

# 가변 블록길이를 갖는 적응형 리드솔로몬 복호기의 설계

중신회원 송 문 규\*, 준회원 공 민 한\*\*

## Design of an Adaptive Reed-Solomon Decoder with Varying Block Length

Moon-Kyou Song\*, Min-Han Kong\*\*

요 약

본 논문에서는 수정 유클리드 알고리즘을 기반으로 임의의 메시지 길이  $k$  뿐 아니라 임의의 블록 길이  $n$ 를 갖는 RS 부호를 복호할 수 있는 적응형 RS 복호기를 설계한다. 설계된 복호기는 임의의 길이를 갖는 단축형 RS 부호의 복호 전에 영들을 추가하지 않아도 되므로 단축형 RS 부호에 특히 유리하다. 또한 이들 RS 부호의 오류정정 능력  $t$ 의 값을 매 부호어 블록마다 실시간으로 변화시킬 수 있으므로 응답 채널이 유용한 경우 채널의 시변 잡음 레벨에 적응적으로 오류 정정 능력을 변화시킬 수 있다. 제시된 복호기 구조는 수정 유클리드 알고리즘에 기반한 4단계 파이프라인 처리를 수행한다: (1) 신드롬 계산 (2) MEA 블록 (3) 에러크기 계산 (4) 복호기 실패 검사. 각 단계는 가변 길이의 RS 복호에 적합한 구조를 갖도록 설계된다. 수정 유클리드 알고리즘(MEA) 블록의 새로운 구조를 제시하고, 에러의 크기 계산을 위한 다항식 평가를 위해 역순 출력을 갖는 다항식 평가 회로를 채용한다. MEA 블록은 연산 셀들의 멀티플렉싱 기법과 배속의 전용 클럭 기법(overclocking)을 적용하여 간단한 하드웨어로써 처리 속도를 유지하도록 하였다. 최대 오류정정 능력이 10인  $GF(2^8)$  상의 적응형 RS 부호를 VHDL로 설계하고, FPGA에 성공적으로 합성하였다.

키워드: 채널코딩

### ABSTRACT

In this paper, we design a versatile RS decoder which can decode RS codes of any block length  $n$  as well as any message length  $k$ , based on a modified Euclid's algorithm (MEA). This unique feature is favorable for a shortened RS code of any block length since it eliminates the need to insert zeros before decoding a shortened RS code. Furthermore, the value of error correcting capability  $t$  can be changed in real time at every codeword block. Thus, when a return channel is available, the error correcting capability can be adaptively altered according to channel state. The decoder permits 4-step pipelined processing : (1) syndrome calculation (2) MEA block (3) error magnitude calculation (4) decoder failure check. Each step is designed to form a structure suitable for decoding a RS code with varying block length. A new architecture is proposed for a MEA block in step (2) and an architecture of outputting in reversed order is employed for a polynomial evaluation in step (3). To maintain the throughput rate with less circuitry, the MEA block uses not only a multiplexing and recursive technique but also an overclocking technique. The adaptive RS decoder over  $GF(2^8)$  with the maximal error correcting capability of 10 has been designed in VHDL, and successfully synthesized in a FPGA.

\* 원광대학교 전기전자및정보공학부 교수(mksong@wonkwang.ac.kr), \*\* 원광대학교 제어계측공학과 박사과정  
논문번호 : 020292-0705, 접수일자 : 2002년 7월 5일

\* 본 논문은 정보통신부의 정보통신기술기초연구지원사업으로 수행한 연구결과입니다.

## I. 서론

최근 정보의 양과 종류의 다양화로 부선을 통한 광대역 데이터의 고속 전송 요구가 높아지고 있다. 광대역 데이터의 고속 전송을 위해서는 유용한 대역폭을 효율적으로 활용하여 데이터 손실에 대처할 수 있는 효율적이고 적응성을 갖는 에러정정 기법이 요구된다. 이는 채널의 시변 잡음레벨과 요구되는 BER(bit error rate) 성능에 따라 최적의 부호를 선택하는 적응형 순방향 에러정정(adaptive forward error correction; AFEC) 기법에 의하여 실현될 수 있다<sup>[1-10]</sup>. AFEC는 Reed-Solomon (RS) 부호의 다양한 부호율( $k/n$ ;  $n$ 은 부호어의 길이,  $k$ 는 메시지 길이)과 에러정정 능력  $t$ 를 적용하여 달성할 수 있으며<sup>[1-7]</sup>, 적응형 RS 부호의 적용 예로서 IEEE 802.16<sup>[12]</sup>, MCNS/DOCSIS<sup>[13]</sup>, ADSL<sup>[11]</sup> 그리고 VDSL<sup>[15]</sup> 등이 있다. 에러정정 능력이 고정된 기존의 고정형 FEC 기법은 최악의 채널 환경에서 원하는 평균 성능을 얻기 위해 설계되었으나 많은 응용에서 채널은 전체 사용시간 중 단지 매우 작은 시간동안만 최악의 상태로 존재한다. 따라서 대역 효율성이 낮고, 다양한 BER 성능 요구를 충족시킬 수 없다<sup>[7][11]</sup>. 이에 반해 AFEC 기법은 대역 효율성과 전력 효율성 사이의 절충이 용이하고 유연성이 있으며, 다양한 전송 성능 요구에 대처할 수 있다<sup>[3-6][9]</sup>.

몇 개의 최근 논문에서 가변형(versatile) RS 복호기에 대하여 연구된 바 있다<sup>[16-20]</sup>. 이들 논문에서는 RS 복호기의 가변성을 위해 블록길이  $n$ 을 고정시킨 채 단지 메시지 길이  $k$ 만을 변화시켜 에러정정 능력  $t$ 의 변화를 허용하는 구조를 제시하였다. 그러나  $n$ 이 고정인 이들 구조에서는 단축형 RS의 복호를 위해 0을 부가하여 블록길이를  $2^n-1$ 로 해주어야 한다<sup>[12]</sup>.

본 논문에서는 매 블록마다 실시간으로 블록길이  $n$ 을 변화시킬 수 있는 적응형 RS 복호기의 구조를 설계하고 VHDL을 이용하여 구현하였다. 구현된 적응형 RS 복호기는 메시지 길이  $k$ 뿐만 아니라 블록 길이  $n$ 도 가변할 수 있도록 설계하여 문헌 [16][17][18]의 구조에서 단축형 RS 부호를 위해 0을 부가해야 하는 단점을 제거하였다. 또한 매 부호워드 블록마다 파라미터  $n$ ,  $k$  그리고  $t$ 를 실시간으로

가변할 수 있다. 그러므로 응답 채널(return channel)이 유용하면 이 복호기는 채널의 상태에 따라 에러정정 능력을 적응적으로 변화시킬 수 있으므로 ARQ(Automatic Repeat Request) 시스템과 병합적으로 사용할 수 있다<sup>[2]</sup>.

## II. 복호기의 설계

본 논문에서는 가장 일반적인  $GF(2^8)$  상에서 정의되는 RS 부호의 복호기를 설계한다.  $GF(2^8)$ 을 구성하기 위한 원시 다항식은  $p(x)=x^8+x^4+x^3+x^2+1$ 로 설정하였다. 블록 길이가  $n=255$ 인 RS 부호는 원시 RS 부호라 하고, 순회 부호이다. 그 외의 블록 길이를 갖는 경우는 단축형 RS 부호라 하며 일반적으로 순회 부호가 아니다. 기존에 설계된 가변형 RS 복호기의 경우 대부분  $n$ 값을 고정하고,  $k$ 값만을 가변하도록 설계되어 단축형 RS 부호의 복호를 위해 복호 전에 0을 첨부하여 부호의 길이를 정해진  $n$ 값으로 맞추어야 한다. 본 논문에서는  $k$ 값뿐만 아니라  $n$ 값을 가변으로 할 수 있는 적응형 RS 복호기의 구조를 제시한다. 이 구조를 사용하면 단축형 RS 부호의 복호를 위해 0을 첨부하는 과정이 필요하지 않게 된다. 또한 본 논문에서 설계한 복호기 구조는  $1 \leq n \leq 255$ 과  $0 \leq t \leq t_M$ 의 완전한 범위에서 파라미터 선택의 유연성을 제공할 수 있다. 본 논문에서는  $t_M=10$ 으로 설계하였으나, 설계된 부호기 및 복호기의 구조는 10이상의  $t_M$ 에 대하여 쉽게 확장할 수 있다.

본 논문에서는 복호를 위해 MEA(modified Euclid's algorithm)를 적용하였으며, 설계된 적응형 RS 복호기의 구성도는 그림 1에 보인다. 이 적응형 RS 복호기는 (1) 신드롬 계산, (2) MEA 블록, (3) 에러의 위치와 크기의 계산, (4) 복호기 실패 검증 등의 4단계 파이프라인 구조를 갖는다. 수신된 부호어가 Chien 탐색을 통해 정정된 후 복호기 실패 여부를 검사한다. 이 과정은 실제적으로 신드롬 계산과 동일하다. 정정후 계산된 신드롬이 0이 아니면, 복호기 실패가 선언된다. 복호 과정과 복호기 구조에 대해서는 이후의 절에서 상세히 다루기로 한다.

### 2.1. 신드롬 계산

신드롬은 다음과 식(1)로 주어진다.

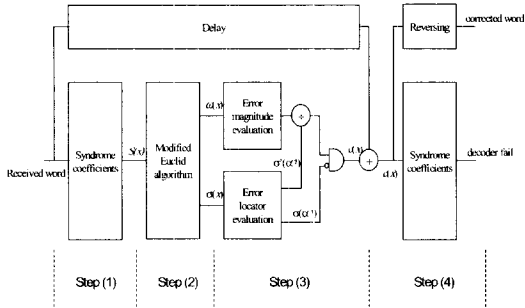


그림 1. 적응형 RS 복호기의 구성도  
Fig. 1 Block diagram of the adaptive RS decoder

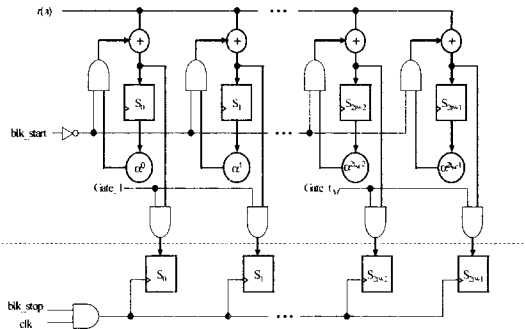


그림 2. 신드롬 계산 구조  
Fig. 2 An architecture of syndrome computation

$$S_j = y(\alpha^{b+i}) = \sum_{i=0}^{n-1} y_i \alpha^{(b+i)j}, \quad (1)$$

$$0 \leq j \leq n-k-1$$

이는 다음과 같이 재귀적으로 표현이 가능하다.

$$S_i = ((\dots ((r_{n-1} \alpha^i + r_{n-2}) \alpha^i + r_{n-3}) \alpha^i + \dots + r_2) \alpha^i + r_1) \alpha^i + r_0 \quad (2)$$

이 표현을 이용하여 신드롬을 계산하는 시스틀릭 어레이 구조는 그림 2에 보인다.

신드롬 계산 구조에서는 항상  $2t_M$ 개의 신드롬을 계산한다. 그러나 매 블록마다 변경될 수 있는 RS 복호기의 오류 정정 능력의 현재 설정값을  $t_0$ 라 할 때, RS 복호에 필요한 신드롬은  $2t_0$ 개이므로  $2t_M$ 의 출력 중에서  $2t_0$ 개 이상의 출력은 게이트 신호에 의하여 차단된다. 'blk\_start'는 수신 부호어의 시작을 나타내는 신호로서, 부호 블록의 입력이 시작되면  $2t_M$ 개의 레지스터를 0으로 초기화한다. 'blk\_stop'은

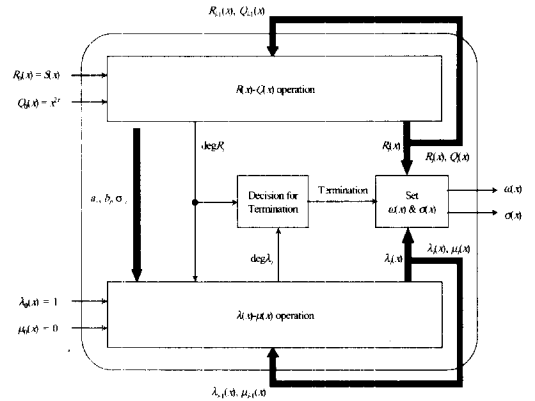


그림 3. MEA의 기본 셀 구조  
Fig. 3 The architecture of the basic cell in MEA block

수신 부호어의 마지막 심볼을 나타내는 신호로서 이 신호에 맞추어 계산된  $2t_0$ 개의 신드롬 값을 출력한다.

### 2.2. MEA 블록

MEA의 절차와 원리는 문헌 [21]에 잘 설명되어 있으며, MEA의 새로운 기본 셀 구조는 그림 3에 보인다.

신드롬  $S(x)=0$ 이면, 에러가 발생하지 않은 것이므로, MEA를 수행할 필요가 없고 에러 평가자 다항식과 에러 평가자 다항식은 각각  $\sigma(x)=1$  및  $\omega(x)=0$  으로 정하면 된다.  $S(x) \neq 0$ 이면, 다음과 같이 초기화를 시키고  $x^{2t}$ 와  $S(x)$ 를 기반으로 MEA를 수행한다.

$$R_0(x) = S(x), \quad Q_0(x) = x^{2t} \quad (3)$$

$$\lambda_0(x) = 1, \quad \mu_0(x) = 0$$

각 레지스터의 초기값이 문헌 [21]과 다름을 유의하라. 그 이유는  $Q_i(x)$ 의 차수가 항상  $R_i(x)$ 의 차수보다 크거나 같게 유지하므로써 오직  $Q_i(x)$ 의 차수만으로 MEA의 종결 조건을 검사하기 위함이다. 식 (3)과 같이 레지스터를 초기화시킨 후 다음의 반복 연산을 수행한다.

$$R_i(x) = [\sigma_{i-1} b_{i-1} R_{i-1}(x) + \frac{\sigma_{i-1} a_{i-1} Q_{i-1}(x)}{-x^{||i-1|}} + \sigma_{i-1} a_{i-1} Q_{i-1}(x) + \sigma_{i-1} b_{i-1} R_{i-1}(x)] \quad (4)$$

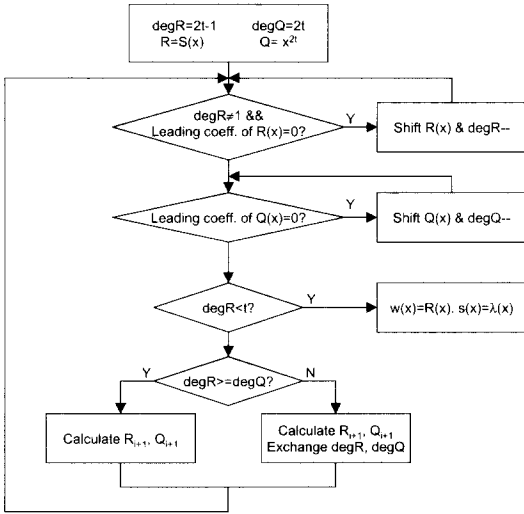


그림 4. MEA 각 단계의 계산 순서도  
Fig. 4 Operational flowchart for the iterative calculation of the MEA

$$\lambda_i(x) = [\sigma_{i-1}b_{i-1}\lambda_{i-1}(x) + \sigma_{i-1}a_{i-1}\mu_{i-1}(x)] - x^{||i-1||}[\sigma_{i-1}a_{i-1}\mu_{i-1}(x) + \sigma_{i-1}b_{i-1}\lambda_{i-1}(x)] \quad (5)$$

$$Q_i(x) = \sigma_{i-1}Q_{i-1}(x) + \overline{\sigma_{i-1}}R_{i-1}(x) \quad (6)$$

$$\mu_i(x) = \sigma_{i-1}\mu_{i-1}(x) + \overline{\sigma_{i-1}}\lambda_{i-1}(x) \quad (7)$$

여기서  $a_{i-1}$ 과  $b_{i-1}$ 은 다항식  $R_{i-1}(x)$ 와  $Q_{i-1}(x)$ 의 최고차항의 계수이고,

$$l_{i-1} = \deg(R_{i-1}(x)) - \deg(Q_{i-1}(x)) \quad (8)$$

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

$$\sigma_{i-1} = 0 \quad \text{if } l_{i-1} < 0 \quad (9)$$

이다.

반복 연산 수행 도중 다항식  $R_i(x)$ 의 차수가  $l$ 보다 작으면 반복 연산을 중지하고, 에러 위치자 다항식과 에러 평가자 다항식을 각각  $\sigma(x) = \lambda_i(x)$  및  $w(x) = R_i(x)$ 로 결정한다.

그림 4는 MEA의 각 단계에서 수행되는 계산 과정의 순서도를 나타낸다.

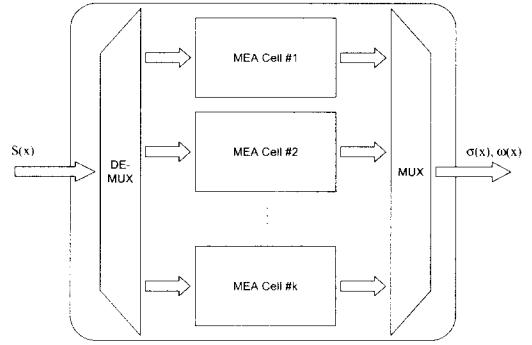


그림 5. MEA 셀의 멀티플렉싱  
Fig. 5 Multiplexing of MEA cell

현재 설정된 부호의 에러정정 능력을  $t_0$ 라 할 때 우리의 구조에서 MEA의 반복 계산을 완료하기 위해 2개 이상의 계수가 동시에 소거되지 않는 정규적인 경우를 고찰할 경우  $2t_0(2t_0+1)$  클럭이 소요된다. 이전의 설계 [21]에서는 MEA 연산의 매 단계마다 오직 하나의 차수만이 감소하도록 설계되었다. 최고차 항의 계수와 함께 그 다음 차수의 항의 계수들이 동시에 소거되면, 결국 최고차항의 계수가 0인 다항식을 가지고 그 다음 단계의 연산을 수행해야 했다. 우리는 MEA 연산에서 동시에 2개 이상의 차수가 소거되는 경우 불필요한 연산 단계를 생략할 수 있는 구조로 설계하였다.

또한  $2t_M$  길이의  $R(x)$  레지스터를  $2t_0$ 개의 신드롬 값들로 초기화한 후 MEA 과정을 시작하기 전에 신드롬 다항식의 최고차 계수  $S_{2t_0-1}$ 를  $R(x)$  레지스터 열의 최상위 레지스터로 천이하기 위하여  $(2t_M-2t_0)$  클럭이 요구된다. 물론 이 시간은 초기화 시  $R(x)$  레지스터 열의 최상위 레지스터에  $S_{2t_0-1}$ 를 초기화하면 제거할 수 있다. 그러나 이 경우 이미 언급한 바와 같이  $R(x)$  레지스터의 초기화를 위한 라우팅이 매우 복잡해질 것이다.

종결 조건이 만족되어 MEA의 수행이 끝나면,  $R(x)$ 와  $\lambda(x)$  레지스터 열의 내용을 다음 과정으로 전달하기 위해서 전술한 바와 같이 계수들을 레지스터의 MSB부터 내림차순으로 정렬해야 한다. 이를 위해 부가적으로  $t_0$  클럭이 요구된다. 따라서 우리의 설계에서 완벽한 MEA 수행을 위하여 총  $4t_0^2 + t_0 + 2t_M$  클럭이 요구된다.

구현된 적응형 RS 복호기의 원활한 파이프라인 동작을 위하여 MEA 수행 시간은 부호 블록의 길이  $n$ 에 의존하는 다른 단계의 처리 시간보다 작거나 같아야 한다. 즉,  $n \geq n_{min} = 4t_0^2 + t_0 + 2t_M$  이어

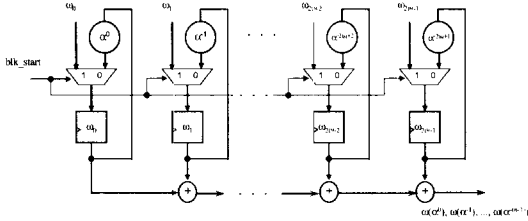


그림 6. 에러 평가자 다항식 평가를 위한 시스톨릭 어레이  
Fig. 6 A systolic array for polynomial evaluation of  $\omega(x)$

야 한다.  $n < n_{min} = 4t_0^2 + t_0 - 2t_M$ 의 경우를 수용하기 위하여 그림 5에서 보인 바와 같이 복수의 MEA 셀들을 멀티플렉싱한다.

만일  $k$ 개의 MEA 셀을 멀티플렉싱할 경우 블록 길이가  $n \geq n_{min} = (4t_0^2 + t_0 + 2t_M)/k$ 인 경우에 대해서도 복호가 가능하다. 이는 복호가 가능한  $n$ 의 최소값  $n_{min}$ 을 멀티플렉싱된 MEA 셀의 수에 따라서 보다 낮은 값으로 확장할 수 있음을 의미할 뿐 복호시간의 단축을 의미하지는 않는다. 또한 MEA 블록을 심볼 클럭보다 몇 배 빠른 클럭으로 동작하도록 설계한다. 즉, MEA 클럭이 심볼 클럭보다  $p$ 배 빠른 경우  $n_{min}$ 은  $(4t_0^2 + t_0 + 2t_M)/p$ 까지 감소할 수 있다. 이와 같이 고속의 전용 클럭 기법(overclocking)을 적용함으로써 요구되는 MEA 셀의 수를 줄일 수 있고, 복잡도와 MEA 수행 시간에 있어 현저한 감소를 이룰 수 있다.

### 2.3. 다항식 평가 블록

MEA 블록의 결과로 얻어진 에러 평가자 다항식  $\omega(x)$ 와 에러 위치자 다항식  $\sigma(x)$ 는 다항식 평가 블록으로 전달된다. 다항식 평가 블록에서는  $a^i, i=0, 1, \dots, n-1$ 에 대하여  $\omega(x), \sigma(x)$  및 미분형 에러 위치자 다항식  $\sigma'(x)$ 를 평가한다. 그림 6은 다항식  $\omega(x)$ 의 평가를 위한 구조를 보인다.  $\sigma(x)$ 와  $\sigma'(x)$ 의 평가를 위해서도 이와 동일한 구조를 적용할 수 있다.

이 구조에서 에러의 위치와 크기를 계산하는 다항식 평가 결과는 부호어가 수신된 순서와 반대의 순서로 발생된다. 복호기에서 현재 설정된 부호어의 길이를  $n_0$ 라 할 때 최초로 수신된 심볼을  $n_0-1$ 번째 심볼이라 하고, 마지막으로 수신된 심볼을 0번째 심볼이라 하면, 이 구조는 부호어의 길이  $n_0$ 에 상관없이 항상 0번째 심볼부터 임의의 변화가능한  $n_0-1$ 번째 심볼까지에 대한 다항식 평가 결과를 출력으로 제공한다. 이와 같이 수신된 부호어와 역순으로 평

가 결과를 출력해주는 방식은 단축형 RS 부호에 대하여 매우 유리하다. 이러한 구조에서는 역순 출력으로 인해 수신 부호어 저장을 위한 버퍼 메모리로서 FIFO형 메모리를 사용하는 것이 불가능하지만, RAM형 메모리로 대체될 수 있다. 또한 역순으로 평가된 부호어의 순서를 재정렬하기 위해 부호 블록 길이에 해당하는 여분의 시간이 필요하지만, 이는 복호기 실패 여부를 나타내는 신호를 생성하기 위해 정정된 부호어를 검증하는 시간과 공유될 수 있다. 따라서 본 논문에서는 수신된 부호어와 역순으로 평가하는 기법을 채택한다. 이러한 구조는 블록 길이를 가변으로 할 수 있는 복호기에 대해 적합하며, 이 때 사용하는 RAM형 메모리도 가변 블록 길이의 복호기 설계에서 MEA 블록의 멀티플렉싱된 구조에 잘 정합된다.

$i$ 번째( $i=0, 1, \dots, n_0-1$ ) 심볼에 대한  $\sigma(x)$ 의 평가 결과가  $\sigma(a^i)=0$ 이라면, 이 심볼은 오염된 심볼이다. 이때 수신된 부호어에서 오염된 심볼에 대응하는 에러의 크기  $e_i$ 를 빼서 정정을 수행한다. 대응하는 에러의 크기  $e_i$ 는 Forney 알고리즘을 사용하여  $a^i, i=0, 1, \dots, n-1$ 에 대한 다항식  $\omega(x)$ 와  $\sigma'(x)$ 의 평가 결과에 의하여 다음과 같이 계산된다.

$$e_i = -a^{-i(b-1)} \frac{\omega(a^{-i})}{\sigma'(a^{-i})} \quad 0 \leq i \leq n-1 \quad (10)$$

여기서  $\sigma'(x)$ 는

$$\begin{aligned} \sigma'(x) &= \sigma_1 + \sigma_3 x^2 + \dots + \sigma_{t-1} x^{t-2} \\ &= (\sigma_1 x + \sigma_3 x^3 + \dots + \sigma_{t-1} x^{t-1}) x^{-1} \quad (11) \\ &= \sigma_{odd}(x) x^{-1} \end{aligned}$$

과 같이 표현할 수 있으므로, 특별히  $b = 0$ 인 경우 에러의 크기  $e_i$ 는

$$e_i = -\frac{\omega(a^{-i})}{\sigma_{odd}(a^{-i})} \quad i = 0, 1, 2, \dots, n-1 \quad (12)$$

과 같이 구할 수 있다. 따라서,  $b = 0$ 의 경우 에러의 크기를 계산하기 위해 식 (10)에서  $a^i$ 에 의한 역원을 계산할 필요가 사라진다.

### 2.4. RAM 메모리 구조

RAM 메모리는 에러의 위치와 크기를 계산하는 동안 수신된 부호어의 저장을 위해 사용되었다. 사

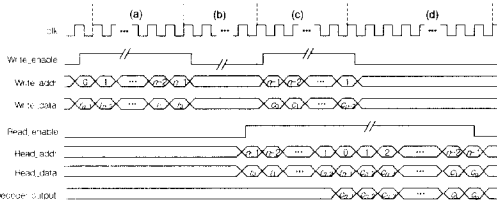


그림 7. RAM의 입출력 타이밍 다이어그램  
Fig. 7 Timing diagram for the RAM-type buffer memory

용되는 RAM은 멀티플렉싱 된 MEA 셀의 수와 동일하다. 수신된 부호어의 시작과 끝을 나타내는 blk\_start 신호와 blk\_stop 신호에 의하여 write\_enable 신호가 활성화되어 RAM에 수신된 부호어를 저장한다. 가장 먼저 수신된  $n_0-1$ 번째 심볼은 0번지에 저장하고 가장 나중에 수신된 0번째 심볼은  $n_0-1$ 번지에 저장한다. 저장된 부호어는 MEA 종결 후 에러가 발생한 심볼에 계산된 에러의 크기를 더하여 정정을 수행할 때 읽혀진다. 이때는  $n_0-1$ 번지의 0번째 심볼부터 읽어 에러가 발생한 심볼의 정정을 수행한다. 에러가 정정된 심볼은  $n_0-1$ 번지에 재 저장한다. 따라서 읽기와 쓰기 동작을 동시에 할 수 있는 dual-port RAM이 요구된다. 1번지에 저장된 심볼까지 정정을 수행한 다음 0번지에 저장된  $n_0-1$ 번째 심볼은 정정 후 재 저장하지 않고 바로 출력한다. 이후 1번지부터 다시 출력하여 역순으로 평가된 부호어를 정순으로 출력한다. 이에 대한 RAM의 입출력 타이밍 다이어그램은 그림 7에 나타냈다.

그림 7의 (a)에서는 신드롬을 계산하는 동안 수신된 부호어를 RAM 메모리에 저장한다. (b)는 MEA 수행을 나타내며, (c)에서는 수신된 부호어 차수의 역순으로 에러를 정정한다. (d)는 역순으로 평가된 부호어를 정순으로 출력함과 동시에 복호 실패를 검사한다. (c)와 (d) 구간에서  $m_{k-2}$  심볼의 정정 후  $m_{k-1}$  심볼은 재 저장하지 않고 바로 정정된 부호어로 출력된다.

### III. 회로합성 및 실험결과

설계된 적응형 RS 복호기는 VHDL을 이용하여 구현하였으며, FPGA에 성공적으로 합성하였다.

#### 2.1. 복호기의 회로 합성 결과

본 논문에서는 예로서  $t_M=10$ 까지의 에러 정정 능력을 갖는 적응형 RS 복호기를 설계한다. 이 구

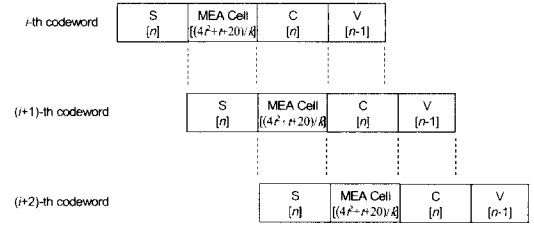


그림 8. 4단계 파이프라인 동작에서 각 단계별 소요시간  
Fig. 8 Timing relationship among the pipelined steps

표 1. RS 복호기의 비교

특징	Shayan, et. al.[17]	Huang[19]	제안된 회로
가변성	$n=255,$ $m=8,$ 임의의 $k$	$n=2^m-1,$ $m \leq 8,$ $t \leq 8$	$13 \leq n \leq 255,$ $m=8,$ $1 \leq t \leq 10$
지연시간 (clock)	$n(n+1)$	$3mn+4mt+4$ $m$	$3n-1+$ $(4t_o^2+t_o+20/p$ $k)$
데이터율 (bit/clock)	8	1	8
Bit rate (Mbit/sec)	83	48	240
게이트 수	340,500	43,987	211,296
입출력 형태	bit-parallel	bit serial	bit-parallel

조는 10이상의  $t_M$ 에 대하여 쉽게 확장할 수 있다. 본 논문에서 설계된 복호기는 2개의 MEA 셀을 멀티플렉싱하였으며, 버퍼 메모리로서  $2 \times 255$  어레이의 8비트 RAM을 사용하였다. 구현된 RS 복호기의 4단계 파이프라인 동작에서 각 단계별 소요시간은 그림 8에 보인다.

그림 8에서 S는 신드롬 계산 과정, MEA 셀은 각 셀에서 에러 위치와 다항식과 에러 평가자 다항

식을 계산하는 과정, C는 에러의 위치와 크기 계산 및 에러의 정정 그리고 V는 역순으로 정정된 부호어를 정순으로 정렬하고 동시에 정정된 부호어를 검증하는 과정을 나타낸다. 각 파이프라인 과정에 소요되는 심볼 클럭의 수를 함께 표시하였다. 설계된 복호기는 그림 8과 같이 연속적으로 입력되는 부호어에 대해서 파이프라인 동작에 의해 복호가

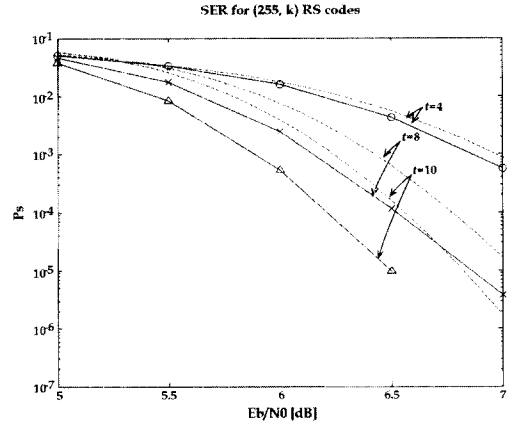
가능하다.  $p$ 는 심볼클럭에 대한 MEA 블록을 위해 사용된 클럭의 비를 나타낸다.  $p$ 와  $k$ 를 곱한 값이 클수록 블록 길이  $n$ 이 작은 값의 단축형 RS 부호에 대해서 복호가 가능할 것이다.

설계된 적응형 RS 복호기는 Leonardo Spectrum을 사용하여 합성하였으며, P&R과 시뮬레이션을 위해 Quartus II를 사용하였다. 타겟 디바이스는 Altera사의 APEX EP20K1500EBC-1X로 이며, 합성결과 적응형 RS 복호기 구현에 18,441개의 로직 셀이 소요되었다. 이는 로직 셀 당 12게이트<sup>[22]</sup>로 가정할 경우 약 211,296 게이트에 해당한다. 합성결과 임계경로는 34.25ns이었으며, 이를 근거로 25 Msps에서 35Msps까지 시뮬레이션한 결과, 합성된 복호기는 최대 30Msps에서 동작함을 확인하였다. 표 1은 기존에 제안된 가변형(versatile) RS 복호기와 제안된 회로의 비교 결과를 나타낸다.

### 3.2. 실험 결과

설계된 적응형 RS 복호기의 실험을 위해 타겟 디바이스인 EP20K1500EBC-1X가 탑재된 DSP 개발 보드를 사용하였으며, 복호기 이외에 RS232 통신기능을 포함한 32bit Nios 임베디드 프로세서를 구성하였다. 컴퓨터에서 메시지를 발생 시켜 RS 부호화를 수행한 후, BPSK로 변조한 다음 SNR을 5~7dB까지 0.5dB씩 증가하면서 AWGN 잡음을 첨가하여 9600bps로 보드에 전송한다. 임베디드 프로세서는 수신 부호어를 저장하고 한 블록의 수신 이 완료되면, 수신 부호어를 적응형 RS 복호기에 전달한다. 수신 부호어의 복호결과가 출력되면 임베디드 프로세서는 복호된 부호어를 저장하며, 한 블록의 복호가 완료되면 다시 컴퓨터에 전송한다. 복호된 데이터를 수신한 컴퓨터는 송신 메시지와 복호된 메시지를 비교하여 심볼 에러율(SER)과 비트 에러율(BER)을 산출한다. 위의 과정은 블록길이를  $n=255$ 로 고정하고, 에러정정 능력  $t=4, 8, 10$ 인 경우에 대하여 각각 10000블록씩 수행하였다. 그림 9는 식 (13)과 (14)로 주어지는 RS 부호의 SER과 BER의 상한<sup>[23,24]</sup>과 실험 결과로 얻어진 SER과 BER을 비교한 것이다. 그림 9에서 점선은 계산된 성능의 상한값을, 실선은 실험 결과를 나타낸다.

$$P_s \leq \frac{1}{2^m - 1} \sum_{j=t+1}^{2^m - 1} j \binom{2^m - 1}{j} p^j (1-p)^{2^m - 1 - j} \quad (13)$$



(a)

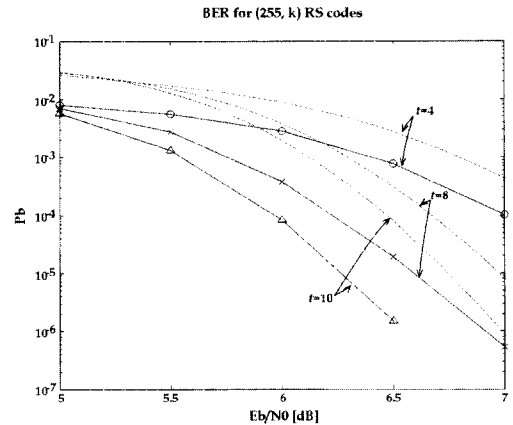


그림 9  $t=4, 8, 10$ 인 경우의 (a) 심볼 에러율 (b) 비트 에러율(AWGN 채널, BPSK 변조)  
Fig. 9 (a) Symbol error rate (b) bit error rate for  $t=4, 8, 10$  (AWGN channel, BPSK)

$$P_b \leq \sum_{i=t+1}^{2^m - 1} \frac{i}{2(2^m - 2)} \binom{2^m - 1}{i} p^i (1-p)^{2^m - 1 - i} \quad (14)$$

여기에서,  $m$ 은 심볼 당 비트수이며,  $p$ 는 채널에서 각 부호어 심볼에 대한 전송 에러율이다. 그림 9에서 알 수 있듯이 실험에 의해 결정된 SER과 BER이 모두 이론적인 상한보다 낮은 값을 보임을 알 수 있다.

그림 10은 232x232 크기의 lena 영상을 RS 부호화한 후 SNR=6dB의 잡음을 첨가하여 복호기에서 복호한 결과를 보인 것이다. 영상의 각 행을 RS 부호의 한 블록으로하여  $t=10$ 인 (253, 232) RS 부



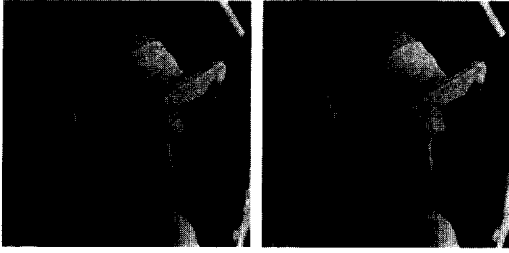


그림 10 Lena 이미지; (a) 복호기 입력 (b) 복호기 출력  
 Fig. 10 The Lena image; (a) decoder input (b) decoder output

호를 적용하였으며, 복호 결과 영상에서 리던던시 심볼은 보이지 않았다.

#### IV. 결 론

본 논문에서는 임의의 메시지 길이  $k$  뿐 아니라 임의의 블록 길이  $n$ 를 갖는 RS 부호를 복호할 수 있는 적응형 RS 복호기를 설계하였다. 설계된 RS 복호기는 단축형 RS 부호의 복호 전에 영들을 추가하지 않아도 되므로 단축형 RS 부호에 특히 유리하다. 또한 이들 RS 부호의 오류정정 능력  $t$ 의 값을 매 부호어 블록마다 실시간으로 변화시킬 수 있으므로, 응답 채널이 유용한 경우 채널의 시변 잡음 레벨에 적응적으로 오류 정정 능력을 가변할 수 있도록 활용될 수 있을 것이다. 제시된 복호기 구조는 수정 유클리드 알고리즘에 기반한 4단계 파이프라인 처리를 수행한다: (1) 신드롬 계산 (2) MEA 블록 (3) 에러크기 계산 (4) 복호기 실패 검사. 각 단계는 가변 길이의 RS 복호에 적합한 구조를 갖도록 설계되며, 특히 수정 유클리드 알고리즘(MEA) 블록은 다음과 같은 특징을 갖는 구조를 제시하였다. 이 MEA 수행 회로는  $2t_M$  길이의 레지스터 열을 사용하여 가변  $t$  값에 따라 레지스터 열의 길이도 또한 가변되도록 설계하였다. 또한, MEA 수행에 요구되는 레지스터 열의 길이를 단축하여 복호지연을 단축하였다. 또한 제시된 MEA 블록은 연산 셀들의 멀티플렉싱 기법과 심볼 클럭보다 빠른 전용의 MEA 클럭 기법(overclocking)의 사용으로 연산 시간을 단축하였다. 에러의 크기 계산을 위한 다항식 평가를 위해 역순 출력을 갖는 다항식 평가 회로를 채용했다. 구현된 적응형 RS 복호기는 완벽한 파이프라인 동작이 가능하다.  $GF(2^8)$  상의 적응형 RS 부호를 VHDL로 설계하고, FPGA에 합성하여 실험에 성공하였다. 구현된 복호기는 211,

296개의 게이트가 사용되었고, 30Msps의 최대 동작주파수를 갖는다.

#### 참 고 문 헌

- [1] I. F. Akyildiz, I. Joe, H. Driver and Y. L. Ho, "An Adaptive FEC Scheme for Data Traffic in Wireless ATM Network," IEEE Trans. Networking, vol. 9, no. 4, pp. 419-426, Aug. 2001.
- [2] A. Shiozaki, K. Okuno, K. Suzuki, T. Segawa, "A Hybrid ARQ Scheme with Adaptive Forward Error Correction for Satellite communications," IEEE Trans. on Computers, vol. 39, no. 4, pp. 482~484, Apr. 1991.
- [3] S. Cho, A. Goulart, I. F. Akyildiz and N. Jayant, "An Adaptive FEC with Provisioning for Real-Time Traffic in LEO Satellite Networks," in Proc. IEEE Int. Conf. Commun. ICC'01, vol. 9, pp. 2938 -2942, 2001.
- [4] N. Nikaein, H. Labiod and C. Bonnet "MA-FEC: A QoS-Based Adaptive FEC for Multicast Communication in Wireless Networks," in Proc. IEEE Int. Conf. Commu. ICC'00, vol. 2, pp. 954-958, 2000.
- [5] S. J. Li et al. "Adaptive Reed-Solomon Coding for Wireless ATM Communication," IEEE Southeastcon 2000. pp.27-30.
- [6] A. Almulhem, F. El-guibaly, T. A. Gulliver, "Adaptive Error Correction for ATM communications using Reed-Solomon codes," in Proc. IEEE Southeastcon'96. pp. 227-239, 1996.
- [7] M. B. Pursley and C. S. Wilkins "Adaptive-Rate Coding for Frequency-Hop Communications over Rayleigh Fading Channel," IEEE Journ. Sel. Areas Commun., vol. 17, pp. 1224~1232, July 1999.
- [8] B. Vucetic, "An Adaptive Coding Scheme for Time-Varying Channels," IEEE Trans. commun., vol. 39, no. 5, pp. 653-663. May 1991.
- [9] T. Tuan and K. Park, "Multiple Time Scale Redundancy Control for QoS-sensitive Transport of Real-time Traffic," in Proc. IEEE INFOCOM 2000. vol. 3, pp. 1683-1692, 2000.



[10] D. Cygan, E. Lutz, "A Concatenated Two Stage Adaptive(CTSA) Error Control Scheme for Data Transmission in Time-Varying Channels," IEEE Trans. Commun., vol. 43, no. 2/3/4, pp. 795-803, Apr. 1995.

[11] M. A. Hasan and V. K. Bhargava, "Architecture for a low complexity rate-adaptive Reed-Solomon encoder," IEEE Trans. Computers, vol. 44, no. 7, pp. 938~942, July 1995.

[12] P802.16/D5-2001, IEEE Standard for Local and Metropolitan Area Networks - Part 16: Standard Air Interface for Fixed Broadband Wireless Access Systems 2001, Oct. 2001.

[13] Data-Over-Cable Service Interface Specifications, Radio Frequency Interface Specification, S P-RFiv2.0-I01-011231, Dec. 2001.

[14] ITU COM 15-131-E, Draft New Recommendation G.992: Asymmetrical Digital Subscriber Line(ADSL) Transceiver for Approval, March 1999.

[15] T1E1.4/2000-152, Very-High-Speed Digital Subscriber Lines(VDSL) Metallic Interface, Part 1: Functional Requirements and Common Specification, May 2000.

[16] Andrew d. Preez, F. Swarts, F. Agdhasi, "A Flexible Reed-Solomon codec," IEEE Africon., vol. 1, pp. 93-98. 1999.

[17] Y. R. Shayan, T. Le-Ngoc, "A Cellular Structure for a Versatile Reed-Solomon Decoder," IEEE Trans. Comput., vol. 46, no. 1, pp. 80-85, Jan. 1997.

[18] Y. R. Shayan, T. Le-Ngoc, V. J. Bhargava, "A Versatile Time-Domain Reed-solomon Decoder," IEEE J. Select. Areas Commun., vol. 8, no. 8, pp. 1535-1542, Oct. 1990.

[19] J. C. Huang, C. M. Wu, M. D. Shieh, C. H. Wu, "An Area-Efficient Versatile Reed-Solomon Decoder for ADSL," IEEE ISCAS'99, vol. 1, pp. 517-520, 1999.

[20] 정인택, 이승수, 송상섭, "고속 DSL 모뎀을 위한 Programmable RS 부호 설계," 한국통신학회 논문지, vol. 23, no. 9B, pp. 1314-1319, September 2001.

[21] Howard M. Shao, and Irving S. Reed, "On the VLSI Design of a Pipeline Reed-Solomon

Decoder Using Systolic Arrays," IEEE Trans. Computers, vol. 37, no. 10, Oct. 1988.

[22] Application note 112, Gate Counting Methodology for APEX20K Devices, ver 1.01, Sep. 1999.

[23] B. Sklar, *Digital Communications: Fundamentals and Applications*, Prentice Hall INC., 2001.

[24] R. E. Ziemer, R. L. Peterson, *Introduction to Digital Communication*, Prentice Hall INC., 1992.

송 문 규(Moon-Kyou Song)

중신회원



1988년 2월 : 고려대학교 전자공학과 졸업

1990년 2월 : 고려대학교 전자공학과 공학석사

1994년 2월 : 고려대학교 전자공학과 공학박사

1994년 3월~현재 : 원광대학교

전기전자및정보공학부 부교수

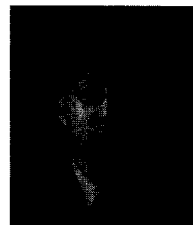
1999년 9월~2000년 8월 : 캐나다 빅토리아 대학교 전기및컴퓨터공학과 방문교수

2001년 1월~현재 : 한국통신학회 논문지 편집위원

<주관심분야> 무선통신, 디지털 통신시스템 설계, 채널 코딩

공 민 한(Min-Han Kong)

준회원



2001년 2월 : 원광대학교 전기공학부 졸업

2003년 2월 : 원광대학교 제어계측공학과 공학석사

2003년 3월~현재 : 원광대학교 제어계측공학과 박사과정

<주관심분야> 디지털 통신시스템 설계, 채널코딩