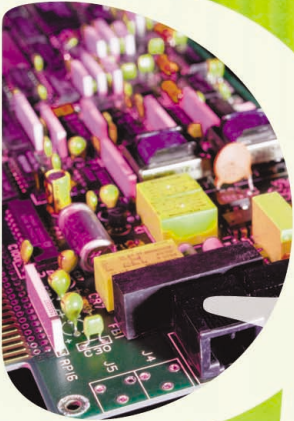


채널 코덱 설계 방법론



M

>> Design Methodology

채널 코덱에 대한 최근 동향을 알아보고, 그것을 채널 코덱 설계에 어떻게 연결하여 발전시킬 것인가, 설계의 관점에서의 채널코덱, 그에 따른 채널 코덱의 일반적인 설계 방법론, 구체적인 예로 고속 비터비 복호기와 고속 터보 복호기의 구체적인 설계 방법과 향후 채널 부호에 대하여 간략히 살펴보기로 한다.

채널 코덱 설계 방법론

전인산 책임연구원_한국전자통신연구원 모뎀SoC설계팀(isjeon@etri.re.kr)

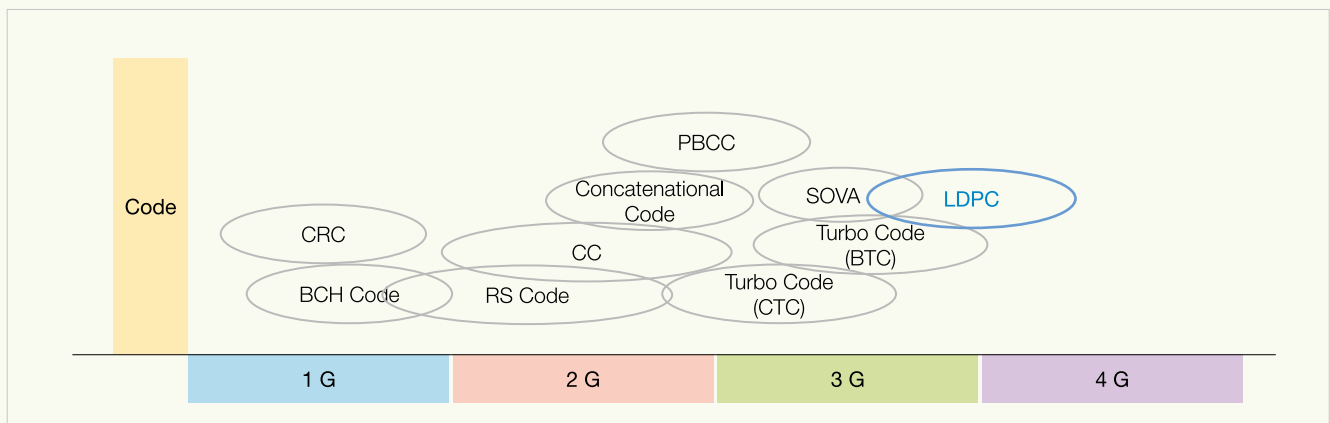
김혁 선임연구원_한국전자통신연구원 모뎀SoC설계팀(haggy@etri.re.kr)

1. 서론

디지털 통신 시스템에는 채널상에서 발생하는 오류를 정정하기 위하여 채널코덱을 사용한다. 채널코덱을 분류하는 방법은 여러가지가 존재하나 크게 블록 부호와 길쌘부호(Convolutional Code) 그리고 이들을 서로 연결하여 사용하는 연접부호(Concatenation Code)로 나누어 볼 수 있다. 블록 부호는 오류정정이론에 그 기반을 두고 발전한 것으로 간단한 선형순회부호로부터 시작하여 BCH(Bose-Chaudhuri-Hocquenghem) 부호 및 RS(Reed-Solomon) Code 등이 있다. 길쌘부호는 기존의 길쌘부호와 이를 변형한 터보 부호가 있다. 터보 부호는 길쌘부호의 변형으로 출현하였으나, 현재에는 블록부호를 사용한 터보 부호도 나타났다. 이 둘을 구분하기 위하여 길쌘부호를 사용한 터보부호를 CTC(Convolutional Turbo Code)라하고 블록부호를 사용한 터보 부호를 BTC(Block Turbo

Code)라 한다. 그리고 이전에 만들어 졌었으나 그 복잡도로 인하여 잊혀져 있다가 새롭게 재 발견된 LDPC(Low Density Parity Check) 부호도 있다. 이 중 터보 부호와 LDPC부호는 반복부호라고 부르기도 하며, C. E. Shannon에 의해 발표된 채널용량(channel capacity) 근처까지 오류정정이 가능한 강력한 오류정정 부호로 알려져 있다.

오류정정부호는 다양한 표준에서 채택하고 있으며 통신 시스템의 세대 진화에 따라서 오류정정 부호도 발전을 하여 왔다. <그림 1>은 각 세대별 통신시스템에서 채택한 오류정정부호를 보여준다. 여기서 PBCC는 Packet Binary Convolutional Code이고 CC는 Convolutional Code 이고 SOVA는 Soft Output Viterbi Algorithm이다. 3세대 통신시스템에서 사용되고 있는 오류정정부호의 특징은 터보 부호의 출현이다. 터보 부호는 연성복호를 통하여 한번 복호한 결과를 다시 복호기의 입력에 인가하여 다시 복호를 수행하는 반복복호이다. 이 반복복호과정을 통하여 이론적으로 채널용량까지 오류정정이 가능한 강력한 오류정정 부호가 3세대 통



<그림 1> 통신시스템에 사용된 주요 오류정정 부호



신시스템에서 사용되는 오류정정 부호의 특징이라 할 수 있다. 그러나 이것을 실제 구현하는데 있어서는 오류마루 현상이 나타나는데 이것을 향상시킨 것이 LDPC 부호이다. 모든 통신시스템의 발전의 바탕에는 SoC(System-On-a-Chip)의 진화가 그 밑바탕에 존재한다. 채널코덱의 경우도 그동안 복잡성으로 인하여 구현을 시도도 하지 못했던 것이 이제는 가능할 뿐 만 아니라 나아가 병렬처리까지 가능한 수준에 이르고 있다. 그러한 의미에서 채널 코덱도 전체 SoC의 설계의 동향을 크게 벗어날 수 없으므로 SoC에 대한 최근 동향을 살펴보고 그것을 채널코덱의 설계에 어떻게 연결하여 발전시킬 것인가를 고려하는 것은 자연스러운 일이라 하겠다. 2장에서는 설계의 관점에서 채널코덱을 살펴보고 3장에서는 그에 따른 채널 코덱의 일반적인 설계 방법론에 대하여 살펴보고, 그 구체적인 예로 고속 비터비 복호기 와 고속 터보 복호기의 구체적인 설계 방법을 살펴보고 향후 채널 부호에 대하여 간략히 살펴보고 결론을 내린다.

II. 설계관점에서 채널코덱

NRE(non-recurring engineering) 비용의 증가는 반도체의 모든 분야에 영향을 끼친다. 설계 분야도 예외는 아니어서 플랫폼과 IP(Intellectual Property)기반의 SoC, Structured ASIC, ASIP(Application Specific Instruction Processor) 등 여러가지 다른 모델들이 시도되고 있다. 또한 메모리, 디지털 설계 그리고 아날로그 설계 등 이질적인 요소를 하나의 칩에 올리는 것보다 각각 최적화하여 하나의 패키지에 넣는 SiP(Silicon-In-Package)가 SoC의 대안으로 나타나고 있다. 그러한 메가 트렌드에 의해 디지털 설계도 영향을 받을 것은 자명하다. 이러한 영향 아래에 있는 디지털 설계는 그 설계 방법에 따라서 다음과 같이 나누어 볼 수 있다. 먼저 CPU나 DSP와 같은 ISA(Instruction Set Architecture)를 가지는 설계, 프로토콜을 효율적으로 처리하기 위한 프로토콜 지향 설계 그리고 연산알고리즘을 효율적으로 처리하기 위한 연산 지향 설계로 나눌 수 있다. 이러한 관점에서 채널 코덱은 연산 지향 설계에 속한다.

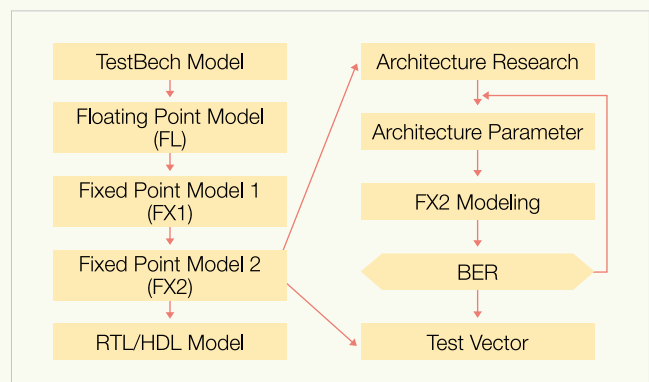
또한 같은 연산알고리즘의 설계에 있어서도 알고리즘이 결정되면 상세한 설계구조까지 결정되는 설계가 있는가 하면 알고리즘에서 아키텍처를 어떻게 설계하느냐에 따라서 성능이 크게 좌우되는 아키텍처 중심의 설계가 있다. 채널 코덱의 경우 아키텍처 중심의 설계가 이루어져야 한다. SoC설계에 있어서 가장 중요한 특징 중의 하나는 IP재활용과 플랫폼기반의 설계이다. 채널코덱도 SoC의 관점에서는 하나의 IP이다. SoC시대 이전의 IP는 고정된 IP나 매개변수화된 IP가 주종을 이루었다. 그러나 이들 IP는 기존의 사용자가 재사용하는 경우 인터페이스 설계에 많은 시간이 걸리고 병렬처리나 파이프라인의 변화를 주어 IP를 향상하는 것은 어렵다.

반면 SoC 시대의 IP는 이러한 것을 극복하여 기존의 IP 수준을 넘어선 가상 부품(Virtual Component) 수준으로 발전할 것을 요구하고 있다. 특히 부동소수점 모델, 고정소수점 모델, RTL/HDL(Register Transfer Level/Hardware Description Language) 모델 그리고 마지막으로 FPGA/ASIC (Field Programmable Gate Array/Application Specific Integrated Circuit) 까지 모든 수준의 모델이 부품수준으로 지원을 요구하고 있다. 채널 코덱도 가상부품의 조건을 만족해야 한다.

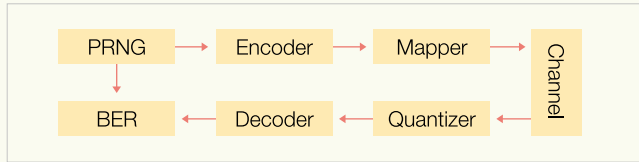
III. 일반적인 채널 코덱의 설계 흐름도

채널코덱의 설계 흐름은 <그림 2>에 나타나 있다. 먼저 주어진 오류정정 부호의 사양에서 주어진 매개변수를 가지고 채널 코덱의 성능을 측정할 수 있는 테스트 벤치 환경을 만든다. <그림 3>에서 보는 바와 같이 임의의 랜덤한 데이터를 생성하기 위한 PRNG(Pseudo Random Number Generator), 표준이나 시스템 규격에서 주어진 부호기, 논리값을 신호값으로 변경하기 위한 맵퍼로 송신부를 구성한다. 채널 모델로는 일반적으로 AWGN(Additive White Gaussian Noise)을 사용하나 Rician 또는 Rayleigh 모델도 사용하며 필요에 따라서는 표준에서 제공하는 모델을 사용하기도 한다. 수신부에서는 채널을 거친 후 잡음이 추가된 신호를 받아서 연성판정을 수행하기 위한 양자화를 수행하고 그것을 복호기의 입력에 인가하여 복호를 수행하여 추정된 복호값을 얻는다. 이것을 원래의 값과 비교하여 BER(Bit Error Rate)이나 PER(Packet Error Rate)과 같은 성능을 얻는다.

이상의 모델은 먼저 부동소수점으로 구성되어 채널코덱 알고리즘이 정상적으로 동작하는지를 검증한다. 부동소수점 모델을 만드는 목적은 테스트 벤치환경이 제대로 만들어 졌는가를 검사하는 것과 채널코덱의 알고리즘



<그림 2> 채널 코덱 설계 방법론



〈그림 3〉 테스트 벤치환경

움이 정상적으로 동작하는지를 검증하는데 있다. 여기서는 부동소수점 모델을 FL이라 명한다. 채널 코덱에서 부동소수점 모델은 일반적으로 C/C++ 언어를 사용하여 구현한다. 기존의 통신시스템 분야에서 많이 사용되는 설계 툴인 SPW나 Matlab을 사용하여 구현 할 수도 있으나 채널 코덱의 BER의 측정범위가 10⁻⁴에서 10⁻⁸ 사이의 값을 가지므로 채널코덱의 성능을 측정하기 위한 시뮬레이션 시간이 너무 길어지기 때문에 일반적으로 사용되지 않는다. 나아가 C/C++ 프로그램을 작성하는 경우에도 시뮬레이션의 속도를 향상할 수 있도록 가능한 공통으로 사용되는 함수들은 최적화를 수행한다. 이것을 바탕으로 하여 상세 설계에 사용될 하드웨어 비트수의 범위를 알기 위하여 부동소수점의 다른 것은 변화하지 않고 고정된 비트 수를 인가한 고정소수점 모델인 FX1을 만든다. FX1을 사용하여 고정소수점의 성능을 측정하고 측정된 성능이 부동소수점성능의 값과 비교하여 시스템 성능을 만족할 수 있도록 근사적인 비트 크기를 결정한다.

채널코덱은 앞서 언급한 바와 같이 아키텍처 중심의 알고리즘을 고려한 설계가 그 관건이다. 아키텍처 설계시 최대 처리 속도 뿐 아니라 채널 부호의 특성에 따라서 여러가지 중요한 설계의 요소가 존재하며 이것에 의하여 아키텍처가 결정된다. 그런데 FX1은 알고리즘을 구현한 것으로 아키텍처를 고려하고 있지 않다. 그러므로 아키텍처가 고려된 고정 소수점 모델인 FX2를 만든다. 이것을 위해서는 기존에 발표된 아키텍처를 면밀히 분석한 후 향상된 아키텍처를 유도하고 그로부터 아키텍처 매개변수를 추출한다. 이 변수를 사용하여 아키텍처를 고려한 고정소수점 모델 FX2를 만든다. 또한 필요에 따라서는 알고리즘이나 알고리즘 내에 포함된 수식을 아키텍처 구현에 맞게 수정 또는 간소화하는 것이 필요하다. 이 모델을 테스트 벤치환경의 FX1을 대치한 후 성능을 측정하여 만족하지 않으면 아키텍처 매개변수를 수정하여 만족하는 성능을 얻도록 한다. 또한 이것은 설계의 다음 단계인 RTL/HDL의 모델과 가능한 세부 블록까지 동일하도록 마이크로 아키텍처 설계를 수행하는 것을 권고하며 이렇게하면 RTL/HDL 설계의 시간을 단축할 뿐 아니라 검증을 아주 용이하게 할 수 있고 RTL/HDL 수준에서 BER성능을 측정하는 것도 가능하다. 또한 FX2 모델을 사용한 테스트 벤치에서 RTL/HDL 설계를 검증하기 위한 테스트 벡터를 생성한다. 그러므로 FX2의 설계가 실제 채널코덱의 설계의 가장 핵심이며 설계의 승패를 나누는 기준이라 볼 수 있다.

RTL/HDL 설계과정은 그 동안 여러 차례 설계를 통하여 검증완료 된 유산 IP(Legacy IP)를 사용하여 설계한다. 유산 IP를 사용하면 HDL 코드

설계 시간 뿐 아니라 HDL 코드를 검증하는데 소요되는 시간을 현저히 줄일 수 있는 장점을 가지고 코드의 신뢰도도 높일 수 있다. 채널 코덱의 설계에 사용되는 유산 IP는 복잡도에 따라서 Pico IP, Micro IP 그리고 Macro IP로 나눌 수 있다. 채널코덱 수준의 IP는 Micro IP 보다도 복잡도가 높은 Mega IP로 분류하고 있다. 먼저 Pico IP는 정수 연산이 가능하고 비트 크기가 매개변수화 되어 있는 것으로 기본논리회로, 단순 사칙 연산기, 단순 카운터 등의 기능을 수행하는 IP를 말한다. 그리고 Micro IP는 Pico IP 처럼 비트 크기를 임의로 할당할 수 있고 제네릭을 사용하여 특정 기능을 집합적으로 수행하는 IP로 다중단을 갖는 Mux, 연산기, ACS(Add-Compare-Select)와 같은 복합연산기, 그리고 일반화된 CRC 등이 있다. 그리고 Macro IP는 특정 단독 기능을 수행하는 IP로 메모리 모델 등이 있다. 이 과정은 채널코덱 지향적인 유산 IP를 사용한다는 것 외에는 일반적인 RTL/HDL 설계과정과 동일하다.

이 이후의 설계 과정은 일반적인 IC 설계과정과 동일하게 FPGA를 통하여 에뮬레이션을 수행하여 검증을 종료하고 칩을 제작한다.

IV. 채널코덱의 설계 예

상기의 채널코덱의 설계 방법론을 사용하여 설계한 채널코덱의 예로 비터비 복호기와 터보 복호기에 대하여 살펴보고 차세대 초고속 무선 통신 표준의 선택사항으로 채택되고 있는 LDPC부호에 관하여 간단히 살펴본다.

1. 비터비 복호기

비터비 복호기는 크게 두 종류가 사용된다. 상대적으로 채널환경이 나쁜 이동통신시스템의 경우에 구속장이 9인 길쌈부호를 사용하며 채널환경이 좋은 IEEE Std 802.11 계열의 WLAN(Wireless Local Area Network) 과 WiMAX(Worldwide Interoperability for Microwave Access) 또는 mobileWiMAX로 불리는 IEEE Std 802.16 및 16e 계열의 WMAN(Wireless Metropolitan Area Network)의 경우 구속장이 7인 길쌈부호를 사용한다. 길쌈부호의 경우 부호기는 구현에 있어서 문제가 되지 않지만 비터비 복호기는 구속장의 2의 지수승에 비례하여 연산복잡도를 갖는다. 비터비 복호기의 FX2 모델을 만들기 위한 주요 아키텍처 매개변수는 전송율, 예상동작 주파수, 그리고 예상 임계 경로의 지연시간이다. 이것을 기반으로 하여 ACS의 수를 결정하며 역추적 길이와 그에 따른 구조를 결정한다. 만일 K = 9인 경우 전송율 TP 가 384Kbps이고 동작주파수 F가 30MHz이고 ACS의 임계경로가 33.3nsec를 보다 크지 않다면 필요한 ACS수인 Nacs는 수식 (1)과 같다.

$$Nacs = \text{ceiling}(TP * 2^{(K-1)} / F) \quad (1)$$



여기서 ceiling 함수는 ceiling내의 값보다 큰 수중 가장 작은 정수를 나타 내며 이 경우 Nacs는 4가 된다. 만일 임계경로가 33.3nsec 보다 크면 Nacs는 8개가 필요하다. 일반적으로 HDL설계의 제어의 복잡성을 피하기 위하여 ACS의 수는 2의 지수승으로 정한다. 역추적 값이 D의 설정은 부동소수점의 경우 구속장의 4~6배 정도의 값이 적절한 값으로 알려져 있는데 구현시 제어의 복잡성을 피하고 고정소수점에의한 성능저하를 고려하여 D = 64정도로 수행한다. 그 외에도 아키텍처 설계에 있어서 양자화된 연성판정값과 그를 사용한 가지 메트릭 계산상에 정규화와 경로메트릭을 계산하는데 있어서 정규화 등을 매개변수로 고려하여야 한다. 물론 모든 매개변수를 하드웨어 설계에 할당할 비트 크기와 함께 고려해야한다. 이러한 매개변수에서 얻어진 아키텍처 매개변수의 값을 결정하므로써 고정소수점 아키텍처 설계인 FX2를 완료한다. 이렇게 설계된 384Kbps를 지원하는 비터비복호기의 칩은 <그림 4>에 나타나 있다.

2. 터보 복호기

터보 부호는 채널용량까지 정보전달이 가능하도록 강력한 오류정정능력을 가진것으로 처음에는 CTC(Convolutional Turbo Code)가 사용되다가 BTC(Block Turbo Code)도 표준에서 채택하여 사용하고 있다. 대표적인 이동통신시스템의 표준인 HSUPA(High Speed Uplink Packet Access)로도 불리는 3GPP의 R6에서는 구속장 K=40이고 부호율 R=1/3인 전형적인 CTC를 채택하고 있으며, 대표적인 WMAN인 IEEE Std 802.16 및 16e는 구속장 K가 40이고 구성부호의 부호율이 1/2인 변형된 구성부호를 사용하는 CTC와 패리티 검사부호로 확장된 해밍부호를 가지는 BTC를 선택사항으로 채택하고 있다.

여기서는 CTC 복호기에 대하여 MAP(Maximum A Posteriori) 설계방법론을 다룬다. 터보 인터리버의 설계는 각 표준마다 서로 다른 것을 채택하고 있고 인터리버는 채널코덱의 설계와 그 특성이 다르므로 여기서 다루지 않는다. CTC 복호기의 FX2를 모델링하기 전에 하드웨어의 설계를 용이하게 하기 위하여 MAP 알고리즘에 자연로그나 상용로그를 취하는 대신에 본 설계에 있어서는 이진로그를 취하고 그에 따라 수식을 변경한다. 이렇게 하는 경우 수식이 하드웨어와 동작과 같은 이진표기로 기술되므로 설계가 용이하고 FX2에서의 BER 성능도 상용로그나 자연로그를 취하는 것보나 좋게 나타난다. 또한 로그로 처리하게 됨에 따라서 음수의 값이 존재할 수 없기 때문에 가지 메트릭을 계산한 후 정규화를 통하여 모두 양수화를 한다. 그리고 순방향 상태 메트릭과 역방향 상태 메트릭을 동일한 하드웨어로 설계할 수 있도록 수식을 수정한다. 이렇게 변환된 이진 터보 복호 알고리즘의 수식은 다음과 같다 [8].

$$D_k^{i,m} = (\log_2 e) \frac{2}{\sigma^2} (x_k^i + y_k^i Y_k^{i,m}) \quad (2)$$

$$A_k^m = \sum_{j=0}^J (D_k^{j,b(j,m)} + A_{k-1}^{b(j,m)}) \quad (3)$$

$$B_k^m = \sum_{j=0}^J (B_{k+1}^{F(j,m)} + D_{k+1}^{j,f(m)}) \quad (4)$$

$$L_k = \sum_{m=0}^{2^V-1} (A_k^{l,m} + B_k^{s(m)}) - \sum_{m=0}^{2^V-1} (A_k^{0,m} + B_k^m) \quad (5)$$

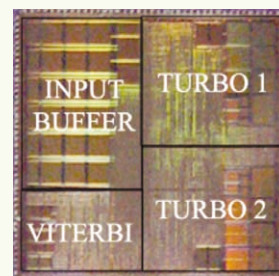
이러한 일련의 일들은 FX2보다 이전에 할 수 있지만 FL과 FX1에서는 기존의 방식에 따라 테스트 벤치환경을 만들고 그것을 기준으로 사용하여 야 하기 때문에 FX2에서 처리하는 것이 타당하다.

터보 복호기의 FX2 모델을 만들기 위한 주요 아키텍처 매개변수는 전송율, 예상동작 주파수, 그리고 예상 임계 경로의 지연시간이다. 이것을 기반으로 하여 MAP의 핵심 연산블럭인 상태메트릭 계산기의 수를 결정하며 슬라이딩 윈도우의 길이를 결정한다. 만일 구속장 K = 4인 경우 전송율 TP 가 384Kbps이고 동작주파수가 F가 30MHz이고 2함수의 임계경로가 33.3nsec를 보다 크지 않다면 필요한 상태 메트릭 계산기의 수인 Ns는 수식 (6)과 같다 .

$$N_s = \text{ceiling}(TP * 2^{(K-1)*2} * N_{itr}/F) \quad (6)$$

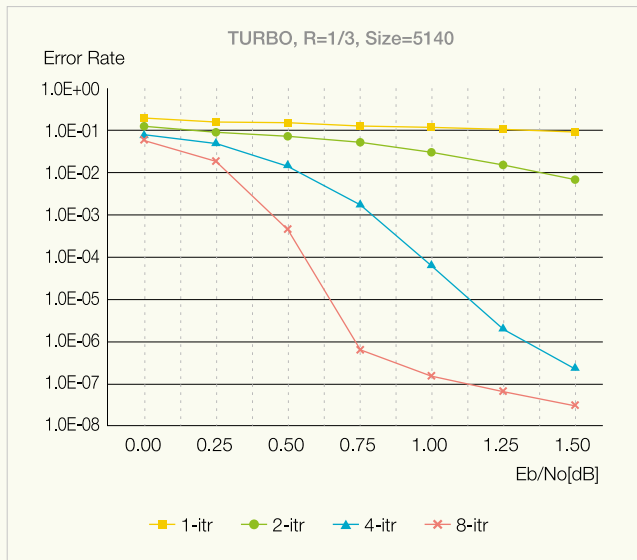
여기서 ceiling 함수는 ceiling내의 값보다 큰 수중 가장 작은 정수를 나타내며, 두번째 2는 상태 메트릭을 계산하는데 순방향과 역방향에 대하여 각각 계산하는 것을 나타내며, Nitr은 반복 복호 횟수를 나타낸다. 8번 반복이 필요한 경우 Ns는 4가되어 4개의 상태 메트릭 계산기가 필요하게 된다. 만일 임계경로가 33.3nsec 보다 크면 Ns는 8개가 필요하다. 일반적으로 HDL설계의 제어의 복잡성을 피하기 위하여 Ns의 수도 2의 지수승으로 정한다.

슬라이딩 윈도우의 크기는 시뮬레이션을 통하여 256 정도 결정한다. 앞서 언급한 바와 같이 로그 영역에서 음수의 정의는 없으므로 가지 메트릭 계산에서 정규화를 수행하여 모두 양수화를 하며 상태 메트릭 값의 증가를 막기 위하여 상태 메트릭의 정규화도 수행한다. 이러한 모든 값의 각 부분의 비트수를 적절하게 할당하면서 각 아키텍처의 매개변수를 결정한다. 이렇게 설계된 384Kbps를 지원하는 터보 복호기의 칩사진은 다음 <그림 4>와 같다. <그림 4>의 채널코덱 칩은 한 개의 비터비 복호기와 2개의 터보 복호기를 포함하고 있다.



<그림 4> WCDMA 채널코덱 칩 사진

이렇게 설계된 터보 복호기 칩의 성능은 <그림 5>에 나타난 바와 같이 AWGN 채널환경에서 부호율 R이 1/3이고 프레임의 크기가 5140비트일 경우 8번 반복복호를 하였을 때 EbNo이 0.75dB에서 BER이 10⁻⁶보다 낮은 것을 알 수 있다.



<그림 5> 터보 코덱 칩 성능

3. 차세대 채널코덱의 설계

차세대 광대역 통신시스템은 초고속의 오류정정을 요구하고 있다. 차세대 무선랜 표준인 IEEE Std 802.11n의 경우 길쌈부호를 사용하여 135Mbps에서 243Mbps 정도의 전송율을 처리하는 것이 필수사항이고 LDPC를 사용하는 경우 540Mbps에서 630Mbps 정도의 전송율이 선택 사항으로 논의 되고 있다. 또한 LDPC의 경우 IEEE 802.16e D8에 선택 사항으로 채택되었으며, Huhges 등에서 제안한 LDPC가 Philips 등에서 제안한 터보 계열의 부호를 누르고 유럽의 위성 DVB-S2의 표준 초안에 채택되어 LDPC가 광대역 통신시스템의 차세대 오류정정 부호로서 그 위치를 확고히 할 것으로 보여진다.

이러한 초고속의 채널코덱의 설계도 앞서 언급한 설계 방법론과 유사한 방법으로 설계될 수 있다. 다만 나노 공정이 진행됨에 따라서 게이트의 지연보다 선로의 지연이 길어지기 때문에 초고속 병렬 구조에서 선로 지연을 줄이기 위한 시스템릭 어레이 구조에 대한 연구가 아키텍처와 알고리즘 수준까지 반영될 것으로 보여진다.

그에 따라서 FX2의 설계에 선로 지연효과를 고려한 아키텍처설계가 중요한 이슈 중의 하나가 될 것이다.

V. 결론

본 고에서는 디지털 통신시스템의 핵심요소 중의 하나인 채널코덱의 동향을 살펴보고 SoC 설계 동향이 채널 코덱에 미치는 영향을 간략하게 살펴 보았다. 이를 바탕으로 하여 채널코덱의 설계 방법론에 대하여 설명하였고 그 예로 고속 비터비 복호기와 고속 터보 복호기의 설계 방법을 구체적으로 설명하였다. 그리고 통신시스템에서 차세대 오류정정의 표준에 따른 채널코덱 설계 방법론의 변화를 간략하게 살펴 보았다.

이상에서 살펴본 바와 같이 오류정정부호는 그 특성상 여러 표준시스템에 유사한 부호가 채택되는 것이 일반적인 사항이다. 그러므로 채널코덱을 다양한 IP 형태로 지원한다면 국내 IT-SoC의 발전에 큰 기여를 할 것이다. Ⓜ

▶ 참고문헌

- [1] IEEE Std 801.11/801.11a/802.11b, "WLAN MAC and PHY specification", IEEE, 1999
- [2] IEEE Std 801.11g, "WLAN MAC and PHY specification", IEEE, 2003
- [3] IEEE Std 801.16, "Air Interface for Fixed Broadband Wireless Access Systems", IEEE, 2004
- [4] IEEE Std 801.16e/D8, "Air Interface for Fixed and Mobile Broadband Wireless Access Systems", IEEE, May. 2005.
- [5] 3GPP TS 25.212/222, "Multiplexing and channel coding (FDD/TDD)", V6, 3GPP, Dec. 2004.
- [6] 전태현, "차세대 무선 LAN 표준화동향", 주간기술동향, 1179호, 2004년 12월
- [7] Draft ETSI EN 302 307, "DVB: Second Generation", V1.1.1, June. 2004
- [8] 전인산, "상대 메트릭을 갖는 터보 복호기 및 그를 이용한 계산방법", 국내특허 KR-1004354340000, 2004년 6월