



### 工學博士 學位論文

# 효율적인 위성/지상 전송을 위한 저전력 고속 LDPC 복호 알고리즘 및 FPGA 구현 에 관한 연구

A Study of an FPGA Design and Analysis of Low Power High-Speed LDPC Decoding Scheme for Efficient Satellite/Terrestrial Transmission System

指導教授 鄭 智 元

2012年 8月 韓國海洋大學校 大學院

電波工學科

金 慜 赫

本 論文을 金額赫의 工學博士 學位論文으로 認准함.



#### 2012年 8月

韓國海洋大學校 大學院

#### 電波工學科

#### 金额赫



그림 목차iv
표 목차vi
기호vii
약어viii
ABSTRACTix
제 1 장 서론1
제 2 장 효율적인 위성/지상 전송시스템의 부호화 알고리즘
제 2.1 절 LDPC 부호화 방식5
2.1.1 Long-size LDPC 부호화 방식6
2.1.2 Short-size LDPC 부호화 방식16
제 2.2 절 e-RS 부호화 방식
제 2.3 절 Cross Layer 부호화 방식24
제 3 장 LLR 기법을 이용한 효율적인 부호화 방식 제안27
제 4 장 위성/지상 저전력 고속 전송을 위한 LDPC 복호 알고리즘
제 4.1 절 저복잡도 알고리즘
4.1.1 Early Stop 알고리즘33
4.1.2 Early Detection 알고리즘 제안
제 4.2 절 고속 복호 알고리즘40
4.2.1 HSS 및 VSS 알고리즘41
4.2.2 Self-Correction 알고리즘 기반의 CNU 알고리즘 제안54



제	5 7	상 저전력 고속 LDPC 복호기 FPGA 구현	54
	제	5.1 절 기존의 LDPC 복호기 구조	54
	제	5.2 절 HSS 기반의 고속 복호기 구조 제안	57
		5.2.1 고속 rotator 알고리즘	59
		5.2.2 구현을 위한 최적의 복호기 구조	70
	제	5.3 절 FPGS 구현 및 비교분석7	'3

제 6	장	결	론	77
-----	---	---	---	----

참고문헌
------





## 그림 목차

그림 2.1 패리티 검사 행렬의 부 행렬	.6
그림 2.2 LDPC 전체 복호 과정	.7
그림 2.3. 부호율에 따른 성능 곡선(반복횟수=40, N=64800)1	0
그림 2.4. 각 부호화 별 반복 횟수 변화에 따른 성능 곡선(N=64800)1	6
그림 2.5.802.11n에 제시된 페리티 검사 행렬(N=1944, Z=81, rate=1/2)1	7
그림 2.6. 802.11n LDPC 부호화기의 구조1	9
그림 2.7. 다양한 부호화율에서 BER 성능 곡선2	0
그림 2.8. CRC 기반의 삭제된 MPE-FEC 메모리 형태2	4
그림 2.9. MPE-FEC 프레임 구조 및 가상 인터리버2	5
그림 3.1. LLR 기반의 복호 방식 블록도2	8
그림 3.2. δ 값의 변화에 따른 MPE-FEC의 성능	)
그림 3.3 제거된 심볼들 중 에러 심볼의 평균 개수	1
그림 3.4 기존의 방식과 제안한 방식의 성능 비교	2
그림 4.1 Early-stop 알고리즘 블록도	4
그림 4.2 두 가지 방식의 early-stop 알고리즘 블록도	5
그림 4.3 early stop 알고리즘의 BER 성능곡선	7
그림 4.4 Bipartite 그래프	8
그림 4.5 $T_c$ 의 변화에 따른 BER 성능곡선	9
그림 4.6 기존의 LDPC 복호기의 구조4	1
그림 4.7 HSS 복호 방식의 흐름도4	12
그림 4.8 HSS 방식의 노드 연산 구조4	14
그림 4.9 메모리 연결 충돌의 예4	-6
그림 4.10 HSS의 반복횟수에 의한 성능 비교(부호화 율=1/2)4	7
그림 4.11 HSS의 반복횟수에 의한 성능 비교 (부호화 율=1/4)4	7
그림 4.12 HSS의 반복횟수에 의한 성능 비교 (부호화 율=2/3)4	8
그림 4.13 HSS의 반복횟수에 의한 성능 비교 (부호화 율=3/4)44	8
그림 4.14 HSS의 반복횟수에 의한 성능 비교 (부호화 율=4/5)49	9



그림 4.1	15 HSS의 반복횟수에 의한 성능 비교 (부호화 율=8/9)	49
그림 4.1	16 VSS 방식의 초기화 단계	.51
그림 4.1	17 VSS 방식의 비트 노드 업데이트	.51
그림 4.1	18 VSS 방식의 체크 노드 업데이트	.52
그림 4.1	19 VSS 방식의 구현을 위한 구조	.52
그림 4.2	20 VSS 방식에서 복호과정의 예	.53
그림 4.2	21 VSS 방식에 대한 LDPC 복호기 구조	54
그림 4.2	22 α값에 따른 NMS 방식의 성능	.56
그림 4.2	23 SP 방식과 NMS 방식의 성능	.57
그림 4.	.24 NMS 방식 적용시 CNU의 구조	58
그림 4.2	25 기존 방식의 CNU 구조	58
그림 4.2	26 SP, MS 방식과 NMS 방식의 성능	59
그림 4.2	27 Self-Correction 방식의 구조	.60
그림 4.2	28 SP 방식과 SC-SP 방식의 성능	61
그림 4.2	29 MS 방식과 SC-MS 방식의 성능	61
그림 4.3	30 NMS 방식과 SC-NMS 방식의 성능	62
그림 5.1	1q그룹으로 분할된 비트/체크 노드에 대한 Tanner 그래프6	54
그림 5.2	2 일반적인 LDPC 복호기의 구조	.65
그림 5.3	3 Message 메모리의 구조	.66
그림 5.4	4 Rotation 구조	67
그림 5.5	5HSS 방식을 기반으로 한 복호기의 구조	68
그림 5.6	6 제안된 rotator 구조	70
그림 5.7	7 HSS 기반의 LDPC 복호기의 구조	71
그림 5.8	8 CNU 연산과정	72
그림 5.9	9 SC-NMS 방식에 대한 최대 클럭 속도	75
그림 5.1	10 SP 방식에 대한 최대 클럭 속도	75



## 표 목차

표 2.1 부호화율에 따른 최적의 반복횟수	17
표 2.2 802.11n에서 제공하는 각 부호화율에 따른 파라메타	21
표4.1 강판정 알고리즘 일 때의 평균 정지된 반복 횟수	36
표 4.2. 패리티 검사 식 알고리즘 일 때의 평균 정지된 반복 횟수	36
표 4.3 LDPC 복호기에 필요한 계산량	39
표 4.4 early detected 된 수	40
표 4.5 HSS 알고리즘 적용시 요구되는 반복횟수	50
표 5.1 부호화 파라메타	73
표 5.2 구조 파라메타	73
표 5.3 구현 시 클럭 소모 량과 속도	74
표 5.4 CNU 연산 방식에 따른 클럭 속도와 복호 throughput	76





기호

 $\alpha$ : normalization factor c(x): 송신 다항식 dc: LDPC 복호에서 각 체크 노드와 연결된 엣지 수(row weight)  $d_{a}^{*}$  : 각 체크 노드 당 early detection 한 엣지의 평균 수 δ :LDPC 복호 데이터의 신뢰도 기준 값 δfn : 비트 노드의 초기값 dv : LDPC 복호에서 각 비트 노드와 연결된 엣지 수(column weight) e(x): 오류 다항식 f(x): 삭제 다항식 Γ(x): 삭제 위치 다항식 L : 비트 노드의 LLR 값 Λ(x): 오류 위치 다항식 M,m:H 행렬에서의 세로 길이 N,n:H 행렬에서의 가로 길이(LDPC), 부호화된 전체 심볼 수(RS) N\* : 비트 노드를 early detection한 평균 수 q: 체크 노드의 그룹 수 r<sub>n</sub> : 수신 데이터 r(x): 수신 다항식 **σ** : 잡음 분산 t:RS부호의 에러 정정 능력 u: 체크 노드에서 비트 노드로 연결된 엣지 v : 비트 노드에서 체크 노드로 연결된 엣지 v<sub>n</sub> : LDPC 복호 데이터의 연판정 값 Z:802.11n에서의 H 행렬을 이루는 부 행렬의 크기



## 약어

- AWGN : Additive White Gaussian Noise
- BNU : Bit Node Update
- CNU : Check Node Update
- CRC : Cyclic Redundancy Check
- DVB : Digital Video Broadcasting
- DVB-H: DVB-Handheld
- DVB-SSP : DVB-Satellite Services to Portable
- DVB-S2 : DVB-Satellite 2
- DVB-T: DVB-Terrestrial
- e-RS : erasure RS
- FEC : Forward Error Correction
- FPGA : Field Programmable Gate Array
- HSS : Horizontal Shuffle Scheduling
- IP : Internet Protocol
- LDPC : Low Density Parity Check
- LLR : Log Likelihood Ratio
- LTF : Long Term Fading
- LUT : Look Up Table
- MPE-FEC : Multi-Protocol Encapsulation-FEC
- MS : Min Sum
- NMS : Normalization MS
- OMS : Offset MS
- PL-FEC : Physical Layer-FEC
- RS : Reed Solomon
- SC : Self Correction
- SP : Sum Product
- UL-FEC : Upper Layer-FEC
- VHDL : VHSIC Hardware Description Language
- VSS : Vertical Shuffle Scheduling



#### ABSTRACT

Efficient support for digital video broadcasting (DVB) services in mobile systems is crucial issue for mobile operators and broadcasters. Currently several DVB systems have been developed internationally, to meet the requirements of broadcasters. The representative standards are DVB-SSP and DVB-S2 specifications for mobile and broadband multimedia services by using hybrid satellite and terrestrial transmission schemes. The standards for satellite and terrestrial transmission such as DVB-S2 and DVB-T2 use the Low Density Parity Check(LDPC) code to correct transmission errors.

In a long-term fading environment, the MPE-FEC(Multi-Protocol Encapsulation-Forward Error Correction) scheme which adopts the erasure Reed Solomon (e-RS) code and the virtual interleaver is used to compensate the fading effects.

The e-RS can correct more erroneous code symbols than the conventional nonerasure Reed Solomon code, which is one of the advantages of e-RS code. The erasure information can be obtained from CRC error detection. However, a critical factor must be considered for dealing with a long-term fading channel. In fact, at the end of every MPE-FEC section, there is a CRC-32 field calculated over the entire section, to detect erroneous sections at the receiver side. If there is only one bit error in an IP packet, the entire IP packet is considered as unreliable bytes, even if it contains correct bytes. This implies that if there is one real byte error in a IP packet of 512 bytes, 511 correct bytes should be erased from the frame. This causes performance degradation in erasure RS decoding scheme.

This paper has proposed new efficient decoding algorithms for MPE-FEC code and the DVB-S2 LDPC code. A new MPE-FEC decoding algorithm based on LLR infomation has been proposed to increase the decoding performance. This paper proposed a new MPE-FEC decoding algorithm based on LLR. If the number of erased bytes is less than the error correction capability, an erasure RS code is used, otherwise, a non-erasure decoder that buffered another received symbol memory is used. The main difference between the decoding methods based on CRC and LLR analyzed in this paper in terms of performance, is the manner in which the erasure information is obtained and it is utilized.



In the study of LDPC decoding algorithm, two kinds of LDPC decoding algorithm have been studied to reduce the computational complexity and decoding speed. There are two kinds of low computational complexity algorithms such as early edge detection algorithm and early stop algorithm. Also, in the high-speed LDPC decoding algorithms, Horizontal Shuffle Scheduling(HSS) and Vertical Shuffle Scheduling(VSS) algorithm can be used. In case of CNU methods, SP (Sum Product) method is often used in LDPC decoding. However SP needs LUT (Look UP Table) that is critical path in LDPC decoding speed [4]. In this paper, a new SC-NMS method is proposed. The SC-NMS method needs only normalized scaling factor instead of LUT, which reduces the complexity of computation for SP approximation. As a result, the number of iteration can be reduced to the half of the conventional algorithm, algorithm with the same performance of the SP algorithm.

Finally, in this paper, the hardware structure of high-speed LDPC decoder impelementation has been proposed and its throughtput has been also investigated. The LDPC decoder has been implemented based on HSS algorithm by using FPGA chip xc5vlx155t model. A new memory structure and data permutation method has been proposed and verified by real hardware experiment. This new structure has been compared with conventional structure in the point of decoding speed. According to the result, the decoding throughput of the proposed structure based on SC-NMS algorithm has been improved by 40%.



### 제1장서론

언제, 어디서나 시청 가능한 방송서비스를 위해 차세대 방송통신은 지상 파 방송과 위성과의 융합에 관한 연구가 활발히 진행 중에 있으며, 특히 위성을 이용한 쌍방향 통신, 즉 무선 인터넷과의 융합, 이동체에 대한 위성 방송 및 인터넷 통신을 하기 위한 연구가 제안되고 있다. 이동체에 대한 위성방송 및 인터넷 서비스를 지속적으로 제공하기 위해 기존의 DVB-S2 표준화에 DVB-H 와 DVB-T 를 결합한 새로운 Digital Video Broadcasting-Satellite Services to Portable(DVB-SSP) 표준화에 대한 연구와 함께, 이는 이동 체에 대해서 위성을 이용한 통신 방식을 규정하고 있으며, physical layer(PL) 와 upper layer(UL) 의 두 단계로 부호화 및 복호화 방식인 cross layer 부호 화 방식을 적용하고 있다. PL 에서 정정하지 못한 오류를 UL 에서 정정함 으로써 오류 정정 능력을 한층 강화 시킨 것이다[1][2][3].

본 논문은 크게 세가지로 나눌 수 있다. 첫째로 효율적인 위성/지상 하이 브리드 전송을 위한 부호화 방식인 DVB-SSP의 cross layer 부호화 방식의 알고리즘에 대하여 제 2장에서 분석한다. 둘째로는 현재 DVB-SSP에서 적 용중인 CRC 기반의 cross layer 부호화 방식의 성능면에서의 비효율성 극복 을 위해 새로운 형태인 LLR 기반의 cross layer 부호화 방식을 제 3장에서 제안한다. 마지막으로 PL-FEC(PL-Forward Error Correction) 부호화 알고리즘 인 LDPC 부호는 많은 계산 량과 반복 횟수로 인해 높은 전력이 필요하고, 고속화가 힘들다. 이를 극복하기 위해 제 4장에서는 LDPC 복호기의 복잡 도와 복호 속도를 개선하기 위한 저전력 고속 알고리즘을 분석, 제안하였 고, 제 5장에서는 이를 적용시킨 LDPC 부호의 복호기를 FPGA 설계 및 구 현하여 기존의 방식과 비교 분석 하였다.

최적의 부호화 방식 측면에서 이동중인 물체에 대해 Internet Protocol(IP) 데이터를 전송하기 위한 DVB-SSP 시스템에서 적용되고 있는 부호화 방식 은 UL-FEC로는 erasure RS(e-RS) 부호와 가상 인터리버를 결합한 DVB-H 의 MPE-FEC(Multi-Protocol Encapsulation-FEC) 구조를 사용한다. e-RS 부호의



1

장점은 기존의 RS 부호보다 더 많은 오류 정정이 가능하다는 점이다. PL-FEC에 적용되는 LDPC 부호는 기존의 DVB-S2 표준안에서 적용되고 있는 부호화 알고리즘을 적용시켰다. 기존의 방식 경우, e-RS 부호 후 cyclic redundancy check(CRC) 를 포함하여 가상 인터리버를 거친 후에 LDPC 부호 화되어 전송된다. 수신측에서는 LDPC 복호후 CRC 검사를 수행하여 각 IP 패킷 내에 오류가 발생했음이 판단되어 지면 그 IP 패킷 전체를 삭제한 후 e-RS 복호를 거치게 된다. 이 때, CRC 검사에서는 수신된 IP 패킷 내에 단 1 bit 의 오류에도 IP 패킷 모두를 삭제를 하게 된다. 이것은, 만약 IP 패킷 의 크기가 512 바이트이고, 그 512 바이트 중 단 1bit 만이 오류라 하더라 도, 1 bit를 제외한 나머지 정상적인 데이터도 삭제됨을 뜻한다. 이는 e-RS 복호시 복호 능력의 한계 안에 있음에도 불구하고, 오류 한계를 벗어나는 원인이 될 수 있으므로 복호의 비효율성을 나타낼 수 있으며 또한 성능 열 화의 원인이 된다[4][5].

따라서 본 논문에서는 기존의 CRC 방식의 구조에서 크게 벗어나지 않되 오류능력 범위 한계에 들어오면 e-RS 복호기를 이용하여 복호하며, 오류 한계 범위를 벗어나면 삭제 시키지 않고 수신된 심볼 자체를 이용하여, RS 복호기를 이용하는 효율적인 LLR 방식의 복호 알고리즘을 제안하였다. 이 는 삭제된 심볼의 개수가 오류 정정 능력을 벗어난다 할지라도 오류 없는 심볼이 삭제된 경우의 수가 있으므로, 삭제전의 수신 심볼을 이용하여 기 존의 RS복호 방식을 이용하면 오류정정이 가능할 확률이 있다는 것에서 착안하였다. 제안한 방식은 기존의 CRC기반의 IP 패킷을 삭제시키지 않고 LDPC 복호기의 LLR을 기반으로 각 심볼별로 삭제 시키는 방식으로 성능 이 기존의 방식에 비해 향상됨을 알 수 있다.

차세대 지상/위성 전송 방식은 대용량, 고속 및 저전력 전송에 주안점을 두고 있으므로 본 논문에서는 cross layer 기반의 전송시스템에서 고속 및 저전력 부호화 방식을 연구하였다. Cross layer 기반의 전송 시스템에서 전력 및 고속적인 측면에서 제일 고려되어야 할 모듈은 LDPC 부복호 알고리즘 부분이다. 따라서 논문에서는 cross layer 부호화 방식의 PL-FEC로 적용되고



2

있는 LDPC부호의 복호기 설계를 위한 많은 계산 량 및 복호 속도를 개선 하기 위한 알고리즘을 분석, 제안 한다.

DVB-S2와 DVB-SSP에 적용되는 샤논의 채널 용량 한계에 근접한 LDPC 부호는 터보 부호에 비해 복호화의 복잡도가 낮을 뿐 아니라 좋은 거리 특 성으로 오류마루 현상이 나타나지 않고, 완전 병렬 처리로 고속 처리가 가 능한 장점이 있다[7][8].

100Mbps급 이상의 LDPC부호를 설계하기 위해서는 부호화 관점에서는 높은 복잡도가 LDPC 부호의 중요한 문제점이었으나 최근에 삼각행렬 분해 법, Linear-congruence 방법을 사용하여 부호화기를 간단하게 하였다. 특히 DVB-S2기반 부호화 알고리즘은 높은 부호화기의 복잡도를 간단하게 구현 하기 위해 검사행렬에서 "1"의 위치를 주소로 생성하여 주소를 이용하여 간단하게 구현되고 있다[9]. 구현의 문제점은 복호부에 있으며, 이를 어떻 게 100Mbps급 이상의 전송률을 가지게 하는가에 있다. DVB-S2에서 제안하고 있는 알고리즘은 큰 블록 사이즈(N= 64800)에 따른 많은 계산 량 및 많은 반복 횟수를 요구하고 있다. 따라서 복호시에 계산 량을 줄이고, 반복 횟수를 줄여 저전력 고속 복호기를 위한 새로운 방식과 설계가 중요하다. 그러므로 본 논문에서는 100Mbps급 이상의 LDPC 부호의 저전력을 위한 저 계산 량 알고리즘, 많은 반복 횟수를 줄이기 위한 고속 복호 알고리즘, 이를 적용시킨 최적의 복호기 구조 설계, VHDL 설계를 위한 최적 파라메 타 설정에 대하여 연구, 제안하였다.

LDPC 복호 시 고속화를 위해 병렬 처리를 많이 하게 된다. 이렇게 병렬 처리가 많아 지게 되면, 계산 량이 증가하게 되고, 이는 구현시 하드웨어의 복잡도 증가를 초래하여 critical path의 발생과 클럭 주파수의 감소 원인이 된다. 본 논문에서는 이러한 복잡도 증가에 대한 해결책으로 반복횟수 감 소를 위한 early stop 알고리즘을 분석하였다. 그러나 반복횟수의 감소만으 로는 LDPC 복호기의 저전력의 효과를 가져올 수 없으므로, 복호시 각 노 드에서 계산하는 계산량을 줄여야만 저전력 효과를 가져올 수 있으므로 본 논문에서는 각 노드에서 어느 일정한 값이 넘어서면 신뢰성이 있다고 판단



하여 더 이상 계산하지 않는 early detection 알고리즘을 제안한다. 또한 많 은 반복횟수를 줄여 고속 복호기를 구현하기 위해 horizontal shuffle scheduling(HSS) 알고리즘과 vertical shuffle scheduling(VSS) 알고리즘에 대해 연구하였다[10]. 복호기 구현 시 클럭 주파수 감소의 주 원인으로는 check node update(CNU) 연산을 꼽을 수 있다. 기존의 연산 방식은 sum-product(SP) 방식이다. 이 연산 방식에서는 채널 offset에 대한 보상으로 look up table(LUT)을 사용하여 특정 값에 대한 연산이 필요하다. 복호기 클럭 주파 수 감소의 가장 큰 원인은 바로 LUT 값의 연산에 있고, 이를 대신해 간단 한 연산으로 성능의 하락을 가져오지 않을 연산 방식의 계발이 필요하다. 본 논문에서는 다양한 CNU 연산 방식들을 비교 분석하였고, 성능의 큰 하 락 없이 보다 간단히 연산하여 클럭 주파수를 보다 빠르게 만들 수 있는 새로운 연산 방식을 제안한다.

마지막으로 본 논문에서는 저전력 고속화 알고리즘을 바탕으로 FPGA 설 계를 하였다. 이 때 반복 횟수를 줄여 고속화를 위해 사용된 알고리즘은 HSS 알고리즘이다. 본 논문에서는 HSS 알고리즘을 적용하여 구현을 하였 을 경우 그에 따른 최적의 설계 구조와 고속화를 위한 최적의 check node update(CNU) 방식을 제안하고, 기존의 방식을 사용하였을 때와 복호 속도 를 비교하였다. 그 결과 시스템 클럭 속도를 약 40% 빨리 할 수 있었고, 전체적인 복호 throughput 또한 40% 향상시킬 수 있었다. 또한, 최대의 클 럭 속도를 사용 하였을 때의 최종 복호 속도는 약 408Mbps임을 알 수 있 다.



## 제 2 장 효율적인 위성/지상 전송시스템의 부 호화 알고리즘

최근 디지털 방송과 이동 통신의 융합 추세 및 이동 통신 시스템의 비약 적인 발전에 힘입어 디지털 방송 시스템은 고정 수신만을 고려하여 설계되 었던 초기 시스템에서 이동성뿐만 아니라 휴대성을 고려한 시스템으로 진 화되어 가고 있다. 유럽의 지상파 디지털 방송시스템의 표준안인 DVB(Digital Video Broadcasting) 시스템은 위성, 케이블, 지상파 등 모든 전 송에서 디지털 TV 및 데이터 방송을 위한 완전한 해결방안을 제시하였다. DVB 표준으로는 위성방송의 대표적인 DVB-S, DVB-S2와 지상파 방송의 대표적인 DVB-T가 있다. 또한 고정 수신만을 고려하여 수신되었던 초기 시스템에 이동성 및 휴대성으로 진화되고 있는 가운데 DVB-T를 기반으로 하는 DVB-H의 표준화로 진행되고 있다. 현재 국내에서는 위성 및 지상파 DMB (Digital Multimedia Broadcasting)를 통한 방송 서비스가 제공되고 있으 며, 유럽에서는 기존의 DVB-T (Digital Video Broadcasting-Terrestrial) 표준을 기반으로 소형 휴대 단말기에 적합한 DVB-H(handheld)의 표준화를 진행하 고 있다. 이러한 위성과 지상을 결합한 전송 시스템으로는 DVB-SSP가 있 다. DVB-SSP에서는 PL-FEC로 LDPC 부호를 적용하고 있고, UL-FEC로는 e-RS 부호를 적용하고 있다. 본 장에서는 이러한 표준안에서 적용되고 있는 오류정정부호의 사례를 분석하고자 한다.

#### 제 2.1 절 LDPC 부호화 방식

위성/지상의 cross layer 부호화 방식을 적용하는 DVB-SSP에서는 PL-FEC 로 LDPC 부호를 적용 중이다. LDPC 부호는 1962년 Gallager에 의해 만들어 진 샤논의 채널 용량 한계에 근접한 부호로 DVB-SSP에 적용 가능한 LDPC 부호는 한 프레임의 길이에 따라 long size LDPC와 short size LDPC로



나눌 수가 있다.

#### 2.1.1 Long Size LDPC

Long size LDPC를 사용하는 대표적인 규격은 DVB-S2이다. DVB-S2 규격 의 LDPC 부호의 길이는 N=64800 이다.

DVB-S2 규격 LDPC 코드의 패리티 검사 행렬이 저 밀도로 형성되어 있 다 할 지라도 일반적인 생성 행렬은 부호화를 위하여 필요하다. 물론, 일반 적인 선형 부호는 패리티 검사 행렬을 알고 있기에 생성 행렬은 Gaussian elimination method(GEM)를 사용하여 간단하게 유도될 수 있다. 그러나 그 결과 생성 행렬의 밀도는 더 이상 낮지 않다. 이 것은 저장용량과 부호화 의 복잡도 문제를 야기한다. 따라서 여기서는 패리티 검사 행렬을 다음의 형태로 적용시킨다.  $H_{(N-K)xV} = \left[A_{(N-K)xK}B_{(N-K)x(N-K)}\right]$ 여기서 B는 다음 그 림 2.1에서 나타난 형태의 부 행렬로 표현할 수 있다.



행렬 A는 cycle-4를 피하고, cycle-6을 최소화 시키면서 저밀도 행렬을 구 성하며, 각각의 초기치도 무작위적으로 분포시킨다. LDPC부호기는 코드워 드 크기  $n_{ldpc}, c = (i_0, i_1, ..., i_{k_{upc}-1}, p_0, p_1, ..., p_{n_{ubpc}-k_{upc}-1})$ 에 정보블록크기  $k_{ldpc}, i = (i_0, i_1, ..., i_{k_{ubpc}-1})$ 를 부호화한다. 코드워드의 전송은  $i_0$ 와



 $p_{n_{ldpc}-k_{ldpc}-1}$ 이 주어질 때 시작한다.

부호기는  $k_{ldpc}$  정보비트의 모든 블록  $(i_0, i_1, \dots, i_{k_{dpc}-1})$  에 의해 $n_{ldpc} - k_{ldpc}$ 패리티비트  $(p_0, p_1, \dots, p_{n_{ldpc}-k_{ldpc}-1})$ 를 결정한다.

DVB-S2에 제시한 복호 알고리즘은 기존의 복호 알고리즘과 거의 동일하 나, LUT를 만드는 방법 및 Check Node 확률을 구하는 방법상에서 약간의 차이를 가진다.

전체적인 복호 과정을 표현하면 그림 2.2과 같다.



Fig 2.2 Total Decoding Process of LDPC

LDPC의 복호 과정을 나누어 보면 수신비트에다가 채널 추정 값을 구하 는 초기화 과정, Check node 확률을 구하는 CNU(check Node Update), 비트 확률을 구하는 BNU(Bit Node Update)로 세가지 단계로 나눌 수 있다.



단계 1: Initialization

첫번째 단계는 초기화 단계로, 식 (2.1)과 같이 수신데이터에 채널 추정 값인  $-\frac{2}{r^2}$ 를 곱해준  $\delta f_n$ 을 각 비트 노드에 저장한다.

$$\delta f_n = -\frac{2}{\sigma^2} r_n$$
 (2.1)  
 $r_n : 수신 데이터$ 

단계 2: Check Node Update(CNU)

하나의 체크 노드에서  $d_c$ 개의 row weight를 가진다고 가정할 때, 각각의 체크 노드로 들어오는 엣지 값은 아래 식으로 구할 수 있다.

ABIIME IIA.

$$g(a,b) = sign(a) \times sign(b) \times min(|a|, |b|) - LUT_a(a,b)$$
(2.2)

여기서,

$$LUT_g(a,b) = ln(1+e^{-|a+b|}) - ln(1+e^{-|a-b|})$$
(2.3)

를 나타낸다.

따라서  $LUT(x) = ln(1 + e^{-x})$ 의 look up table 하나만 있으면 위의 CNU를 계산할 수 있다.

예를 들어, 그림 2.2에서  $u_{0,0}$ 을 구하기 위해서는 이전 체크노드로 입력된 자신의 엣지 값인  $v_{0,0}$ 를 제외한 나머지  $v_{0,2}, v_{0,3}$ 의 값으로 식 (2.2)에 대입 하여 구할 수 있다.

각각의 비트노드들은 각각 연결된 모든 엣지의 값을 더함으로서 복호 값 인 log likelihood ratio(LLR) 값을 구할 수 있다. 이는 식 (2.4)로 나타낼 수 있다.

$$L_n = \sum u_{m,n} \tag{2.4}$$



CNU에서 각 체크 노드로 연결된 비트들에 대한 업데이트가 이루지고 각 비트 노드의 LLR 값을 구한 후, 마지막 단계로 LLR 값과 비트 노드로 입력된 각 엣지 값들을 이용하여 다시 체크 노드로 보내는 엣지의 값을 구 하기 위한 BNU 연산을 해야 한다. BNU 연산 역시 CNU와 마찬가지로 비 CNU 연산 후, 비트 노드로 입력된 자신의 엣지 값을 제외한 다른 엣지 값 으로만 연산을 행한다. 이러한 연산 방법은 식 (2.5)와 같이 간단히 행할 수 있다.

$$v'_{m,n} = L_n - u_{m,n} \tag{2.5}$$

위의 세 단계를 중 처음 초기화 단계 후에는 두 번째와 세 번째 단계인 CNU와 BNU만을 반복하며, 복호 데이터를 구한다.

그림 2.3은 DVB-S2 규격의 LDPC에 대한 반복 횟수 40회, 블록 사이즈 64800일 때, 부호화율에 따른 성능 곡선을 나타낸다.



그림 2.3. 부호율에 따른 성능 곡선(반복횟수=40, N=64800) Fig 2.3 The performance by coding rate(iteration=40, N=64800)



그림 2.4은 각 부호화 율에서 반복횟수의 변화를 주어 시뮬레이션 한 그 림이다.



(b) 부호화율 2/3





(d) 부호화율 4/5





(f) 부호화율 7/8





(h) 부호화율 9/10





(j) 부호화율 1/3





(l) 부호화율 3/5

그림 2.4. 각 부호화 별 반복 횟수 변화에 따른 성능 곡선(N=64800) Fig 2.4 The performance for coding rates in according to iterations(N=64800)

그림 2.4 에서 알 수 있듯이 각 부호화 율에 따른 반복의 횟수는 변화를 주어야 한다.



예를 들어 부호화율 1/2인 경우 60회와 70회일 때의 성능 곡선을 비교하 여보면 성능이 거의 같으므로 이때 반복은 60회로 정할 수 있으며 부호화 율 2/3인 경우 50회와 60회에서 두 성능 곡선이 일치해지므로 50회로 정한 다. 다음 표 2.1 는 각 부호화율에 따른 최적의 반복 횟수이다.

> 표 2.1 부호화율에 따른 최적의 반복횟수 Table 2.1 The optimal number of iteration according to code rates.

부호화율	최적의 iteration 수
1/2	60
2/3	50
3/4	60
4/5	40
5/6	50
7/8	50
8/9	40
9/10	40

#### 2.1.2 Short Size LDPC

Short size LDPC를 사용하는 대표적인 규격은 802.11n 이다. 차세대 무선 랜 기술인 802.11n에서도 BCC(Binary Convolutional Code)와 더불어 세가지 변조 기법과 네 가지 부호화 율 그리고 세가지 부호어 길이를 가지는 LDPC부호 또한 오류정정 부호기술로 채택되어 LDPC에 대한 연구가 활발 히 진행 중이다. 표준안에서는 부호화 블록 사이즈는 N=648부터 1944까지 다양한 사이즈를 권고하고 있으며 반복횟수는 8~12회를 요구하고 있다. 그 림 2.5은 본 연구에서 사용한 802.11n에 제시된 페리티 검사 행렬 H를 나 타낸 것이다.

1945



3 -1 28 -1 0 -1 -1 -1 55 7 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 30 -1 -1 -1 24 37 -1 -1 56 14 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 62 53 -1 -1 53 -1 -1 3 35 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1 40 -1 -1 20 66 -1 -1 22 28 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1 0 -1 -1 -1 8 -1 42 -1 50 -1 -1 8 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 H =69 79 79 -1 -1 -1 56 -1 52 -1 -1 -1 0 -1 -1 -1 -1 0 0 -1 -1 -1 0 65 -1 -1 -1 38 57 -1 -1 72 -1 27 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 64 -1 -1 -1 14 52 -1 -1 30 -1 -1 32 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 45 -1 70 0 -1 -1 -1 77 9 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 24 -1 61 -1 60 -1 -1 27 51 -1 -1 16 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 그림 2.5. 802.11n에 제시된 페리티 검사 행렬(N=1944, Z=81, rate=1/2) Fig 2.5 The parity check matrix of 802.11n (N=1944,Z=81,rate=1/2)

페리티 검사 행렬의 각 성분은 Z사이즈를 가지는 단위 행렬의 우 순환 행렬로 이루어져 있다. 각 성분의 숫자는 우 순환 이동의 횟수를 의미한다. 그림 2.5의 페리티 검사 행렬에 사용되는 데이터는 972 비트이고, 부호화 과정을 거쳐 출력되는 데이터는 1944 비트로 부호화 율은 1/2이 된다.

입력되는 데이터의 집합을 m이라 하면,  $m = [m_0, m_1, ..., m_{k_b-1}]$ 로 표현 할 수 있다. 이때, m의 각 성분은 Z 개의 데이터 집합이고, 페리티 검사 행 렬의 가로 길이를  $n_b$ 라 하고 세로의 길이를  $m_b$ 라 하면,  $k_b = n_b - m_b$ 가 된 다. 또한, 데이터의 집합 m이 부호화 과정을 거친 후의 데이터를  $c_b$ 라 하 면,  $c_b = [m_0, m_1, ..., m_{k_b-1}, p_0, p_1, ..., p_{m_b-1}]$ 로 표현 할 수 있다.

결과적으로 부호화가  $H \cdot c_b = 0$ 의 특징을 가지므로, 다음과 같이 표현할 수 있다.

$$\begin{bmatrix} h_{0,0} & \cdots & h_{0,k_{b}-1} & 1 & \cdots & \cdots & -1 \\ h_{1,0} & \cdots & h_{1,k_{b}-1} & -1 & \cdots & \cdots & -1 \\ \vdots & & \vdots & & \vdots & & & \vdots \\ h_{x,0} & \cdots & h_{x,k_{b}-1} & 0 & \cdots & \cdots & -1 \\ \vdots & & \vdots & \vdots & & & \vdots \\ h_{m_{b}-1,0} & \cdots & h_{m_{b}-1,k_{b}-1} & 1 & \cdots & \cdots & 0 \end{bmatrix} x \begin{bmatrix} m_{0} \\ \vdots \\ m_{b_{k}-1} \\ p_{0} \\ \vdots \\ p_{m_{b}-1} \end{bmatrix} = 0 \quad (2.6)$$

#### 위 식(2.6)을 계산하면 다음과 같다.



$$\sum_{j=0}^{k_{b}-1} h_{0,j} m_{j} + \prod_{1} p_{0} + p_{1} = 0 \quad \text{(Oth row)}$$

$$\sum_{j} h_{i,j} m_{j} + p_{i} + p_{i+1} = 0 \quad i \neq 0, x, m_{b} - 1$$

$$\sum_{j} h_{x,j} m_{j} + p_{0} + p_{x} + p_{x+1} = 0 \quad \text{(xth row)}$$

$$\sum_{j} h_{m_{b}-1,j} m_{j} + \prod_{1} p_{0} + p_{m_{b}-1} = 0 \quad ((m_{b} - 1) \text{ th row})$$
(2.7)

위의 식(2.7)을 모두 더하면,

$$p_0 = \sum_{i=0}^{m_b - 1k_b - 1} h_{i,j} m_j = \sum_{i=0}^{m_b - 1} \lambda_i, \quad \lambda_i = \sum_{j=0}^{k_b - 1} h_{i,j} m_j$$
(2.8)

을 얻을 수 있다.  $p_0$ 을 구하면,  $p_1$ 부터는 다음 식(2.9)로 구할 수 있다.

$$p_{1} = \lambda_{0} + \prod_{i} p_{0},$$

$$p_{i+1} = \lambda_{i} + p_{i} (i \neq 0, x, m_{b} - 1),$$

$$p_{i+1} = \lambda_{i} + p_{i} + p_{0} (i = x),$$

$$p_{m_{b}-1} = \lambda_{m_{b}-1} + \prod_{i} p_{0}$$
(2.9)

그림 2.6은 위 식들을 이용하여 부호화기를 만든 것이다. 이 구조를 이용 하면 보다 쉽고 간단하게 부호화기를 설계 할 수 있다.





그림 2.6. 802.11n LDPC 부호화기의 구조 Fig 2.6 The structure of encoder

Short size LDPC의 복호화 과정은 long size LDPC의 복호화 과정과 같다. IEEE 802.11n 표준안 기반의 시스템은 4개의 서로 다른 부호화율 1/2, 2/3, 3/4 5/6을 지원하며, 각각의 부호화 율에서는 세 개의 서로 다른 z 사이즈를 지원한다. 그리고 성능향상을 위해 irregular LDPC 부호를 사용한다. 802.11n 에서 제공하는 각 부호화에 따른 부호화어 크기 N, 정보 데이터의 크기 K 그리고 각 행과 열에서 '1'의 개수를 나타내는 row weight와 column weight 의 크기는 아래 표2.2과 같다. 그림 2.7는 AWGN에서 부호화 블록 사이즈 N이 1944 이고 변조 방식이 BPSK, 반복횟수 8 일 때 성능 그래프이다.



그림 2.7. 다양한 부호화율에서 BER 성능 곡선 Fig 2.7 The BER performance according to coding rate



그림 2.3과 그림 2.7을 비교해 본 결과 부호화 율 1/2을 보면 그림 2.3에 서는 1dB이상일 때 에러가 발생하지 않고, 그림2.7에서는 3dB 이상에서 에 러가 발생하지 않는다는 것을 알 수 있다. 이는 LDPC 부호의 블록 사이즈 가 클수록 성능이 더 좋아 지는 것이다. 본 논문에서는 이를 바탕으로 cross layer 부호화 방식에서 PL-FEC로는 위성 표준인 DVB-S2의 long size LDPC 부호를 적용시켰다.

Coding rate	Coded	Information	Row weight	column weight
R	Word N	Size K	$d_c$	$d_{v}$
	648	324	8	12
1/2	1296	648	8	11
	1944	972	8	11
	648	432	11	8
2/3	1296	864	11	8
	1944	1296	11	8
	648	486	15	6
3/4	1296	972	15	6
	1944	1458	15	6
	648	540	22	4
5/6	1296	1080	22	4
	1944	1620	20	4

표2.2802.11n에서 제공하는 각 부호화율에 따른 파라메타

Table 2.2 The parameter according to code rate in 802.11n

#### 제 2.2 절 e-RS 부호화 방식

Cross layer 부호화 방식에서 UL-FEC로 e-RS 부호가 사용되어 진다. 기존 의 일반 RS 부호에서 부호화 과정이 끝난 패리티를 포함한 전체 심볼의



개수를 n, 정보 데이터 심볼의 개수를 k, 에러 정정 능력을 t라 하면 RS(n,k,t)라 한다. 이 때, 복호 단에서 에러가 난 심볼의 수를 x라 하면, 식 (2.10)을 만족 시키면, 모든 에러를 정정 가능하다.

$$t = \frac{n-k}{2}, x \le t \tag{2.10}$$

Erasure RS(e-RS)부호의 경우 일반 적인 RS 부호의 복호와 달리 수신 심 볼의 신뢰성을 판단하여 신뢰성이 낮아 에러일 확률이 큰 심볼을 모두 삭 제한다. 그 후 복호화 과정에서 삭제한 심볼을 복구하면서 에러를 정정하 는 방식이다. e-RS 복호 시 삭제된 심볼의 수를 t라 하고, 이중에 에러 심 볼의 수를 x라 하면, 식 (2.11)을 만족할 경우 모든 에러를 정정 가능하다.

$$t = n - k, x \le t \tag{2.11}$$

물론 식 (2.11)이 성립하여 모든 에러를 정정 하기 위해서는 모든 에러 심볼이 삭제가 되어지는 전제가 바탕이 되어야 한다.

e-RS 부호는 에러심볼이 *n*-*k*가 이하일 때 모두 정정할 수 있으므로, 이러한 관점에서 본다면 최적의 정정부호이다.

수신 다항식 모델은 오류와 삭제, 두 부분으로 구성이 된다. 오류는 랜덤 하게 발생하고, 그 크기 또한 완전히 복호되기 전까지는 수신단에서 알지 못한다. 삭제는 오류와 유사하지만, 수신단에서 그 위치를 알 수 있다는 것 이 오류와 다른 점이다[11][12].

복호시 오류의 발생위치를 아는 것이 최우선시 되어야 하는 몇 가지 이 유가 있다. 수신기는 삭제선언을 하는 dead-zone으로 강판정기를 사용할 수 있게 한다. 이는 RS 복호에 있어 연판정 입력에서 강판정 출력으로 가기 위한 과정이다. 그리고 몇몇 심볼들이 전송되지 않았을 때, 수신기에서는 그 심볼들이 삭제되었음을 사전에 알아야만 하기 때문이다.

만약 우리가 오류 다항식을 e(x)로 표현을 하고, 삭제 다항식을 f(x)라고



표현을 할 때, 수신 다항식은 식 (2.12)와 같이 표현할 수 있다.

$$r(x) = c(x) + e(x) + f(x)$$
(2.12)

식 f(x)는 식 (2.13)과 같이 다르게 표현을 할 수 있다.

$$f(x) = \sum_{i=1}^{\nu} f_i x^{J_i}$$
(2.13)

이 때 삭제 심벌 위치 J<sub>i</sub>는 알고 있지만, 삭제 크기 f<sub>i</sub>는 알지 못한다. 복호의 과정은 아래와 같이 진행된다. β 개의 오류와 ρ개의 삭제가 있다고 가정을 한다면, 식 (2.14), 식 (2.15)과 같이 놓을 수 있다.

$$E_1 = a^{I_1} E_2 = a^{I_2} \dots E_\beta = a^{I_\beta}$$
(2.14)

$$F_1 = a^{J_1} F_2 = a^{J_2} \dots F_{\rho} = a^{J_{\rho}}$$
(2.15)

그리고 삭제 위치 다항식을 식 (2.16)와 같이 정의할 수 있다.

$$\Gamma(x) = \begin{cases} \Pi_{i=0}^{\nu} (1 - F_i x) & \nu \neq 0 \\ 1 & \nu \neq 0 \end{cases}$$
(2.16)

오류와 삭제를 정정하기 위해 BM알고리즘을 사용한다. Chien search 알고 리즘을 사용할 때 우리는 ^(x)의 근을 구할 수 있고, 그에 따라 오류의 위 치를 알 수 있게 된다. 오류와 삭제의 크기를 결정하기 위해서는 먼저 오 류와 삭제의 위치 다항식을 결합해야 한다.



$$\Psi(x) = \Lambda(x)\Gamma(x) \tag{2.17}$$

$$\Omega(x) = \Lambda(x)[1 + \Xi(x)] \mod x^{2t+1}$$
(2.18)

Forneys 공식을 사용하면 오류와 삭제의 크기를 식 2.19, 식 2.20과 같이 계산할 수 있다.

$$e_{\Gamma x} = \frac{-E_x \Omega(E_x^{-1})}{\Psi'(E_x^{-1})}$$
(2.19)

$$f_{\Gamma x} = \frac{-F_x \Omega(F_x^{-1})}{\Psi'(F_x^{-1})}$$
(2.20)

### 제 2.3 절 Cross Layer 부호화 방식

DVB-SSP에서 CRC 기반의 복호 블록도는 그림 2.8과 같으며, UL-FEC와 PL-FEC를 결합시킨 cross layer 부호화 방식을 적용시킨 시스템으로, 이는 FIFTH 제안 모델의 단점을 개선시킨, 즉, DVB-S2 모델을 독립적으로 사용 하면서 UL-FEC에 e-RS 부호를 적용시킴으로써 deinterleaving의 메모리를 감소시킬 수 있는 장점을 가지고 있다[4].





그림 2.8. CRC 기반의 삭제된 MPE-FEC 메모리 형태 Fig. 2.8. MPE-FEC memory based on the CRC method.

CRC 기반의 MPE-FEC의 cross layer 부호화 방식의 구조는 그림 2.8과 같 다. UL-FEC는 IP(Internet Protocol) 패킷 데이터를 입력으로 하는 DVB-H의 MPE-FEC 부분인, e-RS(255,191,64) 부호와 가상 인터리버, PL-FEC는 DVB-S2의 부호화율 1/2, 블록 사이즈 64800을 갖는 LDPC 부호이다.



그림 2.9. MPE-FEC 프레임 구조 및 가상 인터리버 Fig. 2.9. Frame structure of MPE-FEC & Virtual interleaver.

그림 2.9는 MPE-FEC 프레임을 나타낸다. MPE-FEC 메모리는 DVB-H의 입력인 IP 패킷 데이터가 입력되는데 IP 데이터의 최대 길이는 헤더를 포


함한 4080 바이트이며, UL-FEC 후에 12바이트의 헤더와 4바이트의 CRC를 포함한 최대 4096 바이트이다[13]. Frame을 구성하는 하나의 심벌은 8비트 로 구성되며, column의 개수는 255개의 심벌로 고정되어 있고, column의 길 이는 최고 1024개의 심벌까지 유동적으로 설정 가능하다. 따라서 총 프레 임의 크기는 최대 약 2M 비트를 가질 수 있다. 191개의 심벌로 이루어진 프레임 왼쪽의 부분은 정보 비트 부분이고, 64개의 심벌로 이루어진 프레임 의 오른쪽 부분은 RS 부호화 과정으로 생겨난 RS 패리티 부분이다. IP 데 이터가 최대 4080바이트로 입력되면 프레임의 가장 왼쪽에 위치한 column 을 채우면서 오른쪽 방향의 column에 데이터를 저장하고, 저장되지 않은 부분이 생긴다면 이 부분은 '0'으로 채워 넣는다. 데이터를 모두 저장하고 난 후에, row 단위로 191개의 데이터 심벌을 이용하여 64 심벌의 RS 패리 티를 만든다. 부호화가 끝난 데이터는 column 방향으로, 즉, 가장 왼쪽의 column부터 가장 오른쪽의 column까지 블록에 입력된 순서 차례대로 전송 된다. 전송되는 IP 데이터는 RS 부호 후 CRC 헤더를 붙여 PL-FEC 부분으 로 전송된다[14].

수신단 측면에서 보면, LDPC 복호된 심볼은 강판정(hard-decision)되어 IP 패킷으로 나눈 뒤 CRC-32 과정을 통해, 오류가 존재하면 IP 패킷 모두를 삭제시켜 MPE-FEC 메모리에 저장한 후, 메모리에 데이터가 모두 채워지면 가로로 e-RS 복호화 과정을 수행한다.

그러나 이러한 복호 과정에서 수신된 데이터 중에 1 bit 의 오류가 있을 지라도 IP 패킷 모두를 삭제함으로써 오류 없이 전송된 데이터도 erasure 시킴으로써, 복호시 비효율성을 나타낼 수 있다.



25

# 제 3 장 LLR 기법을 이용한 효율적인 부호화 방식 제안

그림 2.9의 MPE-FEC 프레임을 다시 한번 살펴 보면, frame을 구성하는 하나의 심벌은 8비트로 구성되며, column의 개수는 255개의 심벌로 고정되 어 있고, column의 길이는 최고 1024개의 심벌까지 유동적으로 설정 가능하 다. 다시 말하자면 row의 길이는 255심벌로 고정되어 있고, row의 개수는 최고 1024개의 심벌까지 유동적으로 설정 가능하다. 따라서 총 프레임의 크기는 최대 약 2M 비트를 가질 수 있다.

191개의 심벌로 이루어진 프레임 왼쪽의 부분은 정보 비트 부분이고, 64 개의 심벌로 이루어진 프레임의 오른쪽 부분은 RS 부호화 과정으로 생겨 난 RS 패리티 부분이다.

데이터가 입력되면 프레임의 가장 왼쪽에 위치한 column을 채우면서 오 른쪽 방향의 column에 데이터를 저장하고, 저장되지 않은 부분이 생긴다면 이 부분은 '0'으로 채워 넣는다.

데이터를 모두 저장하고 난 후에, row 단위로 RS 부호화를 실행한다. 191 개의 데이터 심벌을 이용하여 64 심벌의 RS 패리티를 만든다. 부호화가 끝 난 데이터는 column 방향으로, 즉, 가장 왼쪽의 column부터 가장 오른쪽의 column까지 블록에 입력된 순서 차례대로 전송된다.

RS 부호화 과정의 전후로 하여 인터리버와 디인터리버 과정을 수행하지 만, 프레임으로 입력되는 순서와 출력되는 순서가 같아 가시적으로 인터리 버 과정이 들어나지는 않기 때문에 이를 '가상 인터리버'라고 한다.

제 2.3절과 같이 기존의 방식은 e-RS(255,191,64)에서 삭제된 심볼 수가 64개 이하이면 오류정정이 가능하다. 삭제된 심볼 수가 64개를 초과하면, 정정 범위를 벗어나기 때문에 수신된 심볼을 강판정하여 복호하지 않는다 [15][16]. 본 논문에서 제안한 복호 방식은 기존의 방식을 그대로 수용하되 에러 정정 능력 범위 한계에 들어오면 e-RS 복호기를 이용하여 복호하며,



에러 정정 능력 한계 범위를 벗어나면 삭제 시키지 않고 수신된 심볼 자체 를 이용하여, 일반적인 RS 복호기를 이용하는 복호 방식이다. 이는 삭제된 심볼의 개수가 에러 정정 능력을 벗어난다 할지라도 에러가 발생하지 않은 심볼이 삭제된 경우의 수가 있으므로, 삭제전의 수신 심볼을 이용하여 기 존의 RS 복호 방식을 이용하면 오류정정이 가능할 확률이 있다. 이러한 방 식은 기존의 표준안을 크게 변형시키지 않으면서 적용 가능하므로 표준안 측면에서 보면 효율적이다.



그림 3.1.LLR 기반의 복호 방식 블록도

Fig. 3.1. Structure of MPE-FEC method based on LLR.

그림 3.1에서는 본 논문에서 제안한 부호화 방식의 블록도를 나타낸다. 부호화 방식은 그림 2.8의 CRC 기반의 부호화 방식과 같다. 복호 시 LDPC 복호 후, 그림 2.8에서는 CRC-32 체크 후 MPE-FEC 메모리에 저장시킨 후 삭제된 심볼의 수가 64개 이하이면 e-RS 복호기를, 64개 이상이면 RS 복호 기를 이용하여 복호하는 방식이지만, 그림 3.1의 제안한 구조에서는 LDPC 복호 후 데이터를 강판정하지 않고, 강판정 전의 LLR 값을 그대로 출력한 다. 출력된 LLR 값은 CRC 체크가 아닌 신뢰성 검사를 통해 삭제여부를



판단한다. 이 때, 신뢰성이 낮은 데이터를 가진 심볼은 IP 패킷 전체가 아 닌 그 심볼만을 삭제하게 된다. 이러한 방식을 사용하면 에러가 발생하지 않은 심볼을 그대로 유지시킬 수 있다. e-RS 복호는 CRC 기반의 부호화 방 식에서와 동일하다. MPE-FEC 메모리에 저장된 데이터들 중 삭제된 데이터 의 위치는 알 수 있으므로, 삭제된 심볼 수가 e-RS 복호기의 에러 정정 능 력 범위를 벗어났는지의 여부는 알 수 있다. 이를 이용하여 e-RS 복호기의 에러 정정 능력을 내에 삭제된 심볼 수가 존재 한다면 기존의 방식과 같이 e-RS 복호 과정을 거치고, 만약 에러 정정 능력 범위를 벗어난다면, e-RS 복호가 아닌 일반 RS 복호를 실행한다. 이를 위하여 심볼의 삭제가 되지 않은 데이터를 따로 저장을 시킨다. e-RS 복호기 대신 RS 복호기를 사용하 는 이유는 심볼의 삭제 기준이 에러의 판단 여부가 아닌 LLR 값의 신뢰성 만을 가지고 수행하기 때문이다. 즉, 신뢰성이 낮아 삭제가 된 심볼이라도 에러 심볼이 아닐 경우가 존재하기 때문이다.

이는 하드웨어적인 측면에서는 기존의 방식에 비해 MPE-FEC 메모리를 복호단에서 하나 더 추가를 해야 하며, CRC 체크를 하지 않고, 일정한 신 뢰성에 대한 기준 값을 정해 비교를 해야 하는 연산이 들어가게 된다. 하 지만 CRC 기반의 구조와 크게 변화가 없기 때문에 하드웨어 측면에서도 거의 비슷하다.

LDPC복호기에서 마지막 데이터는 비트 노드 데이터이다. 수신된 데이터 는 체크 노드 업데이트 되어지고, 이 값들은 다시 비트노드로 입력되어 업 데이트 되어진다. 마지막 반복이 끝난 후 비트 노드의 데이터는 e-RS 복호 기와 RS 복호기로 입력된다. 이 때, 신뢰성이 낮은 심볼을 찾기 위한 최적 의 기준을 정하는 것은 성능과 복잡도 측면에서 매우 중요하다. 식(3.1)에 서 신뢰성이 낮아 제거되어질 심볼을 찾기 위한 임계값을 δ라 하면

If, 
$$|v_n| \ge \delta$$
, not erased  
If,  $|v_n| < \delta$ , erased (3.1)



28

ν<sub>n</sub> 을 LDPC 복호기의 비트 노드 데이터의 연판정 값이라하면, δ 값을 정하는 것은 매우 중요하다. 만약 δ 값을 매우 크게 정할 경우, 제거해야 할 심볼 수가 최대 오류 정정 능력보다 커질 수 있다. 반대로 δ 값이 매 우 작으면, 오류 정정 능력도 적어지게 된다. 그러므로 본 논문에서는 컴퓨 터 시뮬레이션을 통하여 δ의 값에 따른 성능을 알아보았다. 그림 3.2에서 는 δ의 값을 1부터 6까지 변화를 시켜 그에 따른 성능을 나타내었다. 시 뮬레이션 결과에 따라 최적의 δ 값은 4이다.



그림 3.2.  $\delta$  값의 변화에 따른 MPE-FEC의 성능.



다음 단계로는 세로 방향의 255 바이트 중에 얼마나 많은 제거해야 할 데이터가 있는지 알아야 한다. 에러 심볼의 개수 N<sub>e</sub>를 δ 값 보다 낮은 신 뢰 할 수 없는 제거 해야 할 데이터의 수라고 하자. N<sub>e</sub>가 오류 정정 능력보다 적다면, e-RS 복호기를 사용하고, 그렇지 않으면, 기존의 RS 복호기를 사용 한다. 본 논문에서 제안한 방식은 매우 단순하지만 표준에 나타난 하드웨



어의 큰 구조적 변화없어 시스템에 최적이다. 또한, 이 방법은 CRC 연산이 필요없고, IP 패킷 제거에서 1 비트 오류를 방지한다. 그림 3.3에서는 CRC 기반의 부호화 방식과 제안한 LLR 기반의 부호화 방식에 대한 삭제 심볼 들 중 에러 심볼의 평균 수를 나타낸 것이다.



Fig.3.3. Aaverage number of non-error symbols within erased symbols in MPE-FEC memory.

그림 3.3을 보면 Es/No가 높아질수록 에러 심볼의 평균 수가 제안한 방 식에 가까워 지는 것을 알 수 있다. 이러한 결과는 CRC 기반의 부호화 방 식에서 많은 비 에러 심볼들을 삭제 한다는 것이 되고, 또한, 제안한 방식 인 LLR 기반의 부호화 방식이 좀더 효율적이라는 것을 나타낸다. 식 (3.2) 는 삭제 심볼의 개수를 구하는 방법을 나타내었다.

$$N_{erased} = N_{non-error} + N_{real-error}$$
(3.2)

기존의 방식에서는 IP 패킷에서 단지 1bit만 에러가 발생하더라도 전체 IP 패킷이 삭제 되어진다. 이는 e-RS에서의 오류마루현상을 야기할 수 있 다.



그림 3.4은 CRC 기반의 MPE-FEC방식과 제안한 LLR 기반의 MPE-FEC 방식의 성능을 비교한 것이다. 그림에서 "Conventional"로 표현된 방식은 그 림 2.8의 CRC 기반의 복호 방식을 나타낸다. 성능을 비교하기 위해 부호화 율 3/4인 e-RS(255,192,64)과 블록 사이즈 64,800의 부호화율 1/2의 LDPC 부 호[4]를 사용하였고, IP 패킷 사이즈는 1,024 바이트를 사용하였다. 채널환경 은 LTF(Long Term Fading) 채널이다.



Fig.3.4. Performance comparison in long term fading channel.

LTF 채널에서는 기존의 방식은 잘못된 수신 데이터는 정정하지 못한다. 그 이유는 기존의 방식에서는 제거되는 데이터 중에서도 올바른 데이터가 많이 포함되어 오류 정정 능력을 벗어나 버리기 때문이다. 그러므로, LTF 채널에서는 기존의 방식에 비해 제안한 방식의 성능이 훨씬 뛰어나다.



# 제 4 장 위성/지상 저 전력 고속 전송을 위한 LDPC 복호 알고리즘

### 제 4.1 절 저 전력 알고리즘

Cross layer 부호화 방식에서 PL-FEC로 사용되는 LDPC코드의 복호화는 많은 반복횟수와 계산량으로 높은 복잡도가 문제가 되고 있고, 이러한 복 잡도로 인해 많은 전력이 필요로 되어 진다. 이에 본 논문에서는 복잡도를 줄여 저 전력의 LDPC 복호기를 설계할 수 있는 두 가지 방법을 적용하여 연구하고 제안하였으며, 시뮬레이션하여 일반적인 복호화 방법과 성능을 비교, 분석해 보았다. 복잡도를 줄일 수 있는 두 가지 방법이란, 첫째로 early stop 알고리즘에 대해 연구 하였다[20]. Early stop알고리즘은 터보 부호 등 반복부호에서 적용할 수 있는 방법들을 이용하여 LDPC 복호기에 적용 시켰으며, 본 연구에서는 그전의 복호 비트와 현재의 복호 비트를 비교하 여 일치하면 더 이상 반복을 하지 않은 방식을 사용하여 성능 분석 하였다. 둘째, 비트 노드 계산과 체크 노드 계산시 일정한 신뢰도 값보다 크면 다 음 반복시 계산을 하지 않는 early detection 알고리즘에 대해 연구 하였다. Early detection 알고리즘은 국외에서 belief propagation의 개념으로 접근을 하 고 있다[17][18]. 기존의 방식과의 차이점은 기존의 방식은 비트노드의 early detection을 하여 제거 하였는데 반해 본 연구에서는 비트노드와 체크 노드 두개의 노드를 동시에 제거하였으며, 최적의 신뢰성 있는 값을 설정하여 성능 열화 없이 계산량을 줄일 수 있다.

## 4.1.1 Early Stop 알고리즘



LDPC 부호에서 Shannon 의 한계에 근접하기 위해서는 많은 블록 사이즈와 많은 횟수의 반복 회수를 요구한다. 많은 횟수의 반복회수는 전력 소모량을 증가시킬 뿐만 아니라 복호 속도 또한 느리게 한다. DVB-S2 표준 안에서는 각 부호화율에 반복횟수를 정해 놓고 있으며, 약 40~70 회 설정하고 있다. LDPC 와 같은 반복 부호는 일정한 반복 후에는 더 이상 성능이 개선되지 않는 현상이 있으며, 이는 복호 속도 저하 및 전력 소모량만 증가시키는 결과만 맞고 있다. 따라서 적절한 early-stop 알고리즘을 이용하여 더 이상의 성능 개선이 없으면 반복을 중지시키는 알고리즘을 그림 4.1 과 같이 구성해야 한다.



본 논문에서는 두 가지 알고리즘을 분석하였다. 각 반복이 끝나면 LLR 값을 강판정하여 비교하는 방식과 패리티 검사 식을 이용하는 방식이다.

#### A. 강판정 알고리즘

각 부호화율에 따른 반복을 마치면 LLR 을 강판정하여 복호 비트를 판정한다. 반복을 거치면서 정정된 비트가 있다면 LLR 의 비트에 변화가 있을 것이고 더 이상 정정할 것이 없다면 반복이 계속되더라도 더 이상의 변화는 없을 것이다. 따라서 각 반복일 때의 LLR 을 강판정하여 그 다음 반복과 비교 더 이상의 변화가 없을 경우 복호를 종료하고 최종적으로



나온 LLR 을 이용하여 복호한다. 그림 4.2(a) 는 강판정 알고리즘의 블록도 이다.

#### B. 패리티 검사 식 알고리즘

비트 노드 업데이트 후 각 비트 노드에서 체크 노드로 가는 값과 체크 노드에서 그 비트 노드로 가는 값의 합을 강판정 한다. 구해진 N 개의 값 중 정보비트 부분(K 개)을 다시 재 부호화 하면 새로운 패리티 데이터 (M 개)가 생성된다. 이 패리티 데이터와 강판정으로 구해진 패리티 데이터를 비교하여 같다면 반복을 중지한다. 그림 4.2(b) 는 패리티 검사 식 알고리즘의 블록도 이다.



(a) Hard decision algorithm (b) Parity check equation algorithm



표 4.1 강판정 알고리즘 일 때의 평균 정지된 반복 횟수 Table 4.1 The average number of iteration according to  $E_b / N_0$  and speed increment rate at the hard decision method.

	Hard decision		
$E_b / N_0$	Iteration(60 ই)	Increment of decoder speed	
0.8	49.52	17.46%	
0.825	45.88	23.53%	
0.85	43.11	28.15%	
0.875	40.53	32.45%	
0.9	38.53	35.78%	

표 4.2. 패리티 검사 식 알고리즘 일 때의 평균 정지된 반복 횟수 Table 4.2 The average number of iteration according to  $E_b / N_0$  and speed increment rate at the parity check equation method

E / N	Parity check equation	
$\boldsymbol{E}_{b}$ / $\boldsymbol{N}_{0}$	Iteration(60 ই)	Increment of decoder speed
0.8	48.56 194	19.07%
0.825	45.04	24.93%
0.85	42.15	29.75%
0.875	39.86	33.57%
0.9	37.79	37.07%

위의 표 4.1 과 4.2 는 두 가지 방식의 early stop 알고리즘을 적용하였을 시 반복 횟수이며 그에 따른 속도의 증가량을 나타내고 있다. 패리티 검사 식 방식을 사용하였을 때 약 1 회 반복을 적게 함을 알 수 있다. 그림 4.3 에서 보듯이 반복 60 회 한 BER 성능과 위의 2 가지 알고리즘을 적용하였을 때의 성능 곡선이 거의 일치함을 알 수 있다.





그림 4.3 early stop 알고리즘의 BER 성능곡선 Fig. 4.3 BER curve of two types of early stop algorithm.

## 4.1.2 Early Detection 알고리즘 제안

복호 과정의 복잡한 계산량을 줄이는 또 다른 방법은 early detection 방식이다. 이 Early detection 방식은 높은 LLR 값을 지닌 비트 노드나 체크 노드는 신뢰할 수 있다는 개념을 기반으로 한다. 따라서 결정된 비트 노드나 체크 노드는 다음 반복에서 계산 하지 않는다. LDPC 복호기에 사용되는 종전의 sum product(SP) 알고리즘은 뛰어난 BER 성능을 가지나 많은 계산 과정이 필요하다. 따라서 계산량을 줄일 수 있는 방법이 필요하게 된다. 비트 노드나 체크 노드를 업데이트 할 때 일정한 기준 값을 넘게 되면 그 값은 더 이상 반복 이후의 업데이트를 하지 않더라도 신뢰할 만한 값으로 사용 가능 하게 된다. Early detection 된 노드를 bipartite 그래프로 보면 그림 4.4 와 같다.





그림 4.4 Bipartite 그래프 Fig. 4.4 Bipartite graph.

LDPC 복호기에서 early detection 하기 위한 과정은 다음과 같다.

Step 1. initialization
모든 Detected Check node, bit node 를 0 으로 초기화 시킨다.
Step 2. Check node updates
체크 노드의 값이 일정한 기준 값( $T_c$ ) 이상 일 때 Detected
check 를 1 로 설정하고 그렇지 않을 경우 check node update 공식에
따라 계산한다.
Step 3. Bit node updates
비트 노드의 값이 일정한 기준 값( $T_b$ ) 이상 일 때 Detected
node 를 1 로 설정하고 그렇지 않을 경우 bit node update 공식에
따라 계산한다.

이 때 기준값( $T_c$ ,  $T_b$ )을 너무 높게 잡으면 계산량이 급격하게 줄어드나 성능이 떨어지며 기준값을 낮게 잡으면 반대로 성능열화는 없으나 계산량이 거의 줄어 들지 않으므로 이 기준값을 적당한 값으로 설정하는 것이 중요하다. 그림 4.5 에서  $T_b$ =18 이상일 때 일반적인 LDPC 복호기와 성능이 거의 일치함을 알 수 있다.





그림 4.5  $T_b$ 의 변화에 따른 BER 성능곡선

Fig. 4.5 Simulation results according to the change of  $T_{\rm b}$  .

표 4.3 는 LDPC decoder 에서 요구되는 계산량의 나열한 것이다.

표 4.3 LDPC 복호기에 필요한 계산량

Table 4.3 Decoding complexity for conventional and early detected method.

	Number of operations for conventional decoder	Number of operations for early detection method
Check nodes(g(a,b))	$(M \times (d_c - 1)) \times N_r$	$(M \times (d_c - d_c^* - 1)) \times N_r$
Bit nodes(additions)	$(N \times (d_v - 1)) \times N_r$	$((N-N^*)\times d_v)\times N_r$

여기서  $d_c^*$ 는 각 체크 노드 당 early detected 한 엣지의 평균 숫자 이고,  $N^*$ 는 비트 노드를 early detected 한 평균 숫자이다.  $T_b = 18$ ,  $T_c = 10$ 으로 고정하였을 때 표 4.4 는  $E_s/N_0$ 가 1dB 일 때 각 반복 횟수에 따른  $d_c^*$ 와



N<sup>\*</sup>의 숫자를 보여준다. 이 때 N=64800, K=32400, M=32400, 이며 d<sub>c</sub>=7,
d<sub>v</sub>=13 이다. 표 4.4 에서 볼 수 있듯이 early detected 방식의 계산량은
기존의 방식과 비교했을 경우 체크 노드 계산 시 50%, 비트 노드 계산 시
99% 정도의 감소를 가져온다.

N <sub>r</sub>	$d_c^*/M$	$N^{*}$	
10	0.387	491.84	
20	1.484	19048.59	
30	2.619	62372.86	
40	3.29	64334.73	
50	3.6068	64335.46	
60	3.852	64335.46	

표 4.4 early detected 된 수 Table 4.4 The number of early detected edges.

### 제 4.2 절 고속 복호 알고리즘

본 절에서는 LDPC 복호기의 고속화 방안에 대해 소개한다. DVB-S2 기반 의 LDPC 부호는 '0'과 '1'만을 이용한 페리티 체크 메트릭스 H에 의해 만 들어 진다. H 메트릭스에서 '1'의 위치는 굉장히 sparse, 랜덤하게 분포되어 있다. 이러한 성질에 의해 LDPC 부호는 좋은 성능을 가지나, 하드웨어 구 현에 있어서 문제가 되고, 큰 블록 사이즈와 많은 반복횟수로 인해 고속화 가 힘들다. 본 연구에서는 이를 극복하기 위해 Horizontal Shuffle Scheduling(HSS) 복호 방식, Vertical Shuffle Scheduling(VSS) 복호 방식에 대해



연구하였고, 또한 실제 LDPC 복호기를 실제 구현 시 클럭 속도에 가장 큰 영향을 주는 Check Node Update(CNU) 연산 방식에 대하여 계산량을 줄이고 보다 쉬운 구현을 위한 새로운 알고리즘과 구조를 제안하였다.

## 4.2.1. HSS 알고리즘 및 VSS 알고리즘



일반적인 LDPC 복호기의 구조는 다음 그림 4.6과 같다.

기존의 LDPC 복호기의 복호 순서는 우선 수신데이터를 이용하여 비트 노드를 초기화 한 후 각각의 체크 노드에 연결된 비트 노드 값을 이용하여 체크 노드 업데이트를 한다. 체크 노드 업데이트 후 다시 각각의 비트 노 드에 대해 업데이트를 하고, 이러한 연산을 계속 반복한다. 기존의 복호 방 식에 의해 체크 노드 업데이트 연산이 모두 끝난 후 비트 노드 업데이트를 하기 때문에 한번의 반복에도 많은 지연이 발생하여 고속의 LDPC 복호를



그림 4.6 기존의 LDPC 복호기의 구조 Fig. 4.6 The structure of convectional LDPC decoder

할 수 없다.

이를 극복하기 위해 본 절에서는 horizontal shuffle scheduling(HSS) 복호 방법과 vertical shuffle scheduling(VSS) 복호 방법을 연구하였다.

HSS 방식은 기존의 방식과는 달리 체크 노드 업데이트 연산을 하면서 비트 노드 업데이트 연산을 동시에 하는 것이 가능하다. HSS 복호 방식의 흐름도는 다음 그림 4.7과 같다.



그림 4.7 HSS 복호 방식의 흐름도 Fig. 4.7 Tanner graph of HSS algorithm

LDPC 복호 과정은 반복에 의한 복호이다. 그러므로 적어도 각 노드들이 한번씩 업데이트가 되면 한번의 반복이 끝나는 것이다. 그리고 전체 복호 과정은 지정된 반복 횟수만큼 반복이 되거나 그렇지 않으면 충분히 신뢰성 있는 데이터를 구했을 때 끝나게 된다. 그 후, 비트 노드의 값을 이용하여



각 비트를 결정한다. HSS 방식을 이용한 LDPC 복호 과정 중 각 비트 노드 의 값은 다음 식에 의해 구할 수 있다.

$$S_i = LLR_i + \sum_{i=1}^{d_v} u_i \tag{4.1}$$

여기서  $S_i$ 는 비트 노드의 최종 값을 나타내고, LLR은 수신 데이터를 나 타낸다. i는 비트노드이고,  $d_v$ 는 비트 노드에 연결된 edge의 수이다. 그리고  $u_i$ 는 체크 노드 업데이트를 통해 얻어진 각 edge의 값이다.

위 그림 4.4를 통해 간단한 예를 들어 보면, 첫 번째 체크 노드 업데이트 를 하기 위해  $v_0$ ,  $v_2$ ,  $v_8$  각각의 edge 값을 가지고 체크 노드 업데이트를 한다. 이 때, 각 edge의 값은 수신 데이터  $LLR_0$ ,  $LLR_1$ ,  $LLR_4$ 이고, 체크 노 드 업데이트 방법은 다음 식으로 알 수 있다.

$$u_j = \bigoplus_{k=1, k \neq j}^{d_c} v_k \tag{4.2}$$

v는 비트 노드에서 체크 노드로 향하는 edge를 나타내고, d<sub>c</sub>는 체크 노 드에 연결된 edge의 수이다. 위 식에서 ⊕는 다음과 같이 구할 수 있다.

$$|v_i| \oplus |v_j| = \min(|v_i|, |v_j|) - \text{offset}$$

$$(4.3)$$

$$\operatorname{sign}(|v_i| \oplus |v_j|) = \operatorname{sign}(v_i) \times \operatorname{sign}(v_j)$$
(4.4)

체크 노드 업데이트의 값을 이용하여, 각 비트 노드의 값은  $S_0 = LLR_0 + u_0$ ,  $S_1 = LLR_1 + u_2$ ,  $S_4 = LLR_4 + u_8$ 이 된다. 그 후, 두 번째 체크 노드 업데이트를 위해  $v_3$ ,  $v_5$ ,  $v_6$ ,  $v_8$ 을 가지고 온다. v는 v'를 이전 반복에서의 edge 값이라 하면 다음 식에 의해 구해질 수 있다.

$$v_i = S_i - v'_i \tag{4.5}$$



두 번째 체크 노드의 업데이트가 끝나면 각 edge의 값을 이용하여 다시, S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>, S<sub>5</sub> 가 구해지고, 세 번째 체크 노드에 대한 업데이트를 하여 S<sub>0</sub>, S<sub>1</sub>, S<sub>3</sub>, S<sub>7</sub> 역시 업데이트 되어 진다. 이와 같이 한 번의 반복이 끝나 게 된다. 이러한 과정이 반복되면서 모든 반복이 끝나거나, 신뢰성 있는 데 이터가 나올 때, 복호 과정은 끝나게 된다.



구현을 위한 HSS 방식의 구조는 다음 그림 4.8과 같다.

DVB-S2 규격의 LDPC 부호는 메트릭스 구조 상 부분화 시켜서 병렬연산 이 가능하다. 즉, H 메트릭스에서 부분화 되는 만큼 병렬로 연산하여 속도 를 증가시킬 수 있다. 하지만, 이렇게 나누게 되면 HSS 방식을 적용하여 구현 할 경우 부분화 된 집합 내에서 한번에 두 개 이상의 비트 노드와 체 크 노드가 연결이 되는 경우가 발생한다. 식 4.1 부터 식 4.5 까지를 보면 HSS 방식은 한번에 하나씩 비트 노드와 체크 노드간에 업데이트를 실행함 을 알 수 있다. 그래서 한번에 두 개 이상의 비트 노드와 체크 노드가 연 결이 되면 각 데이터 간에 메모리 연결 충돌이 발생하여 HSS 연산에서 에 러가 발생할 수 있다[21]. 그림 4.9는 이러한 경우를 보여준다. 이 경우 이 러한 메모리 연결 충돌을 방지하기 위하여 따로 두 개의 임시 프로세서를 사용할 수 있다. 간단한 예를 들어 한 번에 두 개의 비트 노드와 체크 노



드가 연결되었다면, 그림 4.9에서 첫 번째 비트 노드를 S 라 하고, 이 비트 노드에서 체크 노드로 가는 세 개의 edge 값을 각각  $v_0, v_1, v_2$ , 체크 노드에 서 비트 노드로 업데이트 되는 세 개의 edge 값을 각각  $u_0, u_1, u_2$ 라 하자.

$$S' = LLR + u'_0 + u'_1 + u'_2 \tag{4.6}$$

이라 할 때, 여기서  $u'_{0}$ ,  $u'_{1}$ ,  $u'_{2}$ 는 이전 반복에서의 edge 값이고, S'는 이 전 반복에서의 비트노드 값이다. 현재 반복에서의  $u_{2}$ 를 구하기 위해 세 번 째 연결된 체크 노드로 가는 edge  $v_{2}$ 를 구하게 위해서는 다음과 같은 식이 나와야 한다.

$$S = LLR + u_0 + u_1 + u'_2 \tag{4.7}$$

위 식은 정상적인 상태에서 HSS 방식을 적용하였을 때 나올 수 있다. 하 지만 그림 4.9에서처럼 두 개의 edge가 한 번에 계산 되어 질 때는 위와 같은 식이 나올 수 없다. 그래서 따로 두 개의 임시 프로세서를 사용한다 면, 다음과 같이 구할 수 있다. 임시 프로세서 두 개를 각각 *S*<sub>1</sub>, *S*<sub>2</sub>라 하면,

$$v_{0} = S' - u'_{0}, \qquad v_{1} = S' - u'_{1}$$

$$S_{1} = LLR + u_{0} + u'_{1} + u'_{2}$$

$$S_{2} = LLR + u'_{0} + u_{1} + u'_{2}$$

$$S = S_{1} + S_{2} - S' \qquad (4.8)$$

과 같이 구할 수 있다.





그림 4.9 메모리 연결 충돌의 예

Fig. 4.9 The example of memory access conflict

HSS 방식은 한 번의 반복에서 비트 노드가 row Weight 만큼 업데이트 되 기 때문에 기존의 방식에 비해 좋은 성능을 가진다. 이는 기존의 알고리즘 보다 요구되는 반복횟수가 많이 줄어듦을 의미한다. 다음 그림 4.10부터 4.15까지는 기존의 알고리즘과 HSS 방식의 반복 횟수에 의한 성능을 비교 한 것이다.



그림 4.10 HSS의 반복횟수에 의한 성능 비교(부호화 율=1/2) Fig. 4.10 The comparison of convectional algorithm and HSS algorithm according to number of iterations at 1/2 code rate





그림 4.11 HSS의 반복횟수에 의한 성능 비교(부호화 율=1/4) Fig. 4.11 The performance comparison between conventional and HSS algorithm for number of iteration(code rate = 1/4)



그림 4.12 HSS의 반복횟수에 의한 성능 비교(부호화 율=2/3) Fig. 4.12 The performance comparison between conventional and HSS algorithm for number of iteration(code rate = 2/3)





Fig. 4.13 The performance comparison between conventional and HSS algorithm for number of iteration(code rate = 3/4)



그림 4.14 HSS의 반복횟수에 의한 성능 비교(부호화 율=4/5) Fig. 4.14 The performance comparison between conventional and HSS algorithm for number of iteration(code rate = 4/5)





그림 4.15 HSS의 반복횟수에 의한 성능 비교(부호화 율=8/9) Fig. 4.15 The performance comparison between conventional and HSS algorithm for number of iteration(code rate = 8/9)

성능비교에 사용된 데이터는 약 100만개이고, AWGN 환경에서 시뮬레이 션을 하였다. 시뮬레이션의 모든 부호화 율에서 한 블록의 크기는 64800이 다. 기존의 방식은 반복 횟수를 고정 하였고, HSS 방식은 반복 횟수를 조정 하면서 성능을 비교하였다.

성능 비교 결과 1/2의 경우 기존의 방식과 HSS 방식의 반복 횟수 30회 에서 거의 비슷한 성능을 보임을 알 수 있다. 이는 HSS 방식을 사용할 때, 한 번의 반복에서의 속도 증가 효과가 있음과 동시에 반복 횟수 역시 줄일 수 있으며, 기존의 방식에 비해 성능 열화도 거의 발생하지 않음을 의미한 다.

아래 표는 각 부호화 방식에서 HSS 알고리즘 적용시 요구되는 반복횟수 를 나타낸다.



표 4.5 HSS 알고리즘 적용시 요구되는 반복횟수

부호화 율	기존방식 (회)	HSS(ই)	감소량(%)
1/4	40	20	50
1/2	60	30	50
2/3	50	25	50
3/4	60	30	50
4/5	40	25	38
8/9	40	25	38

Table 4.5 The number of iteration for HSS algorithm

표 4.5에서 1/4, 1/2, 2/3 등의 비교적 낮은 부호화 율에서는 반복 횟수의 감소량이 절반으로 떨어지는 것을 알 수 있다. 이는 HSS 방식을 적용 시 복호 속도를 두 배 상승 시킬 수 있음을 나타낸다.

HSS 방식은 체크 노드를 중심으로 복호화 하는 방식이라면 Vertical Shuffle Scheduling(VSS) 방식은 HSS 방식과는 반대로 비트 노드를 중심으로 Scheduling 하는 방식이다.

VSS 방식의 복호 연산은 3단계로 나눌 수 있다. 그림 4.16은 첫 번째 단 계인 초기화 단계를 나타낸다.



그림 4.16 VSS 방식의 초기화 단계 Fig. 4.16 Initialization step of VSS algorithm

첫 번째 단계인 초기화 단계에서는 체크 노드 연산을 위하여 수신 데이 터를 모두 'S check'이라는 체크 노드 연산을 위한 메모리에 저장한다. 모든 수신 데이터가 'S check'으로 저장 된 후 다음 그림 4.17과 같은 두 번째 단



계인 비트 노드 업데이트를 한다. 이 때, 각 edge의 값은 체크 노드 연산이 되어진 값이다.



그림 4.17 VSS 방식의 비트 노드 업데이트 Fig. 4.17 The bit node update of VSS algorithm

첫 번째 비트 노드의 초기 수신 데이터를  $LLR_0$ 라 하고, 이 때 업데이트 되는 두 edge 값을  $u_0, u_1$  이라 하면 첫 번째 비트 노드  $S_0$ 는

$$S_0 = LLR_0 + u_0 + u_1 \tag{4.9}$$

이 된다.

첫 번째 비트 노드가 업데이트 되어 지면, 그림 4.18과 같이 세 번째 단 계인 체크 노드 업데이트를 한다. 이 때 역시 각 edge의 값은 비트 노드 연산이 되어진 값이다.



그림 4.18 VSS 방식의 체크 노드 업데이트 Fig. 4.18 The check node update of VSS algorithm



첫 번째 비트 노드에서 'S check'로 입력되는 각 edge의 값은 다음과 같 이 연산 할 수 있다. 각각  $v_0, v_1$  이라 하면,

$$v_0 = S_0 - u_0, \ v_1 = S_0 - u_1 \tag{4.10}$$

라 할 수 있다. 각 edge 값이 'S check' 메모리에 입력 되면서, 다시 체크 노드 연산이 되고, 이 후 두 번째 단계인 비트 노드 업데이트와 세 번째 단계인 체크 노드 업데이트가 반복 되어진다.

구현을 위한 VSS 방식의 구조는 다음 그림 4.19와 같다.



Fig. 4.19 The structure for implementation of VSS algorithm

다음 그림 4.20은 VSS 방식에서 복호를 하기 위한 과정을 간단한 예시를 통하여 보여준다.





Fig. 4.20 The example of decoding flow for VSS algorithm

그림 4.20의 복호 과정은 다음과 같다. 첫 번째 반복에서는 체크 노드의 데이터를 위해 첫 번째 비트 노드에 연결된 모든 체크 노드로 비트 노드의 값을 업데이트(v0,v6) 한다. 이 때, 이미 체크 노드에는 비트 노드의 값보다 큰 값이 존재 하기 때문에 비트 노드로 전송되는 체크 노드 업데이트 값 (u0,u6)은 원래의 비트 노드 값이 전송되고, 체크 노드는 처음의 비트 노드 값을 가지게 된다. 그 다음 두 번째 비트 노드와 연결된 모든 체크 노드로 다을 가지게 된다. 그 다음 두 번째 비트 노드와 연결된 모든 체크 노드로 두 번째 비트 노드 값(v1,v3,v7)을 업데이트 한다. 이 때, 체크 노드에 존재 하는 이전의 값과 함께 체크 노드 업데이트를 하게 되고, 각 edge의 값을 다시 두 번째 비트 노드로 업데이트 한다. 이러한 과정을 거쳐 그림 4.17의 6번 그림까지 연산이 되면 첫 번째 반복이 끝나게 된다.



첫 번째 반복이 끝나면 모든 체크 노드의 값은 한 번의 반복을 통한 업 데이트 값을 가지게 되고, 이러한 값을 가진 채로 두 번째 반복부터 다시 그림 4.20의 과정을 반복한다. 이러한 과정을 거치게 되면 HSS 방식과 같 이 한번의 반복으로 더 많은 반복 효과를 가지게 된다.

다음 그림 4.21는 VSS 방식에 대한 복호기 구조에 대한 그림이다.



그림 4.21 VSS 방식에 대한 LDPC 복호기 구조 Fig. 4.21 The structure for LDPC decoder of VSS algorithm

## 4.2.2. Self-Correction 기반의 CNU 알고리즘 제안

DVB-S2 규격의 LDPC 부호는 메트릭스 구조 상 부분화가 가능하여 부분 화 된 크기만큼 병렬연산이 가능하다. 즉, H 메트릭스에서 부분화 되는 만 큼 병렬로 연산하여 속도를 증가시킬 수 있다. 하지만 많은 병렬로 인하여 동시 연산량이 많아지게 되고, 이는 클럭 주기가 늘어나도록 하는 원인이



된다. 클럭 주기의 늘어남의 가장 큰 원인 중 하나는 CNU 연산에 있다[22]. LDPC 복호기에서 CNU 연산은 식 (4.11)과 같다. 절대 값 연산과 비교 연 산, 그리고 마지막으로 offset을 빼는 연산을 해야 한다. 이와 같은 방식을 SP 방식이라고 한다. SP 방식을 적용시키면서 LDPC 복호의 고속화를 위해 클럭의 주기를 짧게 하게 되면 CNU에서 critical path가 발생하게 된다. critical path를 극복하기 위해 offset을 빼는 연산을 제거한 min sum(MS) 방 식을 사용할 경우 성능이 약 0.5dB 저하가 발생하게 된다[20][23].

이러한 이유 때문에 성능의 큰 저하 없이 CNU을 단순화 시키고, 계산량 을 줄일 수 있는 방안을 많이 연구하고 있다. 본 절에서는 SP 방식과 MS 방식에 대한 단점을 극복하고 좀더 효율적인 구현을 하기 위하여 CNU 연 산 방식에서 normalized MS(NMS) 방식과 offset MS(OMS) 방식에 대하여 연 구하였다.

SP: 
$$v_i \oplus v_j = sign(v_i) \times sign(v_j) \times \min(|v_i|, |v_j|) - offset$$
 (4.11)

$$MS: v_i \oplus v_j = sign(v_i) \times sign(v_j) \times \min(|v_i|, |v_j|)$$
(4.12)

NMS 방식에서 CNU의 출력은 normalization factor 'α'에 의해 결정된다. 여기서 'α'는 MS 방식의 성능 저하를 보상하기 위한 요소이다. NMS 방식 을 적용하기 위한 식은 다음과 같다.

$$M_{c \to v}^{new} = \alpha \times \underset{v' \in v_c/v}{sign}(M_{v \to c}) \times \underset{v' \in v_c/v}{min}|M_{v \to c}|$$

$$(4.13)$$

이 때, α 의 값은 0 < α ≤ 1 이어야 한다.

NMS 방식과 함께 SP, MS 방식을 대신 할 수 있는 CNU 연산 방법으로 OMS 방식이 있다. OMS 연산을 위한 수식은 다음과 같다.



$$M_{c \to v}^{new} = \underset{v' \in v_c/v}{sign}(M_{v' \to c}) \times \max(\underset{v' \in v_c/v}{min}|M_{v' \to c}| - \beta, 0)$$

$$(4.14)$$

CNU로 입력되는 모든 edge 값들 중 절대 값이 가장 작은 값을 찾은 후, 그 절대 값에서 offset β를 뺀 뒤 그 값과 0 중 큰 값을 찾는 방식이다. 이 때, β는 항상 양수여야 한다.



위의 그림4.22을 보면 α값이 비록 0.75, 0.5에서 비교적 낮은 EbNo에서 성능이 0.875일 때 보다 좋으나 EbNo가 높아 지면 오류마루현상이 발생함 을 알 수 있다. 이와 반대로 낮은 EbNo일 때 0.875는 다른 값들에 비해 성 능이 큰 차이가 없으나 높은 EbNo에서는 오류마루현상 없이 좋은 특성을 가지는 것을 알 수 있다. 그러므로 α는 0.875가 가장 적합한 값으로 볼 수 있다.

그림 4.23은 SP 방식과 NMS 방식의 성능을 나타내고 있다.





Fig. 4.23 The performance of SP and NMS algorithm

위 그림은 SP 방식과 NMS 방식의 FER을 구하여 비교한 것이다. 그림에 서 보면 SP 방식과 NMS 방식은 약 0.2dB의 성능 차이를 가지는 것을 알 수 있다. 비록 NMS 방식이 약 0.2dB가 성능이 안 좋지만, critical path를 없 애고, 클럭 주기를 줄일 수 있기 때문에 구현에 있어서는 고속화를 위해 NMS 방식이 더 효율적임을 알 수 있다.

NMS 방식 적용시 CNU의 구조는 아래와 같다.



그림 4.24 NMS 방식 적용시 CNU의 구조 Fig. 4.24 The structure of CNU using NMS algorithm



위 그림 4.24는 부호화 율 1/2에서의 구조이다. 부호화율 1/2에서 dc는 7 이므로 CNU의 입력 데이터는 7개이다. 7개의 데이터가 입력 되어지면 식 4.13과 같이 계산하게 된다. 하드웨어 구조상 α를 곱하는 것은 힘들다. 하 지만 α의 특성상 1보다 작은 0.875를 사용하였으므로 이는 오른쪽 shift 3을 한 후 MIN 블록에서 나온 출력 값에서 빼주게 되면 NMS의 출력을 얻을 수 있다.

NMS 방식은 구조 상으로도 복호 속도를 높일 수 있는 알고리즘이다.

CNU 계산 방법은 한 클럭에 데이터 하나씩을 불러와 계산 후 다시 하 나씩 출력을 하는 것이다. 이에 대한 구조는 그림 4.25에서 볼 수 있다.



Fig. 4.25 The structure of conventional CNU

그림 4.26은 구현을 위한 integer 시뮬레이션을 하였을 때, SP, MS, NMS에 대한 성능을 나타낸 것이다.





그림 4.26 SP, MS 방식과 NMS 방식의 성능 Fig. 4.26 The performances of SP, MS and NMS algorithm

SP 연산 방식에서 채널 offset에 대한 보상치인 LUT 값을 제거한 MS 방식과 SP 방식과의 성능 차를 극복하기 위해 적용한 NMS 방식 역시 MS 방식 보다 좋은 성능을 가지지만 SP 방식과는 큰 차이를 보이는 것을 알 수 있다.

본 논문에서는 연산에 대한 복잡도를 늘리지 않으면서 SP 방식과 NMS 방식간의 성능 차를 극복하기 위해 NMS 방식에 결합 할 수 있는 selfcorrection(SC) 방식을 결합하여 분석하였다. LDPC 복호는 각각의 부호화율 에 따라 각기 다른 반복 횟수만큼 반복을 하여 마지막 반복이 끝났을 때 LLR 값에 따라 복호하여 데이터를 얻게 된다. 그러나 데이터에 에러가 발 생하게 되거나 혹은 잡음이 심할 경우 CNU 계산과정에서 반복을 거치면 서 체크 노드 계산을 위해 비트 노드에서 들어오는 값의 부호가 바뀌게 되 는 현상이 발생하게 된다. SC 방식은 CNU 과정에서 이전의 반복 시의 엣 지 값과 현재 반복 시의 엣지 값에서 값의 부호를 비교하여 부호가 다르다 면 현재의 값을 '0'으로 만들어주는 것이다. 즉, CNU 과정에서 체크 노드를



계산하기 위해서 체크 노드에 연결되어 있는 비트 노드로부터 값을 받게 되는데, 이전 반복에서의 값과 비교하여 부호가 다르다면 체크 노드로 들 어오는 엣지 값을 '0'으로 만들어주는 것이다. 이전 반복에서의 엣지 값이 다음 반복에서 부호가 바뀌었다는 것은 그 엣지 값은 잡음이 심하거나 오 류가 발생했을 가능성이 커서 신뢰할 수 없는 신호이기 때문에 그 값을 0 으로 만들어 주는 것이다. 이렇게 CNU 계산과정에서 신뢰할 수 없는 신호 를 0으로 만들어주면서 CNU 계산을 더욱 효율적으로 하여 성능의 향상을 가져올 수 있다.

다음 그림 4.27은 SC 방식의 구조를 나타내고 있다.



그림 4.27 Self-Correction 방식의 구조

다음 그림 4.28에서 그림 4.30은 부호화율 1/2, 블록 사이즈는 64800에서 SP 방식, NMS 방식, MS 방식의 성능과 SC 방식을 적용한 성능을 비교한 그림이다.



Fig. 4.27 The structure of Self-Correction algorithm



그림 4.29 MS 방식과 SC-MS 방식의 성능

Fig. 4.29 The performance comparison between MS and SC-MS algorithm




그림 4.30 NMS 방식과 SC-NMS 방식의 성능 Fig. 4.30 The performance comparison between NMS and SC-NMS algorithm

그림 4.28에서 보면 SP 방식과 SC 방식을 적용한 SP 방식은 성능차이가 거의 없으나 SC 방식을 적용한 SP 방식에서 성능이 향상되었다는 것을 알 수 있다. 또한 그림 4.29와 그림 4.30의 결과를 보면 NMS 방식과 MS 방식 에 SC 방식을 적용하여 성능을 비교한 결과 기존의 NMS 방식과 MS 방식 보다 약 0.5dB의 성능이 개선되었다는 것을 알 수 있다. 전체적으로 세가 지 방식에 SC 방식을 적용함으로써 신뢰할 수 없는 신호에 대해 더 효율 적으로 CNU 계산을 하기 때문에 성능이 개선됨을 확인할 수 있다. 이에 더하여 부호 값만을 저장하는 메모리 하나와 비교 연산만을 더 추가하면 되기 때문에 SC 방식을 적용한다 하더라도 연산에 대한 복잡도는 거의 중 가하지 않는다.

다음 그림 4.31은 SC 방식을 적용한 SP 방식, MS 방식, NMS 방식의 성 능을 비교한 그림이다.





그림 4.30 SC 방식을 적용한 SP 방식, MS 방식, NMS 방식의 성능 Fig. 4.30 The performance comparison between SC-SP and SC-NMS and SC-MS algorithm

SC 방식을 적용한 SP 방식은 SC 방식을 적용한 MS 방식, NMS 방식보 다 약 0.1 dB 만큼 뛰어났고, SC 방식을 적용한 MS 방식과 NMS 방식은 거 의 비슷한 성능을 보이지만 NMS 방식이 MS 성능보다 조금 더 성능이 좋 다는 것을 확인할 수 있다. 기존의 SP방식은 MS방식, NMS 방식과 비교해 성능이 약 0.5~0.6 dB만큼의 차이가 있었지만 SC 방식을 적용한 결과를 보 면 그 차이는 약 0.1 dB로 줄어들었다. 이는 NMS 방식과 MS 방식에서 SC 방식을 적용함으로써 복호과정에서 큰 효과를 발휘하여 복호를 하였다는 것을 확인할 수 있다.



# 제 5 장 저 전력 고속 LDPC 복호기 FPGA 구현

본 장에서는 저 전력, 고속의 LDPC 복호기 구현을 위한 최적의 설계를 위해 제 4장에서의 HSS 알고리즘을 효율적으로 적용시키기 위한 복호기의 구조를 연구하였다. 본 장의 처음은 LDPC 복호기에서 기존에 제시된 복호 기 구조에 대하여 분석 하였고, 두 번째로는 저 전력 고속를 위해 HSS 방 식을 기반으로 한 LDPC 복호기의 최적의 구조를 제안하였다. 끝으로 기존 의 LDPC 복호기와 본 논문에서 제안한 고속 LDPC 복호기의 복호 속도에 대해 비교, 분석해 보았다.

# 제 5.1 절 기존의 LDPC 복호기 구조

그림 5.1은 q그룹으로 분할된 비트 노드와 체크 노드에 대한 Tanner 그래 프이다.



그림 5.1 q그룹으로 분할된 비트/체크 노드에 대한 Tanner 그래프 Fig. 5.1 Tanner graph of bit nodes and check nodes for group q



DVB-S2 규격의 LDPC 부호의 H 매트릭스 특성 상 360개의 노드를 하나 의 그룹으로 묶어서 분할 한다[25]. 위 그림은 부호화 율 1/2에서의 Tanner 그래프이고 비트 노드는 데이터의 순서대로 360개씩 그룹을 만들고, 체크 노드의 경우 q 만큼 떨어진 노드들로 360개씩 그룹을 만들 수 있다. 그림 5.2는 기존의 LDPC 복호기의 구조이다.



Fig 5.2 The structure of conventional LDPC decoder

그림 5.2의 구조를 보면, 우선 수신 데이터는 Info 메모리에 저장이 되고, 저장된 데이터를 출력하여 CNU 과정을 거친다. CNU 연산이 된 각 엣지 값들은 Message 메모리에 모두 저장이 되어지고, CNU 연산이 모두 끝나게 되면 Message 메모리에서 모든 엣지 값들을 출력하여 BNU 연산을 한다. BNU 연산이 끝나게 되면 이 때 연산된 모든 엣지 값 역시 Message 메모 리에 저장을 하고, 다시 CNU 연산을 반복하게 되는 구조이다.

Info 메모리에서는 수신 데이터는 데이터의 순서대로 360개씩 그룹별로 저장이 되어 지고, Message 메모리 역시 각각의 엣지의 그룹별로 메모리의 한 주소에 360개씩 저장이 되어진다. Message 메모리 구조는 그림 5.3과 같



이 구성되어 진다.



그림 5.3 Message 메모리의 구조

Fig. 5.3 The structure of Message memory

그림 5.3의 메모리는 부호화율 1/2에서의 구조이다. dc는 각 체크 노드에 연결된 엣지의 수를 나타내고, q는 360개씩 그룹 지어진 체크 노드 그룹의 수를 나타낸다. 즉 그림 5.3에서 dc=7, q=90 이므로 메모리의 총 주소는 630 개이다.

그림 5.2의 구조를 보면 Message 메모리와 Info 메모리에서 데이터가 출 력되면 Rotation으로 입력되는 것을 볼 수 있다. DVB-S2 부호화율 1/2의 H 메트릭스를 보면 모든 체크 노드와 비트 노드들은 순서대로 연결되어 있지 않고, 상당히 random하게 연결되어 진다. 이러한 메트릭스 특성에 의해 각 데이터 그룹들은 permutation 인덱스를 사용하여 데이터를 메트릭스에 대응 하도록 재배치 시켜야 한다. 이 때, 그림 5.2에서 이용되는 것이 Rotation이 다. 그림 5.4는 Rotation의 구조이다.





그림 5.4 Rotation 구조

Fig 5.4 The structure of Rotation

Rotation에 사용되는 permutation 인텍스는 360개의 데이터에 대한 재배치 순서를 나타내므로 0~359의 수가 표시된다. 즉, 구현상으로 총 9비트로 표 시되어 진다. 그림 5.4의 Rotation은 9비트의 permutation 인텍스를 각각 3비 트씩 나누어 3단계에 거쳐 데이터를 재배치 시킨다. 재배치 시키는 방식은 360개의 데이터를 우 순환 쉬프트 연산을 함으로써 적용된다. 예를 들어 permutation 인텍스가 359인 "101100111"이라면, 우선 최하위 세 비트 "111" 에 대한 우 순환 쉬프트 연산이 행해진다. 그 후, 다음 세 비트 "100"에 대 한 우 순환 쉬프트 연산이 행해지는데, 이 때, 단순히 '4'만큼의 우 순환 쉬프트 연산이 행해지는 것이 아니라 실제로는 "100000=32" 만큼의 우 순환 인 위프트 연산이 행해지게 된다. 마지막으로 최상위 세 비트인 "101"에 대한 우 순환 쉬프트 연산이 행해지게 된다. 이것 역시 실제로는 "101000000=320" 만큼의 우 순환 쉬프트 연산이 이루어 진다. 이러한 Rotation 과정을 거친 후 각 360개의 데이터는 CNU 나 BNU의 연산 순서 로 재 배치 되어지고, 각각의 연산이 이루어 지게 된다.

CNU와 BNU 연산이 반복이 되고, 반복이 끝난 후 각 비트 노드의 LLR 값은 Decision 메모리에 저장이 되어 복호 비트를 결정하게 된다.



### 제 5.2 절 HSS기반 고속 LDPC 복호기 제안

그림 5.2의 복호기 구조는 모든 CNU 연산이 끝난 뒤 BNU 연산을 하게 되고, 이는 2장에서의 블록 사이즈 64800의 LDPC 복호의 반복횟수를 필요 로 한다. 즉, 부호화 율 1/2의 경우 60회의 반복 횟수가 필요하다. 고속화를 위해서는 반복 횟수를 줄일 필요가 있다. 본 논문에서는 LDPC 복호기의 고속화를 위해 반복 횟수를 절반으로 줄일 수 있는 HSS 방식을 기반으로 한 LDPC 복호기를 구현하였다.

그림 5.5는 HSS 방식을 기반으로 한 LDPC 복호기의 구조이다.



그림 5.5 HSS 방식을 기반으로 한 복호기의 구조 Fig. 5.5 The structure of decoder based on HSS algorithm

제 5.1절의 일반적인 LDPC 복호기와는 달리 permutation 인덱스와 edge address 인덱스가 사용되었고, 비트 노드의 값들을 저장할 Sj 메모리가 사용 되었다. 그리고 BNU 연산을 위한 블록은 제외 되었다.

HSS 방식을 적용한 LDPC 복호기는 BNU 계산을 위한 블록을 따로 만들 지 않는다. 그 이유는 CNU 블록에서 나온 출력 값들을 바로 Sj 메모리에



업데이트 시키기 때문에 BNU를 위한 연산만을 따로 하지 않기 때문이다. 즉, LDPC 복호를 위한 연산 블록으로는 CNU 만을 필요로 한다. CNU를 계 산하기 위해서는 edge address 메모리와 permutation 인덱스 메모리가 필요하 다. Edge address 메모리는 CNU를 계산하기 위해 몇 번째 Edge 메모리 주소 를 불러와야 하는가는 정하기 위해 필요하다. 부호화 율 1/2 같은 경우 360 개씩 데이터를 분할하면 CNU 계산에 필요한 같은 컬럼에 위치한 데이터 들이 90개의 블록 중 몇 번째 블록에 있는지 그 위치를 나타내 주는 것이 다. 360개씩 데이터를 분할 하면 각 블록은 우 순환 소 정방 행렬의 형태를 가진다. 이 때, 얼마만큼 순환 이동이 되었는지를 나타내는 것이 permutation 인덱스 메모리이다. 즉, permutation 인덱스 메모리는 그 블록에 서 몇 번째부터 읽어야 하는지를 지시한다.

### 5.2.1 고속 Rotator 구조

수신된 데이터는 그림 5.5에서 Info A와 Info B 메모리에 차례대로 입력되 고 이 데이터 들은 다시 Sj 메모리로 입력된다. Edge address 메모리의 edge 위치 값을 이용하여 Sj 메모리에서 데이터를 출력하게 되고, 출력된 데이터 는 rotator로 입력된다. Rotator는 permutation index 메모리의 값을 이용하여 하나의 병렬 데이터 그룹내의 데이터들을 순서에 따라 재배치 시키는 역할 을 한다.

그림 5.4에서 기존의 rotation은 permutation index가 9비트로 이루어져 있으므로, 한 클럭에 3비트씩 shift연산을 처리하였다. 이 때, 세 번째 rotator 의 경우 한 클럭에 최대 320번의 shift연산을 하게 되어 이에 의해 역시 클 럭 속도의 감소를 초래하고, critical path의 발생 원인이 된다.

본 논문에서는 이를 극복하기 위해 새로운 rotator를 적용하였다. 그림5.6 은 본 논문에서 제안한 rotator이다.





그림 5.6 제안된 rotator 구조 Fig 5.6 The structure of proposed rotator

제안된 rotator 역시 index는 0부터 359까지 총 9비트로 구성되어 진다. 기존의 rotator와는 달리 데이터가 입력되면 첫번째 rotator에서는 0, 16~112 까지의 우 순환 shift 연산을 실행한다. 그 후, 두 번째 rotator에서는 0, 4~12, 128~140의 우 순환 shift 연산을 실행한다. 제 번째 rotator에서는 0~3, 128~131까지의 우 순환 shift 연산을 실행한다. 즉, 위의 구조를 사용할 경 우 한 클럭당 최대 140번의 shift 연산이 필요되어 지며, 이는 기존의 구조 에 비해 한 클럭당 필요 되어지는 연산의 수가 약 55%정도 감소되어져 더 욱 빠른 클럭 속도를 사용할 수 있는 것이다.

제안된 구조를 사용할 수 있는 이유는 index의 최대치가 359이기 때문이 다. 이를 이 진수로 나타내면 "101100111"이다. Index(8 downto 0)라 하면, index의 값이 256보다 클 때 index(7)의 값은 항상 '0'가 된다. 두 번째 rotator에서 index(7)의 값이 '0'이고, index(8)의 값이 '1'이라면, index(7)의 값 을 '1'로 만들어 rotator를 실행한 후, 세 번째 rotator에서 다시 한 번 index(7)의 값을 '1'로 만들어 rotator를 실행하게 되면 index(8)이 '1'일 때, 256번의 shift연산을 하지 않아도 rotator를 구현할 수 있는 것이다.

## 5.2.2 구현을 위한 최적의 구조

HSS 기반의 고속 LDPC 복호기의 최적의 구조는 그림 5.7과 같다.





그림 5.7. HSS 기반의 LDPC 복호기의 구조 Fig 5.7 The structure of LDPC decoder based on HSS

그림에서 Info\_mem은 수신데이터 메모리로 한 주소 당 360개의 수신 데 이터를 저장한다. 첫 번째 반복 시 Info\_mem에서 Sj\_mem으로 처음 데이터 가 입력 되어지고, 이 때는 Sj\_index를 사용하지 않는다. Sj 메모리에 모든 데이터가 저장된 후 Sj\_index 메모리에 저장되어 있는 8비트의 index 데이 터가 출력이 되고, 이 index 데이터는 Sj 메모리의 주소 값이 되어 Sj 데이 터가 출력된다. 이렇게 출력된 Sj 데이터는 rotator로 입력되어 진다. Rotator 의 동작 순서는 5.2.1에서와 같다. Rotator에 의해 우 순환 shift 되어진 360 개의 데이터는 CNU 되어진다. 또한, SC 알고리즘을 위하여 첫 번째 반복 때는 각 데이터의 sign 비트 360개를 sign 메모리에 저장한다. 첫 반복 때는 SC 알고리즘을 사용하지 않고, 원래의 데이터로만 CNU 연산을 실행한다.



이 때, CNU 연산은 다음 그림 5.8과 같다.



#### 그림 5.8 CNU 연산과정

#### Fig 5.8 The calculation procedure of CNU

위의 그림 5.8는 부호화율 2/3, dc=6 일 때의 연산과정을 나타낸 그림이다. CNU 블록에서 연산되어 진 데이터는 각각의 edge 값이므로 출력 순서대 로 Edge 메모리에 저장이 된다. 또한, CNU 연산된 새로운 edge 데이터와 CNU의 입력 데이터가 add 연산을 통하여 역 rotator로 입력되어 진다. 역 rotator는 데이터가 CNU 입력 전에 우 순환 shift 되어진 것을 다시 원래의 데이터 순서대로 re-shift 시키는 것이다. 이 때 역시, permutation index가 필 요하다. 역 rotator의 permutation index는 rotator의 permutation index와 구조가 같다. 하지만 역 rotator는 우 순환 shift 연산과는 반대로 좌 순환 shift 연산 을 필요로 한다. 역 rotator를 거쳐 원래의 데이터 순서대로 되돌려진 360개 의 데이터는 Sj index에 따라 Sj 메모리에 입력되어 진다. 이렇게 첫 번째 반복이 끝나고, 두 번째 반복부터는 Info 메모리를 사용하지 않고, Sj 메모 리의 데이터를 Sj index에 따라 출력시킨다. 이 360개의 데이터는 rotator를 거쳐 CNU 연산이 되어지는데, 두 번째 반복부터는 CNU 연산 블록에 입력 되기 전에 edge 메모리에서 이전 반복 시의 edge 데이터를 출력시켜 Si 데 이터에서 뺀 값을 CNU 연산 블록으로 입력시킨다. 또한, 이 때의 sign 비 트와 sign 메모리에 저장되어진 이전 반복 때의 sign 비트를 xor 연산을 시 킨 뒤 Sj 데이터와 같이 입력시킨다. Sign 비트의 xor 연산은 SC(Self-



Corecction) 알고리즘을 적용시키기 위함이다. 이후의 과정은 첫 번째 반복 과 동일하다.

### 제 5.3 절 구현 결과

제 5장 2절에서 HSS 기반의 LDPC 고속 복호기 구현을 위한 전체 구조 와 연산과정을 살펴 보았다. 본 논문에서 CNU의 기본 연산은 NMS 방식을 사용하였고, SP 방식과의 성능 차를 극복하기 위해 SC 알고리즘을 적용하 였다.

본 논문에서 제시된 복호기의 구조를 사용할 경우 한 번의 반복에 사용 되는 클럭 수는 662클럭이다. 다음 표들은 복호기 parameter를 나타내었다.

표 5.1 부호화 파라메타

Table 5.1. Code parameters

Size of LDPC	64800			
Code rate	0.5			
Max degree of check node dc	7			

표 5.1은 구현에 사용된 LDPC 부호의 파라메타이다. 블록 사이즈는 64800이고, 부호화 율은 1/2을 적용하였다. 이 때, row weight는 7이다.

#### 표 5.2 구조 파라메타

#### Table 5.2 Architecture Parameter

Level of parallelism P	360
Clock frequency (Mhz)	200
Latency of rotation	3
Max iteration	30
Bit size of message	6
Bit size of intrinsic	5



표 5.2에서 보면 구현을 위한 구조에서는 하나의 데이터 그룹에는 360개 의 데이터가 병렬로 연산이 되어지고, rotation에 의해 지연되는 클럭이 3클 럭이다. HSS 방식을 적용하였으므로, 부호화 율 1/2에서 총 30회의 반복을 하였고, 수신단에 입력되는 데이터는 5비트로 연판정 되었으나, 복호기의 연산에 의해 6비트로 늘어나는 것을 알 수 있다. 또한 이 구조를 적용하는 FPGA 칩은 200MHz의 클럭에서 동작 하는 것을 알 수 있다.

표 5.3 구현 시 클럭 소모 량과 속도

Number of group check M	90
8 1 8 1	
Number of group variable N	180
C I	
Number of group edges E	630
Nb cvcle check node	660 clock cycles
	5
Total iteration	662 clock cycles
Total nit iteration	19860 clock cycles
Decoding throughput	653 M bits/s
2 cooling monghter	

Table 5.3 clock cycles and decoding throughput

LDPC의 데이터는 360개씩 그룹으로 병렬 처리 되어진다. 부호화 율이 1/2 이므로 최대 반복 횟수는 30회이다. 수신데이터의 양자화 비트수는 5비 트이나, 연산 시 데이터의 비트수는 6비트이다. 표5.3에서는 전체 블록 사 이즈가 64800이고, 360개의 데이터가 그룹을 이루므로 부호화율 1/2에서 체 크노드의 그룹은 90개이고, 비트 노드의 그룹은 180개이다. 총 엣지의 수는 630개 이고, 체크노드 연산에 사용되어 지는 총 클럭 수는 660개이다. 한 번의 반복에 소모되는 클럭수는 662 클럭이고, 30회 반복이므로 총 클럭 수 는 19860 클럭이 소모되었다. 이 때, 복호 throughput은 653Mbps이다.

본 논문에서 사용된 FPGA 칩은 xc5vlx155t이고, 기본 클럭 주파수는 200Mhz이다. 본 연구에서 구현 시 사용한 clock frequency는 200MHz이나, 구현 툴에 의한 시스템 최대 클럭은 다음과 같다.



```
All values displayed in nanoseconds (ns)

Timing constraint: Default period analysis for Clock 'clk'

Clock period: 3.941ns (frequency: 253.743MHz)

Total number of paths / destination ports: 3092805 / 126198

Delay: 3.941ns (Levels of Logic = 6)

Source: CNU182/mes_LIF02/lifo_reg_0_4 (FF)

Destination: CNU182/secon_FLUT/cnu_data_4 (FF)

Source Clock: clk rising

Destination Clock: clk rising
```

그림 5.9 SC-NMS 방식에 대한 최대 클럭 속도

Fig. 5.9 Maximum clock speed for SC-NMS

즉, 본 연구에서 제시한 복호기 구조에서 사용할 수 있는 최대 clock frequency는 253.743MHz이다. 250MHz clock frequency를 사용한다면, 표 6.3에 의해 decoding throughput은 약 816M bits/s이다. 이러한 결과는 SP 방식이 아 닌 NMS방식을 기반으로 SC 알고리즘을 사용하였을 때 나타난 것이다.

다음은 SP방식을 사용하였을 경우의 결과이다.

Timing constraint: Default period analysis for Clock 'clk' Clock period: 5.618ns (frequency: 177.999MHz) Total number of paths / destination ports: 17735695 / 132678		
Delay: 5.618ns (Levels of Logic = 10) Source: CNU359/secon_FLUT/cnu_data_3 (FF) Destination: CNU359/secon_FLUT/cnu_data_5 (FF) Source Clock: clk rising Destination Clock: clk rising		
그림 5.10 SP 방식에 대한 최대 클럭 속도		

#### Fig. 5.10 Maximum clock speed for SP

위의 결과에서 알 수 있듯이 SP방식을 사용하였을 경우, 사용할 수 있는 최대 clock frequency는 177.999MHz이다. 이 때의 decoding throughput은 약



583M bits/s이다.

속도를 비교해본 결과 SC 알고리즘을 이용한 NMS방식을 사용할 경우 약 230Mbits/s 좋아짐을 알 수 있다. 즉, SP방식에서 채널 offset에 대한 보 상으로 LUT를 이용한 것은 많은 연산량과 함께 clock frequency의 감소를 야기시키는 것을 알 수 있다.

다음 표 5.4는 각각의 CNU연산 방식에 따른 클럭 속도와 복호 throughput을 나타낸 것이다.

표 5.4 CNU 연산 방식에 따른 클럭 속도와 복호 throughput

Table 5.4 The clock speed	and decoding	throughput	according to	CNU.
	U	01	U	

	SP	SC-NMS
부호화율	0.5	0.5
최대 클럭 속도	177.999MHz	253.743MHz
복호 throughput	약 583Mbits/s	약 816Mbits/s

표 5.4에서 채널 offset 연산의 유무에 대한 클럭 속도의 차이는 SC-NMS 방식이 약 40% 정도 더 향상 됨을 알 수 있다. 이 때, 복호 속도는 408Mbps이다.



# 제 6 장 결론

차세대 방송통신은 언제, 어디서나 서비스 가능한 기술을 요구하고 있으 며 이를 위해 지상파 방송과 위성과의 융합에 관한 연구가 활발히 진행 중 에 있고, 특히 위성을 이용한 쌍방향 통신, 즉 무선 인터넷과의 융합, 이동 체에 대한 위성 방송 및 인터넷 통신을 하기 위한 연구가 제안되고 있다. 이동체에 대한 위성방송 및 인터넷 서비스를 지속적으로 제공하기 위해 기 존의 DVB-S2 표준화에 DVB-H 와 DVB-T 를 결합한 새로운 DVB-SSP 표 준화에 대한 연구와 함께, 이는 이동체에 대해서 위성을 이용한 통신 방식 을 규정하고 있으며, PL 과 UL 의 두 단계로 부호화 및 복호화 방식인 cross layer 부호화 방식을 적용하고 있다. PL 에서 정정하지 못한 오류를 UL 에서 정정함으로써 오류 정정 능력을 한층 강화 시킨 것이다.

이와 같이 위성 통신 시스템의 PL에서 적용되고 있는 채널 부호로는 LDPC 부호가 있다. 본 논문에서 연구한 LDPC에는 프레임의 길이에 따라 Long size LDPC와 short size LDPC가 있고, 그 특성이 되는 parity check matrix 와 성능을 분석하였다.

또한 UL-FEC로 사용되는 e-RS 부호와 가상 인터리버를 결합한 MPE-FEC 구조에 대해 분석한 결과, e-RS 부호의 장점은 기존의 RS 부호보다 더 많은 오류 정정이 가능하지만, 512 바이트의 IP 패킷 중에서 만약 1바이트 의 오류가 존재한다고 판단 되면, 1바이트를 제외한 나머지 511 바이트의 정상적인 데이터도 삭제되는 비효율성이 있다. 이를 해결하기 위해 오류 한계 범위를 벗어나면 삭제 시키지 않고 수신된 심볼 자체를 이용하여, RS 복호기를 이용하는 효율적인 LLR 방식의 복호 알고리즘을 제안하였다. 이 는 삭제된 심볼의 개수가 오류 정정 능력을 벗어난다 할지라도 오류 없는 심볼이 삭제된 경우의 수가 있으므로, 삭제전의 수신 심볼을 이용하여 기 존의 RS복호 방식을 이용하면 오류정정이 가능할 확률이 있다는 것에서 착안하였다. 제안한 방식은 기존의 CRC기반의 IP 패킷을 삭제시키지 않고 LDPC 복호기의 LLR을 기반으로 각 심볼별로 삭제 시키는 방식으로 CRC



기반의 방식과의 성능 비교 분석 결과 LTF 채널에서 기차 속도가 300km/h 일때, CRC 기반의 방식에서는 에러 정정이 되지 않아 오류 마루 현상이 발 생 하였으나 제안한 방식은 좋은 성능을 보이는 것을 확인 하였다.

차세대 지상/위성 전송 방식은 대용량, 고속 및 저전력 전송에 주안점을 두고 있으므로 본 논문에서는 cross layer 기반의 전송시스템에서 고속 및 저전력 부호화 방식을 연구하였다. Cross layer 기반의 전송 시스템에서 전력 및 고속적인 측면에서 제일 고려되어야 할 모듈은 LDPC 부복호 알고리즘 부분이다. 따라서 논문에서는 cross layer 부호화 방식의 PL-FEC로 적용되고 있는 LDPC부호의 복호기 설계를 위한 많은 계산 량 및 복호 속도를 개선 하기 위한 알고리즘을 분석, 제안 하였다. 고속의 LDPC부호를 설계하기 위 해서는 부호화 관점에서는 높은 복잡도가 LDPC 부호의 중요한 문제점이었 으나 최근에 삼각행렬 분해법, Linear-congruence 방법을 사용하여 부호화기 를 간단하게 하였다. 특히 DVB-S2기반 부호화 알고리즘은 높은 부호화기 의 복잡도를 간단하게 구현하기 위해 검사행렬에서 "1"의 위치를 주소로 생성하여 주소를 이용하여 간단하게 구현되고 있다. 구현의 문제점은 복호 부에 있으며, 이를 어떻게 100Mbps급 이상의 전송률을 가지게 하는가에 있 다. DVB-S2에서 제안하고 있는 알고리즘은 큰 블록 사이즈(N= 64800) 및 많은 반복 횟수를 요구하고 있다. 따라서 복호시에 어떻게 반복 횟수를 줄 이고 연산량을 줄일 수 있을지가 중요하다. LDPC 복호시 고속화를 위해 병 렬 처리를 많이 하게 될 경우, 증가하는 연산량에 의해 야기 되어지는 critical path의 발생과 클럭 주파수의 감소 등을 극복하기 위해 복잡도 감소 알고리즘으로는 early detection 알고리즘 제안하고 early stop 알고리즘을 연 구하였고, 고속화 알고리즘으로는 HSS, VSS 알고리즘에 대해 연구하였고, 그에 따른 성능을 분석하였다. 그 결과 반복 횟수를 절반으로 줄일 수 있 었다. 고속화를 위해서는 클럭 주파수의 저하에 가장 영향을 미치는 CNU 연산에 대해 연구가 되어야 한다. 기존의 CNU 연산 방식인 SP 방식에서의 LUT 계산을 하지 않고 성능을 유지 시킬 수 있는 방안으로 SC-NMS 방식 을 제안하였다. SC-NMS 방식을 적용하여 LUT 계산을 하지 않고 메모리와



비교 연산만을 첨가한 결과 SP 방식과 약 0.1dB의 성능차를 가지고 고속의 클럭 주파수를 사용할 수 있게 되었다.

끝으로 이러한 알고리즘들과 고속화를 위한 최적화된 복호기 구조, CNU 연산 알고리즘, 데이터 rotator 등을 제안하였고, 이를 적용한 HSS 방식의 LDPC 복호기의 새로운 구조를 제안하였다.

위에서 제시한 고속 알고리즘을 토대로 VHDL 타이밍 시뮬레이션을 하 였다. 제안한 알고리즘과 구조의 복호 속도 향상도를 분석하기 위해 HSS기 반의 SP방식과 HSS기반, SC기반의 NMS방식의 복호 속도를 비교 분석하였 다. 분석 결과 HSS기반, SC기반의 NMS 방식이 HSS기반의 SP방식 보다 클 럭 속도를 약 40% 빨리 할 수 있으므로, 전체적인 복호 throughput 또한 40% 향상시킴을 알 수 있다. 또 한, SC-NMS 방식을 적용한 최대의 클럭 속도를 사용 하였을 때의 최종 복호 속도는 약 408Mbps임을 알 수 있다.

본 연구에서 제안한 위성/지상 전송을 위한 효율적인 LLR기반의 cross layer 부호화 방식 및 이를 저전력 고속으로 전송하기 위한 LDPC 부호화 방식의 효율적인 알고리즘 연구는 차후 DVB-S2 기반의 실감 방송 서비스, 802.11n 기반의 고속 무선 랜 시스템, 차세대 DVB-H 및 DVB-T 기반의 지 상 전송 시스템에서의 유용한 자료가 되리라 사료된다.



# 참 고 문 헌

- ETSI EN 301 210: "Digital Video Broadcasting (DVB): Framing Structure, Channel Coding and Modulation for Digital Satellite News Gathering (DSNG) and Other Contribution Applications by Satellite".
- [2] ETSI EN 302 304: "Digital Video Broadcasting (DVB); Transmission System for Handheld Terminal".
- [3] ETSI EN 300 744: "Digital Video Broadcasting (DVB): Framing Structure, Channel Coding and Modulation for Digital Terrestrial Television".
- [4] DVB-TM : Study Mission on Satellite Services to Portable devices (SSP) Draft Technical Report, March 2006.
- [5] V. Schena, G. Losquadro and A. Basili, "A satellite network for train environment in the FIFTH Demonstration", IST Mobile & Wireless Telecommunication Summit 2003, Aveiro(Portugal), pp. 603-607, June 15 ~ 18, 2003.
- [6] T. Richardson and R. Urbanke, "Efficient Encoding of Low-Density Parity Check Codes", IEEE Trans. Information Theory, vol. 47, pp. 638-656, Feb 2001.
- [7] R. G. Gallager, "Low-Density Parity-Check Codes," IRE trans. information theory, vol.8, PP.21-28,1962.
- [8] D. J. C. Mackay and R. M. Neal, "Near Shannon Limit Performance of Low-Density Parity-Check Codes," Electron. Letter, Vol.32, PP. 1645-1646, Aug. 1996.
- [9] Digital Video Broadcasting(DVB). "Second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications (DVB-S2)." European Standard (Telecommunications series) ETSI EN 302 307 V1.2.1(2009-08),2009.
- [10] Arthur S., Francois V., David D., Pascal U." DVB-S2 compliant LDPC decoder integrating Horizontal Shuffle Scheduling,", ISPACS2006, pp. 1013-1016.
- [11] J. Chen and M. Fossorier. "Density evolution of two improved bp-based algorithms for LDPC decoding", IEEE communication letters, March 2002.
- [12] S. Sakata, "Parallel BM algorithm and Euclidean algorithm for decoding are identical," Proc. ISITA-2004 (CD-ROM), Parma, Italy, October 10 –13, 2004.
- [13] Jarkko Paavola and Heidi Himmanen "Performance Analysis of MPE-FEC



Decoding Methods at the DVB-H Link Layer for Efficient IP Packet Retrieval", IEEE TRANSACTIONS BROADCASTING, Vol 35, No. 1, pp.263-275, March 2007.

- [14] N. Kawai, et al, "Performance of Multimedia Broadcasting Through ISDB Transmission System", IEEE Transaction on Broadcasting, Vol.42, No.3, p151~158, Sep, 1996.
- [15] A. Hashimoto, et al, "Development of a Transmission System and an Integrated Receiver for Satellite," ISDB IEEE Transaction on Consumer Electronics, Vol. 43 p.337~343 August, 1997.
- [16] Y. Nishida, et al, "Development of an HDTV codec conforming to MPEG-2 MP@HL and its picture quality", ITE of Japan Tech. Rep., Vol 21, No. 30, p31~36 BCS' 97-22, BFO'97-37, May, 1997.
- [17] M.P.C Fossorier, M. Mihaljevic, and H. Imai."Reduced complexity iterative decoding of low-density parity check codes based on belief propagation." IEEE Transactions on communications, 47 : 673-680, May 1999.
- [18] M. Gomes, G. Falcao, V. Silva, V. Ferreira, A. Sengo, and M. Falcao. "Flexible parallel architecture for DVB-S2 LDPC decoders." In Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE, pages 3265-3269, Washington, USA, November 2007.
- [19] J. Dielissen, A. Hekstra, and V. Berg, "Low cost LDPC decoder for DVB-S2." In design, Automation and Test in Europe, 2006. DATE'06. Proceedings, volume 2, pages 1-6, Munich, Germany, March 2006.
- [20] D.E. Hocever. "A reduces complexity decoder architecture via layered decoding of LDPC codes." In Signal Processing Systems, 2004. SIPS 2004. IEEE Workshop on, pages 107-112, Austin, USA, October 2004.
- [21] C.Marchand, J.-B. Dore, L. Conde-Canencia, and E. Boutillon. "Conflict resolution for pipelined layered LDPC decoders." "In Signal Processing Systems, 2009. SiPS 2009. IEEE Workshop on. Pages 220-225, Tampere, Finlande, November 2009.
- [22] O. Eljamaly and P. Sweeney. "Alternative approximation of check node algorithm for DVB-S2 LDPC decoder." In Second International Conference on Systems and Networks Communications (ICSNC 2007), pages 157-162, October



2007.

- [23] Xiao-Yu Hu, E. Eleftheriou, D.-M. Arnold, and A. Dholakia, "Efficient implementations of the sum-product algorithm for decoding LDPC codes." In Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE, volume 2, pages 1036-1036E vol.2,2001.
- [24] V. Savin. "Self-corrected min-sum decoding of LDPC codes." Pages 146-150,2008.
- [25] C. Marchand, L. Conde-Canencia, and E. Boutillon. "Architecture and finite precision optimization for layered LDPC decoders." In Signal Processing Systems, 2010. SiPS 2010. IEEE Workshop on. Accepted, San Francisco, USA, October 2010.
- [26] S.M.Kim, C.S.Park and S.Y,Hwang. "A Novel Partial Parallel Architecture for High-throughput LDPC Decoder for DVB-S2", IEEE Transactions on consumer electronics. Volume56 No. 2, may, 2010.
- [27] S.L.Fogal, S. Dolinar, and K. Andrews. "Buffering requirement for variableiteration ldpc decoder." In Information Theory and Application Workshop, ITA. 2008, pages 523-530, San Diego, CA, January 2008.



# 연구 실적

#### SCI

- Duk-Gun Choi, Min-Hyuk Kim, Jin-Hee Jeong, Ji-Won Jung, Jong-Tae Bae, Seok-Soon Choi, Young Yun "A FPGA Implementation of High-Speed Flexible 27-Mbps 8-State Turbo Decoder" ETRI Journal Volume 29. Number 3. pp.363-370,June 2007
- [2] Ji Won Jung, Xinping Huang, Mario Caron, Min Hyuk Kim, Ki Man Kim, Young Yun, "The phase estimation algorithm of arrival phase difference in space-time coded communication" ETRI Journal, 제28권, 제5호, pp.680-683,October 2006

국제 학술 대회

- [1] Jong-Tae Bae, Seok-Soon Choi, Min-Hyuk Kim, Ji-Won Jung "A Study of interleaver Construction to Overcome Deep Fading for Satellite Communication" 25th AIAA International Communications Satellite Systems Conference, April 2007
- [2] Min Hyuk Kim, Jin Hee Jeong, Ji Won Jung "High-Speed Adaptive Turbo Decoding Algorithm and Its Implementation" Proceedings of 2006 IEEE Information Theory Workshop pp.104-108, October 2006
- [3] Min Hyuk Kim, Duk Gun Choi, Jin Hee Jeong, Ji Won Jung "An Efficient Decoding Algorithm of Iterative Codes for High Speed Satellite Communication System" 24th AIAA International Communications Satellite Systems Conference pp.1655-1660, June2006년
- [4] Jin Hee Jeong, Min Hyuk Kim, Ji Won Jung, Eun A Choi, Duk Gil Oh "Design Architecture of Long-Size LDPC Decoder Using the Partial Parallel Processing in the Digital Video Broadcasting by Satellite" 24th AIAA International Communications Satellite Systems Conference, pp.1630-1635, 2006년
- [5] Nam Soo Kim, Min Hyuk Kim, Jong Tae Bae, Seok Soon Choi, Tae Doo Park, Ji Won Jung "Soft bit Split Methods for Iterative Codes", HUT-ICCE, June.2008
- [6] Tae Doo Park, Min Hyuk Kim, Jong Tae Bae, Seok Soon Choi, Nam Soo Kim, Ji Won Jung "Cross layer coding for Mobility of DVB-S2 system", HUT-ICCE,



June.2008

- [7] Jong Tae Bae, Min Hyuk Kim, Seok Soon Choi, Tae Doo Park, Nam Soo Kim, Ji Won Jung "The Performance Analysis of Advanced T-DMB using Uuequal Error Protection System and Hierarchical Modulation", HUT-ICCE, June.2008
- [8] Min Hyuk Kim, Jong Tae Bae, Seok Soon Choi, Tae Doo Park, Nam Soo Kim, Ji Won Jung "An FPGA Implementation of Channel Coding Algorithm for European CATV down Stream of 1Gbps", HUT-ICCE, June.2008
- [9] Nam Soo Kim, Min Hyuk Kim, Jong Tae Bae, Seok Soon Choi, Tae Doo Park, Ji Won Jung "A Study on MPE-FEC decoding base on LLR method in DVB-SSP system", ITC-CSCC, July.2008
- [10] Seok Soon Choi, Min Hyuk Kim, Jong Tae Bae, Tae Doo Park, Nam Soo Kim, Ji Won Jung "An Efficient Implementation of LDPC Decoder with Partial Parallel Algorithm for DVB-S2 System", ITC-CSCC, July.2008
- [11] Min Hyuk Kim, Jong Tae Bae, Seok Soon Choi, Tae Doo Park, Nam Soo Kim, Ji Won Jung "An Efficient Design Methodology of Serial Concatenate Coding Scheme for CATV Transmission System", ITC-CSCC, July.2008
- [12] Tae Doo Park, Min Hyuk Kim, Jong Tae Bae, Seok Soon Choi, Nam Soo Kim, Ji Won Jung "Hybrid ARQ System using the Cross Layer Coding", ITC-CSCC, July.2008
- [13] Jong Tae Bae, Min Hyuk Kim, Seok Soon Choi, Tae Doo Park, Nam Soo Kim, Ji Won Jung "Hierarchical Transmission Algorithm Combined Coding Method in the T-DMB System", ITC-CSCC, July.2008
- [14] Min Hyuk Kim, Nam Soo Kim, Young Bae Park, Sae Ho Kim, Ji Won Jung, Seok Yeob Kang, Young Yun, Kyu Ho Park, Jin Seob Kim, Ki Jin Kim, Sae Hwan Choi, Kwang Ho Choi "A miniaturized band rejection filter with new compact PBG (photonic band gap) cell employing T-type microstrip line", EMC'09 Kyoto, July.2009
- [15] Sae Ho Kim, Young Bae Park, Seok Yeob Kang, Min Hyuk Kim, Nam Soo Kim, Ji Won Jung, Young Yun, Kyu Ho Park, Jin Seob Kim, Ki Jin Kim, Sae Hwan Choi, Kwang Ho Choi "A Highly Isolated Transmission Line Employing a Periodic Ground Structure on MMIC For an EMC Solution", EMC'09 Kyoto, July.2009



- [16] Nam Soo Kim, Min Hyuk Kim, Tae Doo Park, Chul Seong Kim, Ji Won Jung "Performance Analysis of LDPC codes for Invariance of Channel Estimation", WICOM, Sept.2009
- [17] Min Hyuk Kim, Tae Doo Park, Nam Soo Kim, Chul Seong Kim, Ji Won Jung "Design Methodology of Concatenate Code for extended RS combined with Rotational Invariant TCM code", WICOM, Sept.2009
- [18] Chul Seong Kim, Min Hyuk Kim, Tae Doo Park, Nam Soo Kim, Ji Won Jung "Hybrid ARQ System using the Double Layer Coding", WICOM, Sept.2009
- [19] Tae Doo Park, Min Hyuk Kim, Nam Soo Kim, Chul Seong Kim, Ji Won Jung "Low Computational Algorithm for Space Time Coded System with LDPC codes", WICOM, Sept.2009
- [20] Chul Seong Kim, Min Hyuk Kim, Tae Doo Park, Chi Hong Min, Ji Won Jung "Hierarchical Transmission Algorithm in the Advanced T-DMB", ICCE2010, Jan.2010
- [21] Tae Doo Park, Min Hyuk Kim, Chul Seong Kim, Ji Won Jung "A Cooperative Communication System using Cross-Layer Coding Method base on Hybrid-ARQ", ICUFN2010, June. 2010
- [22] Chul Seong Kim, Min Hyuk Kim, Tae Doo Park, Ji Won Jung "A New MPE\_FEC Decoding Algorithm", ICSSC2010, Aug. 2010
- [23] Tae Doo Park, Min Hyuk Kim, Chul Seong Kim, Ji Won Jung "Analysis of Turbo Codes for Next Generation DVB\_RCS system", ICSSC2010, Aug. 2010
- [24] Min Hyuk Kim, Tae Doo Park, Chul Seong Kim, Ji Won Jung "An FPGA Design of Low Power LDPC Decoder for High-speed Wireless LAN", ICCT2010, Nov. 2010
- [25] Tae Doo Park, Min Hyuk Kim, Chul Seong Kim, Ji Won Jung "Performance Analysis of turbo φ codes with m=3\_final", ICCT2010, Nov. 2010
- [26] Chul Seong Kim, Min Hyuk Kim, Tae Doo Park, Ji Won Jung "Rate-Compatible Turbo Product Codes with Extended BCH codes", ICCT2010, Nov. 2010
- [27] ] Tae Doo Park, Min Hyuk Kim, Beong Soo Lim, Ji Won Jung "Rate-Compatible Turbo Product Codes with Non-Symmetry Block Codes for DVB-RCS NG Systems", APCC2011, Oct. 2011



국내 학술 등재지

- [1] Min-Hyuk Kim, Jong-Tae Bae, Seok-Soon Choi, In-Ki Lee, Ji-Won Jung "A FPGA Implementation of High-Speed Adaptive Turbo Decoder" 한국 통신 학 회 논문집 제32권 제4호, 2007년 4월
- [2] 배종태, 최석순, 김민혁, 장대익, 정지원 "반복 부호의 다치 변조방식 적 용을 위한 최적의 비트 분리 방법 및 성능평가" 한국 통신 학회 논문집 제32권 제3호, 2007년 3월
- [3] Seok-Soon Choi, Jong-Tae Bae, Min-Hyuk Kim, Eun-A Choi, Ji-Won Jung "Low Latency Algorithms for Iterative Codes" 한국 통신 학회 논문집 제32권 제3 호, 2007년 3월
- [4] 정지원, 정진희, 김민혁, "DVB-S2기반 LDPC 복호기의 멀티레벨 변조방
   식 적용을 위한 디매핑 알고리즘" 한국전자파학회 논문지 제17권, 제7
   호, pp.615-622, 2006년 7월
- [5] 정지원, 김민혁, 정진희, "A 18-Mbp/s, 8-state, High-Speed Turbo Decoder" 한 국전자파학회지, 제6권, 제3호, pp.147-154, 2006년 6월
- [6] 배종태,김민혁,최석순,정지원,천승용,도경철 "수중통신채널에서 고려되는 채널 부호의 성능 분석" 한국음향학회지, 286-295, 2008년08월
- [7] 김민혁,최석순,정지원,이성로,조한나,최명수 "Compensation Algorithm of Arrival Time Mismatch in Space-Time Coded System", 한국해양정보통신학회 지, 353-357, 2008년09월
- [8] 박태두,김민혁,김남수,정지원 "반복부호의 멀티레벨 변조방식 적용을 위 한 비트분리 알고리즘", 한국해양정보통신학회지 1654-1665, 2008년09 월
- [9] 김남수,배종태,김민혁,정지원 "차세대 T-DMB 방송의 부가 서비스 제공
   을 위한 비트 분리 방법 및 Double Binary Turbo 부호 적용 연구", 한국 전자파학회논문지, 1065-1074, 2008년11월
- [10] 최석순,배종태,김민혁,정지원,이성로,최명수 "이동형 DVB-S2 기반 상위 계층 부호화 방식 성능 분석", 한국전자파학회논문지, 1075-1085, 2008년 11월



[11]김민혁,박태두,김남수,정지원,이성로,최명수 "차세대 T-DMB 방송의 부가 서비스 제공을 위한 계층적 전송방식에 대한 연구", 한국통신학회논문 지,997-1005,2008년12월

- [12]박태두,김민혁,김남수,김철승,정지원 "고속 이동체에서 위성 광대역 인 터넷 서비스를 위한 Cross Layer 부호화 방식", 한국통신학회논문지, 485-497, 2009년05월
- [13]김남수,김민혁,박태두,김철승,정지원 "수중통신채널에서 다중경로 극복
   을 위한 오류정정부호에 대한 연구", 한국마린엔지니어링학회지, 738-745,
   2009년07월
- [14]박태두,김민혁,김남수,김철승,정지원,이성로 "DVB-SSP 기반 혼합형 MPE-FEC 복호 알고리즘", 한국통신학회논문지, 848-854, 2009년09월
- [15]김민혁,박태두,김남수,김철승,정지원,천승용 "혼합된 위성 및 지상 전송 시스템에서 새로운 상위계층 복호 알고리즘", 한국전자파학회논문지, 835-842,2009년10월
- [16]김철승,김민혁,김남수,정지원 "LLR 기반의 MPE-FEC 상위계층 복호 방 식", 한국해양정보통신학회지, 2227-2234, 2009년12월
- [17]김민혁,박태두,김철승,정지원,이성로,정민아 "802.11n 규격에서의 저복잡 도 LDPC 복호 알고리즘", 한국통신학회논문지, 148-154, 2010년02월
- [18]김민혁,박태두,김철승,정지원 "수중 다중경로 채널에서 효과적인 채널추 정을 위한 비트 분리 방법", 한국해양정보통신학회지, 2207-2214, 2010년 10월
- [19]박태두,김민혁,김철승,정지원 "Cross Layer를 이용한 Hyblid ARQ System 분석", 한국전자파학회논문지, 1229-1234, 2010년11월
- [20]박태두,김민혁,김철승,정지원 "Cross-Layer 부호기법을 이용한 Hybrid-ARQ 기반의 협력통신 시스템 연구", 한국통신학회논문지, 889-895, 2010 년11월
- [21]김철승,김민혁,박태두,정지원 "무선 랜 규격에서의 고속 알고리즘을 이 용한 LDPC 복호기 구현", 한국해양정보통신학회지, 2783-2790, 2010년12



86

월

- [22]박태두,김민혁,김철승,정지원 "MIMO 채널에서 LLR 추정을 위한 저 계 산량 알고리즘", 한국해양정보통신학회지, 2791-2797, 2010년12월
- [23]박태두,김민혁,정지원 "DVB-RCS Next Generation을 위한 Third-dimention Turbo Code 분석", 279-285, 2011년02월
- [24]박태두,김민혁,임병수,정지원 "차세대 DVB-RCS 시스템을 위한 저 계산 량 연판정 e-BCH 복호 알고리즘", 한국전자파학회논문지, 705-710, 2011 년07월
- [25]박태두,김민혁,임병수,정지원 "LDPC 부호를 이용한 MISO 채널에서 수 중통신 채널 성능 분석", 한국통신학회논문지, 878-885, 2011년07월
- [26]김민혁,박태두,임병수,이인기,오덕길,정지원 "DVB-RCS Next Generation을 위한 터보 부복호화 방식 분석", 한국통신학회논문지, 537-545, 2011년09 월
- [27]김민혁,박태두,임병수,정지원 "DVB-RCS NG 시스템을 위한 연판정 e-BCH 부호의 구현을 위한 최적 양자화 비트 수 결정", 한국해양정보통 신학회지, 1895-1902, 2011년09월

국내 학술 대회

- [1] 배종태, 김민혁, 최석순, 정지원 "T-DMB 방송을 위한 계층적 변조방식 이 적용된 DVB-RCS기반 터보 부호에 관한 연구" 2007 한국통신학회 하 계종합학술발표회 논문집 pp.313, 2007년 7월
- [2] 최석순, 배종태, 김민혁, 정지원, 최은아 "Train Interruption 채널에서 이 동성 DVB-S2 시스템 성능 분석" 2007 한국통신학회 하계종합학술발표 회 논문집 pp.314, 2007년 7월
- [3] 김민혁, 배종태, 최석순, 정지원 "CATV 하향 스트림 채널 코딩 기법 구 현" 2007 한국통신학회 하계종합학술발표회 논문집 pp.526, 2007년 7월
- [4] 김민혁, 정진희, 최석순, 배종태, 정지원 "전송량 향상을 위한 MIMO 수 중 통신용 S/W 모델" 한국 마린엔지니어링학회 후기학술대회 논문집, pp.195-196, 2006년 11월



- [5] 정진희, 김민혁, 정지원 "위성통신에서 Deep Fading 극복을 위한 인터리 버 구조 연구" 한국 마린엔지니어링학회 후기학술대회 논문집, pp.197-198,2006년 11월
- [6] 정진희, 최덕군, 김민혁, 최은아, 이인기, 정지원 "멀티레벨 변조 방식 적용을 위한 디맵핑 알고리즘" 한국통신학회 제 16회 통신정보 합동학 술대회 논문집 pp.01-05, 2006년 3월
- [7] 정진희, 김민혁, 정지원 "효율적인 부분 병렬 연산을 이용한 LDPC 부 호화 알고리즘" 한국통신학회 제 16회 통신정보 합동학술대회 논문집 pp.01-05, 2006년 3월
- [8] 김민혁, 최덕군, 정진희, 정지원 "선박인터넷 고품질 서비스를 위한 DVB-RCS기반 터보부호성능분석" 한국마린엔지니어링 학회 후기 학술 대회 논문집, pp.61-62, 2005년 11월
- [9] 김남수,김민혁,배종태,최석순,박태두,정지원 "이동형 DVB-S2 시스템을 위한 상위 계층 부호화 방식 성능 분석", JCCI, 2008년04월
- [10] 박태두,김민혁,배종태,최석순,김남수,정지원 "반복 부호의 8PSK 변조 방식 적용을 위한 최적의 비트 분리 방법", JCCI, 2008년04월
- [11] 김민혁,배종태,최석순,박태두,김남수,정지원 "CATV 하향 스트림 채널 코딩 시스템 구현", JCCI, 2008년04월
- [12] 배종태,김민혁,최석순,박태두,김남수,정지원 "차세대 지상파 DMB의 부
   가서비스 전송을 위한 효과적인 채널부호 알고리즘에 대한 연구", JCCI,
   2008년04월
- [13] 김남수,김민혁,박태두,김철승,정지원 "3-path 수중채널에서 통신 성능 향상 기법", 한국정보처리학회춘계학술발표대회, 2009년04월
- [14] 김민혁,박태두,김남수,김철승,정지원 "CATV 하향 스트림 적용 시스템 에서 동기 검출 방안 및 FPGA 설계", 한국정보처리학회춘계학술발표대 회, 2009년04월
- [15] 김철승,김민혁,박태두,김남수,정지원 "LLR 방식의 MPE-FEC 복호 알고 리즘", 한국정보처리학회춘계학술발표대회, 2009년04월



- [16] 박태두,김민혁,김남수,김철승,정지원 "MIMO 채널에서 LLR 추정을 위
   한 저 계산량 알고리즘", 한국정보처리학회춘계학술발표대회, 2009년04
   월
- [17] 김남수,김민혁,박태두,김철승,정지원 "수중 통신 채널에서 채널 추정 오차 분석", 한국통신학회하계학술발표대회, 2009년06월
- [18] 김민혁,박태두,김남수,김철승,정지원 "802.11n 규격에 기반을 둔 LDPC 복호의 계산량 감소 알고리즘", 한국통신학회하계학술발표대회, 2009년 06월
- [19] 김철승,김민혁,박태두,김남수,정지원 "위성통신에서 혼합형 상위계층 복호방식", 한국통신학회하계학술발표대회, 2009년06월
- [20] 박태두,김민혁,김남수,김철승,정지원 "시공간 부호화 알고리즘에서 LLR 추정 알고리즘", 한국통신학회하계학술발표대회, 2009년06월
- [21] 김철승,김민혁,박태두,김남수,정지원,천승용,손권 "수중 다중경로 채널 에서 LLR기법을 이용한 채널추정방식", 한국음향학회수중음향학학술발 표회, 2009년09월
- [22] 박태두,김민혁,김철승,정지원 "수중 다중경로 채널에서 OFDM 기반의 반복 부호 성능 분석", 한국수중음향학회,2010년06월
- [23] 김민혁,박태두,김철승,정지원 "무선 랜 규격에서 LDPC 부호의 고속화 및 저 계산량 알고리즘 성능 분석과 구현에 관한 연구", 대한전자공학 회하계종합학술대회, 2010년06월
- [24] 김철승,김민혁,박태두,정지원 "DVB-RCS Next Generation을 위한 Turbo code 부복호화 방식 분석", 대한전자공학회하계종합학술대회, 2010년06월
  [25] 박태두,김민혁,김철승,정지원 "수중통신에서 적합한 채널 부호의 성능

분석", ADD40주년기념학술대회, 2010년08월

[26] 박태두,김민혁,임병수,정지원,천승용,손권 "MISO 채널에서 수중통신을 위한 LDPC 부호 성능 분석", 한국음향학회,2011년05월



학부 2년, 석사 2년, 박사 4년, 총 8년이라는 긴 시간의 마지막을 장식하 려 합니다. 8년이라는 시간 동안 연구실에 있으면서 많은 것들을 보고, 배 우고, 익혀나가며 조금씩 성장한다는 느낌을 갖고, 이제는 어디를 가든 한 사람 이상의 몫을 해낼 자신감과 능력을 갖추게 된 것 같습니다. 제가 이 러한 인재가 될 수 있도록 많은 지도와 사랑을 주신 지도 교수 정지원 교 수님께 깊은 감사의 말씀 올립니다. 교수님 사랑합니다. 그리고 박사 과정 의 마무리를 아름답게 할 수 있도록 많은 지도를 해주신 ETRI의 오덕길 팀장님, 코메스타의 조용훈 이사님, 김기만 교수님, 윤영 교수님 깊은 감사 드립니다.

학과를 잘 이끌어 주시고 저를 포함한 모든 전파인들에게 많은 가르침을 주시는 김동일 교수님, 조형래 교수님, 강인호 교수님, 민경식 교수님 감사 합니다. 건강하십시오. 김은정 조교, 대학원 생활 하면서 많은 도움 고맙다. 승목이랑 평생 행복하게 잘 살아라.

낳아 주시고, 길러 주시고, 항상 응원해 주시며, 지금껏 기다려 주신 아버 지, 어머니 효도하겠습니다. 감사하고, 또 감사합니다. 사랑합니다. 하나 밖 에 없는 누나와 멀리서 물 건너 온 매형 앤드류 고맙습니다. 이쁜 우리 조 카 재하야 사랑한다. 고모들과 작은 아버지들 모두 사랑합니다. 감사합니다. 진우형, 성우형, 희정이 누나, 희원이 누나, 준석 매형, 두 형수님들 감사합 니다. 사랑합니다. 사촌 동생들 해인, 성경, 선후, 성윤이 고맙다. 사랑한다. 조카 현정이, 선연이, 정운이, 서준이 사랑한다. 삼촌이 맛난거 많이 사주마. 형남아 결혼 축하한다. 못가서 미안하다. 청운이, 승원이 파이팅 하자! 고맙 다. 이모님들 감사합니다.

처음 연구실을 들어왔을 때, 저에게 아낌없이 모든걸 가르쳐 주신 인기형, 덕군이 형 고맙습니다. 형님들 덕분에 지금의 제가 있습니다. 처음 뭣도 모 르고 들어온 나에게 신경써주고 잘해준 진회야 고맙다. 연구실에서 처음 방장 노릇하느라 힘들 때 많이 도와주고 같이 마셔준 종태, 석순이 고맙다. 사랑한다. 안좋은 내 성격 때문에 연구실에서 많이 고생한 태두, 남수, 철 승이 모두 고맙다. 지금의 연구실이 있기까지 그 기반을 닦아주신 연구실 선배님들, 상명이 형, 상우 형, 성준이 형, 태길이 형 감사합니다. 다음에는 제가 술 한잔 사겠습니다. 앞으로 나 없이 연구실을 이끌어 나가야 될 후 배들, 군열, 병수, 해찬, 태훈이 열심히 해서 훌륭한 사람이 되거라. 고맙다. 언제나 나를 응원해 주고, 걱정해 주는 친구들, 필식이, 영진이, 재우, 현 우, 재용이, 용철이 고맙다. 자주 못보지만 한번씩 볼때마다 많이 힘이 되 었다. 앞으로도 지금처럼 서로 믿고, 응원해 주도록 하자. 사랑한다. 그리고 민규와 동호, 고등학교 친구들은 너희 둘 밖에 없는 것 같다. 앞으로도 자 주 연락하면서 지내자. 고맙다. 사랑한다.

군 재대 후 복학했을 때, 학부 생활 적응 잘 할 수 있도록 도와준 선배님 들, 승재 형, 재우 형, 덕진이 형, 세영이 형, 동환이 형, 경식이 형, 석봉이 형, 필수 형, 영배 형, 기대 형, 모두 감사합니다. 보고 싶고 그립네요. 선배 이자 친구 수찬이, 성욱이, 정표, 내가 너희들에게 잘 했는지는 모르겠지만,



언제나 너희들을 좋아하고, 고맙고, 감사하게 생각한다는 것은 알아 주길 바란다. 징그러운 후배들 경학이, 재영이, 경관이, 용훈이, 승구 고맙다. 사 랑한다. 뒤 늦게 친해져 더 아쉬움이 있는 항상 생각나고 반가운 동기 철 희 고맙다. 사랑한다. 늦게 들어왔지만 누구보다 친해진 진옥이, 종호 고맙 다. 사랑한다.

대학원에 있으면서 친한 사람들이 학교를 떠나감에 따라 다른 연구실들과 의 교류가 많이 없어져서 아쉬웠습니다. 회사 다니시며 다시 학교로 돌아 오신 DSP의 외형이 형 많은 조언 감사합니다. 마이크로파의 동수를 비롯하 여 이동통신, RF, DSP, 안테나, MMIC 모든 대학원 생들 파이팅! 하시길 바 랍니다.

이 감사의 글을 쓰면서 많은 사람들을 다시 떠올리게 되고, 그와 함께 많 은 추억들이 떠올랐습니다. 어렸을 때 친구들과의 추억, 학부 때 선후배들 과의 추억, 대학원 때 연구실에서의 추억들이 떠오르면서 기분이 묘해집니 다. 조금 먹먹해 지기도 합니다. 인생이 90까지라면 저는 이제 1/3을 조금 지나 왔습니다. 나머지 2/3의 인생 역시 많은 추억과 행복으로 채울 수 있 도록 더욱 열심히 달려보려 합니다. 그 동안 저를 지지해 주시고 따뜻하게 지켜봐 주신 모든 분들 감사 드립니다. 반드시 좋은 모습으로 다시 뵙도록 하겠습니다. 감사하고 또 감사합니다. 사랑합니다.



