

논문 2005-42SD-7-5

파이프라인 재귀적인 기술을 이용한 면적 효율적인 Reed-Solomon 복호기의 설계

(Design of an Area-Efficient Reed-Solomon Decoder using Pipelined Recursive Technique)

이 한 호*

(Hanho Lee)

요 약

본 논문은 무선 및 초고속 광통신등 다양한 통신 시스템에서 사용되는 고속 Reed-Solomon (RS) 복호기의 하드웨어 면적을 줄인 새로운 구조를 소개한다. 특히 folding 기술을 이용하여 높은 처리율(throughput)과 적은 하드웨어 복잡도(hardware complexity)를 가지고 있는 새로운 PrME (Pipelined recursive Modified Euclidean) 구조를 제안한다. 제안된 PrME 구조는 일반적으로 사용되는 systolic-array 그리고 완전한 병렬(fully-parallel) 구조와 비교하여 하드웨어 복잡도를 약 80%정도 줄일 수 있다. 제안된 RS 복호기는 1.2 V의 공급전압과 0.13- μ m CMOS 기술을 사용하여 설계하고 구현하였는데, 총 24,600개의 게이트 수, 5-Gbit/s의 데이터 처리율과 클럭 주파수 625 MHz에서 동작함을 보여준다. 제안된 면적 효율적인 PrME구조에 기반한 RS 복호기는 초고속 광통신뿐만 아니라 무선통신을 위한 차세대 FEC구조 등에 바로 적용될 수 있을 것이다.

Abstract

This paper presents an area-efficient architecture to implement the high-speed Reed-Solomon(RS) decoder, which is used in a variety of communication systems such as wireless and very high-speed optical communications. We present the new pipelined-recursive Modified Euclidean(PrME) architecture to achieve high-throughput rate and reducing hardware-complexity using folding technique. The proposed pipelined recursive architecture can reduce the hardware complexity about 80% compared to the conventional systolic-array and fully-parallel architecture. The proposed RS decoder has been designed and implemented with the 0.13- μ m CMOS technology in a supply voltage of 1.2 V. The result show that total number of gate is 393 K and it has a data processing rate of 5 Gbits/s at clock frequency of 625 MHz. The proposed area-efficient architecture can be readily applied to the next generation FEC devices for high-speed optical communications as well as wireless communications.

Keywords: error correction, area-efficient, Reed-Solomon coding, pipelined, recursive.

I. 서 론

리드솔로몬(Reed-Solomon (RS)) 코드는 마그네틱,

정회원, 인하대학교 정보통신공학부
(School of Information & Communication
Engineering, Inha University)

* 본 연구는 대학 IT연구센터(인하 UWB-ITRC)육성
지원사업의 연구결과로 수행되었음.

접수일자: 2005년3월16일, 수정완료일: 2005년7월4일

광 저장매체, 유선 및 위성 통신 등 다양한 응용분야에 널리 쓰이는 Forward Error Correction (FEC) 기술이다. 8 바이트 오류정정(error correction) RS(255,239) 코드는 해저 광섬유 시스템을 위해 국제통신연합(ITU)에 의해 채택되었다.^[1] 현재 가장 일반적으로 사용되는 RS 복호기(decoder) 구조는 오류 't'를 감지하고 정정하는 세 개의 주요한 부분으로 구성되어있다. 첫 번째 부분은 Syndrome Computation(SC)블록이다. SC블록에서

는 신드롬 다항식 (syndrome polynomial) $S(x)$ 를 발생시키고, 수신된 코드워드(code word)의 오류패턴을 표현한다. 다항식 $S(x)$ 는 RS 복호기의 두 번째 부분인 Key-Equation Solver(KES)블록에서 사용된다. KES블록에서는 키 등식 (key equation) $S(x)\sigma(x) = w(x) \bmod x^{2t}$ 을 해결하기 위해 Euclidean 알고리즘 (EA), modified Euclidean 알고리즘(MEA), 또는 Berlekamp-Massey 알고리즘(BMA) 등이 오류위치 다항식(error-locator polynomial) $\sigma(x)$ 와 오류값 다항식 (error-value polynomial) $w(x)$ 을 위하여 사용될 수 있다.^[2] $\sigma(x)$ 와 $w(x)$ 의 두 다항식은 Chien Search 그리고 Forney 알고리즘을 이용하여 오류의 위치에 대응하는 오류들의 크기값을 구하기 위하여 사용된다. 이 블록의 출력은 복호기로부터 읽혀져 나온 오류 정정되어 수신된 코드워드이다. 추가로 복호기가 오류를 감지하고 정정하는 과정을 실행하는 동안 FIFO 메모리가 버퍼(buffer)로 사용된다. FIFO 메모리의 깊이(depth)는 복호기의 총 지연성(latency)과 관련이 있다.

광통신 네트워크 시스템 구축을 위한 초고속 데이터 전송 기술은 높은 데이터율을 얻기 위한 요구와 맞물려 초고속 FEC 구조의 구현을 필요로 하게 되었다. Dense Wavelength Division Multiplexing (DWDM)의 출현과 함께 광전송 시스템은 지난 십년간 급속도로 발전되어 왔으며 8바이트 오류정정능력에 기인한 RS(255,239)코드가 고속(40-Gbits/s 이상) 광전송 시스템에 일반적으로 사용되고 있다. 그러나 광전송 시스템이 급속도로 발전함에 따라 40-Gbits/s 이상의 고속 데이터 전송을 필요로 하게 되었고, 이에 따라 하드웨어 복잡도와 전력소모가 매우 큰 현존하는 대부분의 RS복호기는 시스템 레벨 통합의 어려움을 가져왔다.

본 논문에서는 그림 1에서 보여지는바와 같이 하드웨어 복잡도와 클럭 주파수가 상당히 향상된 Pipelined recursive ME(PrME)구조에 기인한 면적 효율성과 고속처리를 위한 RS(255,239) 복호기의 종합적인 구조를 제시한다.

II. Reed-Solomon 복호기 구조

1. Syndrome Computation 블록

$C(x)$ 와 $R(x)$ 를 각각 코드워드 다항식 그리고 수신된 다항식이라고 두자. 전송된 다항식은 전송도중 채

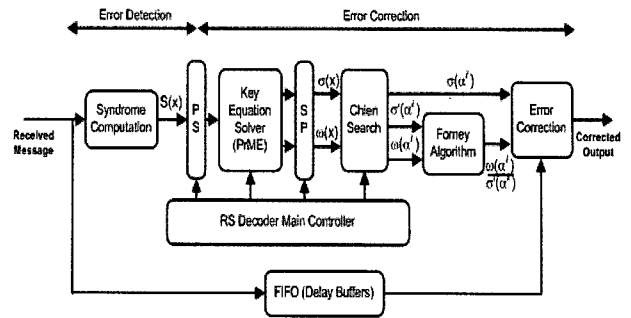


그림 1. PrME구조를 이용한 RS 복호기
Fig. 1. Reed-Solomon decoder using PrME architecture.

널 잡음(channel noise)에 의해 손상되어질 수 있다. 그러므로 수신된 다항식은 다음과 같이 표현되어진다.

$$R(x) = C(x) + E(x) = R_{n-1}x^{n-1} + \dots + R_1x + R_0 \quad (1)$$

이 식에서 $E(x)$ 는 오류 다항식이다.

복호 (decoding) 알고리즘의 첫 번째 단계는 정정 가능한 오류들을 정정할 수 있는 $2t$ syndrome S_i , ($0 \leq i \leq 2t - 1$)를 계산하는 것이다. (t 는 RS 코드에 의해 정정될 수 있는 최대 개수의 오류) 만약 모든 $2t$ syndrome S_i , ($0 \leq i \leq 2t - 1$)가 0이라는 것은 오류가 발생하지 않았다는 것이며 수신된 다항식 $R(x)$ 가 $C(x)$ 로서 유효한 코드워드라는 것을 의미한다. 신드롬 다항식 $S(x)$ 는 다음과 같이 정의될 수 있다.

$$S(x) = S_0 + S_1x + \dots + S_{2t-1}x^{2t-1} = \sum_{i=0}^{2t-1} S_i x^i, \\ S_i = \sum_{j=0}^{n-1} r_j \alpha^{ij} \quad (2)$$

여기서 α 는 원시 다항식(primitive polynomial) $p(x) = x^8 + x^4 + x^3 + x^2 + 1$, $t=8$, 의 근(root)이고 $GF(2^8)$ 에서의 원시 원소(primitive element)이다. RS(255,239)코드에서 α^i ($0 \leq i \leq 254$)는 가능한 오류위치를 의미한다.

그림 2에 보여진 syndrome computation(SC)블록은 잡음이 있는 채널을 통해 전송된 심벌(Symbol)들을 입력으로 받아들이고, 이 심벌 값들을 다항식 계수(coefficient)로 간주한다. 데이터 블록에 포함된 심벌들이 선택된 RS 코드의 데이터 블록에 대하여 유효한 코

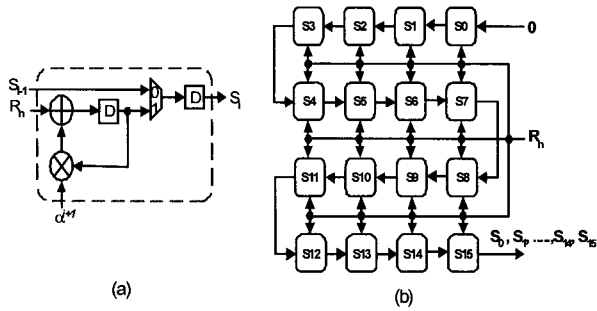


그림 2. (a)신드롬 셀 S_i , (b) 신드롬 연산 블록
 Fig. 2. (a) Syndrome cell S_i ,
 (b) Syndrome computation block.

드워드를 형성하고 있는지 결정한다. 이것은 $2t$ 신드롬 값들에 대한 다항식을 평가해서 그 값이 0인지 아닌지 감지한다. (즉 감지된 값이 0이면 데이터 블록이 코드워드이고 0이 아니면 코드워드가 아니다). 코드워드가 아닌 블록은 채널잡음에 의하여 오류가 발생한 것이다. 그림 2(a)에서 보여 지는 바와 같이 한 부분의 신드롬은 각 사이클마다 α^i 와 곱해지고 수신된 심벌과 누적된다. 그림 2(b)는 16개의 신드롬 셀(syndrome cell)들로 구성된 SC블록을 보여준다. 이 블록은 n 개의 심벌 구간 안에서 신드롬들이 계산되어질 수 있도록 한다. 신드롬 심벌들 S_0, S_1, \dots, S_{15} 는 직렬로 KES 블록으로 출력된다.

2. Key Equation Solver 블록

신드롬 다항식 $S(x)$ 는 KES블록에서 키등식 $S(x)\sigma(x) = \omega(x) \text{ mod } x^{2t}$ 를 계산하기 위하여 사용된다. 이 등식을 풀기 위해서 오류위치 다항식 $\sigma(x) = \sigma_t x^t + \sigma_{t-1} x^{t-1} + \dots + \sigma_0$ 와 오류값 다항식 $w(x) = w_{t-1} x^{t-1} + \dots + w_1 x + w_0$ 를 계산한다.

RS 복호에서의 KES 블록은 EA, ME 또는 BM 알고리즘들을 이용하여 구현할 수 있으며, division-free ME 그리고 고속 ME 구조들이 각각 제안되었다.^{[3][5]}

일반적인 ME 구조는 $2t$ (정정 가능한 최대오류의 2 배수)의 Processing Elements(PEs)로 구성되어 있고 systolic-array 구조로 연결되어 있다. 일반적인 systolic-array ME구조의 하드웨어 크기는 총 RS 복호기 크기의 약 60%를 차지한다.^{[3][5]} 그러므로 RS 복호기 설계에 있어서 중요한 도전은 critical path delay와 총

전력소모를 줄이기 위하여 ME 구조의 하드웨어 복잡도를 최소화하는 것이다. 작은 하드웨어 면적을 가지고 있는 KES블록을 설계하기 위하여, folding 기술을 이용한 PrME구조를 이용하여 하드웨어 복잡도를 줄이고 클락 주파수를 향상시킬 수 있다. 제안한 PrME 구조는 III장에서 자세하게 설명하고자 한다.

3. Chien Search, Forney 알고리즘 및 오류정정 블록

ME알고리즘이 수행된 후, 오류위치 다항식과 오류값 다항식이 double buffered 직렬-병렬(serial-to-parallel) converter로 전송된다. 이 값들은 Chien Search 알고리즘 블록에 전달되고, 여기서 오류 위치 다항식의 근이 계산된다. Forney 알고리즘 블록과 Chien Search 알고리즘은 병렬로 동작하고 각각의 오류 위치에 대응하는 오류들의 크기를 계산한다. 복호 과정의 마지막 단계는 오류들을 정정하기 위하여 이진 XOR 연산을 통해 FIFO buffered 입력 코드워드에 오류 값들을 합하는 것이다. $GF(2^m)$ 에 대한 degree t 차 오류위치 다항식이 $\sigma(x) = x^t + \sigma_{t-1} x^{t-1} + \dots + \sigma_0, \sigma_i \in GF(2^m), 0 \leq i \leq t-1$ 에 의하여 정의된다. 그러면 이런 다항

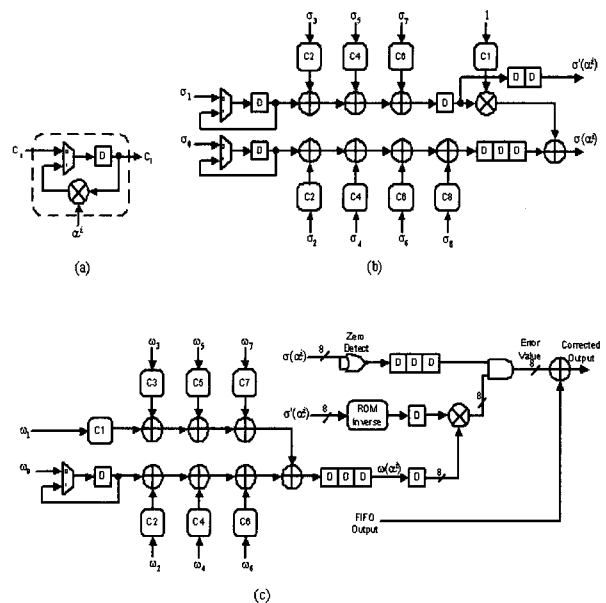


그림 3. (a) Chien search 셀 C_i ,
 (b) Chien search 블록,
 (c) Forney 알고리즘 및 에러 정정 블록

Fig. 3. (a) Chien search cell C_i ,
 (b) Chien search block,
 (c) Forney algorithm and error correction block.

식의 근을 구하는 것은 RS 복호기의 광범위한 연산에 의하여 이루어진다. Chien Search 알고리즘은 $GF(2^m)$ 상의 t 차 오류위치 다항식(t 는 정정 가능한 최대오류의 2배수)의 근을 구하는데 사용될 수 있다. 그러나 Chien Search 알고리즘은 각각 계수 σ_i 와 α 의 거듭제곱과의 곱셈연산을 필요로 한다(α 는 더 이상 약분될 수 없는 $GF(2)$ 상의 t 차 다항식의 근). 오류위치와 오류값을 계산하기 위한 Chien Search 알고리즘과 Forney 알고리즘에 대한 설명은 논문^[5]에 잘 설명되어 있다. 그림 3(b)는 8개의 Chien Search 셀로 구성된 Chien Search 블록의 블록도를 보여주고 있다. Finite-field 덧셈기(adder)는 그림 3(b)에서 보여 지는바와 같이 두개의 Chien Search 셀의 합의 결과를 다음번 덧셈기로 보낸다.

그림 3(c)는 Forney 알고리즘과 오류정정 블록을 보여주고 있고 여기서 오류값을 구한 뒤 심벌들을 정정한다. Galois-field의 나눗셈에 대해서는 제수(divisor)의 역원(inverse element)이 유도된 후 파이프라인 fully-parallel 곱셈기에 의하여 피제수(dividend)의 원소와 곱해진다. $GF(2^8)$ 에서의 non-zero 원소의 역계산을 위한 직접적인 방법은 field 원소들의 역수를 저장할 수 있는 8비트의 255워드들로 이루어진 간단한 look-up 테이블을 사용하는 것이다. 결과적으로 look-up 테이블은 static ROM에 의하여 실현되고, 파이프라인된 곱셈기보다도 더 적은 path delay를 가지고 있다.

4. FIFO 메모리 Buffer 및 Control Logic

각각의 오류 값들이 계산될 때, 이에 대응되는 수신된 심벌이 FIFO메모리로부터 호출되며, FIFO 메모리는 복호 과정동안의 완충역할을 한다. 각각의 오류 값은 정정된 심벌을 만들기 위해 단순히 수신된 심벌에 합해진다. 오류가 발생하지 않은 위치에서 오류 값들은 0이 되므로 합해져도 이 위치에서 수신된 다항식은 바뀌지 않는다.

RS 복호기로 수신된 데이터들은 연속적으로 들어오므로 제어기(controller)는 각 복호 과정을 위한 제어신호를 발생시키는 것이 필요하다. 제어기 시스템의 설계는 주제어기(master controller)로 전달되기 위한 특수한 신호 변경 규약을 갖는 각각의 부제어기(local slave controller)를 구현함으로써 이루어진다.

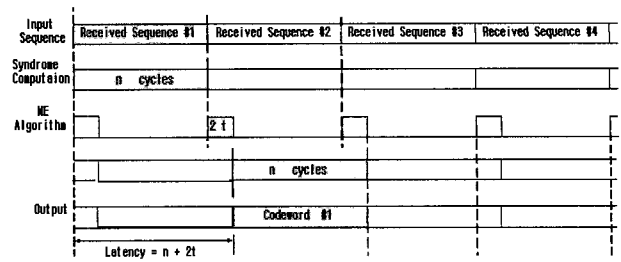


그림 4. Systolic-array ME 구조를 사용한 RS 복호기의 타이밍도

Fig4. 4. Timing chart of RS decoder using the systolic-array ME architecture.

III. Pipelined Recursive Modified Euclidean 구조

이번 장에서는 면적 효율성 및 고속 처리 KES블록의 구현을 위한 folding 방법을 이용한 pipelined recursive modified Euclidean(PrME)구조가 소개된다. 기존의 systolic-array ME 구조는 $2t$ 의 처리요소들(PEs)로 구성되어 있고 systolic-array 구조로 연결되어 있다.^{[3]-[5]} 이와 같은 $2t$ PEs의 systolic-array 구조는 오류위치 다항식과 오류 값 다항식을 계산하고, ME 알고리즘을 연속적으로 실행한다.

ME알고리즘의 systolic-array 구조는 $2t$ 주기의 지연성을 가지고 있다. 그림 4는 systolic-array ME 구조를 가지고 있는 RS 복호기의 타이밍도(timing chart)를 보여주고 있다. 그러나 이 방법은 고속연산에서 매우 큰 하드웨어 비용을 요구하므로 향후 공간 비용(space cost)을 대신한 시간주기(time cycle)를 이용한 면적 효율성 구조를 제안한다. systolic-array ME 구조에서 한 개의 PE를 규칙적으로 16번 사용하면 이상적으로 중첩된 키 등식을 해결할 수 있는데 이 구조는 단지 1/16의 하드웨어 비용만을 필요로 한다.

1. Modified Euclidean Algorithm

ME알고리즘은 키등식 $S(x)\sigma(x) = \omega(x) \text{mod} x^{2t}$ 을 계산함으로써 오류위치 다항식 $\sigma(x)$ 와 오류값 다항식 $\omega(x)$ 을 구하기 위하여 사용된다. 알고리즘을 요약하면 다음과 같다.

Input: $S(x), x^{2t}$

Initialization:

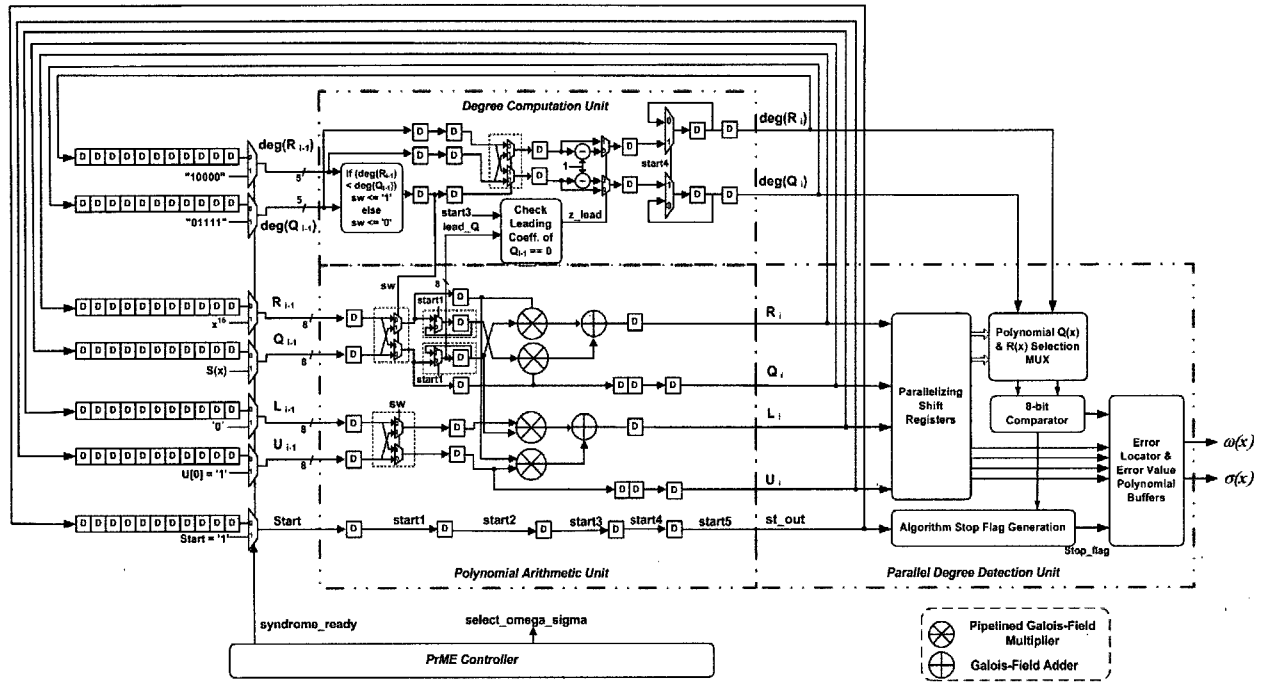


그림 5. 파이프라인 재귀적인 modified Euclidean (PrME) architecture.

Fig. 5. Pipelined recursive modified Euclidean (PrME) architecture.

$$R_0(x) = x^{2t}, Q_0(x) = S(x), L_0(x) = 0, U_0(x) = 1;$$

$$\deg(R_0(x)) = 2t, \deg(Q_0(x)) = 2t - 1$$

$$l_0 = \deg(R_0(x)) - \deg(Q_0(x));$$

Index 'i' is initialized to 0;

Index 'Step' is initialized to 1;

Start Algorithm:

while (Step \leq 2t) **do**

begin

$$\text{Step} \leftarrow \text{Step} + 1$$

$$i \leftarrow i + 1;$$

$$a_{i-1} \leftarrow \text{leading coefficient of } R_{i-1}(x)$$

$$b_{i-1} \leftarrow \text{leading coefficient of } Q_{i-1}(x)$$

if ($\deg(R_i(x)) < i$)

begin

$$R_i(x) = R_i(x);$$

$$Q_i(x) = Q_i(x);$$

$$L_i(x) = L_i(x);$$

$$U_i(x) = U_i(x);$$

Skip the following statements & stop the algorithm.

end

if ($l_{i-1} \leq 0$)

begin

$$R_i(x) = [b_{i-1} R_{i-1}(x)] x^{l_{i-1}} [a_{i-1} Q_{i-1}(x)]; \quad (1a)$$

$$Q_i(x) = Q_{i-1}(x); \quad (2a)$$

$$L_i(x) = [b_{i-1} L_{i-1}(x)] x^{l_{i-1}} [a_{i-1} U_{i-1}(x)]; \quad (3a)$$

$$U_i(x) = U_{i-1}(x); \quad (4a)$$

end

else

begin

$$R_i(x) = [a_{i-1} Q_{i-1}(x)] x^{l_{i-1}} [b_{i-1} R_{i-1}(x)]; \quad (1b)$$

$$Q_i(x) = R_{i-1}(x); \quad (2b)$$

$$L_i(x) = [a_{i-1} U_{i-1}(x)] x^{l_{i-1}} [b_{i-1} L_{i-1}(x)]; \quad (3b)$$

$$U_i(x) = L_{i-1}(x); \quad (4b)$$

end

$$l_{i-1} \leftarrow \deg(R_{i-1}(x)) - \deg(Q_{i-1}(x)); \quad (5)$$

end

Output: $\sigma(x), w(x)$

i 번째 반복에서 a_{i-1} 과 b_{i-1} 는 $R_{i-1}(x)$ 와 $Q_{i-1}(x)$ 의 각각의 계수가 되고, 알고리즘은 $\deg(R_i(x)) < t$, ($\deg(\cdot)$ 은 다항식의 차수)일 때 멈추게 된다.

2. Pipelined Recursive Modified Euclidean 구조

Finite-field 원소의 곱셈연산은 RS 복호기의 VLSI 구현에서 매우 중요한 역할을 한다. 칩의 복잡도와 연산 시간은 finite-field 곱셈기를 어떻게 구현하는가에 많이 의존한다. PrME 구조의 구현에 쓰이는 $GF(2^8)$ 상의 파이프라인 fully-parallel 곱셈기에 대한 설명은 논문 [5]에 잘 설명되어 있으며, 이 곱셈기 구조는 critical path delay의 상당한 감소를 제공한다.

ME 알고리즘에서 하나의 신드롬 다항식이 하나의 코드워드만큼의 시간차를 가지고 계산되며, 그 결과 상당한 부분의 systolic-array가 유향 상태에 있게 된다. 이러한 사실은 데이터 처리율의 감소 없이 좀 더 효율적인 설계가 가능하다는 것을 의미한다. 그림 5는 제안된 PrME 구조의 블록도를 보여주고 있으며, pipelined Degree Computation (DC) unit, Polynomial Arithmetic (PA) unit, Parallel Degree Detection (PDD) unit 그리고 Shift-Registers(SRs)들이 재귀적인 반복(recursive loop)으로 연결되어 있다. 이러한 PrME구조는 오류위치 다항식과 오류값 다항식을 계산한다.

가. Degree Computation

DC unit은 다음의 두 가지 주요 기능을 수행한다. 첫 번째는 5비트 비교기(comparator)를 이용하여 $R_{i-1}(x)$, $Q_{i-1}(x)$ 두 다항식의 차수를 비교하는 것이다. 여기서 등식 1(a,b)와 2(a,b)에서의 $R_i(x)$ 와 $Q_i(x)$, 그리고 등식3(a,b)와 4(a,b)에서의 $L_i(x)$ 와 $R_i(x)$ 의 다항식들이 교환될 필요가 있는지를 결정한다. 그래서 제어회로(control circuit)는 등식(5)에서처럼 l_{i-1} 를 계산한다. 만약 $\deg(R_i(x)) < \deg(Q_i(x))$ 이면 신호 "sw"는 1 (high)이 되고, 그렇지 않으면 0 (low)가 된다. DC unit에서의 두 번째 기능은 다음번 ME 연산을 위해 다항식 $R_i(x)$ 와 $Q_i(x)$ 의 차수를 계산하는 것이다. 이 다항식 차수 값은 각 반복단계의 마지막에 등록되고 다음 반복단계까지 shift-register에 상수로 저장된다. PrME 구조에서 높게 파이프라인된 개의 PE가 재귀적으로 사용되기 때문에 두개의 연

속적인 반복 사이의 의존성을 피하기 위해서 이러한 shift-register의 사용은 매우 중요하다.

나. Polynomial Arithmetic

PA unit은 $R_{i-1}(x)$, $Q_{i-1}(x)$, $U_{i-1}(x)$, $L_{i-1}(x)$ 의 finite-field 연산을 수행하고 각 다항식의 계수를 연속으로(serial) 생성하고 PA unit에 보내진 순서대로 피드백(feed-back)되어 입력된다. 첫 번째 반복에서 병렬-직렬(parallel-serial) converter는 신드롬 블록과 PrME 구조 사이에서 신드롬 다항식을 직렬화하기 위해서 사용된다. PA unit에서 "start"신호는 다항식들의 시작을 알리기 위해서 사용된다. 다시 말하면, "start"신호는 항상 다항식 $R_i(x)$ 와 $Q_i(x)$ 의 leading 계수 a_{i-1} 와 b_{i-1} 를 정렬한다. ME알고리즘의 첫 번째 단계에서 "start"신호에 의해 각 다항식의 leading 계수들이 적절히 유도되는 것과 같은 방법으로 $xQ_0(x)$, $xU_0(x)$ 뿐만 아니라 "start"신호도 한 time unit 만큼 지연된다. 신호 "z_{lead}"는 DC unit에서 다항식 $Q_i(x)$ 의 유도된 계수가 0인지 아닌지를 나타내기 위해서 발생된다. PA unit은 finite-field 곱셈과 덧셈을 처리한다.

하나의 PA unit은 등식 (1)-(4)를 계산하기 위해서 4개의 파이프라인 Galois-field 곱셈기와 두개의 Galois-field 덧셈기 그리고 6개의 MUX를 가지고 있다. ME알고리즘에서 첫 반복단계에서 $R_0(x)$, $Q_0(x)$ 는 각각 x^{2t} , $s(x)$ 로 $L_0(x)$, $U_0(x)$ 는 각각 0과 1로 초기화 된다.

PA unit은 파이프라인 fully-parallel 곱셈기를 사용하고, 클락 주파수의 두드러진 향상을 제공하기 위해 5단의 파이프라이닝 단계(pipelining stage)를 가지고 있다. 각각의 재귀적인 반복단계를 위하여 11단의 shift-register가 각 반복단계의 출력된 값을 저장하기 위해 사용된다. 그러므로 PrME구조는 총 16단의 파이프라이닝 레지스터(register)를 PA unit안에 가지고 있다.

다. Parallel Degree Detection

DC unit에서 다항식 $R(x)$ 또는 $Q(x)$ 의 차수는 제어신호 l_1, z_{lead} 에 따라 각 반복단계마다 1씩 감소한다. 그래서 알고리즘을 멈추기 위한 조건($R(x) < t$ 또는 $Q(x) < t$)을 만족하는지를 감지하기 위해 각 반복단

제마다 stop-flag generation unit은 $R_i(x), Q_i(x)$ 의 현재 차수 값들을 요구한다. PrME구조의 각 반복 단계에서 계산되는 각 다항식 차수가 다항식의 실제 차수와 분명히 다를 수 있다. 그러므로 수신된 코드워드의 오류의 수가 t 보다 작다면 멈추기 위한 조건이 감지되기 전까지 많은 반복단계의 초과계산을 초래할 수 있다. 이러한 상태는 각 반복단계에서 다항식 $R_{i-1}(x)$ 또는 $Q_{i-1}(x)$ 의 차수가 적어도 한번은 감소하였다는 가정 하에 systolic-array ME구조로부터 이어 받은 것이다. 그러나 수신된 코드워드에서 오류의 개수가 t 보다 작다면 오류 값과 오류위치 다항식은 적은 반복단계를 거쳐서 계산될 수 있을 것이다. Systolic-array ME구조에는 $2t$ PE unit이 사용되고 있기 때문에 알고리즘을 완료하는데 $2t$ 의 주기가 필요하다. 그러나 재귀적으로 사용하는 단일 PE unit을 갖는 PrME 구조인 경우에는 알고리즘을 완료하는데 n 주기가 걸린다.

PrME구조에서 전력소모를 최소화하기 위하여 stop 조건이 만족될 때 알고리즘을 멈추고 불필요한 신호들의 toggle을 즉시 중지시킴으로써 저전력 상태로 블록을 유지시키는 것이 중요하다. 그러므로 현재 상태에서 두 다항식 $R_i(x), Q_i(x)$ 의 차수를 확인하기 위해 병렬로 비교해 보는 것이 필요하다. 이런 방법을 이용하면, 각 반복단계의 끝에서 두 다항식중 하나의 차수가 t 이하로 떨어지는지 즉 조건 $R(x) < t$ 또는 $Q(x) < t$ 를 만족하는지 감지할 수 있다. 제안된 PDD구조는 "stop" 신호를 발생시키기 위해 두 다항식 $R_i(x)$ 와 $Q_i(x)$ 의 차수를 병렬로 비교하고 감지한다. PDD unit에는 중요한 네 부분이 있다. 첫 부분에서는 shift-register를 이용하여 두 다항식이 직렬에서 병렬로 전환된다. 각 반복단계의 끝에서 $R(x), Q(x)$ 다항식의 차수를 계산한 5비트 값이 MUX의 선택을 위한 값으로 지정된다.

이 MUX들은 두 다항식 $R_i(x), Q_i(x)$ 의 계수를 정렬하는데 사용한다. $R_i(x)$ 와 $Q_i(x)$ 다항식의 차수 값 5비트 중에서 하위 4비트가 $R_i(x)$ 와 $Q_i(x)$ 다항식의 MUX를 지정하기 위한 값으로 사용된다. 일단 정렬이 되면 상위 8개의 계수들이 0이 아닌 값에 대하여 감지되고 이 값들은 서로 비교된다. 만약 두 다항식의 상위 8개의 계수들이 0이라면 하위8개의 계수들이 비교되고 "stop"신호가 발생한다. "stop" 신호는 두 번째 단계를 위한 PrME구조의 모든 레지스터의 동기 리셋(reset)신호로 사용되고, 저전력 상태의 PA와 DC unit에 입력

표 1. KES블록에서의 Critical path delay와 latency의 비교

Table 1. Comparison of critical path delay and latency for KES blocks.

Architecture	Critical path delay	Latency
Proposed PrME	$3T_{or2}+T_{xnor2}+T_{mux}+T_{ff}$	$2n+12$
Systolic ME ^[5]	$3T_{or2}+T_{xnor2}+T_{mux}+T_{ff}$	$10t$
Parallel ME ^[8]	$T_{mult}+T_{add}+T_{ff}$	$2t+2$
EA ^[6]	$T_{rom}+T_{and}+2T_{mult}+T_{act}+2T_{ms2}+T_{ff}$	$2t$
RiBM ^[7]	$T_{mult}+T_{add}+T_{ff}$	$2t$

된다.

다항식 $R_i(x)$ 또는 $Q_i(x)$ 의 차수가 t 이하이면 $R_i(x) > Q_i(x)$ 인지 조건을 검사하고 비교한다. 즉 이 경우에는 오류위치 다항식 $\sigma(x)$ 는 $Q_i(x)$ 가 되고, 오류 값 다항식 $w(x)$ 는 $R_i(x)$ 가 된다. 다시 말하면 $R_i(x) < Q_i(x)$ 이면 $\sigma(x)$ 는 $U_i(x)$, 그리고 $w(x)$ 는 $Q_i(x)$ 가 된다. 좀 더 신중한 설계는 PDD unit안의 shift-register를 중복 사용하는 대신에 PrME 구조 안에 이미 저장되어 있는 shift-register로부터의 값들을 사용하여 구현하는 것이다. 그러므로 PDD unit를 이용하여 저전력상태로 유지시켜 줌으로써 저전력소비에 있어서 더 많은 이득을 얻을 수 있다.

PrME구조에서 critical path는 DC unit내의 5비트 비교기에 있으므로 $3T_{or2} + T_{xnor2} + T_{mux2} + T_{ff}$ 로 critical path delay를 정의할 수 있다. 이 같은 설계는 일반적인 systolic-array 구조^{[3]-[5]}와 병렬 ME구조^[8]와 비교하여 훨씬 감소된 하드웨어 복잡도와 높은 클럭 주파수의 이득을 얻을 수 있다.

IV. 결과 및 비교

본 논문에서 제안된 PrME구조를 이용한 면적효율성 및 고속처리 RS복호기는 Verilog-HDL로 설계하였고, CADENCE NC-Verilog 시뮬레이터(simulator)로 검증하였다. Verilog-HDL로 설계한 RS 복호기의 결과는 C언어로 설계한 모델과 정확히 일치하였다. 이런 검증 단계 후에는 SYNOPSIS Design Compiler(DC)를 이용하여 적절한 time 및 area constraint와 1.2 V의 공급전

표 2. KES블록에서의 하드웨어 복잡도 비교

Table 2. Comparison of hardware complexity for KES blocks.

Architecture	Multipliers	Adders	D-FFs	MUXes
Proposed PrME	4	2	170	30
Systolic ME ^[5]	8t	8t	78t+4	40t+2
Parallel ME ^[8]	6t+2	3t+1	6t+4	N/A
EA ^[6]	3t+1	4t+1	14t+6	11t+4
RiBM ^[7]	6t+2	3t+1	6t+2	3t+1

표 3. RS(255,239) 복호기의 구현 결과

Table 3. Implementation result of RS(255,239) decoders.

Design	Proposed PrME	Systolic ME ^[5]	Parallel ME ^[8]	EA ^[6]
Syndrome	3,000	3,000	10,000	3,000
KES	17,000	117,500	84,000	44,700
Chien, Forney, Error	4,600	4,600	24,000	4,600
Total # of Gates	24,600	124,600	118,000	55,600
Clock Rate(MHz)	625	625	112	300
Latency (Clock cycle)	522 (n+(2t) ² +12) (0.83μs)	355 (n+12t+20) (0.57μs)	271 (n+16) (1.5μs)	287 (n+32) (0.96μs)
Throughput (Gbit/s)	5	5	2.5	2.4
Efficiency	203.25	40.13	21.19	43.17
Tech.(μm)	0.13	0.13	0.25	0.13

압과 0.13-μm CMOS 기술을 이용하여 합성(synthesize)하고 구현하였다.

표 1에서는 다양한 KES블록의 critical path delay와 지연성(latency)을 비교한 결과를 보여주고 있다. 제안된 PrME구조는 이전의 systolic-array ME구조와 비교하여서는 거의 비슷한 critical path delay를 보여주고 있고, Euclidean과 BM구조 보다는 훨씬 감소한 critical path delay를 보여주고 있다.

표 2에서는 다양한 KES블록의 하드웨어 복잡도를 보여주고 있다. 일반적으로 사용되는 KES블록들과 비교한 결과 본 논문에서 제안한 PrME구조는 단지 4개의 finite-field 곱셈기와 2개의 finite-field 덧셈기, 170개의 D-FF만이 필요하다는 것을 보여주고 있다. 결론

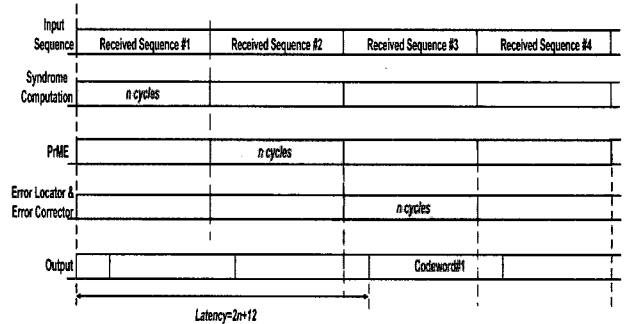


그림 6. PrME 구조를 사용한 RS 복호기의 타이밍도

Fig. 6. Timing chart of RS decoder using the PrME architecture.

적으로 일반적인 ME구조^{[5][8]}, Euclidean^[6] 그리고 BM 구조 보다는 현저히 감소한 하드웨어 복잡도를 보여주고 있다.

표 3은 몇 종류의 RS 복호기의 게이트 수, clock rate, latency, throughput들을 비교한 결과들을 보여주고 있다.

RS 복호기에서 FIFO 메모리를 제외한 하드웨어 복잡도를 비교한 결과 본 논문에서 제안한 RS 복호기는 이전의 systolic-array ME^[5] 및 Euclidean^[6] 구조와 비교하여 각각 20%, 44%의 게이트 수만 필요로 한다는 것을 보여주고 있다. 또한 parallel ME^[8] 구조와 비교하면 제안된 RS 복호기는 20%의 게이트 수를 필요로 한다. 본 논문에서 제안된 RS 복호기는 625 MHz에서 동작하고 0.83 μs의 latency, 5-GBits/s의 처리속도를 갖는다. 이 비교표들로부터 본 논문에서 제안된 RS 복호기가 최근 발표된 복호기들보다 면적과 속도를 고려한 효율성면에서 4-5배정도 우수하다 것을 볼 수 있다.

그림 6은 PrME구조를 이용한 RS 복호기의 타이밍도를 보여주고 있다. Syndrome Computation(SC)블록은 신드롬 다항식을 계산하기 위하여 n클럭 주기만큼의 처리 지연 후 2t 신드롬들을 출력한다. PrME구조는 각 반복단계마다 신드롬들을 받아들이고, 출력을 피드백(feed-back)한다. n주기 후에, PrME구조는 다항식 σ(x)와 w(x)을 출력하고 Chien Search블록으로 병렬로 입력된다. 본 논문에서 제안된 RS 복호기는 코드블록을 연속적으로 생성하는데, 고정된 지연 2n+12 클럭 주기를 가지고 적절히 연산하고 결과를 출력한다.

V. 결 론

본 논문에서는 면적효율성 및 고속처리를 위하여 folding 방법을 이용한 면적효율성 PrME구조를 제안하고 RS 복호기 설계에 적용하였다. 파이프라인 재귀적인 (pipelined recursive) 구조는 단 하나의 처리요소 (processing element)를 가짐으로써 면적효율적인 PrME 구조의 구현을 가능하게 하였다. 제안된 PrME구조는 일반적인 systolic-array 및 fully-parallel 구조와 비교하여 약 80%정도 하드웨어 복잡도를 줄일 수 있다. 제안된 RS 복호기는 총 24,600개의 게이트 수, 5-Gbit/s의 데이터 처리율과 클럭 주파수 625 MHz에서 동작하는 결과를 보여주고 있다. 결론으로써 본 논문에서 제안한 RS 복호기는 현재까지 발표된 복호기중 가장 높은 면적 효율성을 가지고 있는 것 중에 하나이며, 초고속 광통신뿐 아니라 무선통신장비를 위한 차세대 FEC 장치 등에 바로 적용할 수 있다.

참 고 문 헌

- [1] "Forward Error Correction for Submarine System" Telecommunication Standardization Section, International Telecom. Union, ITU-T Recommendation G.975, Oct. 2000.
- [2] S. B. Wicker, "Error Control Systems for Digital Communication and Storage," Prentice Hall, 1995.
- [3] H. M. Shao, T. K. Truong, L. J. Deutsch, J. H. Yuen and I. S. Reed, "A VLSI Design of Pipeline Reed-Solomon Decoder," IEEE Trans. on Computers, Vol. C-34, No.5, pp.393-403, May. 1985.
- [4] W. Wilhelm, "A New Scalable VLSI Architecture for Reed-Solomon Decoders" IEEE Jour. of Solid-state Circuits, Vol34, No.3, Mar. 1999.
- [5] H. Lee, "High-Speed VLSI Architecture for Parallel Reed-Solomon Decoder," IEEE Trans. on VLSI Systems, Vol. 11, No. 2, pp. 288-294, April. 2003.
- [6] H. Lee, "An Area-Efficient Euclidean Algorithm Block for Reed-Solomon Decoder," IEEE computer society Annual Symposium on VLSI, pp. 209-210, Feb. 2003.
- [7] D. V. Sarwate and N. R. Shanbhag, "High-Speed Architecture for Reed-Solomon Decoders," IEEE Trans. on VLSI Systems, Vol 9, No.5, pp.641-655, Oct. 2001.
- [8] L. Song, M-L. Yu and M. S. Shaffer, "10 and 40-Gb/s Forward Error Correction Devices for Optical Communications," IEEE Journal of Solid-State Circuits, Vol. 37, No. 11, pp. 1565-1573, Nov. 2002.

— 저 자 소 개 —



이 한 호(정회원)

1993년 충북대학교 전자공학과 학사 졸업.

1996년 Univ. of Minnesota 전기컴퓨터공학 석사 졸업.

2000년 Univ. of Minnesota 전기컴퓨터공학 박사 졸업.

<주관심분야 : 통신용 VLSI설계, SoC설계>