

HSS 기반 초고속 LDPC 복호를 위한 구조

이 인 기*, 김 민 혁*, 오 덕 길*, 정 지 원°

A High Speed LDPC Decoder Structure Based on the HSS

In-Ki Lee*, Min-Hyuk Kim*, Deock-Gil Oh*, Ji-Won Jung°

요 약

본 논문은 위성방송 전송 규격인 DVB-S2의 기반 채널 부호로 사용되는 LDPC를 고속 복호를 위해 HSS(Horizontal Shuffle Scheduling) 방식을 기반으로 고속 복호기 구조를 연구하였다. 첫째로 HSS방식에서 발생하는 메모리 충돌을 극복하는 방식을 제시한다. 둘째로 고속 복호를 위해 LUT(Look Up Table)을 이용하는 Sum-Product 알고리즘 대신 min값에서 scaling factor를 곱하는 Normalized Min-Sum 알고리즘을 사용하였으며, 성능 향상을 위해 check node에서 bit node로 입력되는 값의 부호를 확인하여 신뢰성 없는 값을 삭제하는 Self-Correct 방식을 제시하여 sum-product 방식에서 발생하는 bottle neck 현상을 하였다. 마지막으로 고속화를 위한 효율적인 메모리 구조를 제안한다.

Key Words : DVB-S2, HSS, LDPC, High Speed, Memory conflict, Memory Structure

ABSTRACT

This paper proposes the high speed LDPC decoder structure base on the DVB-S2. Firstly, We study the solution to avoid the memory conflict. For the high speed decoding process the decoder adapts the HSS(Horizontal Shuffle Scheduling) scheme. Secondly, for the high speed decoding algorithm normalized Min-Sum algorithm is adapted instead of Sum-Product algorithm. And the self corrected is a variant of the LDPC decoding that sets the reliability of a $M_c \rightarrow v$ message to 0 if there is an inconsistency between the signs of the current incoming messages $M_v \rightarrow c$ and the sign of the previous incoming messages $M_{old} \rightarrow c$. This self-corrected algorithm avoids the propagation on unreliable information in the Tanner graph and thus, helps the convergence of the decoder. Start after striking space key 2 times. Lastly, and this paper propose the optimal hardware architecture supporting the high speed throughput.

I. 서 론

광대역 위성 방송에 적용될 수 있는 오류 정정 부호로서 고속 데이터 전송에 효율적이고 성능이 우수한 복호기의 적용이 불가피하다. 또한 DVB-S2에서 제시된 LDPC 부호화 방식은 초고속화 다채널

실감 방송 서비스를 전국 단위로 제공하기 위한 100Mbps급 이상의 초고속 위성 방송 전송 기술로 적합하다. LDPC부호는 높은 부호화 복잡도를 가지며 이를 위해 간편한 부호화 구조를 가지는 설계 기법등이 연구되어 진다^[1]. DVB-S2의 LDPC의 경우는 부호의 간략화를 위하여 패리티 행렬에서 계

* 주저자 : 한국전자통신연구원 위성방송융합연구팀, popularity1@etri.re.kr, 정희원

° 교신저자 : 한국해양대학교 전파공학과 위성통신 연구실, jwjung@hhu.ac.kr, 정희원

* 한국전자통신연구원 위성광대역방송통신연구실

논문번호 : KICS2012-10-479, 접수일자 : 2012년 10월 6일, 최종논문접수일자 : 2013년 1월 10일

단식 형태를 가지는 구조를 제시하였다. 하나 코드 워드의 길이가 64800으로 대단히 크며 고속화를 위해서는 부분 병렬형태로 복호 알고리즘을 수행하여야 한다^[2]. 그리고 고속 복호를 위해서는 LDPC 복호기의 메모리 read/write 연산을 효과적으로 하기 위한 구조를 가져야 한다. 본 논문은 첫째로, 위성 방송 전송 규격인 DVB-S2의 기반 채널 부호로 사용되는 LDPC를 고속 복호하기 위하여 bit node 계산을 check node 계산 후에 하지 않고 check node 계산 중에 수행하는 HSS(Horizontal Shuffle Scheduling) 방식을 연구하여 최적의 반복횟수를 제시한다. 또한 LDPC의 H matrix의 cyclic-4형태로 인하여 HSS방식을 이용시 발생하는 메모리 충돌을 극복하는 방식을 제시한다. 둘째로 고속 복호를 위해 LUT(Look Up Table)을 이용하는 Sum-Product 알고리즘에 비하여 미세한 성능열화는 있으나 min 값에서 scaling factor를 곱하는 Normalized Min-Sum 알고리즘을 이용하여 연산 구조를 간략화 시켰다. 또한 성능 향상을 위해 check node에서 bit node로 입력되는 값의 부호를 확인하여 신뢰성 없는 값을 삭제하는 Self-Correct 방식을 제시하여 sum-product 방식에서 발생하는 bottle neck 현상을 하였다. 마지막으로 고속화를 위한 효율적인 메모리 구조를 제안한다.

II. 본 론

2.1. DVB-S2 복호 알고리즘^[2]

DVB-S2에 제시한 복호 알고리즘은 기존 복호 알고리즘과 거의 동일하나, LUT(Look Up Table)를 만드는 방법 및 Check Node 확률을 구하는 방법상에서 약간의 차이를 가진다. 수신비트에다가 채널 추정 값을 구하는 초기화 과정, Check node 확률을 구하는 CNU(check Node Update), 비트 확률을 구하는 BNU(Bit Node Update)로 세가지 단계로 나눌 수 있다.

step 1 : Initialization

$$Lq_{n \rightarrow m_i} = Lf_n = -L_c \cdot r_n \left(L_c = \frac{2}{\sigma^2} \right),$$

$$n = 0, 1, \dots, N-1, \quad i = 1, 2, \dots, \text{deg}(\text{bit node } n)$$

(1)

step 2. check node update

$$Lr_{k \rightarrow n_i} = g(Lq_{n_1 \rightarrow k}, Lq_{n_2 \rightarrow k}, \dots, Lq_{n_{i-1} \rightarrow k}, Lq_{n_{i+1} \rightarrow k}, \dots, Lq_{n_{dc} \rightarrow k}) \quad (2)$$

$$g(a, b) = \text{sign}(a) \times \text{sign}(b) \times \{ \min(|a|, |b|) \} + LUT_g(a, b)$$

$$= \ln \left(\frac{e^{a+b} + 1}{e^a + e^b} \right) \quad (3)$$

step 3. Bit node update

$$Lq_{n \rightarrow m_i} = Lf_n + \sum_{j \neq i} Lr_{m_j \rightarrow n} \quad (4)$$

여기서 σ 는 채널 잡음의 표준 편차이며, Lf_n 는 초기 확률값, $Lr_{k \rightarrow n_i}$ 는 k번째 Check Node에서 i번째 bit node로 업데이트 되는 값이며, $Lq_{n \rightarrow m_i}$ 는 n번째 bit node에서 m번째 check node로 업데이트 되는 값이다.

2.2. Layered Scheduling Method

기존의 LDPC 복호기의 복호 순서는 우선 수신 데이터를 이용하여 bit node를 초기화 한 후 각각의 check node에 연결된 bit node 값을 이용하여 check node를 업데이트를 한다. check node 업데이트 후 다시 각각의 bit node에 대해 업데이트를 하고, 이러한 연산을 반복회수 만큼 수행한다. 기존의 복호 방식에 의해 check node 업데이트 연산이 모두 끝난 후 bit node 업데이트를 하기 때문에 한번의 반복에도 많은 지연이 발생하여 고속의 LDPC 복호를 할 수 없다. 이를 극복하기 위해 본 논문에서는 Horizontal Shuffle Scheduling(HSS) 복호 방법을 이용하였다. HSS 방식은 기존의 방식과는 달리 check node 업데이트 연산을 하면서 bit node 업데이트 연산을 동시에 하는 것이 가능하다. HSS 복호 방식의 흐름도는 다음 그림 1과 같다. 여기서 v1~v7는 bit node에서 check node로 업데이트 되는 값, u1~u7은 check node에서 bit node로 업데이트 되는 값을 의미한다. 고속 복호를 위해 본문1의 LDPC 복호 연산중 step2와 step3의 복호 알고리즘을 병렬로 가져가기 위해서 Layered scheduling 기법이 사용된다^[3].

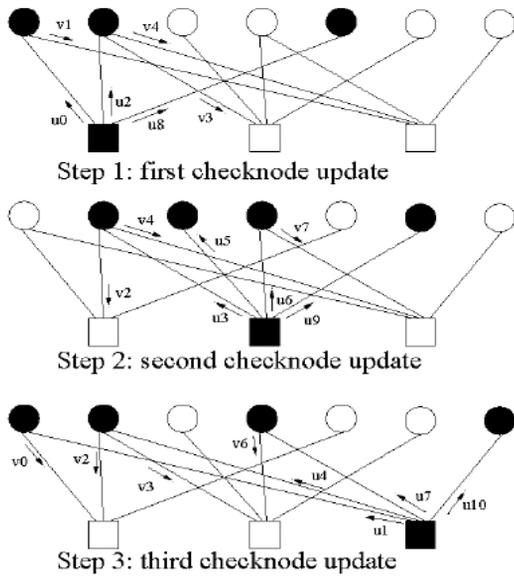


그림 1. HSS 복호 방식의 흐름도
Fig. 1. The flow of HSS decoding scheme

그러나 layered scheduling 기법을 이용하여 부분 병렬을 이용하여 복호 연산을 할 경우 그림 2와 같은 메모리 충돌이 발생한다^{4)[5]}.

즉 고속 복호를 위해 그림 2의 check node group 내의 check node는 병렬 연산을 한다. 그러나 check node group 1의 경우와 같이 cyclic-4 형태가 발생할 경우 동시에 check node group 1내의 각각의 check node의 값을 업데이트 이전 값을 이용하여 계산하게 되어 성능 열화를 발생시킨다. 이를 극복하기 Layered scheduling 기법에서 check node 연산 후 bit node로 업데이트 되는 메시지 u 과 이를 누적한 bit node 누적값 S 는 다음과 같이 계산된다.

$$S = LLR + u_0 + u_1 + u'_2 \quad (5)$$

$$S_1 = LLR + u_0 + u'_1 + u'_2 \quad (6)$$

$$S_2 = LLR + u'_0 + u_1 + u'_2$$

$$S = S_1 + S_2 - S' \quad (7)$$

여기서 S_i 는 bit node i 번째의 전송 비트율의 확률값이며 S' 는 이전 iteration에서 연산된 값을 의미한다.

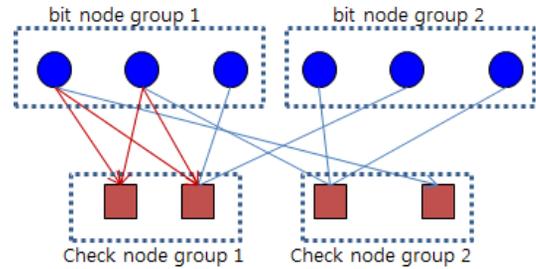


그림 2. 메모리 연결 충돌의 예
Fig. 2. The example of memory conflict

이를 이용한 성능 결과는 아래 그림 3과 같다. check node 연산이후 bit node를 연산하는 기존의 방식과 비교하였을 때 iteration 30회부터 거의 같은 값을 알 수 있다.

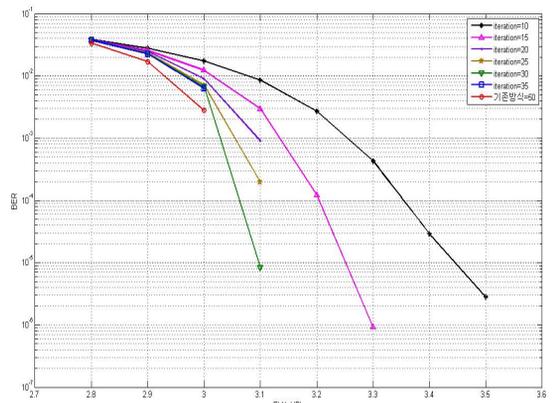


그림 3. 기존 알고리즘과 HSS의 성능 비교(R=2/3)
Fig. 3. The BER graph between HSS and existing algorithm

2.3. Self collected CNU기법

LDPC 복호과정에서의 CNU 계산을 보다 효율적으로 하여 성능을 개선시킬 수 있는 방안으로 Self-Corrected 방식을 제안하였다. 잡음이 첨가된 비트의 경우 CNU 계산과정에서 iteration을 거치면서 check node 계산을 위해 bit node서 들어오는 값의 부호가 바뀌게 되는 현상이 발생하게 된다. Self-Corrected 방식은 CNU 과정에서 check node를 계산하기 위해서 check node에 연결되어 있는 bit node로부터 값을 받게 되는데, 이전 iteration에서의 값과 비교하여 부호가 다르다면 check node로 들어오는 edge 값을 0으로 만들어주는 것이다. 이전 iteration에서의 edge값이 다음 iteration에서 부호가 바뀌었다는 것은 그 edge값은 신뢰할 수 없는 신호이기 때문에 그 값을 0으로 만들어 주는 것이다.

이렇게 CNU 계산과정에서 신뢰할 수 없는 신호를 0으로 만들어주면서 CNU 계산을 더욱 효율적으

로 하여 성능의 향상을 가져올 수 있다.

다음 그림 4는 Self-Corrected 방식의 구조를 나타내고 있다. 그림 5는 sum-product 방식과 Self-Corrected 방식의 성능을 비교한 그래프이며, Self-Corrected 방식이 0.1dB정도 좋다.

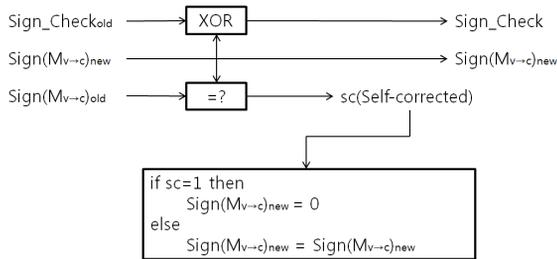


그림 4. Self-Corrected 방식의 구조
Fig. 4. The structure of Self-Corrected scheme

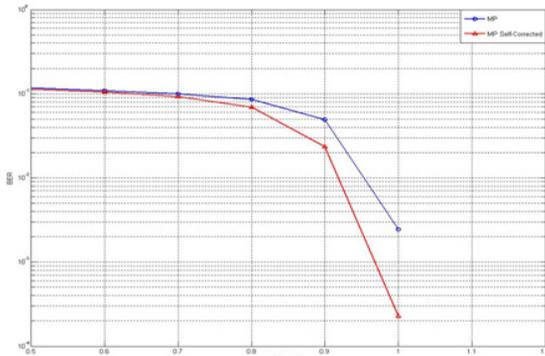


그림 5. MP(sum-product)방식과 SC 방식의 성능비교(R=1/2)
Fig. 5. The BER graph between MP scheme and SC scheme(R=1/2)

2.4. 고속 LDPC Decoder 구조

HSS 기반의 고속 LDPC 복호기의 구조는 그림 6과 같다. 그림에서 Info_mem은 수신데이터 메모리로 한 주소 당 360개의 수신 데이터를 저장한다. 첫 번째 반복 시 Info_mem에서 Sj_mem으로 처음 데이터가 입력 되어지고, 이 때는 Sj_index를 사용하지 않는다. Sj 메모리에 모든 데이터가 저장된 후 Sj_index 메모리에 저장되어 있는 8비트의 index 데이터가 출력되어 되고, 이 index 데이터는 Sj 메모리의 주소 값이 되어 Sj 데이터가 출력된다. 이렇게 출력된 Sj 데이터는 rotator로 입력되어 진다. Rotator에 의해 우 순환 shift 되어진 360개의 데이터는 Check Node 업데이트 된다. 또한,

Self-Corrected 알고리즘을 위하여 첫 번째 반복 때는 각 데이터의 sign 비트 360개를 sign 메모리에 저장한다. 첫 반복 때는 Self-Corrected 알고리즘을 사용하지 않고, 원래의 데이터로만 CNU 연산을 실행한다. 이 때, CNU 연산과정은 다음 그림 7과 같다.

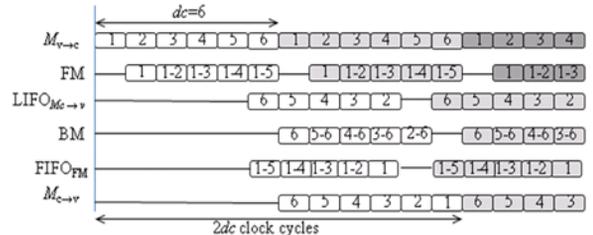


그림 7. CNU 연산과정
Fig. 7. CNU calculation process

CNU 블록에서 연산되어 진 데이터는 각각의 edge 값이므로 출력 순서대로 Edge 메모리에 저장 이 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 필요하다. 역 rotator의 permutation index는 rotator의 permutation index와 구조가 같다. 하지만 역 rotator는 우 순환 shift 연산과는 반대로 좌 순환 shift 연산을 필요로 한다. 역 rotator를 거쳐 원래의 데이터 순서대로 되 돌려진 360개의 데이터는 Sj index에 따라 Sj 메모리에 입력되어 진다. 이렇게 첫 번째 반복이 끝나고, 두 번째 반복부터는 Info 메모리를 사용하지 않고, Sj 메모리의 데이터를 Sj index에 따라 출력시킨다. 이 360개의 데이터는 rotator를 거쳐 CNU 연산이 되어지는데, 두 번째 반복부터는 CNU 연산 블록에 입력되기 전에 edge 메모리에서 이전 반복 시의 edge 데이터를 출력시켜 Sj 데이터에서 빼 값을 CNU 연산 블록으로 입력시킨다. 또한, 이 때의 sign 비트와 sign 메모리에 저장되어진 이전 반복 때의 sign 비트를 xor 연산을 시킨 뒤 Sj 데이터와 같이 입력시킨다. Sign 비트의 xor 연산은 SC(Self-Correction) 알고리즘을 적용시키기 위함이다.

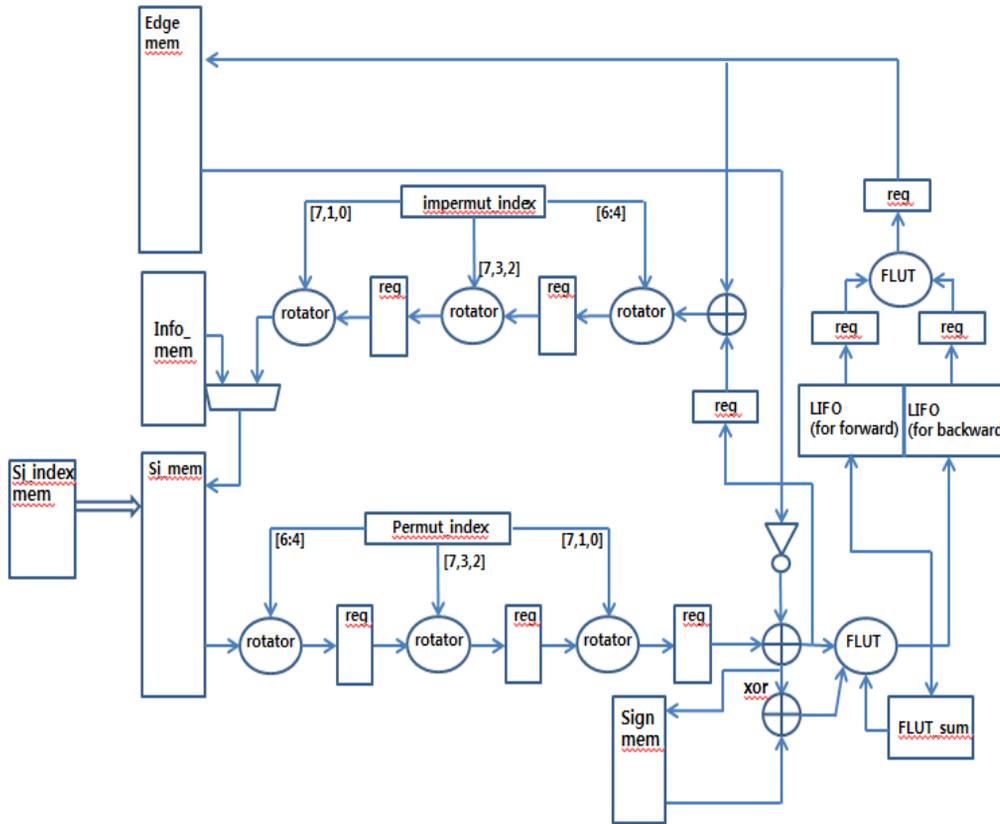


그림 6. HSS 기반의 LDPC 복호기의 구조
 Fig. 6. The structure of LDPC decoder base on the HSS scheme

III. 결 론

초고화질 다채널 실감 위성 방송 서비스를 위해서는 서비스의 연속성이 보장되고 채널을 효율적으로 사용할 수 있어야 한다. 이를 위해서 300Mbps 급 이상의 초고속 위성 방송 전송 기술 및 차세대 위성 방송 확보를 위해 기반 기술의 개발은 위성방송시스템의 주요 목표이다. 본 논문에서는 300Mbps 급 이상의 초고속 위성 방송 전송 기술을 위해 DVB-S2를 기반으로 LDPC 부호의 고속 복호 알고리즘, 최적의 메모리 구조를 제안한다. 특히 본 논문에서는 첫째로 고속 복호 알고리즘 관점에서는 bit node 계산을 check node 계산 후에 하지 않고 check node 계산중에 수행하는 HSS(Horizontal Shuffle Scheduling) 방식을 연구하여 최적의 반복 횟수를 제시하였다. HSS 방식은 check node 계산과 동시에 bit node를 계산하기 때문에 별도의 bit node 계산을 할 필요가 없으므로 고속화 가능하며, 또한 반복 횟수 또한 감소시킬 수 있어 복호 throughput을 증가시킬 수 있다. 시뮬레이션 결과 약 30% 에서 50% 반복횟수를 동일한 성능에서 감

소시킬 수 있음을 확인할 수 있었다. 둘째로 HSS 방식을 적용하기 위해 메모리 충돌 현상 극복하는 방식을 분석하였다. 또한 반복복호중 반복한 edge값과 현재 반복 시 edge값을 비교하여 부호가 바뀌면 신뢰성이 없음을 간주하여, “0”으로 할당하는 SC(Self-Corrected)방식을 도입하여 분석하였다. SC 방식의 적용으로 인해 0.1dB ~ 0.5 dB의 성능이 개선됨을 알 수 있다. 마지막으로 위에서 제시한 첫 번째, 두 번째 알고리즘을 토대로 고속 LDPC 복호를 위한 복호기 구조 및 내부 메커니즘을 제안하였다.

References

- [1] B.W. Chung, J.S. Kim, H.Y. Song, “Efficient Design of Structured LDPC Codes,” *J. KICS*, vol. 31 no. 1C, pp. 14-19, Jan. 2006
- [2] 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.

- [3] M. Gomes, G. Falcao, V. Silva, V. Ferreira, A. Sengo, and M. Falcao. "Flexible parallel architecture for DVB-S2 LDPC decoders." in *Proc. IEEE Global Telecom. Conf., GLOBECOM*, pp. 3265-3269, Nov. 2007.
- [4] C. Marchand, J. B. Dore, L. C.-Canencia, and E. Boutillon. "Conflict resolution for pipelined layered LDPC decoders." in *Proc. Sig. Process. Syst., SiPS 2009. IEEE Workshop*. pp. 220-225, Nov. 2009.
- [5] 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.

이 인 기 (In-Ki Lee)



2003년 8월 한국해양대학교
전파공학과(공학사)
2005년 8월 한국해양대학교
전파공학과(공학석사)
2005년 7월~현재 한국전자통
신연구원 위성방통융합연구
팀 선임연구원

<관심분야> 위성방송통신, 변·복조기술, 채널코딩,
물리계층 FPGA 구현 기술 등

김 민 혁 (Min-Hyuk Kim)



2006년 2월 한국해양대학교 전
파 공학과 (공학사)
2008년 2월 한국해양대학교 전
파 공학과 (공학석사)
2012년 8월 한국해양대학교
전파공학과 박사과정
2012년 11월~현재 한국전자통

신연구원 선임연구원
<관심분야> 위성 통신, 이동 통신, 변·복조 기술,
채널 코딩, FPGA 기술 등

오 덕 길 (Deock-Gil Oh)



1980년 2월 서울대학교 전자공
학과(공학사)
1984년 2월 서울대학교 전자공
학과(공학석사)
1996년 2월 서울대학교 전자공
학과(공학박사)
1982년~현재 한국전자통신연

구원 위성방통융합연구팀 팀장
<주 관심분야> 무선접속기술, 이동통신방송 시스
템, 차세대 위성방송 시스템

정 지 원 (Ji-Won Jung)



1989년 2월 성균관대학교 전
자공학과(공학사)
1991년 2월 성균관대학교 전
자공학과(공학석사)
1995년 2월 성균관대학교 정
보공학과(공학박사)
1991년 1월~1992년 2월 LG

정보통신연구소 연구원
1995년 9월~1996년 8월 한국통신 위성통신연구실
선임연구원
1997년 3월~1998년12월 한국전자통신연구원 초빙
연구원
1996년 9월~현재 한국해양대학교 전파공학과 정교
수
2001년 8월~2002년 8월 캐나다 NSERC Fellowship
(Communication Research Center 근무)
<관심분야> 위성 통신, 이동 통신, 변·복조 기술,
채널 코딩, FPGA 기술 등