효율적 구조의 수정 유클리드 구조를 이용한 Reed-Solomon 복호기의 설계

김동순*, 정덕진
인하대학교 공과대학 전자재료공학과

Implementation of Reed-Solomon Decoder Using the efficient Modified Euclid Module

Dong-Sun Kim, Duck-Jin Chung
Dept. of Electronic Materials & Device Engineering, INHA Univ.

Abstract - In this paper, we propose a VLSI architecture of Reed-Solomon decoder. Our goal is the developmemt of an architecture featuring parallel and pipelined processing to improve the speed and low power design. To achieve the this goal, we analyze the RS decoding algorithm to be used parallel and pipelined processing efficiently, and modified the Euclid's algorithm arithmetic part to apply the parallel structure in RS decoder. The overall RS decoder are compared to Shao's, and we show the 10 % area efficiency than Shao's time domain decoder and three times faster, in addition, we approve the proposed RS decoders with Altera FPGA Flex 10K-50, and Implemeted with LG 0.6 μ processing.

1. 서 론

현대사회에서 신속한 정보의 전달을 위한 통신시스템 및 데이터의 저장을 위한 저장매체의 역할은 매우 큰 비중을 차지하고 있다. 과거의 통신분야에서 부호나 음성신호 등이 주된 정보원이었다면, 오늘날의 통신분야에서는 문자 , 데이터, 음성 및 영상 등의 다양하고 방대한 정보원들이 사용되어지고 있다. 우리가 흔히 사용하는 라디오, 텔레비전, 휴대용 전화기 및 CD-ROM등이 그 좋은 예라 할 수 있으며, 보다 많은 양의 정보를 빠르고 정확하게 전송하기 위한 알고리듬 및 하드웨어의 개발이 매우 중요하게 대두되고 있다. 또한 전송하는 데이터의 신뢰도(Reliability), 보안성(Security) 및 정보의 질(Quality)의 향상을 위해 아날로그 방식보다는 디지털 방식을 이용한 통신시스템이 제안되었고, 이러한 디지털 통신방식을 이용해 기존의 아날로그 방식에 비해 많은 정보를 고속으로 전달 할 수 있게 되었다. 또한 정보의 효율적인 전달을 위해 다양한 전송매체를 이용하게 되었으며, 이러한 통신매체는 전달하는 정보의 오류를 증가시키는 여러 가지 전파 특성을 가지고 있다.[1] 따라서, 저장 또는 전송하고자하는 정보의 신뢰도를 높이기 위해서는 발생하는 오류를 정정해야하는 문제점을 가지고 있다. 본 논문에서는 통신시스템 중에서 정보 전달과정에서 발생할 수 있는 산발오류 및 군집오류 등에 의한 송신측 정보의 왜곡을 막고 이를 정확히 수신 측에 전달하는 Reed-Solomon(RS) 오류정정부호화기를 구현하였으며 검증하였다. 일반적으로 (n,k) RS부호는 m 비트의 심볼단위로 구성되어 있으며, k의 메시지 코드워드에 n-k 개의 잉여항을 추가하여 전송하는 FEC(Forward Error Correction)의 한 방법이다. RS 부호의 복호과정은 key equation을 푸는 과정으로 요약할 수 있고, 이러한 key equation을 푸는 방법으로는 크게 행렬을 이용한 Peterson-Gorenstein-Zeirler 알고리듬과 Berlekamp-Massey 알고리듬 및 Euclid 알고리듬으로 나눌 수 있다.[2]

Peterson-Gorenstein-Zeirler 알고리듬의 경우는 parity를 이용한 행렬연산을 이용한 것으로 많은 나눗셈 과정으로 하드웨어 구현에 많은 어려움이 있으며, Berlekamp-Massey 알고리듬 역시 유한 필드상에서 역수를 구해야 하는 과정이 필요하며, 이러한 유한체내에서 역수를 구하는 과정은 곱셈보다 지연시간이 길어 전체 복호기의 throughput을 결정하는 중요한 부분이다. 따라서, 고속의 역수 계산을 위해 LUT(Look-up table)를 사용하게 되는데, 이러한 LUT의 경우 일반적으로 Rom을 이용해 구현되고, 이러한 Rom은 다른 셀보다 많은 하드웨어 크기를 차지하게 된다. 또한, Berlekamp 알고리듬을 사용하는 경우는 계산된 discrepancy을 가지고, 각 레지스터 B,C,S의 계수들을 순간 순간마다 갱신해야 하는데 이러한 구조는 3개의 곱셈지연과 $\lfloor \log_2 t \rfloor + 2$개의 유한체 가산 지연이 생기게 된다.[5] 따라서, 1 심볼 클럭안에 레지스터의 갱신이 어렵다는 단점이 있다. 본 논문에서는 수정 유클리드 알고리듬을 사용하여 이러한 문제점을 해결 하였으며 기존의 제안 되었던 수정 유클리드 연산부를 개선 함으로써 전체 복호기의 성능을 향상시켰다. 이의 기술을 위해 2장에서는 각 연산부에 대한 알고리듬 및 하드웨어 구현에 대해 설명하고 3장에서는 구현된 복호기의 구현 및 검증에 대해 기술하고 결론을 맺기로 한다.

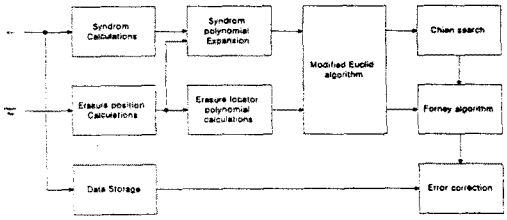

그림 1 전체 Reed-Solomon 복호기의 블럭도

2. Reed-Solomon 복호기의 구현

2.1 Syndrome 계산부

Reed-Solomon 코드의 복호 원리는 송신단에서 코드워드가 생성다항식 g(x)를 코드워드로 나눈 나머지를 잉여항으로 더해 생성되는 것을 이용하여 이루어 진다. 이러한 복호를 Syndrome 기반의 복호라 하고, 이를 위해 생성다항식 g(x)의 근을 실제 수신된 코드워드에 대입 함으써 수신된 코드워드의 이상유무를 판단하는 syndrome을 계산하게 된다. 그 원리는 실제 수신된 코드워드는 $C(x) = x^{n-k}m(x) + R(x)$와 같이 나타낼 수 있으므로, 여기에 생성다항식의 근을 대입하게 되면, $C(\alpha^i) = 0$이므로, $R(\alpha^i) = e(\alpha^i)$가 된다. 즉, 수신된 polynomial에 생성다항식의 근을 대입한 후의 값이 syndrome의 값이 되며, 이를 구현하기 위해 식 (1)과

같은 Honor's rule을 적용하여 구현하게 된다.[7]

$$s_i = [(r_{n-1}x + r_{n-2})x + \cdot \cdot + r_1]x + r_0 \qquad (1)$$

위의 수식을 보면 실제 수신되는 다항식은 코드워드의 길이 N개 만큼의 지연시간(latency)를 가지므로, 실제 클럭만 최대 N clock을 소요하게 된다. 따라서, 본 논문에서는 Honor's rule을 수정하여 병렬처리구조를 사용하였으며, 이를 위해 수정된 Honor's rule를 odd 와 even으로 나누었으며 이는 식(2)와 같고, 수정된 구조는 [그림 2]와 같다.

$$\begin{aligned} s_1 &= [((r_{n-1}x^2 + r_{n-3})x^2 + \cdot + r_3)x^2 + r_1]x \\ s_2 &= [((r_{n-2}x^2 + r_{n-4})x^2 + \cdot + r_2)x^2 + r_0 \end{aligned} \qquad (2)$$

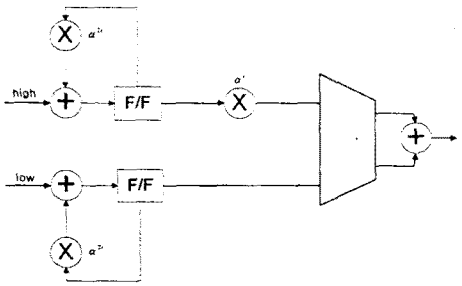

그림 2 수정된 syndrome 계산회로

2.2 수정 Euclid 연산부

Euclid 알고리듬은 key equation을 풀기 위한 방법 중의 하나로, Berlekamp 알고리듬에 비해 이해가 쉽고 하드웨어 구현이 용이하다는 장점을 가지고 있다. 이러한 유클리드 알고리듬은 두 다항식의 최대공약수를 구하는 방법으로 다음과 같은 과정을 거쳐 오류위치 다항식을 구하게 된다.[4]

step 1) 초기화 :

$$\begin{aligned} R_0 &= x^{2t},\ d(R_0) = 2t,\ d(Q_0) = 2t-1 \\ Q_0 &= S(x),\ \lambda_0(x) = 0,\ \mu_0(x) = 1 \end{aligned} \qquad (2)$$

step 2) Compute and Iteration :

$$\begin{aligned} R_i(x) &= a_{i-1}R_{i-1}(x) - b_{i-1}x^{i_{i-1}}Q_{i-1}(x) \\ \lambda_i(x) &= a_{i-1}\lambda_{i-1}(x) - b_{i-1}x^{i_{i-1}}\lambda_{i-1}(x) \\ Q_i(x) &= Q_{i-1}(x) \\ \mu_i(x) &= \mu_{i-1}(x) \end{aligned} \qquad (3)$$

step 3) Compute $R(x), \lambda(x), \mu(x)$

기존에 제안되었던 유클리드 연산부의 경우 $(N-1)^2$의 연산시간이 필요하였으며 이는 전체 복호기의 지연시간 및 동시실행성에 매우 큰 영향을 미쳤다. 따라서, 본 논문에서는 다항식의 차수를 표준화하여 제어신호를 생성하는 방법을 제안하고자 한다. 이 경우 계산된 두 다항식의 차수 만큼을 이동시키는 별도의 회로가 설계되어야 하나 전체 계산시간을 기존의 $(N-1)^2$에서 $2(N-1)$로 줄임으로써 전체 RS 복호기의 구조의 병렬처리 역시 가능하며, Shao가 제시한 겹침 동작의 해소를 위해 사용되어지는 셀의 수 역시 줄일 수 있다. 즉, [그림 3]에서 보는 바와 같이 수정 Euclid 알고리듬을 수행하기 위해 입력되는 두 다항식 $T(x)$와 $\Lambda(x)$의 계수 값들이 동시에 입력되면 입력단의 MUX에서 초기값과 함께 제어신호 발생부로 입력되고, 이 제어 신호부에서는 수정 Euclid 알고리듬의 연산을 위해 필요한 Δ, *leading coefficient*, *shift* 신호를 계산한 후 다항식 연산부로 입력되고 이 다항식 연산부에서는 주어진 다항식의 계수를 이용해 수정 Euclid 알고리듬을 수행하게 된다.

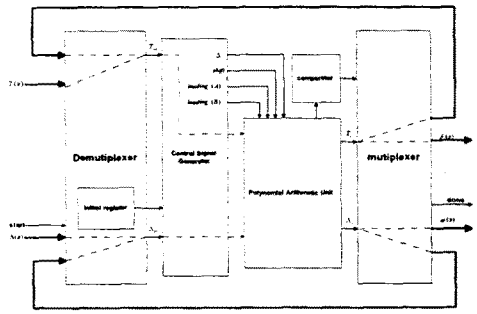

그림 3 제안된 수정 Euclid 연산부

본 논문에서는 두 다항식의 비교와 최고차항의 계수를 제어신호부에서 처리함으로 효율적인 제어신호부의 설계가 매우 중요하다. 이를 위해 본 논문에서는 [그림 4]에서와 같이 다항식의 차수를 표준화시키는 방법을 사용하며, 이를 이용해 두 다항식의 비교와 최고차항의 계수를 효율적으로 구할 수 있다. 제어 신호부는 다항식 연산부의 연산에 필요한 곱셈기의 상수 값과 두 다항식의 차수의 차를 계산한 뒤 래치에 이 제어신호를 저장한 후 입력된 다항식을 가지고 [그림 5]의 다항식 연산부에서 연산을 수행하게 된다. 따라서, 제어신호와 각 연산 값을 동기 시키기 위해 (N-1)의 지연시간과는 별도로 1 clock을 더 소모하게 된다. 하지만 코드워드의 길이 만큼 쉬프트(shift) 시키는 과정이 필요 없으므로 전체 수정 Euclid 알고리듬을 수행하는데 기존의 구조보다 $4t \cdot (t-1)$만큼의 속도 향상을 기대 할 수 있다.

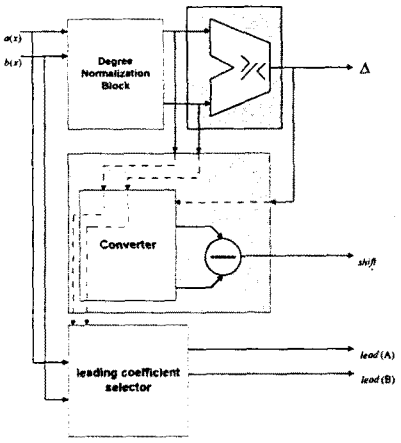

그림 4 수정 Euclid 연산부의 제어

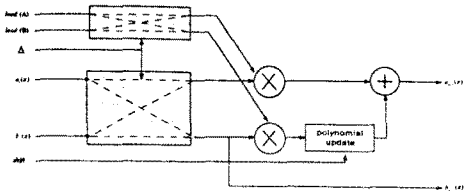

그림 5 수정 Euclid 연산부의 다항식 계산

2.3 오류 위치 및 오류 값의 계산

본 논문에서는 수정 유클리드 연산부에서 계산된 오류 위치 다항식의 근을 구하기 위해 Chien search 알고리듬을 사용하였다. 이러한 Chien search는 Error Locator polynomial과 Error magnitude polynomial을 입력받아 에러가 발생한 위치를 계산하며, Error locator polynomial과 Chien search의 근을 이용한 Forney 알고리즘을 사용하여 그 크기를 계산한다. Chien search의 경우 수정 Euclid 연산부에서 계산된 오류위치다항식이 입력되면 α^i의 값을 대입시켜 그 결과가 0 인 값을 찾으며 이때의 값이 오류위차 다항식의 근이 된다. 최종적으로 오류의 크기를 알려면 식 (4)에서와 같이 오류평가다항식의 미분 값을 알아야 하고 이를 위해 chien search 알고리듬의 수행과 함께 이의 미분값이 동시에 계산되며 chien search 값이 구해지면 그 때의 오류평가다항식의 미분 값을 가지고 오류의 크기를 계산하게 된다. 이의 블록선도는 [그림 6]과 같다.[4]

$$e_{ik} \equiv \frac{-x_k \Omega(x_k^{-1})}{\Lambda(x_k^{-1})} \qquad (4)$$

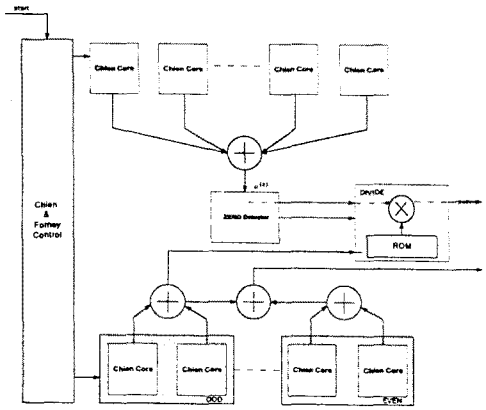

그림 6 오류크기 및 오류 위치 계산부

3. 결 론

본 논문에서는 수정 Euclid 알고리듬을 수행하는 연산부의 구조를 효율적으로 개선함으로써 RS 복호기를 구성하는 각 연산블록들간의 겹침동작을 해소할 수 있었으며, 이를 통하여 전반적인 RS 복호기의 구조를 병렬처리와 pipeline 구조를 효과적으로 적용할 수 있었다. 이의 동작 검증은 ALTERA의 MAXPlus II를 사용하여 게이트 레벨에서의 시뮬레이션을 수행하였으며, 이의 검증을 위해 FLEX 10K-50을 사용하여 동작을 검증하였다. 또한 LG 0.6 μ 공정을 이용해 Full custom으로도 제작하였다.

본 논문에서 설계된 RS 복호기의 비교를 위해 Shao가 제시한 방법에 따라 복호기의 면적을 피승수가 고정된 곱셈기의 면적(A_u)으로 표준화해 비교하였다. 이 경우 상수×상수의 곱셈기는 $2A_u$가 되고 D F/F의 경우는 A_u가 된다.[4] 비교를 위해 RS(255,223) 코드를 사용한 복호기를 기준으로 평가하였으며 이의 결과는 [표 1]과 같다. 본 논문에서 제안한 RS 복호기의 경우 Shao가 제안한 구조보다 전체 크기면에서 시간영역에서는 10%의 작은 면적으로도 약 3배의 속도 향상을 얻을 수 있었다. 이의 검증을 위해 PCB를 이용한 범용 FPGA 테스트 보드를 제작하였고 내부 테스트 벡터를 생성해 주는 부분은 수정의 용이를 위해 VHDL을 사용해 설계하였다. 사용한 테스트 벡터는 all-zero인 코드 워드가 송신된후 x^{12}, x^8, x^3에서 각각 α^4 (0011), α^3 (1000), α^7 (1011)의 오류가 발생했을 경우를 가정하였으며, 이의 로직 분석기를 이용한 검증 결과는 [그림 7]과 같다.

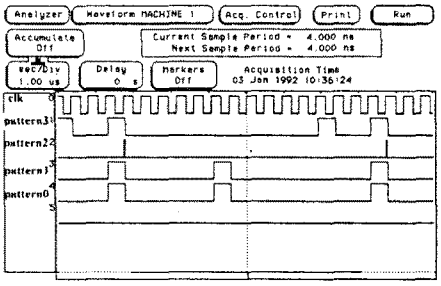

그림 7 logic analyzer로 분석한 화면

표 1 칩의 성능 비교

	Shao's Method		Proposed Method	
	area	delay	area	delay
syndrome circuit	32 A_u	255	80 A_u	128
α^k generation	8 A_u		3 A_u	
expansion circuit 1	64 A_u	255	64 A_u	256
expansion circuit 2	64 A_u		64 A_u	
modified Euclid's	200 A_u	1024	135 A_u	64
evaluation circuit 1	32 A_u	32	32 A_u	32
evaluation circuit 2	32 A_u		32 A_u	
Total	432 A_u	1,566	410 A_u	480

[참 고 문 헌]

[1] F.J MacWilliams and N.J.A. Sloane, "The Theory of Error Correcting Codes," North-Holland, 1981.

[2] 李晩榮, "BCH 부호와 Reed-Solomon 부호," 대우학

술총서 · 자연과학 65, 民音社, 3月, 1990.
[3] R. P. Brent and H. T. Kung, "Systolic VLSI arrays for polynomial GCD computations," DEP. Comput. Sci,. Carneigie-Mellon Univ., Pittsburgh. PA Rep., 1982
[4] H. M. Shao and et. al., " A VLSI design of a pipeline Reed-Solomon Decoder," IEEE Trans. Comput., vol. C-34. pp393-403, May 1985
[5] Kuang Yung Liu, "Architecture for VLSI Design of Reed-Solomon Decoders," IEEE Trans. on computers, Vol . C-33, No 2, pp 178-189, Feb.1984
[6] R. T. Chien, "Cyclic decoding procedures for the Bose-Chaudhuri-Hocquenghem codes," IEEE Trans. Inf. Theory, Vol. IT-10, pp 357-363, Oct. 1964.
[7] GoangSeog Choi, HoonSoon Choi, YoungHwan Kim, " RS Decoder using Modified Euclidean Algorithm for DVD/CD," Proceedings of the ISCPAT(The international Conference on Signal Processing Applications & Technology- Volume 1 , 9/14/97