

논문 2016-53-12-5

비트-직렬 LDPC 복호를 위한 효율적 AT 복잡도를 가지는 두 최소값 생성기

(Efficient AT-Complexity Generator Finding First Two Minimum Values for Bit-Serial LDPC Decoding)

이 재 학*, 선우 명 훈**

(Jea Hack Lee and Myung Hoon Sunwoo[©])

요 약

본 논문은 저면적 비트-직렬 두 최소값 생성기를 제안한다. Min-sum 복호 알고리즘을 적용한 LDPC 복호기에서 두 최소값 생성기가 가장 큰 하드웨어 복잡도를 가지기 때문에, 두 최소값 생성기의 저면적 구현이 매우 중요하다. 하드웨어 면적을 줄이기 위해 비트-직렬 방식의 LDPC 복호기가 제안되었다. 하지만 기존의 비트-직렬 방식의 생성기는 하나의 최소값만 찾을 수 있어 BER 성능이 감소되었다. 제안하는 생성기는 두 최소값을 모두 찾을 수 있어 BER 성능열화를 극복하고 저면적의 LDPC 복호기 구현이 가능하다. 또한 기존의 두 최소값 생성기들과 비교하여 면적-시간 복잡도에서 가장 좋은 성능을 보인다.

Abstract

This paper proposes a low-complexity generator which finds the first two minimum values using bit-serial scheme. A low-complexity generator is an important part for low-area LDPC decoders based on the min-sum decoding algorithm because the hardware complexity of generators utilizes a significant portion of LDPC decoders. To reduce hardware complexity, bit-serial LDPC decoders has been studied. The generator of the existing bit-serial LDPC decoders can find only the first minimum value, and thus it leads to a BER performance degradation. The proposed generator using bit-serial scheme finds the first two minimum values. Hence, it can improve the BER performance. In addition, the area-time complexity of the proposed generator is lower than those of the existing generators finding the first two minima.

Keywords : Low-density parity-check(LDPC) decoder, min-sum algorithm, area-time complexity, low-complexity design, bit-serial scheme

I. 서 론

LDPC(low-density parity-check) 부호^[1]는 뛰어난 오류정정 능력과 병렬구조를 통한 고속 동작의 특징을 가진다. 때문에, LDPC 부호는 플래시 메모리의 ECC(error correction code)^[2-3], 10 Gbps 이더넷 표준

(IEEE 802.3an)^[4], 디지털 방송 표준(DVB-S2)^[5] 등 다양한 표준에서 채널 부호화 방식으로 채택하고 있으며, LDPC 부호/복호 기술은 차세대 고속 통신 시스템 및 단말기의 핵심기술이 될 것이다.

LDPC 복호기에서 체크 노드 유닛(check node unit; CNU)은 행연산을 수행하기 위해 사용된다. 최적의 복

* 정회원, 아주대학교 전자공학과 (Department of Electrical and Computer Engineering, Ajou University)

** 평생회원, 아주대학교 전자공학과 (Department of Electrical and Computer Engineering, Ajou University)

© Corresponding Author(E-mail: sunwoo@ajou.ac.kr)

※ 본 논문은 미래창조과학부 중견연구자지원사업 [2014R1A2A2A01002952, 미래형 실감미디어 지원을 위한 영상처리 전용 프로세서 연구]의 지원에 의하여 연구되었음.

※ 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 대학ICT연구센터 육성지원사업 (IITP-2016- R2718-16-0015)의 결과로 수행되었음.

호 알고리즘인 sum-product 알고리즘을 사용하면 행연산을 위해 초월함수인 Gallager 함수 연산이 필요하다. 하지만 하드웨어로 구현하기 위해 큰 복잡도가 필요하여 min-sum 복호 알고리즘^[6]은 행 연산 입력 중 두 최소값과 각 출력의 부호만을 연산 하여 하드웨어 복잡도를 획기적으로 낮추었다. LDPC 코드^[7~9]는 병렬처리가 가능하여 수 Gbps의 높은 속도의 동작이 가능하다. 하지만, 병렬처리를 많이 할수록 하드웨어 복잡도와 와이어링(wiring) 복잡도가 매우 높아진다. 특히, min-sum 복호 알고리즘의 두 최소값 생성기가 LDPC 복호기에서 가장 큰 하드웨어 복잡도를 가진다^[10~11]. 따라서 LDPC 복호기의 고속의 처리율과 낮은 하드웨어 복잡도를 위해서 두 최소값 생성기의 저면적 구현이 중요하다.

기존의 비트-병렬 방식의 두 최소값 생성기들^[12~14]은 tree 또는 radix 구조를 사용한다. W ($W > 2$)개의 최소값 생성기^[15]는 비트 단위 연산을 사용하였다. 비트-병렬 방식의 생성기는 입력의 수가 증가할수록 하드웨어 복잡도가 기하급수적으로 증가한다. 때문에 긴 코드 길이를 가지는 LDPC 복호기에는 비트-병렬 방식의 구조는 부적합하다.

LDPC 복호기의 하드웨어 복잡도를 줄이기 위해 다양한 전 병렬구조를 사용한 비트-직렬 구조^[7~9]가 제안되었다. 근사 min-sum 알고리즘^[7]은 행연산에서 두 개의 최소값을 찾는 대신 비트-직렬 방식을 사용하여 최소값(\min_1)만 찾고, 두 번째 최소값(\min_2)은 근사치인 $\min_1 + 1$ 를 사용하여 연산을 진행한다. 입력 메시지의 길이가 n 비트라고 하였을 때, 비트-직렬 최소값 생성기는 $n+1$ 클럭 사이클이 필요하다. 따라서 하드웨어 복잡도는 감소하는 반면 BER = 10^{-6} 에서 약 0.5 dB의 성능열화^[9]가 발생한다. 이런 BER 성능 열화를 극복하기 위해 기존의 비트-직렬 방식 최소값 생성기^[7]를 두 번 사용하는 방법^[8]이 제안되었다. 처음에는 최소값을 찾은 뒤, 최소값을 제외한 입력 중에 다시 최소값을 찾아 두 최소값을 찾는 방식이다. 이 방식을 사용한 LDPC 복호는 BER 성능 열화를 극복하는 대신 긴 지연시간 ($2n$ 클럭 사이클)가 필요하여 고속 동작에 부적합하다.

본 논문에서 제안하는 두 최소값 생성기는 n 클럭 사이클 안에 두 개의 최소값을 모두 찾을 수 있다. 따라서 저면적, 고속을 목표로 하는 LDPC 복호기에 적합하다. 또한, 기존의 두 최소값 생성기들과 비교하여 가장 낮은 면적-시간(area-time) 복잡도를 가진다.

II. 기존 비트-직렬 최소값 생성기

기존의 비트-직렬 최소값 생성기^[7]는 m 개의 입력 메시지(X_0, X_1, \dots, X_{m-1}) 중 최소값(\min_1)과 최소값의 위치(\min_{idx})를 최소값 후보를 줄이는 방식을 사용하여 찾는다. Min-sum 복호를 위해 필요한 두 번째 최소값(\min_2)은 $\min_1 + 1$ 로 근사치를 사용한다. n -비트 메시지 X_i 는 다음과 같이 표현 할 수 있다.

$$X_i = x_i(n-1) \times 2^{n-1} + \dots + x_i(0) \times 2^0 \quad (1)$$

비트-직렬 생성기에 입력되는 입력 비트 $x_i(n-t)$ 는 매 사이클마다 X_i 의 MSB(most significant bit)부터 LSB(least significant bit)로 시간 t 에 따라 입력이 된다.

각 메시지 X_i 는 해당 메시지가 최소값 후보군인지 여부를 나타내는 상태 비트 $f_i(t)$ 를 가진다. $f_i(t) = 0$ 은 메시지 X_i 가 최소값의 후보군임을 나타내고, 해당 입력 비트 x_i 를 unmasked 입력 비트라고 표현한다. $f_i(t) = 1$ 은 X_i 가 최소값 후보군에서 제외되었음을 나타내고 x_i 를 masked 입력 비트라고 한다. 연산이 시작되었을 때, 모든 메시지들은 최소값 후보군으로 포함된다. 따라서 모든 상태 비트들은 0으로 초기화된다. 모든 연산이 끝난 후 $f_i(t) = 0$ 으로 유지되는 상태 비트의 메시지 X_i 가 \min_1 으로 결정된다.

그림 1은 입력이 4개($m=4$)인 기존의 비트-직렬 최소값 생성기를 나타낸다. i 번째 메시지(X_i)의 입력 비트($x_i(n-t)$)는 해당 상태 비트($f_i(t)$)와 왼편의 OR 게이트를 통해 연산된다. 만약 상태 비트 $f_i(t)$ 가 1인 경우, 해당 메시지(X_i)는 최소값 후보가 아니다. 따라서 입력비트($x_i(n-t)$)와 상관없이 다음 사이클에 상태

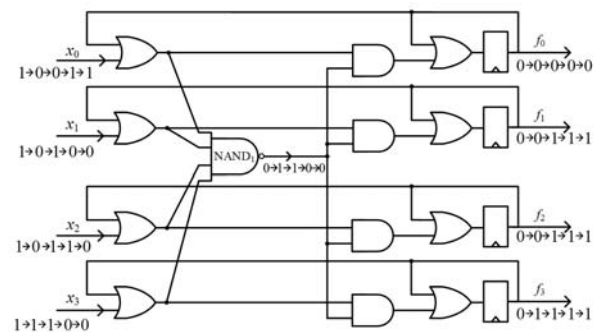


그림 1. 비트-직렬 최소값 생성기
Fig. 1. Minimum-value generator for the bit-serial scheme.

비트($f_i(t+1)$)은 1을 유지한다.

상태 비트 $f_i(t)$ 가 0인 경우, X_i 는 여전히 최소값 후보이며 다음 클럭의 상태 비트는 입력 비트에 의해 결정된다. 시간 t 에서 unmasked 입력 비트 중 0의 개수를 $N_0(t)$ 라고 정의한다. 만약 $N_0(t)$ 가 1 이상일 때, 입력 비트 $x_i(n-t)$ 가 1인 경우 해당 메시지 X_i 는 최소값 후보군에서 제외되고, 다음 사이클에 상태 비트 $f_i(t+1)$ 은 1이 된다. 하지만 $N_0(t)$ 가 0인 경우는 unmasked 입력 비트가 모두 1이 되어 최소값 후보군을 줄일 수 없다. 이 경우 그림 1의 $NAND_1$ 의 출력이 0이 되고 우측의 모든 AND 게이트들의 출력은 0이 된다. 따라서 다음 사이클에 모든 상태 비트의 값은 유지된다 ($f_i(t+1) = f_i(t)$).

간단한 예로 4개의 5-비트 메시지($m = 4$)를 다음과 같이 가정한다.

$$X_0 = 10011_2, X_1 = 10100_2$$

$$X_2 = 10110_2, X_3 = 11100_2$$

이 경우 최소값(\min_1)은 10011_2 이고 최소값의 위치(\min_{idx})는 '0'이 된다. 그림 1의 비트들은 위의 예시에 따른 해당 와이어의 비트 변화를 나타낸다. 연산이 끝난 후, X_0 의 상태 비트($f_0(5)$)가 0이고 나머지는 모두 1로 변화했다. 따라서 X_0 의 값 10011_2 이 \min_1 으로 선

택되며 '0'이 \min_{idx} 가 된다.

기존의 비트-직렬 방식^[7]은 \min_1 만을 찾을 수 있고, \min_2 는 $\min_1 + 1$ 을 근사치로 사용한다. 이러한 근사로 인해 BER 성능 열화를 야기하고 error floor^[8]를 높인다. 이런 BER 성능 열화를 제거하기 위해 [8]은 기존의 비트-직렬 생성기를 두 번 사용하여 두 최소값을 찾는다. 하지만 이럴 경우 긴 지연시간이 필요하여 데이터 처리율이 낮아지는 문제점이 발생한다. 본 논문은 BER 성능을 향상과 저면적 유지를 위해 비트-직렬 방식을 사용하는 두 최소값을 찾는 생성기를 제안하였다.

III. 제안하는 비트-직렬 두 최소값 생성기

이 장은 본 논문에서 제안하는 저면적 비트-직렬 두 최소값 생성기를 설명한다.

1. 제안하는 두 최소값 생성기 알고리즘

그림 2는 제안하는 비트-직렬 생성기의 순서도이고, 3가지(B1, B2, B3) 업데이트 방식이 있다. $\bar{x}(t)$ 는 $x(t)$ 의 반전된 값을 나타낸다. \min_2 를 찾기 위해 \min_1 을 찾았는지 여부를 나타내는 $\min_det(t)$ 신호를 추가하였다. 제안하는 두 최소값 생성기가 \min_1 을 찾을 때는 $\min_det(t) = 0$ 이 되고, \min_2 를 찾을 때는

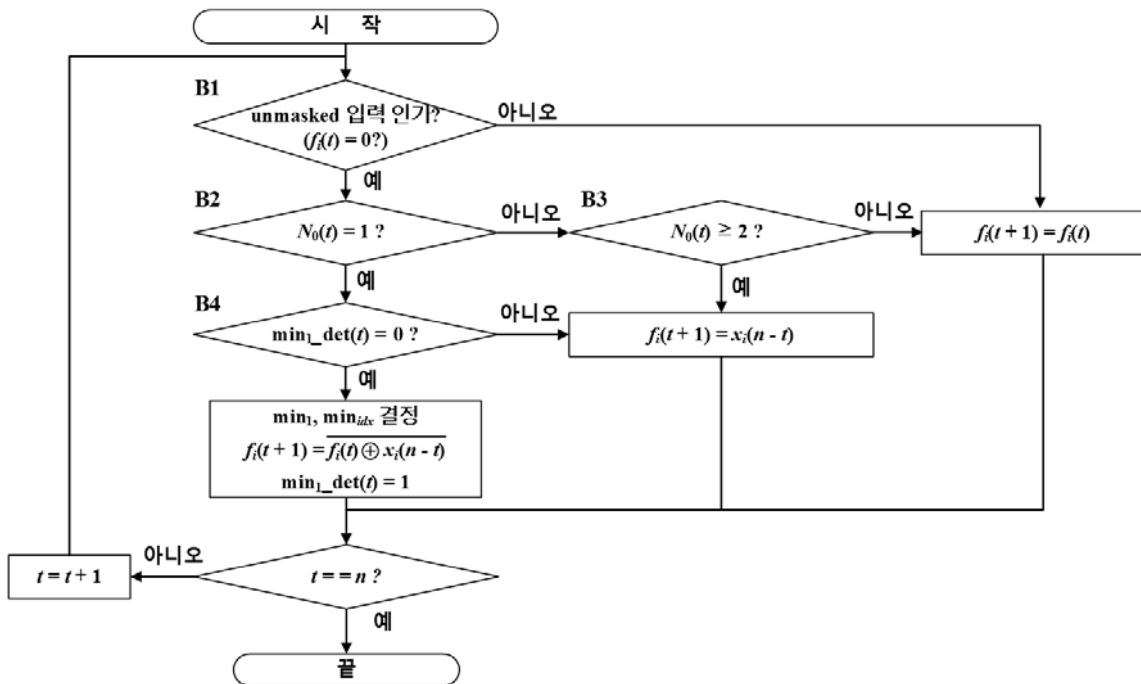


그림 2. 제안하는 비트-직렬 두 최소값 생성기 순서도

Fig. 2. Flowchart of the proposed bit-serial generator for finding the first two minimum values.

$\min_1\text{-det}(t) = 1$ 이 된다. \min_1 을 찾는 중 $N_0(t) = 1$ 이 되면(B3) 최소값 후보 중 1개의 unmasked 입력 비트가 0을 가진다. 이때 해당 입력 비트($x_i(t) = 0$)의 메시지(X_i)가 \min_1 으로 선정이 되고 \min_{idx} 는 i 가 된다. 이후, 제안하는 생성기는 \min_2 를 찾기 위해 $\min_1\text{-det}(t+1)$ 가 1이 된다.

초기화 과정은 모든 상태 비트가 0으로 설정되어 모든 입력을 최소값 후보로 선정한다. 입력은 각 메시지의 MSB부터 LSB로 매 사이클마다 한 비트 씩 입력된다.

상태 비트는 매 사이클 새로 입력되는 입력 비트에 의해 결정된다. $f_i(t) = 1$ 인 경우, 해당 입력 X_i 가 최소값 후보군에서 제외 되었으므로 계속 1로 유지가 된다.

$f_i(t) = 0$ 인 경우, $N_0(t)$ 에 의해서 다음 클럭 사이클의 상태 비트가 변한다. $N_0(t)$ 가 2 이상일 경우, unmasked 입력 비트가 1인 메시지들은 최소값 후보군에서 제외된다. 따라서 해당 메시지의 상태 비트는 다음 사이클에 1로 변환된다. $N_0(t)$ 가 0일 때, unmasked 입력 비트가 모두 1을 나타낸다. 이 경우 최소값 후보군을 줄일 수 없기 때문에 모든 상태 비트의 값은 유지된다.

$N_0(t) = 1$ 일 때, unmasked 입력 비트 중 1개만 0의 값을 가진다. 이때 \min_1 을 찾고 있다면($\min_1\text{-det}(t) = 0$), 해당 메시지를 최소값으로 설정하고, 다음 사이클부터 두 번째 최소값(\min_2)를 찾게 된다. 상태 비트는 \min_1 으로 선택된 메시지의 값만 1로 변하여 다음 사이클에 두 번째 최소값의 후보군에서 제외된다. 제안하는 생성기가 \min_2 를 찾고 있다면($\min_1\text{-det}(t) = 1$), 입력 비트가 0인 메시지가 \min_2 로 결정된다.

n 클럭 사이클 이후 (n -비트 메시지) 제안하는 생성기는 연산을 멈추게 된다. $\min_1\text{-det}(n) = 1$ 인 경우, \min_1 은 이전에 정해졌고 최종적으로 상태 비트가 0인 메시지가 \min_2 로 결정된다. $\min_1\text{-det}(n) = 0$ 일 때, 마지막까지 \min_1 을 결정하지 못하였고 2개 이상의 상태 비트가 0을 유지하는 경우다. 이 경우 2개 이상의 메시지가 같은 최소값을 가지고 있다는 것을 의미한다. 따라서 \min_1 과 \min_2 는 같은 값을 가진다.

그림 3은 2장의 예시를 사용하여 두 최소값을 찾는 방법을 보여준다. 제안하는 생성기 알고리즘을 사용하여 입력 비트에 의해 상태 비트의 변화를 보여준다. 회색 원과 흰색 원은 각각 상태 비트가 0, 1을 표현한다. 원 안의 숫자는 해당 메시지의 입력 비트를 나타낸다.

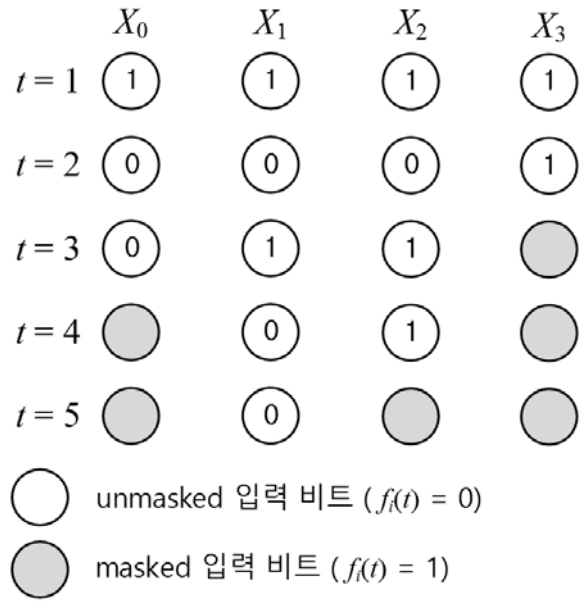


그림 3. 시간에 따른 제안하는 생성기의 입력 비트 및 상태 비트 변화
Fig. 3. Example of the transition of incoming bits and status flags for the proposed generator.

처음($t = 1$)에는 $\min_1\text{-det}(1)$ 가 0이 되고, 모든 메시지가 최소값의 후보가 되기 때문에 모든 상태 비트가 0으로 초기화된다. Unmasked 입력 비트가 모두 1이 되어 최소값 후보군을 줄일 수 없다. 또한, $N_0(1) = 0$ 이 되어 상태 비트는 변하지 않고 그림 2의 B1 연산을 통해 다음 사이클로 넘어간다.

$t = 2$ 에서, x_3 만 1의 값을 가지고 나머지 입력 비트는 모두 0이다. 따라서, X_3 만 최소값 후보군에서 제외되고 상태 비트 f_3 는 통해 1로 변한다. 그림 2의 순서도에 의하면 모든 상태 비트가 B2 연산을 수행하게 된다.

$t = 3$ 에서 $\min_1\text{-det}(3)$ 은 0이고, $N_0(3)$ 가 1이 된다. 유일한 0값을 가지는 입력 비트 $x_0(5-3)$ 의 메시지가 \min_1 으로 선택된다. 따라서 \min_1 은 X_0 , \min_{idx} 는 '0'로 결정된다. 최소값을 찾았기 때문에 다음 사이클에 $\min_1\text{-det}$ 신호는 1로 변하고 제안하는 생성기는 \min_2 를 찾게 된다. 그림 2의 순서도에 따라 B3 연산을 수행하게 된다. 상태 비트는 최소값으로 선택된 f_0 이 1로 변하여 두 번째 최소값 후보군에서 제외되어 \min_2 를 찾는 연산을 수행한다.

$t = 4$ 에서, $N_0(4) = 1$, $\min_1\text{-det}(4) = 1$ 이기 때문에 \min_2 값을 결정 할 수 있다. $x_1(5-4) = 0$ 이기 때문에 \min_2 는 X_1 로 결정이 되고, 해당 상태 비트를 제외하고 모두 1로 변하여 연산이 끝나게 되고, 순서도의

B2 연산으로 상태비트들이 갱신된다.

결과적으로 \min_1 은 X_0 , \min_{idx} 는 '0', \min_2 는 X_1 로 두 최소값을 찾을 수 있다. 지연시간은 5 사이클로 메시지 비트의 길이와 같아 기존의 비트-직렬 최소값 생성기^[7]와 비교하여 한 사이클이 줄어든 지연시간을 가진다.

2. 제안하는 두 최소값 생성기의 하드웨어 구조

제안하는 생성기의 회로는 그림 4(a)와 같이 상태 비트 업데이트 모듈, 제어 모듈, 출력 모듈로 나누어진다. 생성기의 입력은 메시지들의 MSB부터 LSB로 입력이 된다. \min_1 을 찾았을 때 상태 비트의 업데이트를 위한 sel 신호가 추가되었다. \min_1 을 찾지 못했을 때는 sel 신호가 0이 되어 기존의 비트-직렬 최소값 생성기^[7]와 같은 방식으로 상태 비트를 업데이트한다. 반면, \min_1 을 찾으면 sel 신호가 1이 되어, \min_1 의 상태 비트 값이 1로 업데이트 된다.

제어 모듈은 \min_2 를 찾기 위해 $N_0(t)$ 을 사용하여 $\min_{1_det}(t)$ 와 $sel(t)$ 신호를 생성한다. \min_{1_det} 신호는 0으로 초기화 되어 제안하는 생성기는 \min_1 을 찾는다. $\min_{1_det}(t)$ 가 0이고 $N_0(t)$ 가 1일 때, 생성기는 최소값 \min_1 을 결정할 수 있다. 이때 sel 은 1이 되어 최소

값의 상태 비트를 1로 바꾸고 다음 사이클에 \min_{1_det} 가 1로 변환($sel(t)=1, \min_{1_det}(t+1)=1$).

그림 4(b)는 상태 비트 업데이트 모듈을 나타낸다. 와이어 위의 비트는 위 예시에 나온 메시지를 사용한 시간에 따른 비트의 변화를 나타낸다. 그림 1의 기존의 비트-직렬 최소값 생성기^[7]와 비교하여 메시지 당 XNOR게이트와 2×1 MUX가 하나씩 추가된다. 따라서 매우 작은 하드웨어 추가로 두 최소값을 모두 찾을 수 있다.

표 1은 제안하는 생성기의 \min_1, \min_2 찾는 예시를 표현하였다. 흰색과 회색 칸은 각각 unmasked, masked 입력 비트를 나타낸다. $t=1$ 에서 모든 입력 비트가 1을 가진다. 그림 4(b)의 $NAND_1$ 의 출력이 0이 되어 모든 상태 비트가 이전 값을 유지한다. $t=2$ 에서 x_3 만 1을 가진다. 따라서 X_3 는 최소값 후보군에서 제외되어 다음 사이클($t=3$)에서 상태 비트 $f_3(3)$ 가 1로 변환한다.

$t=3$ 에서 $N_0=1, \min_{1_det}=0$ 이 되어 최소값을 결정할 수 있게 된다. unmasked 입력 비트 중 x_0 만 0의 값을 가져 해당 입력 X_0 가 \min_1 으로 결정 된다. 상태 비트 업데이트를 위해 제어 모듈에서 sel 신호가 1이 되어 상태 비트 업데이트 모듈로 전달이 된다.

마지막 사이클($t=5$)에서 $\min_{1_det}(5)=1$ 로 최소

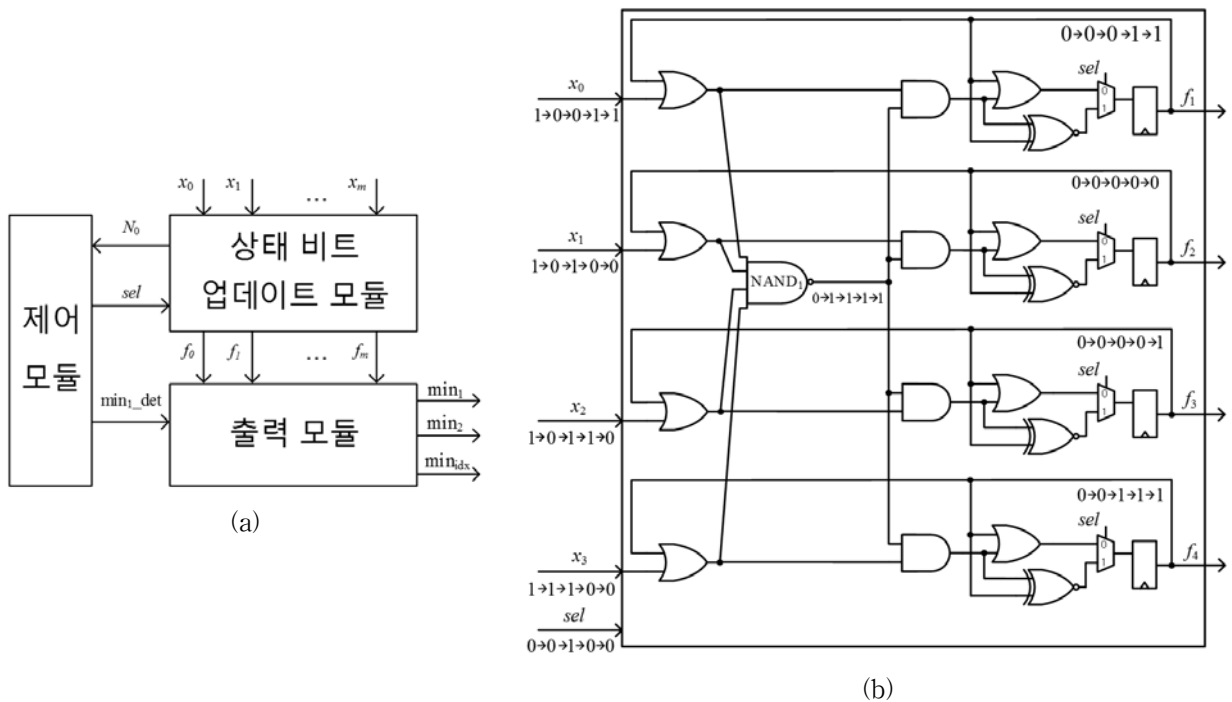


그림 4. 제안하는 두 최소값을 찾는 비트-직렬 생성기. (a) 제안하는 생성기의 블록다이어그램, (b) 상태 비트 업데이트 모듈
Fig. 4. Proposed bit-serial generator for finding the first two minimum values. (a) Block diagram of the proposed generator, (b) Flag update module.

표 1. 제안하는 생성기를 사용한 4개 5-비트 메시지를 활용한 예시

Table 1. Example of bit-serial minimum-value generation for four 5-bit messages using the proposed generator.

	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$
$x_0(5-t)/f_0(t)$	1 / 0	0 / 0	0 / 0	1 / 1	1 / 1
$x_1(5-t)/f_1(t)$	1 / 0	0 / 0	1 / 0	0 / 0	0 / 0
$x_2(5-t)/f_2(t)$	1 / 0	0 / 0	1 / 0	1 / 0	0 / 1
$x_3(5-t)/f_3(t)$	1 / 0	1 / 0	1 / 1	0 / 1	0 / 1
$\min_{1_det}(t)$	0	0	0	1	1
$sel(t)$	0	0	1	0	0
$N_0(t)$	4	3	1	1	1

값(\min_1)이 결정되었을 알 수 있다. 상태 비트가 0을 유지하는 것은 f_1 로 X_1 이 \min_2 로 결정된다. 따라서 제안하는 생성기는 메시지의 비트수와 같은 5 사이클 만에 두 최소값을 모두 결정할 수 있다.

제안하는 생성기는 n 클럭 사이클(n -비트 메시지)에 두 최소값을 찾을 수 있어 기존의 [8]와 비교하여 지연 시간이 반으로 줄어든다. 또한, [7]의 \min_2 근사 방식을 사용하지 않고 정확한 \min_2 를 찾아 BER 성능을 향상 시켰다.

IV. 합성 결과

비트-병렬 구조^[12-15]는 처리율을 높일 수 있는 반면 큰 하드웨어 복잡도를 가진다. 반면에, 비트-직렬 구조는 낮은 처리율을 가지지만 매우 낮은 하드웨어 복잡도를 장점이 있다. 따라서 두 구조를 연산시간 또는 하드웨어 면적으로 비교하는 것은 공정하지 않아 비트-직렬 방식을 사용한 최소값 생성기들^[7-8]과 비교하였다.

제안하는 비트-직렬 두 최소값을 찾는 생성기는 Synopsys사의 design compiler를 삼성 65nm CMOS 공정을 사용하여 합성하였다. 회로들은 일반적으로 면적, 연산 시간, 연산 파워 또는 이들을 결합한 방식을 통해 성능을 비교한다. 주로 사용되는 성능 분석 방법인 면적-시간(AT; area-time) 복잡도^[14-15]를 사용한다. 본 논문은 아래와 같이 AT 복잡도를 사용하여 회로들을 분석하였다.

$$AT \text{ 복잡도}(\mu m^2 \cdot ns) = \text{회로면적}(\mu m^2) \times \text{연산시간}(ns) \quad (2)$$

연산시간은 다음과 같이 구할 수 있다.

$$\text{연산시간}(ns) = \text{지연시간}(clock) / 100 \times \text{동작주파수}(MHz) \quad (3)$$

기존의 비트-직렬 최소값 생성기^[7]는 \min_1 을 찾기 위해 n 클럭 사이클이 필요하다. 하지만 두 번째 최소값으로 사용할 $\min_1 + 1$ 을 연산 위해 추가적인 한 사이클이 더 필요하다. 따라서 모든 연산을 끝내기 위해서 $n+1$ 클럭 사이클이 필요하다. 기존의 비트-직렬 생성기를 활용하여 두 최소값을 찾는 방식이 [8]에서 제안되었다. 이 회로는 두 번 연산을 통해 \min_1 , \min_2 를 찾기 때문에 총 $2n$ 클럭 사이클이 필요하다. 반면에 제안하는 생성기는 \min_1 과 \min_2 를 동시에 찾기 때문에 n 클럭 사이클에 연산이 완료된다.

공평한 비교를 위해 기존의 합성 결과를 입력 36개 ($m = 36$)인 생성기로 삼성 65nm CMOS 공정으로 새로 합성하였다. 제안하는 회로와 기존의 비트-직렬 회로^[7-8]들은 모두 1 GHz 이상에서 동작 할 수 있다. 하지만 높은 동작 주파수는 큰 파워 소모를 야기하므로 동작주파수를 500 MHz로 제한하였다. 비트-직렬 LDPC 복호기의 동작주파수가 500 MHz일 경우 약 30 Gb/s의 처리율을 낼 수 있고, 이는 플래시 메모리^[2-3], 10 Gbps 이더넷^[4], DVB-S2^[5] 등 고속의 처리율을 요구하는 표준을 모두 만족 가능하다.

표 2는 입력이 36개($m = 36$), 각 입력 메시지가 5 비트($n = 5$)일 때의 각 생성기 합성 결과를 나타낸다. [7]의 생성기는 낮은 하드웨어 복잡도를 가지지만 최소값 하나만 생성할 수 있어 근사 min-sum 복호 알고리즘을 사용한다. 따라서, 두 최소값을 모두 생성하여 복호하는 min-sum 알고리즘과 비교하여 BER 성능이

표 2. 비트-직렬 생성기 비교($m = 36, n = 5$)

Table 2. Comparisons of bit-serial generators ($m = 36, n = 5$).

	[7]	[8]	제안하는 생성기
공정(nm)	65	65	65
BER 성능 열화	-0.5 dB	0 dB	0 dB
면적(μm^2)	647.9	830.4	985.3
동작주파수 (MHz)	500	500	500
지연시간(클럭)	6	10	5
연산 시간(ns)	12	20	10
AT 복잡도	7,775 (78.9%)	16,608 (168%)	9,853 (100%)

0.5dB 감소하게 된다. 비트-직렬 방식으로 LDPC 복호기의 하드웨어 복잡도를 감소 시켰지만, BER 성능 열화로 인해 높은 오류 정정 능력은 요구하는 통신에 부적합하다. [8]의 생성기는 기존 [7]의 생성기를 두 번 반복하여 BER 성능 열화를 극복하였다. 하지만, 두 번 반복으로 인해 2배 긴 지연시간이 필요하다. 따라서, [8]을 사용한 LDPC 복호기는 오류 정정 능력은 유지하는 반면, 고속 동작을 요구하는 표준에는 적합하지 않다. 제안하는 생성기는 비트-직렬방식의 장점인 저면적을 유지하면서 두 최소값을 모두 찾아 BER 성능 열화를 극복하였다. 또한, 한번의 연산으로 두 최소값을 찾아내 고속 동작에도 적합하게 설계되었다.

V. 결 론

본 논문은 비트-직렬 방식을 사용한 두 최소값을 찾는 저면적 알고리즘 및 그 회로를 제안하였다. 기존의 비트-직렬 LDPC 복호기는 행연산에 \min_2 대신 근사치를 사용하여 BER 성능 열화가 발생하였다. BER 성능 향상을 위해 제안하는 생성기는 지연시간의 증가 없이 정확한 두 최소값을 찾는다. 따라서 플래시 메모리의 ECC^[2-3], 10 Gbps 이더넷^[4], DVB-S2^[5] 표준 등 저면적/고속 LDPC 복호기의 두 최소값 생성기로 적합하다.

REFERENCES

- [1] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA, USA: MIT Press, 1963.
- [2] J. Kim and W. Sung, "Rate-0.96 LDPC decoding VLSI for soft-decision error correction of NAND flash memory," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 5, pp. 1004 - 1015, May 2014.
- [3] G. Dong, N. Xie, and T. Zhang, "On the use of soft-decision error correction codes in NAND flash memory," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 2, pp. 429 - 439, Feb. 2011.
- [4] IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements Part 3: Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, IEEE Std.802.3an, Sep. 2006.
- [5] A. Morello and V. Mignone, "DVB-S2: The second generation standard for satellite broad-band services," *Proc. IEEE*, vol. 94, no. 1, pp. 210 - 227, Jan. 2006.
- [6] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2004.
- [7] A. Darabiha, A. C. Carusone, and F. R. Kschischang, "Power reduction techniques for LDPC decoders," *IEEE J. Solid-State Circuits*, vol. 43, no. 8, pp. 1835 - 1845, Aug. 2008.
- [8] F. Cai, X. Zhang, D. Declercq, S. Planjery, and B. Vasic, "Finite alphabet iterative decoders for LDPC codes: Optimization, architecture and analysis," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 5, pp. 1366 - 1375, May 2014.
- [9] P. A. Marshall, V. C. Gaudet, and D. G. Elliott, "Deeply pipelined digit serial LDPC decoding," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 12, pp. 2934 - 2944, Dec. 2012.
- [10] J. Li, J. Ma, and G. He, "A memory efficient parallel layered QC-LDPC decoder for CMMB systems," *Integr., VLSI J.*, vol. 46, no. 4, pp. 359 - 368, Sep. 2013.
- [11] F. Gutierrez, G. Corral-Briones, D. Morero, T. Goette, and F. Ramos, "FPGA implementation of the parity check node for min-sum LDPC decoders," in *Proc. Conf. Programmable Logic*, Mar. 2012, pp. 1 - 6.
- [12] C. Wey, M. Shieh, and S. Lin, "Algorithms of finding the first two minimum values and their hardware implementation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 11, pp. 3430 - 3437, Dec. 2008.
- [13] S. Jung, K. Shin. "A design of LDPC decoder for IEEE 802.11n wireless LAN," *The Institute of Electronics Engineers of Korea - Semiconductor and Devices*, vol. 62, no. 5, pp. 31-40. Nov. 2010.
- [14] Y. Lee, B. Kim, J. Jung, and I.-C. Park, "Low-complexity tree architecture for finding the first two minima," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 62, no. 1, pp. 61 - 64, Jan. 2015.
- [15] G. Xiao, M. Martina, G. Masera, and G. Piccinini, "A parallel radix sort-based VLSI architecture for finding the first W maximum/minimum values," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 61, no. 11, pp. 890 - 894, Nov. 2014.

저 자 소 개



이 재 학(정회원)
2009년 아주대학교 전자공학 학사 졸업
2011년 아주대학교 전자공학 석사 졸업
2011년~현재 아주대학교 전자공학 박사

<주관심분야 : SoC 설계, 오류정정코드, 유무선 통신시스템, 저전력 설계, 통신용 ASIP설계, 통신알고리즘>



선우 명 훈(평생회원)
1980년 서강대학교 전자공학 학사 졸업.
1982년 한국과학기술원 전자공학 석사 졸업.
1982년~1985년 한국전자통신연구소(ETRI) 연구원
1985년~1990년 Univ. of Texas at Austin 전자공학 박사.

1990년~1992년 Motorola, DSP Chip Division (미국)
1992년~1996년 아주대학교 전기전자공학부 조교수
1996년~2001년 아주대학교 전자공학부 부교수
2001년~현재 아주대학교 전자공학부 교수
2011년~현재 IEEE Fellow Member
2011년~현재 IEEE CASS Board of Governor
2012년~2014년 대한전자공학회 반도체 소사이어티 회장
2014년~현재 대한전자공학회 반도체 소사이어티 자문위원
2016년~현재 아주대학교 ITRC uDSD 연구센터장
<주관심분야 : ICT SoC 설계, ASIP 설계, 저전력 의료기기 디바이스 설계>