(x+r\Lambda_i)S(x)$이고 이는 $xS(x)+r\Lambda_i S(x)$이다. 이는 앞의 삭제위치자다항식의 계수를 구하는 방법과 동일한 방법을 사용할 수 있으며 (그림 5), 이 때 초기값으로 다만 $S(x)$의 계수를 이용하기만 하면 된다.

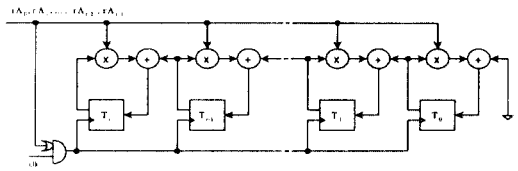

그림 5. 다항식 전개 계산을 위한 시스톨릭 어레이
Fig. 5 A systolic array for polynomial expansion computation

⑤ 수정 Euclid 알고리듬 구현 구조

본 논문에서 채택한 방법은 수정된 Euclid 알고리듬이다. 이 방법에 의하여 회로를 구현하게 되면 결과로서 에러/삭제평가자 다항식(error/erasure evaluator polynomial)과 에러/삭제위치자 다항식(error/erasure locator polynomial)을 손쉽게 얻어낼 수 있다. 이 방법은 입력으로서 수정신드롬 다항식과 삭제위치자 다항식을 취한다. 수정 Euclid 알고리듬의 수행을 위하여 본 논문에서 제안한 회로의 블록도는 그림 6과 같다.

이 부분에서 필요한 레지스터열들은 $R(x)$ 레지스터열, $Q(x)$ 레지스터열, start 레지스터, $\lambda(x)$ 레지스터열, 그리고 $\mu(x)$ 레지스터열 들을 갖는다. 만일 $Q(x)$

의 차수가 $R(x)$의 차수보다 커지게 되면 상호변환 스위치를 이용하여 입력단자를 바꿈으로써 항상 $R(x)$의 차수가 $Q(x)$의 차수보다 같거나 크게 유지한다.

이 회로에 의하여 설계된 수정 Euclid 알고리듬 방법은 동일한 셀을 재귀적으로 연산되어 수행된다. 초기에 $R(x)$의 차수는 4차 $Q(x)$의 차수는 3차로 하고

$$R(x)=x^4, \qquad Q(x)=T(x),$$
$$\lambda(x)=0, \qquad \mu(x)=\Lambda(x) \tag{10}$$

와 같이 초기화시킨다.

$$R_i(x)= [\sigma_{i-1}b_{i-1}R_{i-1}(x)+\overline{\sigma_{i-1}}a_{i-1}Q_{i-1}(x)] - x^{|l_i-1|}[\sigma_{i-1}a_{i-1}Q_{i-1}(x)+\overline{\sigma_{i-1}}b_{i-1}R_{i-1}(x)]$$

$$\lambda_i(x)= [\sigma_{i-1}b_{i-1}\lambda_{i-1}(x)+\overline{\sigma_{i-1}}a_{i-1}\mu_{i-1}(x)] - x^{|l_i-1|}[\sigma_{i-1}a_{i-1}\mu_{i-1}(x)+\overline{\sigma_{i-1}}b_{i-1}\lambda_{i-1}(x)] \tag{11}$$

$$Q_i(x)=\sigma_{i-1}Q_{i-1}(x)+\overline{\sigma_{i-1}}R_{i-1}(x)$$
$$\mu_i(x)=\sigma_{i-1}\mu_{i-1}(x)+\overline{\sigma_{i-1}}\lambda_{i-1}(x) \tag{12}$$

여기서, a_{i-1}, b_{i-1}은 각각 $R_{i-1}(x)$와 $Q_{i-1}(x)$의 최상위 계수이고,

$$l_{i-1}=\deg(R_{i-1}(x))-\deg(Q_{i-1}(x))$$

라 할 때

$$\sigma_{i-1}=1 \qquad \text{if } l_{i-1}\geq 0$$
$$\sigma_{i-1}=0 \qquad \text{if } l_{i-1}<0$$

$R(x)$와 $Q(x)$는 한 개의 차수가 차이가 나거나 혹은 동일한 차수가 되도록 유지한다. $R(x)$에 $Q(x)$의 최고차항의 상수를 곱하여 $Q(x)$에 $R(x)$의 최고차항을 곱하여 덧셈을 하면 $R(x)$의 최고차항이 소거가 된다. 이 때 같은 $R(x)$의 최고차항을 $\mu(x)$에 곱하여 $\lambda(x)$에 더하면 그 효과가 $\lambda(x)$에 반영되어 에러의 위치를 찾는데 도움이 된다. 이 때 만일 $R(x)$와 $Q(x)$의 차수가 동일하건 그렇지 않건 $R(x)$의 차수는 하나씩 감소되어지고 $Q(x)$의 차수는 동일하게 유지하게 된다.

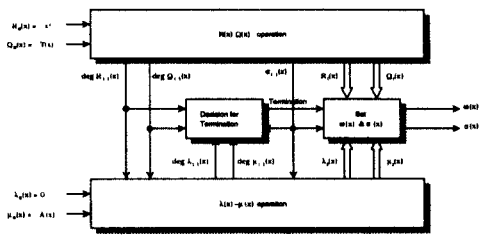

그림 6. 수정 유클리드 알고리듬 연산 셀의 구조
Fig. 6 The architecture of the basic cell in modified Euclid's algorithm

⑥ 에러/삭제평가자다항식 대입값 추출 구조

앞의 수정 Euclid 방법에 의한 결과로 에러/삭제평가자다항식 $\omega(x)$를 얻는다. 에러/삭제 심볼의 위치에 해당되는 값을 알아서 Forney의 알고리듬에 적용을 시켜야만 에러가 발생된 코드워드 $e(x)$를 추출해 낼 수가 있다. 이 과정에서 $\omega(x)|_{x=\alpha^{-j}}$를 대입하여 j번째 에러/삭제 심볼에 대한 정정된 값을 추출해 내는데 이용된다[8]. 이 때

$$\omega(x)=\omega_0+\omega_1 x+\omega_2 x^2+\omega_3 x^3 \tag{13}$$

$$\omega(\alpha^{-j}) = \omega_0+\omega_1(\alpha^{-j})+\omega_2(\alpha^{-j})^2 + \omega_3(\alpha^{-j})^3 = \omega_0+\omega_1(\alpha^{-1})^j+\omega_2(\alpha^{-2})^j + \omega_3(\alpha^{-3})^j \quad ,j=0,\ \ldots,\ 127 \tag{14}$$

$$\omega(\alpha^{-(j-1)}) = \omega_0+\omega_1(\alpha^{-1})^{(j-1)} + \omega_2(\alpha^{-2})^{(j-1)}+\omega_3(\alpha^{-3})^{(j-1)} = \omega_0+\omega_1(\alpha^{-1})^j\alpha^1 + \omega_2(\alpha^{-2})^j\alpha^{2+}\omega_3(\alpha^{-3})^j\alpha^3 \tag{15}$$

이 되므로 계수의 초기값을 각각 ω_0, $\omega_1\alpha^{-1*127}$, $\omega_2\alpha^{-2*127}$, $\omega_3\alpha^{-3*127}$로 하고 이를 회로에 적용한

다(그림 7).

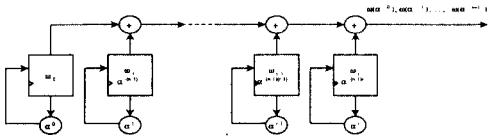

그림 7. 다항식 평가를 위한 시스톨릭 어레이
Fig. 7 A systolic array for polynomial evaluation

⑦ 에러/삭제위치자다항식 대입값 추출 구조

수정 Euclid 방법에 의한 결과로 아래의 에러/삭제위치자다항식을 얻는다.

$$\sigma(x) = \sigma_0 + \sigma_1 x + \sigma_2 x^2 + \sigma_3 x^3 + \sigma_4 x^4 \quad (16)$$

이 다항식에 대한 v개의 근을 $\alpha^{-j_0}, \alpha^{-j_1}, \ldots, \alpha^{-j_{v-1}}$로 할 때 $j_0, j_1, \ldots, j_{v-1}$의 위치에 에러나 삭제 심볼이 위치하게 된다. 이 근을 구하는 방법 역시 앞의 에러/삭제평가자의 값을 구하는 방법(그림 7)과 동일하다 다만 에러/삭제위치자다항식은 계수 5개를 갖는 최대 4차의 다항식이 된다.

⑧ 미분형 에러/삭제위치자다항식 대입값 추출 구조

미분형 에러/삭제위치자 다항식에 $x = \alpha^{-j}$, j=0, ..., 127에 대한 대입값을 추출하는 과정은 앞의 두 가지 방법(그림 7)과 유사하다. 단지 미분형이기 때문에 미분에 대한 대입값을 추출해야 한다. 미분형 에러/삭제위치자다항식은 아래와 같다.

$$\begin{aligned}\sigma'(x) &= \sigma_1 + 2\sigma_2 x + 3\sigma_3 x^2 + 4\sigma_4 x^3 \\ &= \sigma_1 + \sigma_3 x^2\end{aligned} \quad (17)$$

⑨ 에러 정정

수정될 에러/삭제 값은 에러/삭제평가자다항식을 미분형 에러/삭제위치자다항식과 함께 연산하여 구할 수 있다.

$$\sigma(\alpha^{-i}) \neq 0 \text{ 일 때 } e_i = 0 \quad (18)$$

$$\sigma(\alpha^{-i}) = 0 \text{ 일 때 } e_i = \alpha^{-i(120-1)} \frac{\omega(\alpha^{-i})}{\sigma'(\alpha^{-i})}$$

위와 같이 결정된 수정 값은 수신된 해당 심볼과 Galois Field 덧셈을 통하여 얻어진다.

전체적인 회로는 3단의 파이프라인 구조를 갖으며 표 1과 같은 데이터의 흐름을 갖는다. 첫 번째 단에서는 단어의 수신 종료와 동시에 신드롬 생성과 삭제 위치 근 생성이 완료된다. 둘째 단에서는 앞 단에서 넘겨 받은 신드롬으로 수정 신드롬의 생성과 삭제위치 근으로 다항식으로의 전개에 4클럭이 소요되고 수정 Euclid 알고리듬을 이용한 평가자 및 위치자 다항식을 생성하는 데 20클럭이 소요된다. 따라서 마지막 단에서는 수신 단어의 종료 후 25 클럭의 지연 이후부터 정정된 단어의 심볼들이 출력되기 시작한다.

표 1. 4단으로 구성된 파이프라인의 데이터 흐름
Table 1. Data flow of four stage pipeline

파이프라인 단	1	2	3
수신 단어	*i*번째	*i*+1 번째	*i*+2 번째
신드롬 생성	*i*번째 단어		
삭제 위치 근 생성			
수정 신드롬 생성		*i*번째 단어	
삭제위치다항식 생성			
수정Euclid이용 평가자/위치자 다항식 생성			
에러/삭제평가자 다항식 계산			*i*번째 단어
에러/삭제위치자 다항식 계산			
에러 정정			

표 2는 제안된 복호기와 다른 방법들[9]과의 비교를 나타내고 있다. 제안된 복호기의 입출력 속도는 30Mbyte/s 까지 가능하고 Galois Field 곱셈기는 조합회로로 구현되고 있으며, 합성결과 약 25,000게이트가 이용된다.

III. RS 부호기와 복호기 회로의 시뮬레이션 및 합성

위의 설계된 회로의 하드웨어 특성을 유지하도록

표 2. 다른 방법과의 비교
Table 2. Comparison with other methods

	제안된 방법	[9]의 제안 방법		Gold Star	Sharp
		형태1	형태2		
입출력 속도	30Mbyte/s	23Mbyte/s		5Mbyte/s	16Mbyte/s
융통성	×	×	○	×	○
구조	3단 파이프라인	3단 파이프라인	3단 파이프라인	순차적	4단 파이프라인
Galois Field Generator	조합회로	천이 레지스터 이용	마이크로 프로그램된 ROM	천이 레지스터 이용	마이크로 프로그램된 ROM
신드롬과 에러정정 수행	병렬	병렬	병렬	순차	순차

C 언어로 구현하였으며, 임의의 위치에서 최대 4개의 삭제와 최대 2개의 에러를 가지는 가능한 모든 조합의 경우에 대한 에러정정 기능을 확인하였다. 본 연구에서의 Reed-Solomon 부호기와 복호기에 대한 구현은 Mentor Graphics에서 제공하는 VHDL 시뮬레이터를 이용하여 검증하였다. 그림 8은 설계된 복호기의 에러정정 상황을 보이는 시뮬레이션 결과를 보인 것이다. 모두 "00"로 송신된 코드워드에 대하여 지연된 수신신호가 신호 'rbuf(0)'에 나타나고 있으며 신호 'berloc'가 '0' 일 때 삭제에 의하여 신호 'err'에 나타난 에러 값 "01", "02", "1A"가 보정되어 신호 'data'에 송신된 값 "00"으로 복원되고 있다.

그림 8. 에러와 소실의 정정을 보이는 시뮬레이션 파형
Fig. 8 Simulation waveforms showing the correction of the error and erasures

이와 같이 논리 시뮬레이션을 완성한 VHDL 프로그램을 Synopsis사의 회로 합성 툴을 이용하여 Reed-Solomon 부호기와 복호기에 대한 회로합성을 수행하였다. 그 결과 복호기의 합성된 게이트의 수는 약 25,000개이고 가능한 주파수는 약 30MHz 즉 30Mbyte/sec의 속도를 예측할 수 있다. 그림 9는 합성된 RS 부호기의 구성도를 보인 것이다.

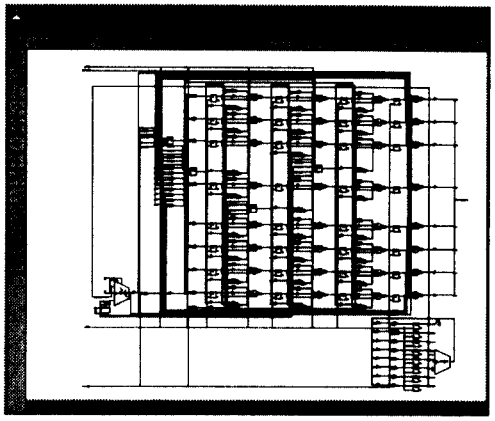

그림 9. Reed-Solomon 부호화기의 합성
Fig. 9 Synthesized circuit of the RS encoder

IV. 결 론

본 연구는 비트 에러 및 셀 손실(삭제)에 대한 정정을 위해 Reed-Solomon 부호 기법을 적용한 것으로 128 심볼(octets)의 블록에서 최대 2개의 심볼(octet) 에러나 최대 4개의 삭제(erasure)를 정정할 수 있는 (128, 124) RS 부호에 대한 부호기와 복호기를 설계하고 먼저 C 언어를 이용하여 그 기능을 검증함으로써 회로도의 설계를 완성하였다. VHDL을 사용하여 시스톨릭 어레이 형태를 사용한 파이프라인 구조로 회로를 설계 및 제안하고 논리 시뮬레이션을 통해 회로의 동작을 검증하였으며 최종적으로 회로 합성에 성공하였다.

본 연구에서 얻어진 Reed-Solomon 부호의 부호화 및 복호화기 설계 기술은 규격에 상관없이 모든 RS 부호의 부호화 및 복호화기의 설계에 활용될 수 있으며, 효율적인 에러 대처방안 개발에 대한 실제적 자료로 활용할 수 있다.

본 연구에서 개발된 C 프로그램과 VHDL 프로그램은 디지탈 통신 시스템에서 연집에러의 대책을 위한 RS 부호화기 및 복호화기에 대한 라이브러리로 활용할 수 있을 것이다. 또한 본 연구를 통해 Reed-Solomon 부호를 요구하는 모든 시스템의 부호화기 및 복호화기 설계 능력을 자체적으로 확보할 수 있었다.

참 고 문 헌

1. S. B. Wicker, *Error Control Systems for Digital Communication and Storage*, Englewood Cliffs, N.J.:Prentice-Hall, 1995.
2. Stephan B. Wicker and Vijay Bhargava, *Reed-Solomon Codes and Their Applications*, IEEE Press, 1994.
3. CCITT Recommendation I.363, "B-ISDN ATM Adaptation Layer(AAL) Specification," 1991
4. E. R. Berlekamp, "Bit-Serial Reed-Solomon Encoder," *IEEE Trans. on Information Theory*, Vol. IT-28, No. 6, pp. 869-874, Nov. 1982.
5. M. A. Hasan and V. K. Bhargava, "Bit-Serial Systolic Divider and Multiplier for GF(2^m)," *IEEE Trans. on Computers*, Vol. 41, pp. 972-980, Aug. 1992.
6. H. M. Shao, T. K. Truong, L. J. Deutsch, J. H. Yuen and I. S. Reed, "A VLSI Design of a Pipeline Reed-Solomon Decoder," *IEEE Trans. on Computers*, Vol. C-34, pp. 393-403, May 1985.
7. Howard M. Shao, and Irving S. Reed, "On the VLSI Design of a Pipeline Reed-Solomon Decoder Using Systolic Arrays," *IEEE Trans. on Computers*. Vol. 37, No. 10, Oct., 1988
8. C. C. Wang, T. K. Truong, H. M. Shao, L. J. Deutsch, J. K. Omura, and I. S. Reed, "VLSI Architecture for Computing Multiplications and Inverses in GF(2^m)," *IEEE Trans. on Computers*, Vol. C-34, pp. 709-717, Aug. 1985.
9. M. H. Lee, S. B. Choi, J. S. Chang, "A High Speed Reed-Solomon Decoder," *IEEE Trans. on Consumer Electronics*, Vol. 41, No. 4, pp. 1142-1149, Nov. 1995.

이 상 설(Sang-Seol Lee) 정회원

1984년 2월:고려대학교 전자공학과(공학사)

1989년 2월:한국과학기술원 전기 및 전자공학과(공학석사)

1994년 2월:한국과학기술원 전기 및 전자공학과(공학박사)

1994년 3월~현재:원광대학교 공과대학 전기공학과 조교수

※주관심분야:컴퓨터구조, 통신 및 응용 VLSI 설계, 테스팅, 영상처리 등

e-mail:slee@wonnms.wonkwang.ac.kr

송 문 규(Moon Kyou Song) 정회원

1988년 2월:고려대학교 전자공학과(공학사)

1990년 2월:고려대학교 대학원 전자공학과(공학석사)

1994년 2월:고려대학교 대학원 전자공학과(공학박사)

1994년 3월~현재:원광대학교 공과대학 제어계측공학과 조교수

1997년 10월~현재:한국전자통신연구원 초빙연구원

※주관심분야:스펙트럼확산 통신, 무선이동통신, 채널코딩, 방송기술 등

e-mail:mksong@wonnms.wonkwang.ac.kr