

# 버퍼 변환을 이용한 저면적 ISDB-T 시간 디인터리버 구조

## A Small-Area ISDB-T Time Deinterleaver Structure with Buffer Transformation

강형주\*

Hyeong-Ju Kang\*

### 요 약

본 논문에서는 저면적 ISDB-T 시간 디인터리버 구조를 제안하였다. ISDB-T는 일본과 중남미에서 많이 사용되고 있는 이동형 TV 표준으로써 긴 인터리빙을 이용하여 다른 표준에 비해 높은 성능을 보이고 있다. 그러나 긴 인터리빙을 구현하기 위해서는 많은 지연 버퍼가 필요하다. 지연 버퍼들은 주소 레지스터가 있어야 하므로 주소 레지스터의 개수도 많아진다. 본 논문에서는 디인터리버의 등가 변환을 통해 주소 레지스터의 개수를 크게 줄이는 구조를 제안하였다. 실험 결과를 통해 디인터리버의 면적을 줄일 수 있음을 확인할 수 있었다.

### Abstract

This paper presents a small-area ISDB-T time deinterleaver structure. ISDB-T is an mobile TV standard that is widely used in Japan and many South American countries. One of the strong points of the standard is the long interleaving depth, which enhance the communication performance. However, long interleaving requires many delay buffers, in other words many pointer registers. This paper reduces the number of pointer registers with the deinterleaver equivalent transformation. The experimental results show that the area is reduced with the proposed structure.

Key words : ISDB-T, Deinterleaver, Convolutional Deinterleaver, ISDB-T Time Deinterleaver

### I. 서 론

최근의 전자 기기 경향에서 이동형 기기는 빨라질 수 없는 화두일 것이다. 전화기와 컴퓨터 등에서 이런 경향을 볼 수 있으며 TV 시청에서도 지난 몇 년 동안 이동형 디지털 TV가 점차 각광받고 있다.

대한민국에서 위성 Digital Multimedia Broadcasting(DMB)로 처음 상업화에 성공한 이후로, 대한민국의 지상파 DMB, 일본의 Integrated Services

Digital Broadcasting-Terrestrial(ISDB-T), 유럽의 Digital Video Broadcasting-Handheld (DVB-H) 등의 표준들이 발표되거나 방송되고 있다[1].

이 중 ISDB-T는 일본에서 2003년에 시험 방송한 후로 일본에서만 서비스되어 오다가 2007년에 브라질에서 채택된 것을 시작으로 많은 중남미 국가들이 채택하였다. 그리고 아프리카의 여러 나라들도 ISDB-T의 채택을 검토하면서 최근 많은 관심을 받고 있다.

\* 한국기술교육대학교 컴퓨터공학부(School of Computer Science and Engineering, Korea University of Technology and Education)

· 제1저자 (First Author) : 강형주

· 투고일자 : 2011년 3월 9일

· 심사(수정)일자 : 2011년 3월 10일 (수정일자 : 2011년 4월 4일)

· 게재일자 : 2011년 4월 30일

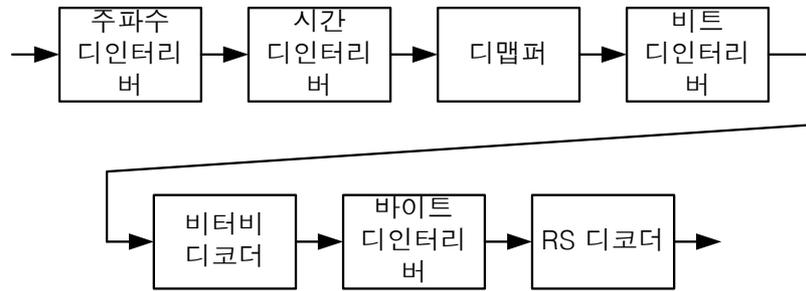


그림 1. ISDB-T 수신기의 FEC  
Fig. 1. FEC of ISDB-T receiver

ISDB-T 표준의 장점 중 하나는 긴 인터리빙(interleaving)을 사용함으로써 통신 환경이 좋지 않을 때 상대적으로 수신 성능이 좋다는 점이다[2]. 인터리빙은 forward error correction(FEC) 방식 중의 하나로써, 송신할 때 데이터의 순서를 바꾸어서 인접한 데이터를 떨어뜨리는 방식을 말한다. 수신할 때는 송신에서와 반대로 순서를 바꿈으로써 원래의 순서를 복원한다. 송신기에서 순서를 바꾸는 과정을 인터리빙이라고 부르고, 수신기에서 순서를 복원하는 과정을 디인터리빙(deinterleaving)이라고 부른다.

채널에서 발생하는 에러는 수신기에 데이터 순서 복원과정만을 지나게 된다. 그러므로 채널에서 인접한 데이터에 발생한 에러는, 수신기에 디인터리빙 과정을 거치면서 서로 떨어지게 되는 것이다. 즉, 연속 에러(burst error)가 임의 에러(random error)로 바뀌는 것이다. 일반적인 에러 정정 코드는 연속 에러를 정정하는 능력에 한계가 있다. 인터리빙 기법을 사용하면 연속 에러가 임의에러로 바뀌므로 연속 에러 정정 능력이 향상된다.

ISDB-T는 길쌈 인터리빙(convolutional interleaving)이라는 긴 인터리빙에 알맞은 방식을 사용한다. 이 방식은 데이터들에 서로 다른 시간만큼 지연시켜서 데이터의 순서를 바꾼다[3]. 데이터들을 지연시키기 위해서는 지연 버퍼가 필요하다. ISDB-T의 경우에는 긴 인터리빙을 위해 지연 버퍼에 많은 양의 데이터를 저장한다. 따라서 일반적인 지연 버퍼의 구현 방법인 쉬프트 레지스터(shift register) 방식을 사용할 수 없다.

쉬프트 레지스터 방식의 대안으로 많이 사용되는 것은 써큘러 버퍼(circular buffer) 방식이다. 써큘러 버퍼 방식은 지연시킬 데이터들을 메모리에 저장하

고, 그 메모리에 읽거나 쓰는 위치를 적절히 제어함으로써 데이터들을 지연시킨다. 즉, 주소를 증가시키면서 읽고 쓰는데, 읽기 주소를 쓰기 주소보다 3만큼 작도록 유지시키면 데이터를 3만큼 지연시킨 효과가 생긴다. 써큘러 버퍼 방식에서는 메모리를 읽거나 쓰는 위치를 저장해야 하며, 지연 양이 일정한 지연 버퍼의 경우에는 지연 버퍼 당 하나의 주소 레지스터로 이 역할을 할 수 있다.

ISDB-T 3-seg 표준은 1152개의 지연버퍼를 사용한다. 즉, 1152개의 주소 레지스터가 필요한 것이다. 이러한 레지스터는 개수가 너무 많으므로 면적을 많이 차지한다. 그러나 이전의 디인터리버에 대한 연구에서는 메모리의 효율적 사용에 집중되어서, ISDB-T 디인터리버의 주소 레지스터를 줄이는 분야는 거의 연구되지 않았다[4]-[7]. ISDB-T 수신기 전체의 설계에 관해 많이 연구되어온 것에 비하면 ISDB-T 디인터리버에 대한 연구는 부족한 상태이다[8]-[10].

본 논문에서는 ISDB-T의 디인터리버에서 주소 레지스터의 개수를 줄여서 면적을 줄이는 구조를 제안한다. 데이터들이 일정한 순서로 입력되는 디인터리버에서는 여러 개의 지연 버퍼를 한 개의 지연 버퍼로 변환할 수 있으며, 이 성질을 이용해서 주소 레지스터의 개수를 줄일 것이다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 ISDB-T 표준과 디인터리버 구조에 대해 설명하고, 3장에서 새 구조를 제안할 것이다. 4장에서 실험결과를 제시한 뒤, 5장에서 결론을 맺을 것이다.

## II. ISDB-T 디인터리버

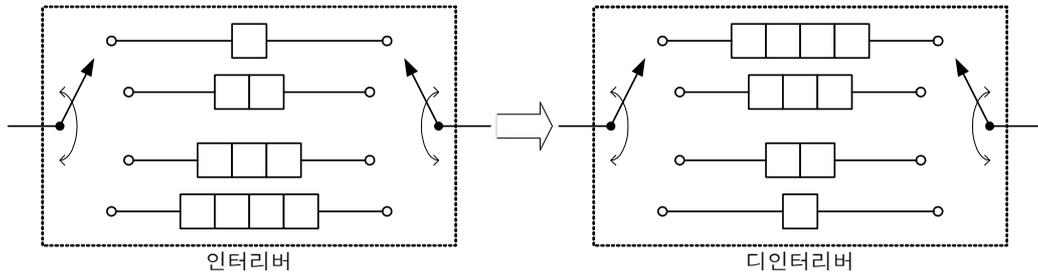


그림 2. 길쌈 인터리빙  
Fig. 2. Convolutional interleaving

우선 ISDB-T 수신기의 FEC 부분을 설명하고, ISDB-T의 인터리버에 대해 검토할 것이다.

2-1 ISDB-T FEC

ISDB-T 수신기의 FEC는 그림 1과 같이 구성되어 있다. 주파수 디인터리버는 주파수 영역에서의 디인터리빙이며, 그 다음의 시간 디인터리버는 시간 영역에서의 디인터리빙이다. 그 다음에는 디맵퍼를 통해 데이터 비트를 복원하고 비터비 디코더와 Reed Solomon (RS) 디코더를 거쳐 에러를 정정한다.

ISDB-T의 FEC는 네 개의 인터리빙을 채택하고 있다. 이 중에서 주파수 인터리빙, 비트 인터리빙, 바이트 인터리빙은 상대적으로 길이가 짧으며 블록 인터리빙(block interleaving) 방식이다. 이에 반해 시간 인터리빙은 길이가 매우 길며 길쌈 인터리빙 방식이다.

블록 인터리빙은 데이터를 블록 단위로 나누고 그 블록 내에서 데이터의 순서를 바꾼다. 예를 들어, 데이터들을 가상의 사각형에 가로 방향으로 쓴 뒤 세로 방향으로 읽는 것이다. 이렇게 하면 인터리빙 전에

인접했던 데이터들이 인터리빙 후에는 사각형의 높이만큼 떨어지게 되는 것이다.

길쌈 인터리빙은 그림 2와 같이 데이터들을 여러 개의 경로로 나눈 뒤, 각 경로마다 서로 다르게 지연시킨다. 지연시키는 양이 다르므로, 인터리빙 전에 인접했던 데이터들이 인터리빙 후에는 경로의 수만큼 떨어지게 된다. 수신기에서는 송신기에서와 같이 여러 경로로 데이터들을 나누고, 송신기에서의 지연을 보상하는 만큼 지연시켜서, 원 데이터의 순서를 복원한다.

2-2 ISDB-T 시간 디인터리버

ISDB-T 시간 디인터리버의 기본 블록은 그림 3과 같다 [1]. 96개의 경로에 대해 지연 버퍼의 길이는  $0 \cdot L$ 부터  $95 \cdot L$ 까지이다. 여기서 L은 0, 1, 2, 4 중 하나의 값이다. ISDB-T 3-seg 시간 디인터리버는 이러한 기본 블록을 그림 4와 같이  $3 \cdot 4$  개 사용해서 구성된다.

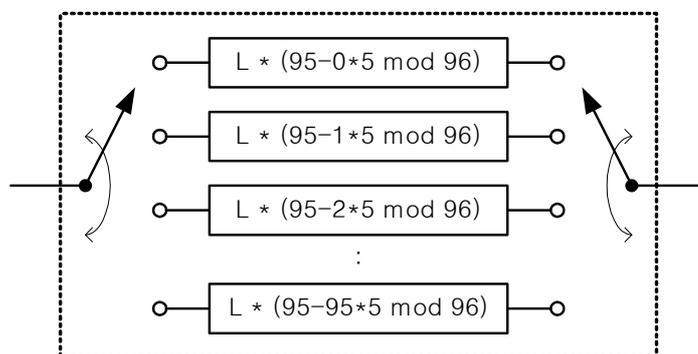


그림 3. ISDB-T 시간 디인터리버의 기본 블록  
Fig. 3. Basic block of ISDB-T time deinterleaver

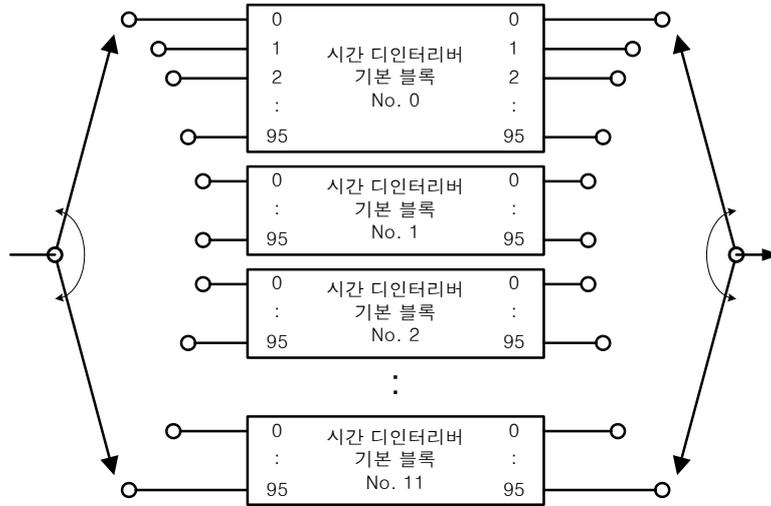


그림 4. ISDB-T 3-seg 시간 디인터리버  
Fig. 4. ISDB-T 3-seg time deinterleaver

Ⅲ. 주소 레지스터의 개수를 줄이는 ISDB-T 시간 디인터리버 구조

이 절에서는 ISDB-T 시간 디인터리버의 구현상의 문제점을 분석한 뒤 저면적 시간 디인터리버 구조를 제안할 것이다.

3-1 시간 디인터리버의 구현

ISDB-T 시간 디인터리버는 총 1152개의 지연 버퍼로 구성되며 지연 버퍼의 길이를 모두 더하면  $54,720 * L$ 이 된다. 데이터의 폭이 2 byte라고 가정하면 지연 버퍼에 저장할 데이터의 양은  $109,440 * L$  byte이다. 이것을 쉬프트 레지스터로 구현하면 그 면적이

매우 커지므로 써큘러 버퍼 방식으로 구현해야 한다.

써큘러 버퍼 방식에서는 데이터들을 메모리에 보관하고, 메모리에 쓰거나 읽는 위치를 적절히 제어함으로써 데이터들을 지연시킨다. 일반적인 버퍼에서는 읽기 위치와 쓰기 위치를 모두 저장해야 하나, 길이가 일정한 지연 버퍼는 읽기 위치와 쓰기 위치의 차이가 일정하므로 하나의 위치만 저장하면 된다. 즉, 지연 버퍼 하나 당 위치 주소 레지스터가 한 개 필요하다.

ISDB-T 시간 디인터리버를 써큘러 버퍼 방식으로 구현하면 1152개의 주소 레지스터가 필요하다. 이를 바로 구현하면 역시 면적을 많이 차지한다.

3-2 시간 디인터리버의 주소 레지스터 개수를 줄이는 구조

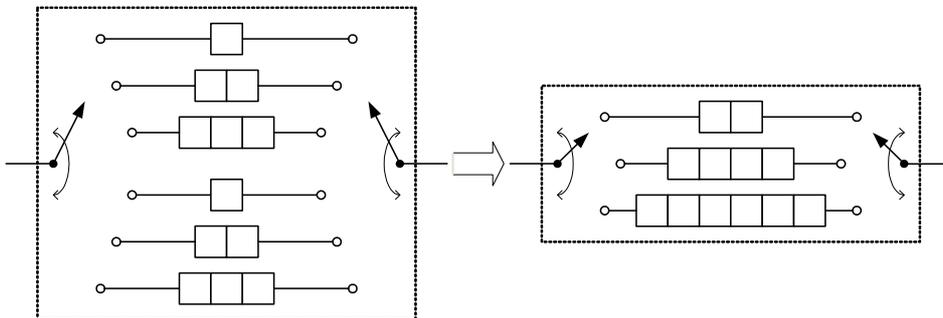


그림 5. 디인터리버 등가 변환  
Fig. 5. Deinterleaver equivalent transformation

매우 커지므로 써큘러 버퍼 방식으로 구현해야 한다.

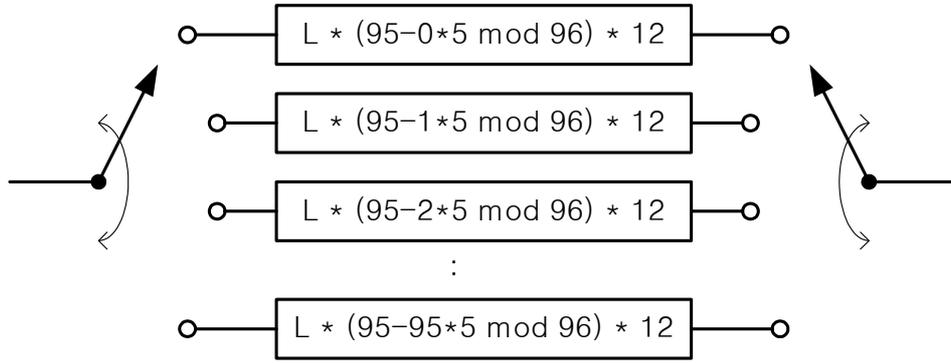


그림 6. 변환된 ISDB-T 시간 디인터리버  
Fig. 6. Transformed ISDB-T time deinterleaver

디인터리버와 같이 지연 버퍼의 길이가 일정한 패턴으로 반복될 경우 그림 5와 같은 변환을 할 수 있다. 그림 5의 왼쪽에서는 데이터를 여섯 개의 경로로 나누고 각각을 1, 2, 3, 1, 2, 3만큼 지연시킨다. 오른쪽의 그림에서는 지연 양이 같은 경로들을 합쳐서, 지연양이 각각 2, 4, 6인 세 개의 경로로 만들었다.

디인터리버의 입력 데이터를  $a(n), n \geq 0$ , 왼쪽 구조의 출력 데이터를  $b(n), n \geq 0$ , 오른쪽 구조의 출력 데이터를  $c(n), n \geq 0$ 으로 표현하자. 왼쪽 그림에서는 6개의 경로로 나누므로 6으로 나눈 나머지 값에 따라 지연 값이 1, 2, 3, 1, 2, 3이 된다. 따라서  $b(n)$ 을 다음과 같이 쓸 수 있다.

$$b(n) = \begin{cases} a(n - 6 \times 1) & \text{if } n \% 6 = 0 \\ a(n - 6 \times 2) & \text{if } n \% 6 = 1 \\ a(n - 6 \times 3) & \text{if } n \% 6 = 2 \\ a(n - 6 \times 1) & \text{if } n \% 6 = 3 \\ a(n - 6 \times 2) & \text{if } n \% 6 = 4 \\ a(n - 6 \times 3) & \text{if } n \% 6 = 5 \end{cases} \quad (1)$$

오른쪽 그림에서는 3개의 경로로 나누므로 3으로 나눈 나머지 값에 따라 지연값이 2, 4, 6이 된다. 따라서  $c(n)$ 을 다음과 같이 쓸 수 있다.

$$c(n) = \begin{cases} a(n - 3 \times 2) & \text{if } n \% 3 = 0 \\ a(n - 3 \times 4) & \text{if } n \% 3 = 1 \\ a(n - 3 \times 6) & \text{if } n \% 3 = 2 \end{cases} \quad (2)$$

$b(n)$ 과  $c(n)$ 에서  $n \% 6 = 0$ 인 경우를 비교해 보면,  $b(n) = a(n-6)$ 이고  $c(n) = a(n-6)$ 이어서 서로 같다. 나머지 경우들도 모두 비교해 보면  $b(n)$ 과  $c(n)$ 이 서로 같

음을 알 수 있다.

그림 5에서 소개한 변환을 구조적인 면에서 살펴보자. 이 변환에서는 전체 지연 버퍼의 길이의 합은 일정하므로 데이터를 저장할 저장공간의 크기가 줄어드는 것은 아니다. 그러나 디인터리버에 있는 경로의 개수를 줄일 수 있다. 그림 5의 왼쪽에서 여섯 개였던 경로가 변환을 한 후에는 세 개로 줄었다. 디인터리버 구조에서 경로의 개수만큼 지연 버퍼가 필요하므로, 경로의 개수를 줄이면 지연 버퍼의 개수도 줄일 수 있다.

ISDB-T 시간 디인터리버의 구현상의 문제점은 지연 버퍼의 개수가 많음에 따라 주소 레지스터의 개수도 많다는 것이므로, 이 변환을 사용하면 주소 레지스터의 개수를 줄일 수 있다. ISDB-T 시간 디인터리버는 같은 구조의 디인터리버 블록이 12개 반복되는 구조이므로 그림 5의 변환을 적용할 수 있다. 그림 6이 변환된 ISDB-T 시간 디인터리버이다. 1152개의 지연 버퍼를 96개의 길이가 12배인 지연 버퍼로 변환하였다. 이와 같이 변환하면 1152개의 주소 레지스터를 96개로 줄일 수 있다.

#### IV. 실험 결과

본 논문에서 제안한 구조와 기존의 일반적인 방식으로 구현한 구조를 합성을 통해 비교하였다. 데이터의 폭은 16비트로, 클럭 주파수는 25MHz로 가정하였다. Verilog-HDL로 레지스터 전송 수준에서 설계를 하고, TSMC의 0.25um 공정으로 합성하였다. 합성틀

표 1. 기존 구조와 제안한 구조의 비교

Table 1. The comparison of the conventional structure and the proposed structure.

	주소 레지스터 개수 (개)	면적 (게이트 카운트)	전력 소모 (mW)
기존 구조	1,152	95,494	67.96
제안한 구조	96	10,908	7.51
비율	8.3%	11.4%	11.1%

은 Cadence 사의 RTL Compiler를 사용하였다. L은 2일 때를 가정하였다.

표 1에서 기존 구조와 제안한 구조를 비교하였다. 기존 구조에서는 주소 레지스터의 개수가 1,152개였던 반면에 제안한 구조는 96개에 불과하다. 따라서 제안한 구조의 합성 면적이 기존 구조에 비해 작다. 표 1에서 제시한 면적은 합성툴에서 제공하는 면적으로 등가의 2-입력 NAND 게이트 카운트로 변환한 것이다. 제안한 구조가 10,908이고 기존의 구조가 95,494이어서, 면적이 1/9로 줄어들었음을 알 수 있다.

합성된 netlist를 이용하여 게이트 수준 시뮬레이션을 한 뒤, 각 노드의 toggle 정보를 RTL Compiler에 제공하여 전력을 측정된 결과도 표 1에서 제시하였다. 면적이 줄어든 만큼 전력 소모도 줄어서 약 1/9로 줄일 수 있었다.

## V. 결 론

본 논문에서는 ISDB-T 시간 디인터리버의 저면적 구조를 제안하였다. 시간 디인터리버에서 필요한 데이터를 저장하기 위해서는 싸클러 버퍼 방식으로 구현해야 한다. 싸클러 버퍼 방식에서는 지연 버퍼 하나 당 한 개의 주소 레지스터가 필요하므로, 많은 주소 레지스터가 필요하게 된다. 본 논문에서는 디인터리버의 등가 변환을 통해 주소 레지스터의 개수를 1/12로 줄였다. 이를 통해 면적과 전력 소모를 1/9로 줄일 수 있었다. 본 논문에서는 ISDB-T의 시간 디인터리버에 국한해서 구조를 제안하였으나, 지연 버퍼의 개수가 많은 다른 디인터리버에도 적용이 가능할 것이다.

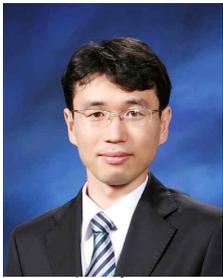
## 참 고 문 헌

- [1] Transmission system for digital terrestrial television broadcasting, ARIB STD B31.
- [2] Y. Wu, E. Pliszka, B. Caron, P. Bouchard, and G. Chouinard, "Comparison of terrestrial DTV transmission systems: the ATSC 8-VSB, the DVB-T COFDM, and the ISDB-T BST-OFDM," *IEEE Trans. Broadcasting*, pp. 101-113, 2000.
- [3] J. L. Ramsey, "Realization of optimum interleavers," *IEEE Trans. Inform. Theory*, pp. 338-345, 1970.
- [4] M. Rim, "A VLSI architecture for convolutional deinterleavers," *Proc. Int. Conf. Consumer Electronics*, pp. 130-131, 1996.
- [5] Y.-N. Chang, "A multibank memory-based VLSI architecture of DVB symbol deinterleaver," *IEEE Trans. Very Large Scale Integration Systems*, pp. 840-843, 2010.
- [6] H. Yang, Y. Zhong, and L. Yang, "An FPGA prototype of a forward error (FEC) decoder for ATSC digital TV," *IEEE Trans. Consumer Electronics*, pp. 387-395, 1999.
- [7] Y. You, J. Wang, M. Yu, and Y. Ye, "Design and implementation of concatenated decoder," *International Workshop on Digital and Computational Video*, pp. 135-142, 2002.
- [8] K. Mori, M. Suzuki, Y. Ohara, S. Matsuo, and A. Asano, "Development of low power ISDB-T one-segment decoder by mobile multi-media engine SoC(S1G)," *Asia and South Pacific Design Automation Conference*, pp. 644-648, 2007.
- [9] K. Fukuda, K. Watanabe, M. Ouchi, N. Tokunaga, and

T. Kamada, "A 9mW OFDM demodulator LSI for narrow-bandwidth ISDB-T," *IEEE International Conference on Consumer Electronics*, pp. 90-91, 2003.

- [10] M. Okada, T. Masaki, T. Iwasaki, and N. Ueno, "Narrowband OFDM receiver architecture for partial reception of ISDB-T signal," *IEEE International Conference on Consumer Electronics*, pp. 202-203, 2003.

### 강 형 주 (姜亨周)



1998년 2월 : 한국과학기술원 전기 및 전자공학과(공학사)

2000년 2월 : 한국과학기술원 전자 전산학과(공학석사)

2005년 2월 : 한국과학기술원 전자 전산학과(공학박사)

2005년 3월 ~ 2006년 1월 (주)매그나칩반도체

2006년 2월 ~ 2009년 2월: (주)지씨티리씨치

2009년 3월 ~ 현재: 한국기술교육대학교 전임강사

관심분야 : VLSI 설계, 통신 모뎀 설계