다항식 표현을 이용한 DCME 알고리즘 설계

종신회원 강 성 진*°, 김 남 용**

Design of Degree-Computationless Modified Euclidean Algorithm using Polynomial Expression

Sung-Jin Kang*°, Nam Yong Kim** *Lifelong Members*

요 약

본 논문에서는 고속 RS(Reed-Solomon) 복호기의 KES(Key Equation Solver) 블록 구현에 사용하는 ME(Modified Euclidean) 알고리즘을 효율적으로 설계할 수 있는 구조를 제안하고 구현하였다. 제안된 구조에서는 각 PE(Processing Element) 블록을 제어하기 위해 새로운 상태변수를 정의하고 다항식으로 표현함으로써, 입출력 신호가 간단해지고, 차수계산회로가 필요 없기 때문에 회로의 복잡도를 줄일 수 있다. 또한, PE 회로가 오류 정정 능력 t와 무관하기 때문에, t가 증가함에 따라 KES 블록의 하드웨어 복잡도가 선형적으로 증가하는 장점을 가진다. 제안된 구조와 기존의 구조를 비교하기 위해, RS(255,239,8) 복호기에 대한 KES 블록을 구현하고, 0.13um CMOS cell library를 이용하여 합성하였다. 실험 결과로부터, 제안된 구조를 이용하여 적은 gate count로 고속 RS 복호기 구현이 가능함을 알 수 있다.

Key Words : Reed-Solomon, Modified Euclidean, RS decoder, Key Equation Solver, DCME

ABSTRACT

In this paper, we have proposed and implemented a novel architecture which can be used to effectively design the modified Euclidean (ME) algorithm for key equation solver (KES) block in high-speed Reed-Solomon (RS) decoder. With polynomial expressions of newly-defined state variables for controlling each processing element (PE), the proposed architecture has simple input/output signals and requires less hardware complexity because no degree computation circuits are needed. In addition, since each PE circuit is independent of the error correcting capability t of RS codes, it has the advantage of linearly increase of the hardware complexity of KES block as t increases. For comparisons, KES block for RS(255,239,8) decoder is implemented using Verilog HDL and synthesized with 0.13um CMOS cell library. From the results, we can see that the propesed architecture can be used for a high-speed RS decoder with less gate count.

I. 서 론

RS(Reed-Solomon) 부호는 연집 오류에 대하여 우수한 오류 정정 능력을 가지고 있어서, 광/자기 저장 매체, 유무선 통신, 방송, 위성 통신 등 많은 통신시스템에서 널리 사용되고 있다. 또한, 최근에는 NAND 플래시 메모리 분야에서도 오류 정정을 하기 위한 연구가 활발히 진행되고 있다.

일반적인 RS(n,k,t) 부호에서 n은 전체 부호어(codeword)의 길이(심볼 개수), k는 정보 심볼의 개수를 의미하며, $t = \lfloor (n-k)/2 \rfloor$는 RS 부호의 오류 정정 능력을 나타낸다[1,6]. RS부호에 대한 복호기는 그

* 한국기술교육대학교 전기전자통신공학부(sjkang@kut.ac.kr), (° : 교신저자)
** 강원대학교 공학대학 전자정보통신공학부(namyong@kangwon.ac.kr)
논문번호 : KICS2011-08-356, 접수일자 : 2011년 8월 17일, 최종논문접수일자 : 2011년 10월 5일

림 1과 같이 신드롬 연산(syndrome computation), 키 방정식 연산(Key Equation Solver, KES), Chien 탐색, Forney 알고리즘, 오류 정정 블록 및 FIFO(First Input First Output)로 구성된다[2-5]. 여기에서 KES 블록이 오류위치 다항식(error locator polynomial, $\sigma(x)$)과 오류값 다항식(error value polynomial, $\omega(x)$)을 찾기 위해 가장 많은 연산을 필요로 하며, 하드웨어 복잡도가 가장 높다.

RS 복호기에 관한 연구는 대부분 KES 알고리즘에 관한 것이며, 많은 복호 알고리즘과 복호기 구조가 연구되어 왔다[1-8]. 이 중에서 수정된 유클리드(Modified Euclidean, ME) 알고리즘은 하드웨어의 규칙성이 우수하여 쉽게 구현이 가능한 장점을 지니고 있다[4].

ME 알고리즘[2]은 차수 계산과 다항식 연산을 수행하는 PE(processing element) 블록을 $2t$개 사용하여 구현할 수 있으며, 이러한 구조는 하드웨어 규칙성 및 경로 지연(critical path)이 작아서 고속으로 동작하는 RS 복호기를 구현할 수 있다[4,5]. [6]에서는 차수 계산이 필요치 않는 DCME(degree computationless ME)를 제안하였지만, 각 기본 셀(basic cell)내의 feedback되는 부분과 모든 셀에 입력되는 leading coefficient a_i, b_i가 feedback되므로 상대적으로 고속 구현이 어렵게 된다. [7]에서는 DCME 알고리즘의 지연시간과 basic cell을 개선하여 E-DCME 알고리즘을 제안하였다. [6,7]의 DCME 구조는 하드웨어 복잡도와 복호지연(decoding latency)면에서 우수하지만, [4,5]에 비해 고속 구현이 어렵다. [8]에서는 [4]의 PE 구조를 개선하여 마지막 PE 블록의 출력신호에서 차수 비교 및 MUX(multiplexer)가 필요 없을 뿐 만 아니라, 차수 계산(degree computation) 회로 대신 FSM(finite state machine)을 사용하여 하드웨어 복잡도를 줄였다. 하지만, 이 구조는 RS(23,17) 복호기에만 적용되며 일반적인 RS 복호기에 적용할 수 없다.

본 논문에서는 [5,8]의 PE 제어를 위한 FSM를 사용하는 대신에, 상태변수를 다항식으로 표현함으로써 간단한 회로로 구현이 가능한 PE 구조를 제안하고 구현한다. 제안된 구조는 입출력 신호가 간단하고, 차수 계산회로가 필요 없기 때문에 회로의 복잡도를 줄일 수 있으며, 기존의 구조와 달리 PE 회로가 오류 정정 능력 t와 무관하기 때문에, t가 증가함에 따라 KES 블록의 하드웨어 복잡도가 선형적으로 증가하는 장점을 가진다.

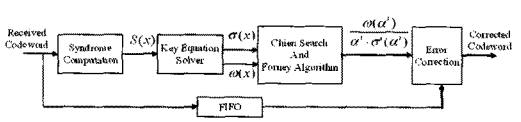

그림 1. RS 복호기의 블록도

본 논문의 구성은 2장에서 제안된 PE 구조를 설명하고, 3장에서는 제안된 PE 구조를 이용한 DCME 알고리즘 구조에 대하여 설명한다. 4장에서는 RS(255,239,8) 복호기를 위한 KES 블록 설계와 성능평가에 관하여 다루고, 5장에서 결론을 맺는다.

Ⅱ. 제안된 PE 구조

[8]에서 제안된 PE 구조는 [4]의 PE 구조에서 크게 두 부분이 개선되었다. 첫 번째는 PA(polynomial arithmetic) 부분에서 다항식 스위치(polynomial exchange)를 PE의 입력단에서 수행하지 않고, 출력 다항식에 대해서 다항식 스위치를 수행하여 최종 PE 블록에서 부가회로 없이 오류위치 다항식 $\sigma(x)$과 오류값 다항식 $\omega(x)$을 얻을 수 있다. 두 번째는 DC(degree computation) 부분에서 복잡한 차수 계산을 하지 않고 $R_i(x)$와 $Q_i(x)$의 차수의 차이를 상태변수로 정의한 후 FSM을 사용하여 제어신호를 발생한다. 이 두 부분으로 인해 PE 구조의 하드웨어 복잡도를 개선할 수 있다.

[4]에서 차수 계산이 필요한 이유는 첫째로 $\deg(R_{i-1}(x)) < \deg(Q_{i-1}(x))$인 경우에 다항식 스위치를 하기 위한 sw신호를 발생하기 위함이며, 둘째로 $\deg(R_i(x)) < t$ 또는 $\deg(Q_i(x)) < t$인 경우에 $stop$신호를 발생하기 위함이다. [8]에서 sw신호를 발생하기 위해 정의한 상태변수는 일반적인 RS 부호에 적용할 수 있지만, $stop$신호는 UWB 시스템에서 사용하는 RS (23,17)에서 유효하지만, 일반적인 RS 복호기에 적용을 할 수 없는 한계를 가지고 있다.

2.1 sw 신호

[8]의 PE 구조에서 다항식 스위치가 일어나는 경우는 $R_{i-1}(x)$와 $Q_{i-1}(x)$의 차수가 같고 $Q_{i-1}(x)$의 leading coefficient가 '0'이 아닌 경우에, 다항식 연산에서 $R_{i-1}(x)$의 최고차항 계수가 제거되어 차수가 1 감소하였으므로, $\deg(R_i(x))$가 $\deg(Q_i(x))$보다 작게 된 경우이다. 즉, 상태 변수를 식 (1)과 같이 정의하면, $\Delta_{i-1}=0$이고 zq신호가 '0'인 경우이다. zq는 $Q_{i-1}(x)$의 leading coefficient가 '0'일 때 '1'이고, 그 외에는 '0'인 신호이다.

$$\Delta_i = |\deg(R_i(x)) - \deg(Q_i(x))| \quad (1)$$

여기에서 i는 i번째 PE 블록을 의미하고, $0 \le i \le 2t$, $0 \le \Delta_i \le t-1$이다. $\Delta_i = t$인 경우는 $\deg(R_i(x)) = 2t$일 때는 $\deg(Q_i(x)) = t$가 되게 하는 신드롬 다항식이 발생할 수 없고, $\deg(R_i(x)) < 2t$일 때는 $\Delta_i = t$가 되기 전에$\deg(Q_i(x))$가 t보다 작아서 *stop* 신호가 발생하기 때문에, $\Delta_i = t$인 경우는 발생하지 않는다. 또한, [8]의 PE 구조는 항상 $R_{i-1}(x)$ 또는 $Q_{i-1}(x)$의 차수가 1씩 감소하도록 되어있기 때문에, 그림 2와 같은 상태도로 표현할 수 있다.

그림 2에서 i번째 PE블록에서 $\Delta_{i-1} = 0$이고 zq=0이면, sw신호가 1이 되어 다항식 스위치가 일어남을 알 수 있다.

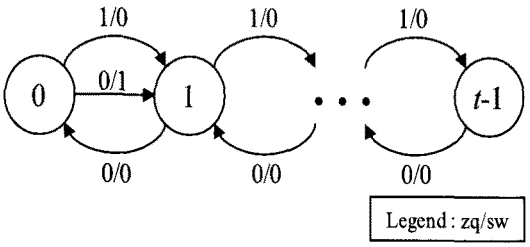

그림 2. Δ_i의 상태도

2.2 *stop* 신호

[8]의 PE 구조에서 *stop* 신호는 i번째 PE 블록의 출력 다항식 $R_i(x)$, $Q_i(x)$의 차수가 오류정정 능력 t보다 작을 때 발생하여, 이후의 PE 블록이 동작하지 않도록 한다. [8]의 PE는 항상 $\deg(R_i(x)) \ge \deg(Q_i(x))$가 성립하므로, $Q_i(x)$의 차수가 t보다 작으면 *stop* 신호를 발생한다. 식 (1)에 정의된 Δ_i로 부터 $R_i(x)$와 $Q_i(x)$의 차수의 차이를 알 수 있지만, $Q_i(x)$의 차수가 t보다 작은지를 알 수가 없다. 따라서, 각 PE에서 $Q_i(x)$의 변화를 관찰하고 있어야 한다. $Q_i(x)$의 차수가 감소하는 경우는 $Q_{i-1}(x)$의 leading coefficient가 '0'인 경우와 다항식 연산 후에 $R_i(x)$의 차수가 $Q_i(x)$의 차수보다 작게 되어 다항식 스위치가 일어나는 경우이다. 따라서 $Q_i(x)$의 차수가 감소하는 횟수를 카운트하기 위한 상태변수 $\mu_{i-1} = \deg(Q_{i-1}(x))$를 정의하면, i번째 PE에서 μ_i는 식 (2)와 같이 됨을 알 수 있다. 만약, $\mu_i = 2$라면 $Q_i(x)$의 차수는 $Q_0(x)$의 차수에서 2만큼 감소했음을 의미한다.

$$\mu_i = \begin{cases} \mu_{i-1} - 1, & \text{if } (zq=1) \text{ or } (sw=1) \\ \mu_{i-1}, & otherwise \end{cases} \quad (2)$$

여기에서, i는 i번째 PE 블록을 의미하고, $0 \le i \le 2t$, $t-1 \le \mu_i \le 2t-1$이다. $\mu_i = t$인 경우는, $Q_i(x)$의 차수가 $Q_0(x)$ 차수로부터 t만큼 감소했다는 의미이며, ME 알고리즘에서 초기치 $Q_0(x)$의 차수는 $2t-1$이므로, $Q_i(x)$의 차수가 $t-1$이 되었음을 나타낸다. 따라서, 이 경우에는 더 이상 PE 블록에서 다항식 연산이 수행되지 않는다. 이로부터 *stop*신호를 PE 블록의 입출력 신호로 사용할 필요가 없음을 알 수 있다. 따라서, *stop*신호는 PE 블록 내부에서 발생되며, 식 (3)과 같이 정의된다.

$$stop = \begin{cases} 1, & \text{if } \mu_{i-1} = t \\ 0, & otherwise \end{cases} \quad (3)$$

2.3 PE 구조

본 논문에서 제안하는 PE구조는 식 (1), (2)에 정의되어 있는 신호를 입출력신호로 사용하고 식 (3)에 정의되어 있는 *stop* 신호는 PE 내부에서만 사용되는 신호이다. Δ_i와 μ_i를 FSM을 이용하여 구현할 수 도 있지만, 하드웨어의 복잡도를 줄이기 위해 식 (1)의 Δ_{i-1}를 식 (4)와 같이 다항식으로 표현한다.

$$C_{i-1}(x) = c_0 x^{2t} + c_1 x^{2t-1} + \cdots + c_{t-1} x^{t+1} \quad (4)$$

여기에서, 계수 c_k은 식 (5)와 같고, $0 \le k \le t-1$이다.

$$c_k = \begin{cases} 1, & \text{if } \Delta_{i-1} = k \\ 0, & otherwise \end{cases} \quad (5)$$

식 (1)과 그림 2로부터 Δ_i는 $\Delta_{i-1} = k$로부터 1이 증가하거나 감소함을 알 수 있고, 이는 $C_{i-1}(x)$의 계수를 좌우로 이동하는 효과와 동일하다. 따라서, 그림 2의 FSM은 식 (4)에서 계수를 shift 연산으로 구현 가능함을 알 수 있다. 즉, $\Delta_{i-1} = k$이면, $C_{i-1}(x) = x^{2t-k}$이므로, $C_i(x)$는 그림 2에서와 같이 $C_i(x) = x^{2t-k-1}$ 또는 $C_i(x) = x^{2t-k+1}$가 되어야 하며, 이는 $C_{i-1}(x)$의 계수를 좌우로 shift 한 결과와 일치하게 된다.

Δ_i와 유사하게 식 (2)의 μ_i를 식 (6)과 같이 다항식으로 표현한다.

$$D_i(x) = d_0 x^{2t} + d_1 x^{2t-1} + \cdots + d_t x^t \quad (6)$$

여기에서, 계수 d_k은 식 (7)와 같고, $0 \le k \le t$이다.

$$d_k = \begin{cases} 1, & \text{if } k = (\mu_{i-1} - (t-1)) \\ 0, & otherwise \end{cases}, \qquad (7)$$

식 (2)로부터 μ_i는 μ_{i-1}와 같거나, 1이 감소되므로, $D_i(x)$는 $D_{i-1}(x)$의 계수를 shift 연산을 함으로써 식 (2)를 구현할 수 있다. 즉, $\mu_{i-1} = n$이면 $D_{i-1}(x) = x^{2t-(n-(t-1))}$이므로, $D_i(x)$는 $D_{i-1}(x)$와 같거나, $D_i(x) = x^{2t-(n-1-(t-1))}$가 된다.

i번째 PE 블록은 $\deg(Q_{i-1}(x)) < t$이면 stop 신호가 발생하므로, $D_{i-1}(x) = x^{2t}$일 때 i번째 PE 블록 내부에서 stop 신호가 발생한다. 따라서, stop 신호는 $D_{i-1}(x)$의 leading coefficient가 1이 될 때 발생하므로, stop 신호를 발생하는 로직이 추가로 필요하지 않음을 알 수 있다.

그림 3은 식 (4), (6)의 다항식을 이용하는 제안된 PE 구조이다. 박스 안의 'D'는 지연소자(filp-flop)을 의미하고, m은 $GF(2^m)$의 m을 의미한다. *sw* 신호는 다항식 스위치에 사용되며 $C_{i-1}(x)$의 leading coefficient가 1이고(즉, $\Delta_{i-1} = 0$), $zq = 0$일 때 발생한다. *ctl1*은 $R_i(x)$와 $L_i(x)$를 제어하는데, *stop* 신호가 발생하거나 또는 $zq = 1$이면, *ctl1* = 1이 되고, $R_i(x) = R_{i-1}(x)$, $L_i(x) = L_{i-1}(x)$가 출력된다. *ctl2*는 $Q_i(x)$와 $U_i(x)$를 제어하는데, *stop* 신호가 발생하거나 또는 $zq = 0$이면, *ctl2* = 1이 된다. *ctlC*는 $C_i(x)$를 제어하는데, $zq = 1$이거나 $C_{i-1}(x)$의 leading coefficient가 1이면, *ctlC* = 1이 된다. *ctlC* = 1이면, $C_i(x)$는 $\Delta_i = \Delta_{i-1} + 1$을 나타내고, *ctlC* = 0이면 $C_i(x)$는 $\Delta_i = \Delta_{i-1}(x) - 1$을 나타낸다. *ctlD*은 $D_i(x)$를 제어하는데, *stop* 신호가 발생하거나 또는 $C_{i-1}(x)$의 leading coefficient가 0이면 *ctlD* = 1이 된다. *ctlD* = 1이면, $D_i(x)$는 $\mu_i = \mu_{i-1}$을 나타내고, *ctlD* = 0이면, $D_i(x)$는 $\mu_i = \mu_{i-1} - 1$을 나타낸다.

그림 3에서 $C_{i-1}(x)$, $C_i(x)$, $D_{i-1}(x)$, $D_i(x)$는 1-bit 신호이므로, [4]와 [5]에 비해 PE 블록을 제어하기 위한 입출력 신호가 매우 간단함을 알 수 있다. 또한 stop신호도 내부적으로 발생하기 때문에, 입출력신호로 나타나지 않음을 볼 수 있다. 또한, 그림 3의 PE 구조는 RS 부호의 오류정정능력 t와 무관하게 설계되므로, t가 변화해도 PE는 변화하지 않는다. RS부호의 Galois Field가 변할때만 $R_i(x)$, $Q_i(x)$, $L_i(x)$, $U_i(x)$의 비트수가 바뀌고, 곱셈기와 덧셈기가 변화된다.

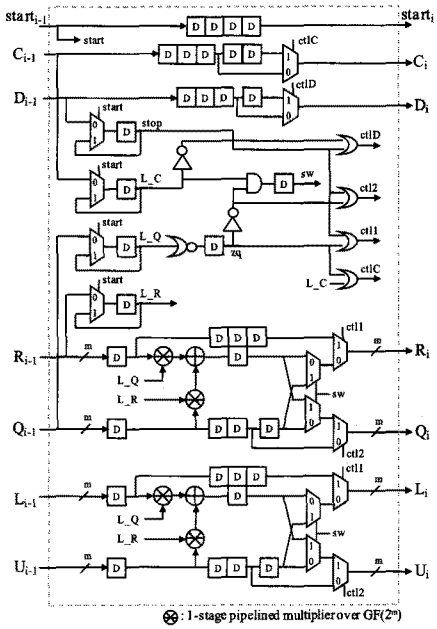

그림 3. 제안된 PE 구조

Ⅲ. DCME 알고리즘 구조

그림 1의 KES 블록을 구현하기 위해서는 PE 블록 $2t$개를 연결하여 사용한다[4]. 그림 4는 본 논문에서 제안된 PE 블록 $2t$개를 이용하여 KES 블록을 구현하기 위한 구조를 보여준다.

그림 4에서 KES 블록 입력 중에서 $R_0(x)$, $Q_0(x)$, $L_0(x)$, $U_0(x)$는 아래와 같은 ME 알고리즘의 초기값을 사용한다.

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

$$L_0(x) = 0, \quad U_0(x) = 1$$

그림 4에서 $C_0(x)$의 초기치는 식 (4)로부터 $C_0(x)$

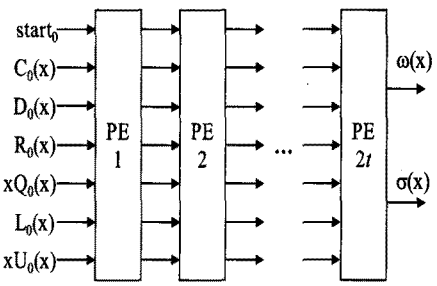

그림 4. KES 블록 구조

$=x^{2t-1}$가 됨을 알 수 있다. 즉, $R_0(x)$의 차수와 $Q_0(x)$의 차수의 차이가 1이므로, $\Delta_0=1$임을 의미한다. $D_0(x)$의 초기치는 식 (6)으로부터 $D_0(x)=x^t$이다. 즉, 다항식 연산이 한번도 수행되지 않았기 때문에 $\mu_0=\deg(Q_0(x))=2t-1$이다. KES 블록의 출력 인 오류위치 다항식 $\sigma(x)$과 오류값 다항식 $\omega(x)$은 다음 식과 같다.

$$\sigma(x)=U_{2t}(x),\ \omega(x)=Q_{2t}(x)$$

표 1은 오류정정능력이 $t=3$경우에 대하여 그림 4의 초기치 입력 시퀀스에 대한 예를 보여준다. $t=3$인 경우에 대한 신드롬을 식 (8)과 같이 주어졌을 때, $C_0(x)=x^5$, $D_0(x)=x^3$, $R_0(x)=x^6$, $xQ_0(x)=xS(x)$, $L_0(x)=0$, $xU_0(x)=x$이므로, 표 1과 같은 시퀀스가 KES블록에 입력된다.

$$S(x)=\sum_{j=0}^{5}S_jx^j \tag{8}$$

표 1. $t=3$인 경우에 대한 KES블록 입력 시퀀스 예

time index	⋯	6	5	4	3	2	1	0	⋯
$start_0$	0	0	0	0	0	0	0	1	0
$C_0(x)$	0	0	0	0	0	0	1	0	0
$D_0(x)$	0	0	0	0	1	0	0	0	0
$R_0(x)$	0	0	0	0	0	0	0	1	0
$xQ_0(x)$	0	0	S_0	S_1	S_2	S_3	S_4	S_5	0
$L_0(x)$	0	0	0	0	0	0	0	0	0
$xU_0(x)$	0	0	1	0	0	0	0	0	0

Ⅳ. 성능평가

제안된 구조의 성능을 평가하기 위해 가장 많이 사용되는 RS(255,239,8) 복호기를 구성하여 [4,5]의 구조와 비교하였다. 식 (9)는 사용된 RS부호의 발생 다항식 $g(x)$이다..

$$\begin{aligned}g(x)&=\prod_{i=1}^{16}(x-\alpha^i)\\&=x^{16}+118x^{15}+52x^{14}+103x^{13}\\&\quad+31x^{12}+104x^{11}+126x^{10}+187x^9\\&\quad+232x^8+17x^7+56x^6+183x^5\\&\quad+49x^4+100x^3+81x^2+44x+79\end{aligned} \tag{9}$$

RS 복호기는 그림 1과 같은 구조로 구성했으며, 수신 코드워드로부터 신드롬을 계산한 후에, 제안된 DCME 알고리즘 구조를 이용하여 오류위치 다항식 $\sigma(x)$과 오류값 다항식 $\omega(x)$을 얻는다. $\sigma(x)$와 $\omega(x)$를 사용하여 Chien 탐색과 Forney 알고리즘을 이용하여 오류 정정을 수행한다[4].

다양한 오류패턴에서 제안된 알고리즘 구조가 유효한지를 검사하기 위해 C언어를 이용하여 RS (255,239,8) 복호기를 구성하고 시뮬레이션을 수행하였다. 그림 5는 AWGN(Additive White Gaussian Noise) 채널에서 BPSK(Binary Phase Shift Keying) 변조를 사용했을 때, RS(255,239,8)부호의 BER(Bit Error Rate) 성능 곡선이다. AWGN채널에서 RS(255,239,8) 부호의 심볼오류확률 P_{es}는 식 (10)과 같고, 비트오류확률 P_{eb}는 식 (11)과 같다[9].

$$P_{es}=\frac{1}{255}\sum_{i=9}^{255}i\binom{255}{i}P_M^i(1-P_M)^{255-i} \tag{10}$$

$$P_{eb}=\frac{2^7}{2^8-1}P_{es} \tag{11}$$

여기에서 P_M은 N개의 코드심볼로 구성된 코드워드에서 한 코드심볼이 오류가 날 확률이므로, $P_M=(1-(1-P_B)^8)$가 된다. AWGN 채널에서 BPSK 변조의 비트오류확률은 $P_B=Q(\sqrt{2E_b/N_0})$이다. 그림 5에서 이론치와 시뮬레이션 결과치가 낮은 E_b/N_0에서 다소 다르게 나타나는 것을 알 수 있다. 이는 식 (11)의 결과가 근사치이며, 상한으로 작용한다고 볼 수 있다. E_b/N_0가 5dB이상이면, 이론치와 시뮬레이션 결과가 일치함을 볼 수 있고, 제안된 구조를 사용하는 RS 복호기가 다양한 오류패턴에서 잘 동작하고 있음을

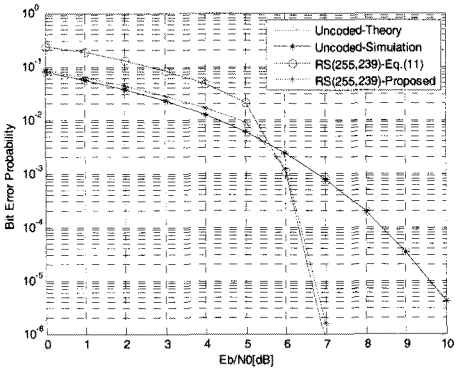

그림 5. RS(255,239,8) 부호의 BER 성능

확인할 수 있다.

다음으로 제안된 PE 구조를 이용하여 RS (255,239,8) 복호기를 위한 KES 블록을 Verilog HDL를 사용하여 구현하였으며, SMIC 0.13um library로 합성하였다. 표 2는 제안된 구조와 기존의 다른 구조와 critical path delay를 비교한 결과이다. 제안된 구조가 가장 작은 critical path delay를 가짐을 알 수 있다. 제안된 구조에서는 다항식 연산(PA)이후에 다항식 스위치를 수행하고, 차수 계산 회로가 다항식 표현을 통해 간단한 회로로 대치되었기 때문에 critical path delay가 작아지게 된다.

표 3은 KES 블록을 합성한 결과를 비교한 결과로서, 제안된 구조의 합성결과와 참고문헌 [4,5]의 결과를 비교한 것이다. 제안된 PE 구조에 대한 결과는 RS(255,239,8) 복호기를 위한 KES 블록을 Verilog HDL를 사용하여 구현한 후, SMIC 0.13um library로 합성하였다. 제안된 구조는 pDCME와 비교했을 때, 각 PE 블록에서 사용된 지연 소자의 개수가 작아서 latency가 작고, 표 2에서와 같이 critical path delay가 작기 때문에 높은 주파수 클록에서 동작할 수 있음을 알 수 있다. 또한 다항식 연산을 통해 차수계산회로가 간단한 shift 회로로 구현하여 gate count가 적음을 볼 수 있다.

제안된 PE 구조는 오류정정능력 t와 무관하기 때문에, $GF(2^m)$가 바뀌지 않으면 t가 변화함에 따라 단순히 $2t$개의 PE를 그림 4와 같이 연결시키면 되기 때문에 KES 블록을 매우 쉽게 설계할 수 있는 특징을 가진다. 따라서 t가 증가함에 따라 KES 블록의 gate count가 선형적으로 증가하는 장점을 가진다. 또한, 제안된 구조에서는 PE 블록을 제어하기 위한 입출력 신호가 다른 구조에 비해 매우 간단하다.

표 2. KES 블록의 Critical path delay 비교

Architecture	Critical path delay
Proposed	$T_{and2}+3T_{xor2}+T_{ff}$
pDCME[5]	$T_{inv}+T_{and2}+3T_{mux2}+T_{ff}$
ME[4]	$3T_{or2}+T_{xnor2}+T_{mux2}+T_{ff}$

표 3. KES 블록의 합성 결과 비교

Architecture	Proposed	pDCME[5]	ME[4]
Technology	0.13um	0.13um	0.13um
Gate count	44,200	46,200	55,500
Clock Rate (MHz)	680	660	625
Latency	8t	10t	10t

V. 결 론

본 논문에서는 KES(Key Equation Solver) 블록 구현에 사용하는 ME(Modified Euclidean) 알고리즘을 효율적으로 설계할 수 있는 구조를 제안하고 구현하였다. 제안된 구조는 입출력 신호가 간단하고, 차수계산회로가 필요 없기 때문에 회로의 복잡도를 줄일 수 있다. 또한, 기존의 구조와 달리, PE 회로가 오류 정정 능력 t와 무관하기 때문에, t가 증가함에 따라 KES 블록의 하드웨어 복잡도가 선형적으로 증가하는 장점을 가진다. 실험 결과로부터, 제안된 구조를 이용하여 적은 gate count로 고속 RS 복호기 구현이 가능함을 알 수 있다.

참 고 문 헌

[1] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*, Englewood Cliffs, NJ, Prentice-Hall, 1995.

[2] H. Shao, T. Truong, L. Deutsch, J. Yuen, I. Reed, "A VLSI design of a Pipeline Reed-Solomon Decoder," *IEEE Trans. on Computers*, Vol.c-34, No.5, pp.393-403, May 1985.

[3] L. Song, M. Yu, M. Shaffer, "10- and 40-Gb/s Forward Error Correction Devices for Optical Communications," *IEEE Journal of Soild-State Circuits*, Vol.37, No.11, pp.1565-1573, Nov. 2002.

[4] Hanho Lee, "High-Speed VLSI Architecture for Parallel Reed-Solomon Decoder," *IEEE Trans. on VLSI Systems*, Vol.11, No.2, pp.288-294, April 2003.

[5] S. Lee, H. Lee, "A High-Speed Pipelined Degree-Computationless Modified Euclidean Algorithm Architecture for Reed-Solomon Decoders," *IEICE Trans. Fundamentals*, Vol.E91-A, No.3, pp.830-835, March, 2008.

[6] J. H. Baek and M. H. SunWoo, "New degree computationless modified Euclid's algorithm and architecture for Reed-Solomon decoder", *IEEE Trans. Very Large Integr. (VLSI) Syst.*, Vol.14, No.8, pp.915-920, Aug. 2006.

[7] J. H. Baek and M. H. SunWoo, "Enhanced degree computationless modified Euclid's algorithm for Reed-Solomon decoders," *Electronics Letters*,

Vol.43, No.3, pp.175-176, Feb., 2007.
[8] 강성진, 김한종 "UWB 시스템을 위한 RS(23,17) 복호기 최적 설계," *한국통신학회논문지*, Vol.33, No.8, pp.821-828, Aug., 2008.
[9] J. Proakis, M. Salehi, *Digital Communications*, McGraw-Hill, 5th ed., 2008

강 성 진 (Sung-Jin Kang) 종신회원

1998년 8월 연세대학교 전자공학과 공학박사
1998년 12월~2000년 1월 ETRI 무선방송기술연구소 선임연구원
2000년 2월~2002년 8월 (주) 이노텔리텍 기술이사
2002년 9월~2007년 2월 KETI 통신네트워크연구센터 책임연구원
2007년 3월~현재 한국기술교육대학교 전기전자통신공학부 부교수
<관심분야> WPAN, WLAN, MODEM SoC

김 남 용 (Nam Yong Kim) 종신회원

1988년 2월 연세대학교 전자공학과 공학석사
1991년 8월 연세대학교 전자공학과 공학박사
1992년 8월~1998년 2월 관동대학교 전자통신공학과 부교수
1998년 3월~현재 강원대학교 공학대학 전자정보통신공학부 교수
<관심분야> Adaptive equlization, RBFN algorithms, ITL algorithms, Odor sensing systems