

# 가변 부호율과 블록 길이를 갖는 연속 가변형 리드솔로몬 복호기

공민한, 송문규  
원광대학교 전기전자 및 정보공학부  
전화 : 063-850-6341 / 핸드폰 : 016-604-9636

## A Continuous Versatile Reed-Solomon Decoder with Variable Code Rate and Block Length

Min-Han Kong, Moon-Kyou Song  
Dept. of Electrical Electronic and Information Engineering, Wonkwang University  
E-mail : [y2kdoli@wonkwang.ac.kr](mailto:y2kdoli@wonkwang.ac.kr)

### Abstract

In this paper, an efficient architecture of a versatile Reed-Solomon (RS) decoder is designed, where the message length  $k$  as well as the block length  $n$  can be variable. The decoder permits 3-step pipelined processing based on the modified Euclid's algorithm(MEA). A new architecture for the MEA is designed for variable values of error correcting capability  $t$ . To maintain the throughput rate with less circuitry, the MEA block uses both the recursive and the overclocking technique. The decoder can decode a codeword received not only in a burst mode, but also in a continuous mode. It can be used in a wide range of applications due to its versatility. A versatile RS decoder over  $GF(2^8)$  having the error-correcting capability of up to 10 has been designed in VHDL, and successfully synthesized in an FPGA chip.

### 1. 서론

RS (Reed-Solomon) 부호는 연집에러에 가장 강력한 부호로 알려져 있으며, CD-ROM, DVD, 위성통신과 같은 분야에 널리 사용되고 있다. 블록 길이  $n$ 과 메시지

길이  $k$ 가 고정된, 즉 에러정정 능력  $t$ 가 고정된 기존의 RS 부호는 최악의 채널 상태에서 평균성능을 얻기 위해 설계된다. 그러나 채널이 최악의 상태로 존재하는 경우는 전체 응용시간의 매우 적은 부분을 차지하므로, 대역 효율성을 저하시킬 뿐 아니라 성능의 변화의 요구에 부합할 수 없다[1,2]. 따라서 최근 시변 잡음 레벨과 원하는 성능에 따라서 적응적으로 최적의 부호를 선택하는 적응형 순방향 에러정정 기법(adaptive forward error correction)기법이 통신 표준에 채택되고 있다. 적응형 순방향 에러정정 기법은 가변 에러정정 능력  $t$ 를 갖는 가변형 RS 부호에 의해 구현될 수 있다[2-3].

본 논문에서는 매 부호 블록마다 실시간으로 메시지 길이  $k$ 와 블록 길이  $n$ 을 가변 할 수 있고, 버스트 모드와 연속 모드에 있어 효율적인 복호를 수행할 수 있는 복호기를 구현한다.

### 2. 가변형 RS 복호기의 구조

부호어 블록 내에서 최대  $t_M$  개의 에러를 정정할 수 있는 가변형 RS 복호기의 구조는 그림 1에 보인다. 복호기의 에러정정 능력  $t$ 와 블록 길이  $n$ 은 매 부호 블록마다  $0 \leq t \leq t_M$  그리고  $1 \leq n \leq 2^m - 1$ 의 범위에서 변화될 수 있다. 복호기는 수정 유클리드 알고리즘(the modified Euclid's algorithm; MEA)을 기반으로 (1) 신드롬 계산, (2) MEA, (3) 다항식 평가를 통한 에

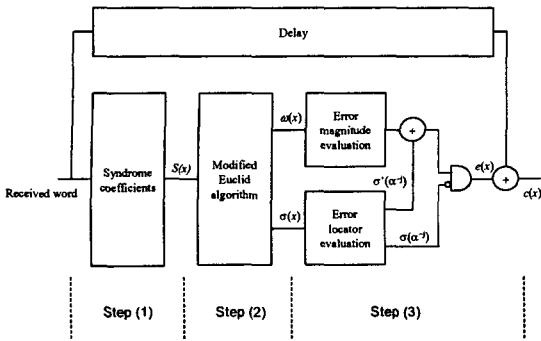


그림 1 가변형 RS 복호기의 구조

러 정정 등의 효율적인 3단계 파이프라인 처리를 수행한다.

각 단계는 분리된 독자적인 클럭을 사용하도록 설계하였으므로, 단계 2 그리고/또는 단계 3에서 고속 클럭을 사용하면 2단계 파이프라인 처리로도 동작할 수 있다. 이러한 특징으로 인해 입출력 클럭이 상이한 경우에도 버퍼를 사용하지 않고서 복호기를 쉽게 적용할 수 있다.

### 2.1 복호기 구조 설계

단계 (1)에서  $2t$ 개의 신드롬 계수로 구성되는 신드롬 다항식  $S(x)$ 를 얻는다. 신드롬 다항식  $S(x)$ 는 단계 (2)에 입력되어 MEA를 이용하여 에러 위치 다항식  $\sigma(x)$ 와 에러 평가자 다항식  $\omega(x)$ 를 계산한다.

변화하는 에러정정 능력  $t$ 를 위하여 문헌 [4]에서 설명한 MEA의 새로운 구조를 사용한다. MEA 블록에서 천이 레지스터의 동작 길이는 하나 감소되었으며,  $t$ 의 서로다른 값에 따라서 변화되도록 하였다. 간단한 회로로써 동작 속도를 유지하기 위해 MEA 블록은 재귀적 기법과 고속 클럭킹 기법을 사용한다. 이로써 연속 모드의 복호에 대해서 보다 더 효율적인 구현이 가능하다. 설계된 복호기는 버스트 모드와 연속 모드에서 모두 복호가 가능하다.

단계 (3)에서는 Chien과 Forney의 알고리즘을 이용하여 에러의 위치와 크기를 계산한다. 만약  $i$ 번째 심볼 위치에서  $\sigma(x)$ 를 평가한 결과인  $\sigma(d^i)$ 가 0이면, 이는 해당 심볼이 오류임을 나타낸다. 그러면, 해당 심볼에서 에러 크기  $e_i$ 를 빼줌으로써 이 에러를 정정해야 한다.

해당 에러 크기  $e_i$ 는 다음과 같이 계산할 수 있다.

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

이는 생성자 다항식의 기저(base)가  $b=0$ 의 특수한 경우에 대해서 Forney의 알고리즘은 다음처럼 간략화된

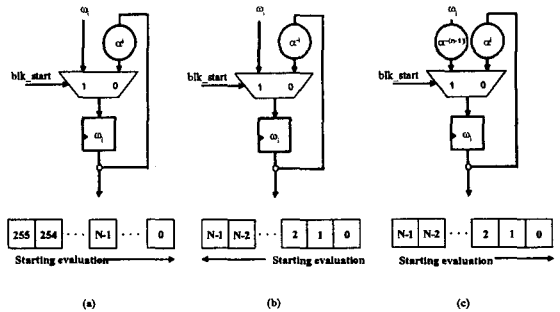


그림 2 다항식 평가회로를 위한 3가지 구조

다.

$$e_i = -\frac{\omega(\alpha^{-i})}{\sigma'_{odd}(\alpha^{-i})} \quad i = 0, 1, 2, \dots, n-1 \quad (2)$$

에러의 위치와 에러의 크기를 계산하기 위해서는  $\sigma(x)$ 와  $\omega(x)$  그리고  $\sigma'(x)$  등의 3가지 다항식을 평가하여야 한다. 이를 위해 그림 2에 보인 바와 같이 다항식 평가를 위한 3가지 후보 구조가 존재한다.

먼저 기법 (a)는 오직 하나의 상수 곱셈기를 사용하여 평가 출력을 정순으로 출력한다. 그러나 이 방법은 항상 길이가  $2^n-1$ 인 부호어를 출력하므로 길이가  $n < 2^n-1$ 인 단축형 부호의 복호를 위해 여분의 시간을 소모하게 된다.

기법 (b)는 블록 길이  $n$ 와 동일한 수의 심볼만을 평가하므로 다항식 평가를 위해 오직  $n$  심볼 클럭만이 소요된다. 이 기법은 기법 (a)에서처럼 오직 하나의 상수 곱셈기만을 사용하므로 구현이 간단하다. 그러나 이 구조는 출력이 역순으로 발생되기 때문에 이 순서를 다시 정순으로 복원하기 위한 부가적인 단계와 버퍼 메모리의 용량 추가가 필요하다. 결과적으로 이는 결국 복호 지연과 복잡도의 증가를 초래한다.

기법 (c)는 기법 (a)에 하나의 가변 곱셈기를 추가하므로써 블록 길이  $n$ 과 동일한 수의 출력을 산출한다. 그 출력은 정순으로 발생된다. 기법 (c)가 다항식 평가만을 위해서 가장 복잡한 회로를 야기하지만, 기법 (a)보다 작은 평가 시간을 소모한다. 또한 이 방법은 평가 출력의 순서를 정순으로하기 위한 부가적인 단계를 필요로 하지 않으므로 전체 복호기의 관점에서 기법 (b)보다 적은 복호 지연 시간을 요구한다. 따라서 기법 (c)를 단축형 부호를 포함하는 가변 블록 길이의 RS 부호를 복호하기 위해 가장 적합한 것으로 선택한다.

그림 3은  $\sigma(x)$ 의 다항식 평가를 위한 시스톨릭 어레이 구조를 보인다.  $\omega(x)$ 와  $\sigma'(x)$ 의 다항식 평가를 위해서도 동일한 구조를 사용한다.

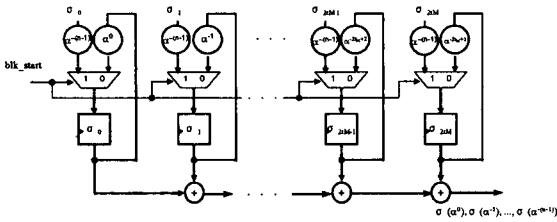


그림 3  $w(x)$  평가를 위한 시스템릭 어레이 구조

### 2.2 처리 시간

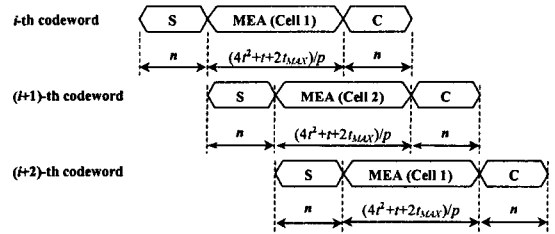
단계 (1)에서 신드롬을 계산하기 위해  $n$  클럭이 소요된다. 단계 (3)에서 다항식 평가를 통해 에러를 정정하기 위해  $n$  클럭이 소요된다. 단계 (1)과 (3)은 수신되는 심볼 클럭을 사용한 것이다. 단계 (2)에서 MEA를 수행하기 위해 요구되는 처리 시간은 다음과 같다. MEA를 시작하기 전에 신드롬 계수를 가장 오른쪽의 레지스터로 천이하는데  $(2t_M - 2t)$  클럭이 소요된다. MEA 반복 연산을 계산하기 위해서는 한 번에 하나 이상의 계수가 소거되지 않는 정상적인 경우  $2t(2t+1)$  클럭이 소요된다. MEA 과정을 마친 후 다음의 다항식 평가 단계를 위한 레지스터에 직렬로 전달되어야 한다. 이를 위해  $t$ 의 클럭이 소요된다.

따라서 우리의 구조에서 MEA 연산을 마치기 위해 총  $t_{MEA} = 4t^2 + t + 2t_M$  개의 클럭이 소요된다.

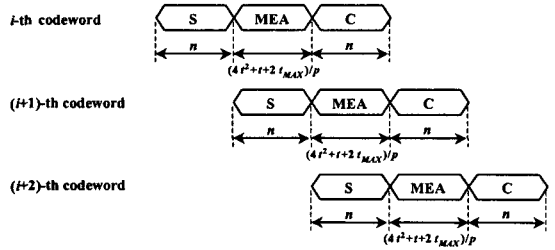
### 2.3 파이프라인 동작

복호기의 파이프라인 동작을 위해 MEA 블록에서 요구되는 처리 시간은 다른 단계에서 요구되는 처리 시간인  $n$  심볼 클럭보다 작거나 같아야 한다. 이 경우 최소의 블록 길이는  $n_{min} = 4t^2 + t + 2t_M$  이 된다.

블록 길이  $n_{min}$ 이  $4t^2 + t + 2t_M$ 보다 작은 경우 충돌을 피하기 위해 문헌 [5]과 같이 다수의 MEA 셀을 멀티플렉싱하면,  $n_{min}$ 은 사용된 MEA 셀의 수에 비례해서 더 낮은 값으로 확장된다. 그러나 이러한 멀티플렉싱 기법은 실제적인 처리 시간  $t_{MEA}$ 은 줄이지 않으면서 단지  $n_{min}$ 의 값만을 감소시킬 수 있다. 만일 MEA 블록에 오버클러킹 기법을 사용하면, 수용가능한 블록 길이 뿐 아니라 처리 시간도 동시에 줄일 수 있다. 심볼 클럭보다  $p$ 배 빠른 클럭을 사용하면, MEA의 처리를 위해  $t_{MEA} = (4t^2 + t + 2t_M)/p$  심볼 시간이 소요되며,  $n_{min}$ 의 값도  $(4t^2 + t + 2t_M)/p$ 로 낮추어진다. 만일  $p$ 의 값에 제한이 있다면, 오버클러킹 기법과 멀티플렉싱 기법을 결합하여 사용 가능하다. 이 경우  $n_{min} = (4t^2 + t + 2t_M)/pk$ 로 감소할 것이며,  $t_{MEA}$ 는  $(4t^2 + t + 2t_M)/p$  심볼 시간이 될 것이다. 여기서  $k$ 는 멀티플렉싱을 위해 사용된 MEA 기본 셀의 수를 나타낸다.



(a)



(b)

그림 4 파이프라인 처리를 위한 시간 관계 (a) 멀티플렉싱( $k=2$ )과 오버클러킹 기법 사용 (b) 오버클러킹 기법 사용

그림 4는 파이프라인 처리를 위한 3 단계들 사이의 시간 관계를 보인 것이다. 그림 4에서 단계 (1)과 단계 (3)은 모두 심볼 클럭으로 동작하고, 단계 (2)는  $p$ 배 고속 클럭을 사용한 경우를 가정하였다. 그림 4(a)는 MEA 수행 시간이  $n$  심볼 시간보다 큰 경우 연속적인 신드롬 다항식을 처리하기 위해서  $k = \lceil t_{MEA}/n \rceil$  개의 MEA 셀을 멀티플렉싱한 것을 보인 것이다. 심볼 클럭보다  $p$  ( $\geq (4t^2 + t + 2t_M)/n$ )배 빠른 전용 클럭이 유용할 경우 그림 4(b)에 보인 것처럼 멀티플렉싱을 적용하지 않아도 된다. 따라서 멀티플렉싱 기법 없이 오버클러킹 기법만을 적용하는 것이 가능한 경우 복잡도와 복호 지연의 관점에서 모두 유리해진다.

### 2.4 가변 파라미터를 갖는 연속적 복호

현재 복호될 부호어 블록의  $n$ 과  $t$ 의 값을 각각  $n_0$ 와  $t_0$ 라 하고, 다음 부호어 블록에 대한  $n$ 과  $t$ 의 값을 각각  $n_1$ 과  $t_1$ 라 하자. 다음 값들이 현재 값들보다 같거나 크다면, 그림 5(a)에 보인 것처럼 연속적인 블록들 간에 충돌은 발생하지 않는다. 그러나 그렇지 않은 경우에는 그림 5(b)에 보인 것처럼 이전 블록에 대한 실행이 아직 끝나기 전에 새로운 부호어 블록이 단계 (2)로 입력될 수 있으므로 충돌이 발생할 수 있다. 따라서 연속 복호를 위해서  $n_1$ 과  $t_1$ 이  $n_0$ 과  $t_0$ 보다 작을 때마다 부호어 블록들 사이에  $(n_0 - n_1) + (t_0 - t_1)(4(t_0 + t_1) + 1)/pk$  심볼 시간의 보호 시간이 요구된다. 그러면, 그림 6(c)에 보인 것처럼 충돌을 피할 수 있으며, 이 보호 시간은 새로운

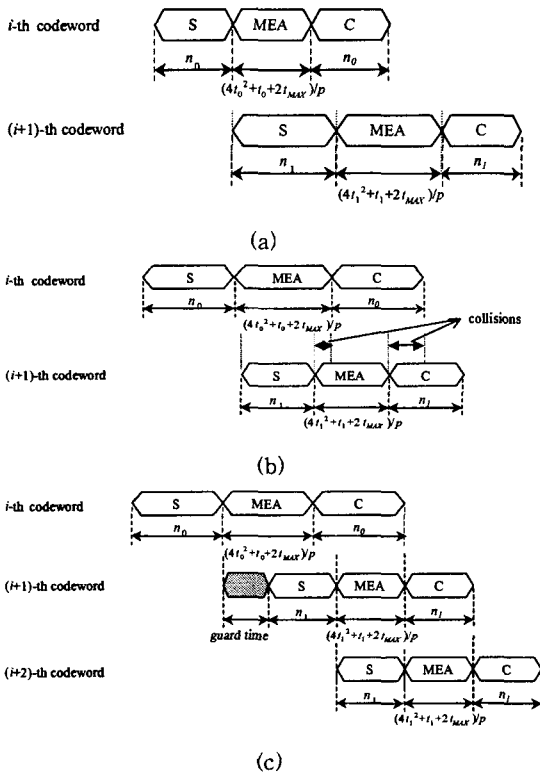


그림 6 파라미터 변화에 따른 타이밍 관계 (a)  $n_0 \leq n_1$   
 (b)  $n_0 \geq n_1$  (c) 보호 시간 사용  $n_0 \geq n_1$

값들이 더 작을 때에만 요구된다.

### 3. 합성 결과

본 논문에서는  $t_M=10$ 을 갖는 가변형 RS 복호기를 설계하고 VHDL로 구현되며, FPGA에 합성하여 검증을 수행하였다. 표 1은 복호기 각 블록별 소요 로직 셀과 메모리 비트수를 나타낸 것이다.

심볼 클럭 주파수를  $f_{sym}$ , 고속의 전용 클럭 주파수를  $f_{over}$ 이라 하면  $p=f_{over}/f_{sym}$ 이므로  $f_{sym} < n \times f_{over} / (4t^2 + t + 2t_M)$  이 만족되어야 한다. 합성 결과  $f_{over} = 38\text{MHz}$ 이므로,  $n=255$ ,  $t=10$ 의 경우 연속 복호가 가능한 가변형 RS 복

표 1 복호기 각 블록별 소요 로직 셀과 메모리 비트수

복호기 블록	로직 셀	메모리 비트
신드롬 계산 블록	603	0
MEA 수행 블록	1,880	0
에러 위치 다항식 평가 블록	3,215	22,440
에러 평가 다항식 평가 블록	2,877	20,400
에러 크기 계산 및 정정 블록	103	2,048
지연 블록	114	4,112

호기에서 심볼 클럭의 최대 주파수는 22.5MSPS이 된다.

### 4. 결론

본 논문에서는 임의의 블록 길이  $n$ 와 메시지 길이  $k$ , 결과적으로 가변 에러정정 능력  $t$ 를 갖는 RS 부호를 복호할 수 있는 가변형 RS 복호기의 효율적인 구조를 제안한다. 복호기는 MEA를 기반으로 한 3단계 파이프라인 처리를 수행한다. 각 단계는 분리된 클럭에 의해 구동될 수 있으며, 단계 2 그리고/또는 단계 3에 고속 클럭을 사용하면 2단계 파이프라인 처리로 동작하는 것이 가능하다. 제안된 가변형 RS 복호기는 버스트 모드 뿐만 아니라 연속 모드로 수신된 부호어를 복호할 수 있으며, 그 가변성으로 인해 다양한 분야에서 사용될 수 있다. GF( $2^8$ ) 상에서 최대 10의 에러정정 능력을 갖는 가변형 RS 복호기를 VHDL로 설계하였으며, FPGA 칩에 성공적으로 합성하였다. 합성결과 복호기 최대 동작 주파수는  $n=255$ 인 경우 22.5MSPS이며, 8,792개의 로직 셀과 49,000비트의 메모리가 사용되었다.

### 참고문헌

- [1] M. A. Hasan, V. K. Bhargava, "Architecture for a low complexity rate-adaptive Reed-Solomon encoder," *IEEE Trans. on Computers*, vol. 44, no. 7, pp. 938~942, July 1995.
- [2] S. J. Li, K. F. Pan, J. S. Yuan, A. J. Vigil, A. L. Berg, "Adaptive Reed-Solomon Coding for Wireless ATM communication", in *Proc. IEEE, Southeastcon'00*. pp. 27~30, 2000.
- [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] M. K. Song, E. B. Kim, H. S. Won, M. H. Kong, "Architecture for Decoding Adaptive Reed-Solomon Codes with Variable Block Length," *IEEE Transactions on Consumer Electronics*, vol. 48, No. 3, pp. 631-637, Aug. 2002.
- [5] 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.