

Reed-Solomon 복호기를 위한 새로운 E-DCME 알고리즘 및 하드웨어 구조

정회원 백재현*, 선우명훈*

New Enhanced Degree Computationless Modified Euclid's Algorithm and its Architecture for Reed-Solomon decoders

Jae-hyun Baek*, Myung-hoon-Sunwoo* *Regular Members*

요 약

본 논문에서는 리드-솔로몬(Reed-Solomon) 복호기를 위한 새로운 E-DCME (enhanced degree computationless modified Euclid's) 알고리즘 및 하드웨어 구조를 제안한다. 제안하는 E-DCME 알고리즘은 새로운 초기 조건을 사용하여 기존 수정 유클리드 알고리즘 및 DCME 알고리즘에 비해 $T_{mult} + T_{add} + T_{mux}$ 의 짧은 최대 전달 지연 (critical path delay)을 갖는다. 시스톨릭 에레이 (systolic array)를 이용한 제안하는 구조는 키 방정식 (key equation) 연산을 위해서 초기 지연 없이 $2t - 1$ 클럭 사이클만을 필요로 하여 고속의 키 방정식 연산이 가능하다. 또한, 기존 DCME 알고리즘에 비해 사용하는 기본 셀의 개수가 적어 하드웨어 복잡도가 낮다. 전체 $3t$ 개의 기본 셀 (basic cell)을 사용하는 E-DCME 구조는 오직 하나의 PE (processing element)를 사용하므로 규칙성 (regularity) 및 비례성 (scalability)을 갖는다. $0.18\mu\text{m}$ 삼성 라이브러리를 사용하여 논리합성을 수행한 결과 E-DCME 구조는 18,000개의 게이트로 구성된다.

Key Words : Reed-Solomon, Key equation, Modified Euclid's algorithm, 오류 정정 코드

ABSTRACT

This paper proposes an enhanced degree computationless modified Euclid's (E-DCME) algorithm and its architecture for Reed-Solomon decoders. The proposed E-DCME algorithm has shorter critical path delay that is $T_{mult} + T_{add} + T_{mux}$ compared with the existing modified Euclid's algorithm and the degree computationless modified Euclid's (DCME) algorithm since it uses new initial conditions. The proposed E-DCME architecture employing a systolic array requires only $2t - 1$ clock cycles to solve the key equation without initial latency. In addition, the E-DCME architecture consisting of $3t$ basic cells has regularity and scalability since it uses only one processing element. The E-DCME architecture using the $0.18\mu\text{m}$ Samsung standard cell library consists of 18,000 gates.

I. 서 론

컨볼루션 (convolutional) 부호와 블록 (block) 부호로 구분되는 오류 정정 부호는 전송된 데이터의 오류 정정을 위해서 통신 시스템 및 디지털 저장

장치에 폭넓게 사용된다^[1]. 리드-솔로몬 (Reed-Solomon) 부호는 랜덤 오류뿐만 아니라 연접 오류에 우수한 정정 능력을 갖고 있다. 일반적인 (n, k, t) 리드-솔로몬 부호에서 k 는 m 비트 정보 심벌의 개수를 나타내며 리드-솔로몬 부호화 후 $n = 2^m - 1$ 개의 심

* 본 연구는 교육인적 자원부 2단계 BK(Brain Korea) 21 과제 및 정보통신부 21st Century Frontier R&D Program의 ubiquitous Computing and Network (UCN) 과제의 지원을 받아 수행되었습니다.

* 아주대학교 전자공학부 SOC 연구실(koguryo@ajou.ac.kr, sunwoo@ajou.ac.kr)

논문번호 : KICS2006-12-538, 접수일자 : 2006년 12월 20일, 최종논문접수일자 : 2007년 7월 21일

벌을 갖는다. $t (= \lfloor (n - k) / 2 \rfloor)$ 는 리드-솔로몬 부호의 오류 정정 능력을 나타낸다. 리드-솔로몬 복호기는 신드롬 (syndrome) 연산, 키 방정식 연산, Chien search 알고리즘, Forney 알고리즘, 오류 정정의 5개 블록으로 구성되며, 이들 블록 중 오류 위치 다항식 (error locator polynomial) 및 오류 크기 다항식 (error value polynomial) 연산을 위한 키 방정식 블록은 가장 큰 하드웨어 복잡도 및 연산 시간을 필요로 한다. 베르캄프-메세이 (Berlekamp-Massey) 알고리즘,^{[1][2][3]} 유클리드 (Euclid) 알고리즘,^{[4][5]} 수정 유클리드 (modified Euclid) 알고리즘^{[6][7][8][9]}가 키 방정식 연산을 위해 사용된다.

베르캄프-메세이 (Berlekamp-Massey) 알고리즘을 기반으로 하는 RiBM (reformulated inversionless Berlekamp-Massey) 구조^[3]은 베르캄프-메세이 알고리즘을 사용하는 기존 키 방정식 연산 블록에 비해 최대 전달 지연 경로가 짧고 하드웨어 복잡도가 낮은 우수한 구조이며, 키 방정식 연산을 위해 $2t$ 클럭 사이클을 필요로 한다. 결과적으로 RiBM 알고리즘^[3]은 키 방정식 연산을 위해 수정 유클리드 알고리즘을 사용하는 키 방정식 연산 블록과 같은 클럭 지연을 갖지만, 짧은 최대 전달 지연 경로를 갖고 있어 고속의 키 방정식 연산이 가능하다.

본 저자에 의해 제안된 수정 유클리드 알고리즘을 기반으로 하는 DCME (degree computationless modified Euclid's) 알고리즘^{[9][10]}은 기존 수정 유클리드 알고리즘과는 달리 다항식의 차수 연산 및 비교 연산을 필요로 하지 않아 하드웨어 복잡도가 낮은 장점을 갖는다. 또한, $3t + 2$ 개의 기본 셀로 구성된 DCME 구조는 하나의 PE만을 사용하므로 VLSI 구현에 적합하며, t 또는 유한체의 변화에 따라 재설계가 용이하다. DCME 구조의 게이트 수는 21,760개이다. 그러나 DCME 구조는 기존 수정 유클리드 알고리즘에 비해 낮은 하드웨어 복잡도 및 짧은 최대 전달 지연 경로를 갖지만 RiBM 알고리즘에 비해 하드웨어 복잡도가 높다는 단점을 갖는다.

본 논문에서는 새로운 E-DCME (enhanced DCME) 알고리즘 및 하드웨어 구조를 제안한다^[11]. 제안하는 E-DCME 알고리즘^[11]은 새로운 초기 조건을 사용하여 기존 수정 유클리드 알고리즘 및 DCME 알고리즘에 비해 $T_{mult} + T_{add} + T_{mux}$ 의 짧은 최대 전달 지연을 갖는 것은 물론 RiBM 알고리즘과 유사한 성능을 갖는다. 또한, $3t$ 개의 기본 셀로 구성된 제안하는 E-DCME 구조는 키 방정식 연산을 위해 $2t - 1$ 클럭 사이클만을 필요로 하여 고

속의 리드-솔로몬 복호기에 적합하다. 0.18 μ m 삼성 라이브러리를 사용하여 논리합성을 수행한 결과 E-DCME 구조는 18,000개의 게이트로 구성된다.

본 논문의 구성은 다음과 같다. II장에서는 기존 수정 유클리드 알고리즘 및 DCME 알고리즘에 대해 설명한다. III장에서는 제안하는 E-DCME 알고리즘 및 하드웨어 구조에 대해 기술하고, IV장에서 제안한 E-DCME 구조와 기존 키 방정식 연산 블록들과의 성능 평가를 수행한다. 마지막으로 V장에서 결론을 맺는다.

II. 수정 유클리드 및 DCME 알고리즘

본 절에서는 기존 수정 유클리드 알고리즘^[6]과 DCME 알고리즘^[9]을 설명한다.

2.1 수정 유클리드 알고리즘

그림 1은 수정 유클리드 알고리즘^[6]의 연산 흐름을 나타낸다. $R_i(x)$, $Q_i(x)$, $\lambda_i(x)$, $\mu_i(x)$, $\deg(R_i(x))$, $\deg(Q_i(x))$ 을 입력받아 키 방정식 연산을 수행한다. 그림 1에서 보인 것처럼 수정 유클리드 알고리즘^[6]은 다항식 $R_i(x)$, $Q_i(x)$ 의 차수와 t 의 비교 연산이 필요하다. 따라서 각 기본 셀은 다항식의 차수 비교를 위해 차수 연산 및 비교 회로를 포함한다. 그

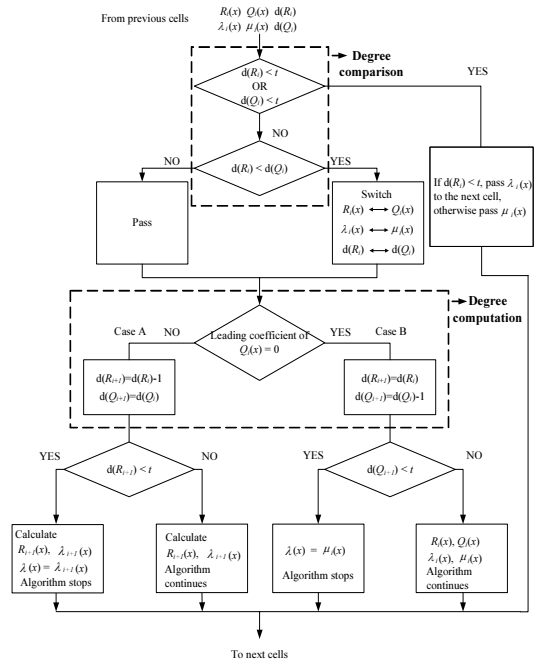


그림 1. 수정 유클리드 알고리즘의 연산 흐름

결과 수정 유클리드 알고리즘⁶⁾은 키 방정식 연산을 위해서 많은 지연 시간을 필요로 하며, 다양한 다항식 연산을 위한 복잡한 제어 과정 및 하드웨어를 필요로 한다.

식 (1)과 (2)는 그림 1의 수정 유클리드 알고리즘을 사용하여 키 방정식 연산을 위한 초기 조건 (initial condition)을 나타낸다.

$$R_0(x) = x^{2t}, Q_0(x) = S(x) = \sum_{i=0}^{2t-1} S_i x^i \quad (1)$$

$$\lambda_0(x) = 0, \mu_0(x) = 1 \quad (2)$$

식 (1)과 (2)의 $R(x)$ 와 $\lambda(x)$ 는 각각 오류 크기 다항식과 오류 위치 다항식을 의미하고, $S(x)$ 는 신드롬 다항식 (syndrome polynomial)이다.

수정 유클리드 알고리즘은 식 (3)~(6)의 다항식 연산을 반복적으로 수행한다.

$$R_{i+1}(x) = [\sigma_i b_i R_i(x) + \overline{\sigma_i a_i} Q_i(x)] - x^{l_i} [\sigma_i a_i Q_i(x) + \overline{\sigma_i b_i} R_i(x)] \quad (3)$$

$$\lambda_{i+1}(x) = [\sigma_i b_i \lambda_i(x) + \overline{\sigma_i a_i} \mu_i(x)] - x^{l_i} [\sigma_i a_i \mu_i(x) + \overline{\sigma_i b_i} \lambda_i(x)] \quad (4)$$

$$Q_{i+1}(x) = \sigma_i Q_i(x) + \overline{\sigma_i} R_i(x) \quad (5)$$

$$\mu_{i+1}(x) = \sigma_i \mu_i(x) + \overline{\sigma_i} \lambda_i(x) \quad (6)$$

a_i 와 b_i 는 각각 $R_i(x)$ 와 $Q_i(x)$ 의 최고차항 계수이고, i 는 다항식 연산의 반복 횟수를 나타낸다. 식 (3)~(6)의 σ_i 와 l_i 는 식 (7)에 의해 주어진다.

$$l_i = \deg(R_i(x)) - \deg(Q_i(x)) \quad (7)$$

$$\sigma_i = 1 \quad \text{if } l_i \geq 0$$

$$\sigma_i = 0 \quad \text{otherwise}$$

식 (7)의 $\deg(R_i(x))$ 와 $\deg(Q_i(x))$ 는 다항식 $R_i(x)$ 와 $Q_i(x)$ 의 차수를 나타낸다. 식 (3)~(6)의 다항식 연산은 $\deg(R_i(x)) < t$ 를 만족할 때까지 반복적으로 수행된다. 연산이 종료되면 다항식 $R(x)$ 와 $\lambda(x)$ 를 얻는다.

2.2 DCME 알고리즘

그림 2는 DCME 알고리즘⁹⁾의 연산 흐름을 나타낸다. i, k, j 는 각각 연산 반복 횟수, 상위 셀의 위치, 하위 셀의 위치를 나타낸다. DCME 알고리즘은 그림 2에서 보인 것처럼 다항식 $R_i(x)$ 와 $Q_i(x)$ 의 최고차항 계수, $Control_Q(C_Q)$ 에 의해서 6가지 Case로 구성된다. 다항식 $R_i(x)$ 와 $Q_i(x)$ 의 최고차항 계수는 시프트 연산의 수행 여부를 결정하고, C_Q 신호는 다항식 교환 연산의 수행 여부를 결정한다. $C_Q = 0$ 이면 다항식 연산과 식 (12), (13)의 교환 연산이 함께 수행된다. 반면에 $C_Q \neq 0$ 이면 교환 연산 없이 다항식 연산만 수행된다. $Control_R(C_R)$ 은 C_Q 와 함께 다항식 연산을 제어한다. DCME 알고리즘은 $\deg(Q_i(x))$ 가 $\deg(R_i(x))$

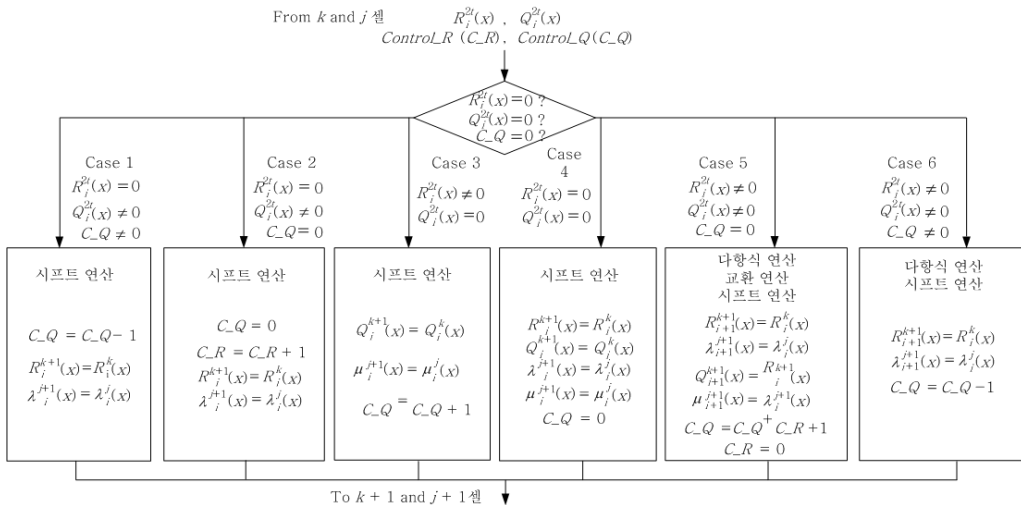


그림 2. DCME 알고리즘의 연산 흐름

보다 크면 C_R 을 1 증가시킨다. C_Q 와 C_R 의 초기 값은 각각 2와 0이다. $2t$ 번의 반복 연산 후 DCME 알고리즘은 종료되고 오류 크기 다항식 $R(x)$ 와 오류 위치 다항식 $\lambda(x)$ 를 얻는다.

DCME 알고리즘^[9]는 식 (8), (9)의 초기 조건을 사용한다. 기존 수정 유클리드 알고리즘의 초기 조건은 $\deg(R_0(x))$ 가 $\deg(Q_0(x))$ 보다 1 크므로 이를 같도록 만들기 위해 $Q_0(x)$ 와 $\mu_0(x)$ 의 차수를 1씩 증가 시킨 것이다.

$$R_0(x) = x^{2t}, Q_0(x) = xS(x) = \sum_{i=0}^{2t-1} S_i x^{i+1} \quad (8)$$

$$\lambda_0(x) = 0, \mu_0(x) = x \quad (9)$$

위의 초기 조건에 의해 $\deg(R_0(x))$ 와 $\deg(Q_0(x))$ 는 $2t$ 로 같다. $\deg(R_0(x))$ 와 $\deg(Q_0(x))$ 가 같으므로 식 (7)의 l_0 는 0이고 식 (3)은 식 (7)의 σ_0 값에 관계없이 $R_i(x) = b_i R_0(x) + a_i Q_0(x)$ 로 간단하게 표현할 수 있다. 즉, $l_i = 0$ 이고 $\sigma_i = 1$ 이면 식 (3)의 x^{l_i} 가 1이므로 식 (3)은 $R_{i+1}(x) = b_i R_i(x) + a_i Q_i(x)$ 으로 표현할 수 있다. 또한, $l_i = 0$ 이고 $\sigma_i = 0$ 이면 식 (3)의 x^{l_i} 는 1이므로 식 (3)은 $R_{i+1}(x) = b_i R_i(x) + a_i Q_i(x)$ 로 표현된다. 따라서 $l_i = 0$ 이면 σ_i 값에 관계없이 식 (3)과 (4)는 식 (10), (11)과 같이 표현 가능하다.

$$R_{i+1}(x) = b_i R_i(x) + a_i Q_i(x) \quad (10)$$

$$\lambda_{i+1}(x) = b_i \lambda_i(x) + a_i \mu_i(x) \quad (11)$$

식 (5)과 (6)의 교환 연산 역시 식 (12), (13)로 표현할 수 있다.

$$Q_{i+1}(x) = R_i(x) \quad (12)$$

$$\mu_{i+1}(x) = \lambda_i(x) \quad (13)$$

따라서 DCME 알고리즘은 식 (7)을 사용할 필요가 없으며 다항식 차수 연산 및 비교 회로를 제거할 수 있다. 즉, DCME 알고리즘은 식 (10)~(13)의 다항식 연산 및 교환 연산만을 반복 수행한다. (10)~(13)의 식은 다항식 차수 연산 및 비교 연산을 필요로 하지 않으므로 (3)~(6)의 식에 비해 훨씬 단순하다. 결과적으로 DCME 알고리즘의 키 방

정식 연산 블록은 기존 수정 유클리드 구조에 비해 짧은 임계 경로 및 낮은 하드웨어 복잡도를 갖는다. 보다 자세한 내용은 참고 문헌 [9], [10]에 기술되어 있다.

III. 새로운 E-DCME 알고리즘 및 하드웨어 구조

본 절에서는 제안하는 E-DCME 알고리즘 및 하드웨어 구조에 대해 설명한다^[11].

3.1 제안하는 E-DCME 알고리즘

제안하는 E-DCME 알고리즘^[11]은 기존의 DCME 알고리즘이 RiBM 알고리즘에 비해 최대 전달 지연 경로가 길고 하드웨어 복잡도가 높은 단점을 개선한다. 식 (14)과 (15)는 제안하는 E-DCME 알고리즘을 위한 새로운 초기 조건을 나타낸다.

$$R_0(x) = x^2 S(x) = \sum_{i=0}^{2t-2} S_i x^{i+2},$$

$$Q_0(x) = xS(x) = \sum_{i=0}^{2t-1} S_i x^{i+1} \quad (14)$$

$$\lambda_0(x) = x^2, \mu_0(x) = x \quad (15)$$

기존 DCME 알고리즘의 초기 조건인 식 (8)과 (9)의 값들 중 신드롬 다항식인 $S(x)$ 만 수신된 데이터의 오류 패턴에 따라서 다른 값을 갖고, 신드롬 다항식을 제외한 다른 값은 오류 패턴에 관계없이 항상 고정되어 있다. 따라서 초기 조건 (8)과 (9)의 첫 번째 다항식 연산이 수행된 후 $R_i(x)$, $Q_i(x)$, $\lambda_i(x)$, $\mu_i(x)$ 는 항상 같은 패턴을 갖는다. 따라서 제안하는 E-DCME 알고리즘은 첫 번째 다항식 연산 결과를 초기 조건으로 하여 불필요한 연산 사이클을 제거하였다. 결과적으로 제안하는 E-DCME 알고리즘은 기존 DCME 알고리즘에 비해 키 방정식 연산을 $2t - 1$ 클럭 사이클 만에 수행할 수 있어 기존 DCME 알고리즘에 비해 1 클럭 사이클을 줄일 수 있다. 제안하는 E-DCME 알고리즘의 다항식 연산 및 교환 연산은 식 (10)~(13)의 기존 DCME 알고리즘^[9]과 동일하다.

그림 3은 제안하는 E-DCME 알고리즘의 연산 흐름을 나타낸다. 제안하는 E-DCME 알고리즘은 오류 크기 다항식 $R(x)$ 의 최고차항 계수와 제어 신호 (CS)에 따라 각기 다른 4가지 연산을 수행한다. $R(x)$ 의 최고차항 계수는 시프트 연산을 수행하지

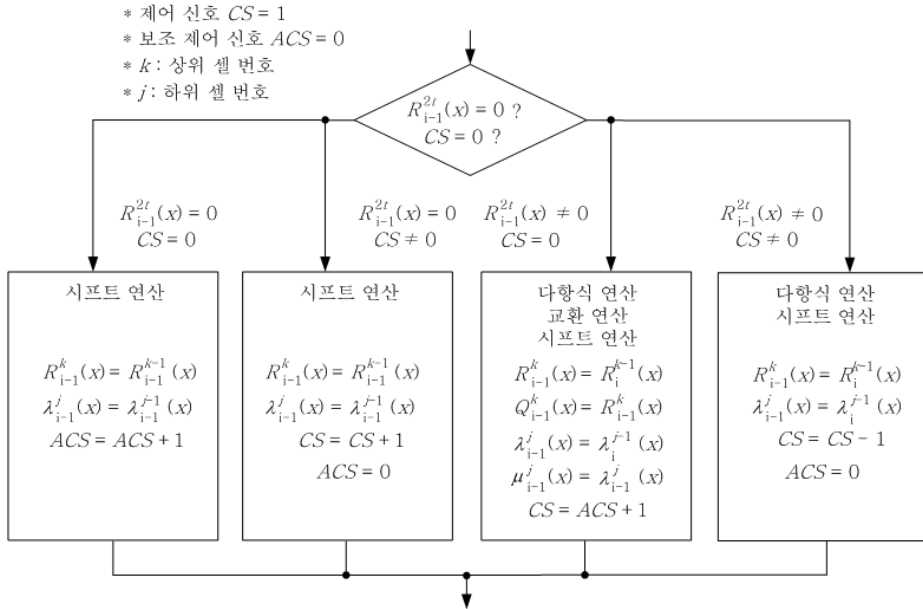


그림 3. 제안하는 E-DCME 알고리즘의 연산 흐름

여부를 결정하고, CS 는 다항식의 교환 연산 수행 여부를 결정한다. 또한, 보조 제어 신호 (ACS)는 CS 신호와 함께 다항식의 연산 흐름을 제어한다. ACS 신호는 $Q(x)$ 다항식의 차수가 $R(x)$ 다항식의 차수보다 큰 경우 그 차이를 나타낸다. 따라서 제안하는 E-DCME 알고리즘은 6가지 각기 다른 연산을 수행하는 기존 DCME 알고리즘에 비해 단순한 연산 흐름을 갖는다.

3.2 제안하는 E-DCME 구조

그림 4는 제안하는 E-DCME 알고리즘의 하드웨어 구조를 나타낸다. 제안하는 E-DCME 구조는 $2t - 1$ 개의 상위 셀과, $t + 1$ 개의 하위 셀, 제어 블록으로 구성된다. 제안하는 E-DCME 구조는 새로운 초기 조건 (14)과 (15)를 사용하여 한 번의 다항식 연산을 수행한 효과를 가지므로 다항식 연산을 위한 한 개의 기본 셀을 제거할 수 있다. 또한, E-DCME 알고리즘은 기존 수정 유클리드 알고리즘과 같이 오류 크기 다항식 $R(x)$ 의 최고차항 계수를 반복적인 연산을 통해서 지속적으로 제거한다. 결과적으로 기존 DCME 구조의 최상위 셀의 출력은 항상 0을 출력하므로, 제안하는 E-DCME 구조는 최상위 셀의 유한체 곱셈기 및 덧셈기를 제거하여 최상위 셀과 제어 블록을 통합하였다. 따라서 E-DCME 구조는 $3t + 2$ 개의 기본 셀을 사용하는 기존 DCME 구조에 비해 적은 기본 셀만으로 키

방정식 연산을 수행할 수 있다.

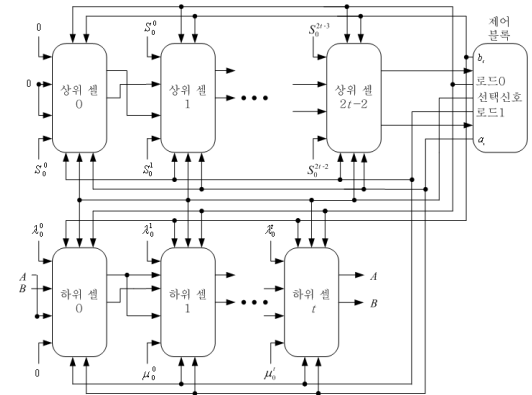


그림 4. 제안하는 E-DCME 구조

그림 5는 제안하는 E-DCME 알고리즘을 위한 새로운 기본 셀을 나타낸다. 기존 DCME 구조는 키 방정식 연산을 위해 더 많은 멀티플렉서를 사용하므로 기존 RiBM 구조에 비해 긴 전달 지연 경로를 갖는다. 반면에 제안하는 E-DCME 구조는 오류 위치 다항식 $R(x)$ 의 시프트 연산 경로와 다항식 연산 경로를 통합하여 멀티플렉서를 제거하였다. 또한 다항식 $Q(x)$ 은 다항식 $R(x)$ 로부터 값을 입력 받으므로 0의 값을 갖지 않는 것에 착안하여 $Q(x)$ 의 시프트 연산을 제거하였다. 따라서 제안하는 E-DCME 구조는 각 기본 셀마다 기존 DCME 구

조에 비해 4개적은 멀티플렉서를 사용하며, $T_{mul} + T_{add} + T_{mux}$ 의 전달 지연 경로를 갖는다. 제안하는 기본 셀은 2개의 멀티플렉서, 2개의 레지스터, 2개의 유한체 곱셈기, 1개의 유한체 덧셈기로 구성된다.

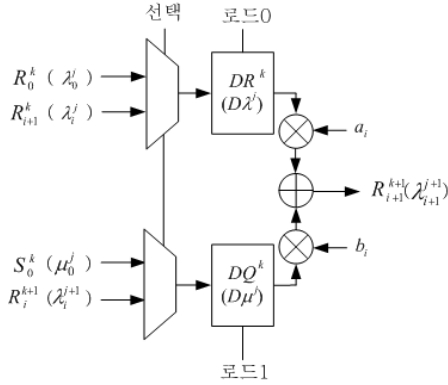


그림 5. 제안하는 기본 셀

IV. 성능 평가

제안하는 E-DCME 알고리즘^[11]의 검증을 위해서 다양한 RS 부호에 대해서 C++ 시뮬레이션을 수행하였으며, VHDL을 사용하여 설계하였다. 설계한 E-DCME 구조는 삼성 0.18 μ m 라이브러리와 SYNOPSISTM을 사용하여 논리 합성을 수행한 결과 18,000개의 게이트로 구성되며, 키 방정식 연산을 위해서 $2t - 1$ 클럭 사이클만을 필요로 한다. 표 1은 제안하는 E-DCME 구조와 기존 수정유클리드 구조, DCME 구조와의 성능 비교를 나타낸다.

표 1. 기존 구조와의 성능 비교

	[8]	[9]	E-DCME
공정	0.16 μ m	0.25 μ m	0.18 μ m
레지스터	50t	6t + 4	6t
멀티플렉서	38t	18t + 12	6t
곱셈기	8t	6t + 4	6t
덧셈기	4t	3t + 2	3t
게이트 수	43,100개	21,700	18,000개
지연시간	3t + 37	2t	2t - 1

$2t$ 개의 기본 셀로 구성되어 있는 기존 ME 구조^{[7][8]}은 50t 레지스터, 38t 2-입력 멀티플렉서, 8t 유한체 곱셈기, 4t 유한체 덧셈기를 사용한다. 각 기본 셀은 레지스터 25개, 2-입력 멀티플렉서 19개,

유한체 곱셈기 4개, 유한체 덧셈기 2개로 구성된다. 기존 ME 구조의 게이트 수는 43,100개이며, 키 방정식 연산을 위해 $3t + 37$ 클럭 사이클을 필요로 한다^{[7][8]}.

$3t + 1$ 개의 기본 셀을 사용하는 DCME 구조^[9]는 $6t + 4$ 레지스터, $18t + 12$ 2-입력 멀티플렉서, $6t + 4$ 유한체 곱셈기, $3t + 2$ 유한체 덧셈기를 사용한다. DCME 구조의 게이트 수는 21,700개이며, 키 방정식 연산을 위해 $2t$ 사이클을 필요로 한다^[9].

반면에 제안하는 E-DCME 구조는 전체 $3t$ 개의 기본 셀을 사용하여 $6t$ 레지스터, $6t$ 2-입력 멀티플렉서, $6t$ 유한체 곱셈기, $3t$ 유한체 덧셈기를 사용한다. 따라서 제안하는 E-DCME 구조는 기존 ME^[8] 및 DCME^[9] 구조에 비해 적은 하드웨어 면적을 필요로 한다. 또한 제안하는 E-DCME 알고리즘은 키 방정식 연산을 위해 $2t - 1$ 클럭 사이클을 사용하므로 기존 구조에 비해 고속의 키 방정식 연산이 가능하다. 제안하는 E-DCME 구조의 게이트 수는 18,000이다.

V. 결론

본 논문에서는 리드-솔로몬 복호기의 키 방정식 연산을 위한 새로운 E-DCME 알고리즘을 제안한다. E-DCME 알고리즘은 사용하는 기본 셀의 개수가 기존 DCME 알고리즘에 비해 적을 뿐만 아니라 각 기본 셀의 하드웨어 구조가 매우 단순하다. 따라서 제안하는 E-DCME 구조는 기존 DCME 구조에 비해 작은 하드웨어 크기를 갖는다. 또한 제안하는 E-DCME 알고리즘은 새로운 초기조건을 사용하기 때문에 키 방정식 연산을 위해 $2t - 1$ 클럭 사이클만을 필요로 하여 고속의 키 방정식 연산이 가능하다. 설계한 E-DCME 구조의 게이트 수는 18,000이다. 따라서 제안한 구조는 짧은 지연 시간 및 낮은 하드웨어 복잡도를 가지므로 PLC, DVB-T, VSB, cable modem, WATM, 위성 통신, 이동 통신, DVD 등 다양한 분야에 활용 가능하다.

참고 문헌

- [1] A. Raghupathy and K. J. R. Liu, "Algorithm-based low-power/high-speed Reed-Solomon decoder design," *IEEE Trans. Circuit Syst. II*, vol. 47, pp. 1254-1270, Nov. 2000.
- [2] J. H. Jeng and T. K. Truong, "On decoding of

both errors and erasures of a Reed-Solomon code using an inverse-free Berlekamp-Massey algorithm," *IEEE Trans. Commun.*, vol. 47, pp. 1488-1494, Oct. 1999.

[3] D. V. Sarwate and N. R. Shanbhag, "High-speed architectures for Reed-Solomon decoders," *IEEE Trans. VLSI Syst.*, vol. 9, pp. 641-655, Oct. 2001.

[4] M. A. A. Ali, A. Abou-El-Azm, and M. F. Marie, "Error rates for non-coherent demodulation FCMA with Reed-Solomon codes in fading satellite channel," in *Proc. IEEE Vehicular Techn. Conf. (VTC'99)*, vol. 1, 1999, pp. 92-96.

[5] T. K. Matsushima, T. Matsushima, and S. Hirasawa, "Parallel architecture for high-speed Reed-Solomon codec," in *Proc. IEEE Int. Telecommun. Symp. (ITS'98)*, vol. 2, 1998, pp. 468-473.

[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. Comput.*, vol. C-34, pp. 393-403, May 1985.

[7] H. H. Lee, M. L. Yu and L. Song, "VLSI design of Reed-Solomon decoder architectures," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS' 2000)*, vol. 5, May 2000, pp. 705-708.

[8] H. H. Lee, "Modified Euclidean algorithm block for high-speed Reed-Solomon decoder," *IEE Electronics Letters*, vol. 37, pp. 903-904, July 2001.

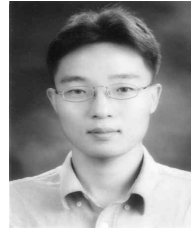
[9] J. H. Baik and Myung H. Sunwoo, "New degree computationless modified Euclid's algorithm and architecture for Reed-Solomon decoder," *IEEE Trans. VLSI Syst.*, vol. 14, pp. 915-920, Aug. 2006.

[10] 백재현, 선우명훈, "새로운 DCME 알고리즘을 사용한 고속 Reed-Solomon 복호기," *전자공학회 논문지 제40권 SD편*, 6호, 81-90쪽, 2003.

[11] J. H. Baik and Myung H. Sunwoo, "Enhanced degree computationless modified Euclid's algorithm for Reed-Solomon decoder," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS' 2006)*, vol. 5, May 2006, pp. 3554-3557.

백 재 현 (Jae-hyun Baik)

정회원



2002년 2월 : 아주대학교 전자공학과 졸업
 2002년 3월~현재 : 아주대학교 전자공학과 석박사 통합과정 <관심분야> 통신 및 신호처리용 SOC 설계

선 우 명 훈 (Myung-hoon Sunwoo)

정회원



1980년 2월 : 서강대학교 전자공학과 졸업
 1982년 2월 : 한국과학기술원 전자공학과 석사
 1982년 3월~1985년 8월 : 한국전자통신연구소 (ETRI)
 1985년 9월~1990년 8월 : Univ. of Texas at Austin 전자공학과 박사
 1992년 8월~1996년 10월 : 아주대학교 전기전자공학부 조교수
 1996년 10월~2001년 9월 : 아주대학교 전자공학부 부교수
 2001년 10월~현재 : 아주대학교 전자공학부 교수
 <관심분야> VLSI 및 Parallel Architecture, 통신 멀티미디어용 DSP 칩 및 SOC 설계