

# OakDSPCore®를 이용한 EVRC 음성코덱의 실시간 구현

김승훈, 이동원, 김상윤, 강상원  
한양대학교 제어계측공학과

## Real-time implementation of the EVRC Codec using OakDSPCore®

Seoung-Hun Kim, Dong-Won Lee, Sang-Yoon Kim, Sang-Won Kang  
Dept. of Control & Instrumentation Eng., Hanyang Univ.  
E-mail) swkang@selab.hanyang.ac.kr

### 요약

본 논문에서는 EVRC 음성 부호화 시스템을 OakDSPCore®를 기반으로 설계된 C&S Technology 사의 CSD17C00 칩을 이용하여 전 과정을 어셈블리어로 실시간 구현하였다.

구현된 EVRC 음성 부호화기는 최대의 계산량을 요구하는 8kbps 일때 잡음제거 알고리즘을 제외한 인코더부분이 평균 22.5MIPS 이며, 디코더부분은 약 3.35MIPS 의 복잡도를 나타낸다. 사용된 메모리양은 프로그램 ROM 10.8Kwords, 데이터 ROM(table) 6.72K words 및 RAM 2.94K words 이다.

구현된 EVRC 음성 부호화기는 북미 표준화 기구인 TIA(Telecommunications Industry Association)에서 제공하는 19 개의 test 벡터들을 모두 통과하였다.

### 1. 서론

국내 및 북미의 디지털 셀룰라 CDMA 시스템의 음성 부호화기로 사용되고 있는 IS-127 EVRC(Enhanced Variable Rate Codec)[1]는 1994년 Bell Lab에서 제안한 RCELP(Relaxation CELP)[2] 알고리즘에 기반을 두고 있다. RCELP 알고리즘은 기존의 CELP 형의 분석·합성 방법과는 달리 낮은 전송률에서 성능의 저하가 없는 Generalized Abs 구조[3]를 이용한다. Generalized Abs 구조는 피치를 파라미터화하여 부호화하며 이때 발생하는 오류를 원신호의 warping 을 통해서 방지하는 방식으로, 음질 저하 없이 피치 신호에 적은 비트를 할당할 수 있으

며, 페루프 분석이 없이 개루프 피치 탐색만으로 결정된 정수 피치를 프레임 단위로 전송하므로 부프레임 단위로 실수 피치를 전송하는 기존의 CELP 방식에 비해 피치에 할당된 비트를 고정 코드북의 여기신호와 같은 중요 파라미터에 비트 할당을 늘림으로서 복호화된 신호의 음질을 향상시킬 수 있는 장점이 있다. 이러한 RCELP 알고리즘을 사용한 EVRC 음성 부호화기는 13kbps QCELP 이상의 음질을 제공하는 것으로 알려져 있으며, 다양한 무선통신 서비스들을 통합한 제 3 세대 이동통신 시스템인 IMT-2000에서 동기방식과 비동기방식중 어떤 방식이 표준안으로 채택되더라도 기존 방식과의 backward compatibility 를 제공하므로 향후 활용도가 매우 높은 음성 부호화기라 할 수 있다.

본 논문에서 사용한 CSD17C00 칩은 DSP 그룹사에서 개발된 저전력 소비형 16비트 고정 소수점 DSPCore인 OakDSPCore®를 기반으로 miscellaneous logic, serial port, compander, host interface 및 timer 의 5가지 peripheral 과 범용 I/O ports로 설계되었고, 최대 40MIPS 의 처리 용량을 갖는다.

본 논문의 구성은 2장에서 EVRC의 기본구조와 특성에 대하여 살펴보고, 3장에서는 실시간 구현에 관해 논하며, 마지막으로 4장에서 결론을 맺도록 하겠다.

### 2. EVRC 음성 부호화기

#### 2.1 EVRC 음성 부호화기

EVRC의 부호화 과정은 기본적으로 Generalized Abs 구조의

RCELP 알고리즘을 이용하여 전체 부호화 과정의 기본 개념도는 그림 1 과 같다. 입력된 신호는 8 kHz 로 샘플링되며 20ms, 160 샘플을 한 프레임으로 하고 53 샘플의 두개 부프레임과 54 샘플의 한개 부프레임으로 구성된다. LP 계수 추출을 위한 320 샘플중 80 샘플을 이후의 프레임에서 취하므로 총 알고리즘 지연시간은 30ms 가 된다.

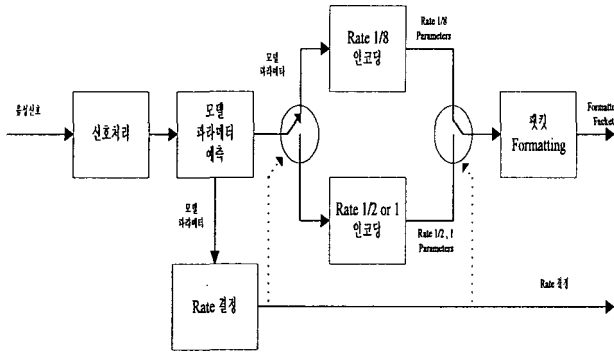


그림 1. EVRC 부호화기의 기본구조

전처리 과정으로 차단 주파수 120Hz 의 고대역 통과 필터를 사용하여 입력신호의 DC 성분을 제거하고 전처리 과정을 통과한 입력 음성 신호는 이전 프레임 80 샘플, 현재 프레임 160 샘플, 이후 프레임 80 샘플의 총 320 샘플을 이용해 LP 계수를 추출한다. 10 차의 linear prediction(LP) 해석필터를 통해 얻어진 LP 계수는 안정도 특성을 가지며 양자화 과정에서 효과적인 line spectrum pair (LSP) 계수[4]로 변환되고, 현재와 이전 프레임에서 얻은 LSP 계수를 보간하여 만든 역필터에 전처리된 음성 신호를 통과시켜 잔여 신호를 얻은후, 이 잔여 신호를 이용하여 개루프 피치 탐색에서 피치 지연값을 추정하고 개루프 장구간 예측 이득값을 구한다. 그리고 RDA(Rate Decision Algorithm)에서 LP 계수 추출과정에서 구한 17 개의 자기상관 계수와 개루프 장구간 예측 이득값 및 영교차률등을 문턱값으로 이용하여 입력신호의 특성에 따라 세 가지 전송률 중에서 최적 전송률이 결정된다. 아래의 표 1 은 각각의 전송률에서 파라미터들의 비트할당을 나타낸 것이다.

표 1. EVRC 의 비트할당

필드	패킷 형태		
	Rate 1	Rate 1/2	Rate 1/8
Spectral Transition Indicator	1		
LSP	28	22	8
피치 지연	7	7	
Delta 지연	5		
ACB 이득	9	9	
FCB Shape	105	30	
FCB 이득	15	12	
프레임 에너지 (reserved)	1		8
Total	171	80	16

입력신호의 특성에 따른 현재 프레임의 비트 전송률이 결

정되면, LSP 계수는 각 전송률에 해당하는 LSP 코드북 탐색과정에서 찾은 인덱스에 의해 프레임 단위로 양자화 되고, 그림 2 에 나타난 Generalized Abs 구조를 기본으로 하여 부프레임 단위로 적응 코드북 탐색, 고정 코드북 탐색, 개루프 이득값 추출등의 과정으로 부호화가 이루어진다.

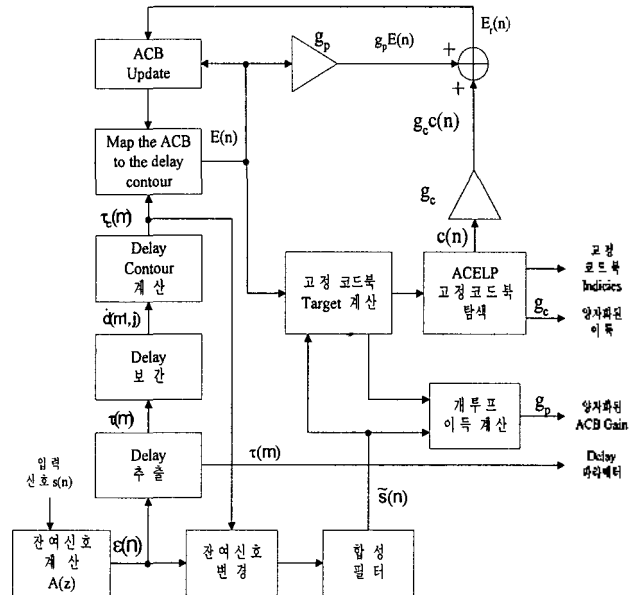


그림 2. Generalized Abs 구조

적응 코드북의 피치주기 contour 는 현재 프레임과 이전 프레임의 피치를 선형 보간하여 구한 선형 피치주기 contour 를 사용한다. 이러한 방법은 피치 주기의 선형적인 보간에 의해 많은 비트를 아낄 수 있는 장점이 있으나 보간된 부분에서 발생하는 피치주기의 작은 차이가 일반적인 Abs 메커니즘의 성능에 심각한 영향을 주 수 있는 단점이 있다. 그래서 generalized Abs 구조에서는 고정 코드북 검색에 사용되는 목적 신호를 만들기 위해 피치 주기를 파라미터화 하여 선형보간을 하며 이 보간된 피치주기 contour 와 이전 프레임에서 수정된 잔여신호로 목적신호를 만들고 이 신호에 최적으로 정합 되도록 원음이 손상 받지 않는 범위 내에서 원신호의 잔여신호를 시간변형을 통한 수정을 수행한다. 시간변형을 통한 잔여신호 변형시 신호의 불연속점을 없애기 위해 잔여신호에 대해 연속적인 time-warp 작용을 수행하는 것이 좋은 방법이지만 변형된 잔여신호와 피치 예측기 기여값 사이의 정합 정밀도를 위해 고정 코드북 기여값을 더하기 전에 부프레임 단위로 warping 작용을 수행한다. 이와 같은 과정을 통해 얻은 잔여 신호는 합성 필터에 통과시켜 수정된 음성 신호를 구하고 가중 필터를 사용하여 수정된 음성신호를 기중한다. 수정된 기중 음성 신호에서 적응 코드북의 기여를 제거하여 고정 코드북 탐색을 위한 목적 신호를 생성한다.

고정 코드북 탐색은 부프레임 단위로 수행되며 ACELP(Algebraic CELP) 코드북을 사용한다. 표 2 는 최대 비트 전송률일 때 최적 여기신호 검출을 위해 EVRC 에서 채택한 고

정 코드북의 구조를 나타낸 것이다. 최대 전송률 일 경우 음성의 여기신호벡터의 요소값으로서 각 부프레임당 8개의 펄스만이 지정된 위치 중에서 +1 또는 -1의 값을 갖게 되며 중간 전송률의 경우 3개의 펄스만 있다. 이러한 대수적(algebraic) 코드북 구조에서는 단 5번의 검색루프를 통해 연쇄적으로 최적 펄스의 위치를 찾게 되므로 계산량의 감소와 함께 우수한 음질을 나타낼 수 있게 한다.

표 2. Ratel 에서 고정코드북의 구조

펄스	부호	위치
T0	± 1	0,5,10,15,20,25,30,35,40,45,50
T1	± 1	1,6,11,16,21,26,31,36,41,46,51
T2	± 1	2,7,12,17,22,27,32,37,42,47,52
T3	± 1	3,8,13,18,23,28,33,38,43,48,53
T4	± 1	4,9,14,19,24,29,34,39,44,49,54

부호화의 마지막 단계에서는 적응 코드북과 고정코드북의 이득값을 양자화하며 양자화된 파라미터들을 패킷화하고 다음 프레임의 목적 신호를 구하기 위해 합성 및 가중필터의 메모리 갱신을 수행한다.

## 2.2 EVRC 음성 복호화기

부호화기에서 양자화된 파라미터들은 패킷 단위로 복호화기로 전송되고 이 패킷을 이용해 복호화 과정이 수행된다. 그림 3은 EVRC 음성 복호화기의 기본 개념도 이다.

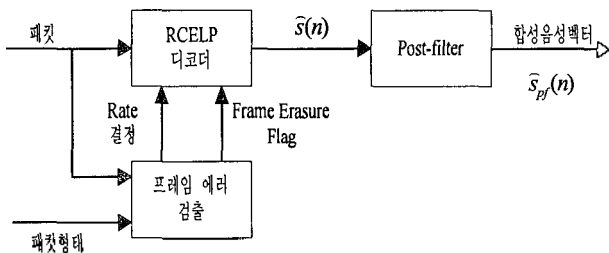


그림 3. EVRC 복호화기의 기본구조

복호화 과정을 살펴보면 다음과 같다. 패킷으로부터 각각의 양자화된 파라미터들을 추출하고 추출된 파라미터들로부터 부호화의 역 과정으로 각 프레임에 이용되는 전송률 및 LP 계수값들이 정해지고 매 부프레임에 적용될 여기신호와 이득값 및 피치 지연값과 이득값이 결정되면 이를 사용하여 각각의 전송률에 따른 합성 음성을 복원한다.

복원된 합성 음성은 보다 나은 음질을 위해 양자화되어 보간된 LP 계수와 보간된 피치지연을 입력으로 하여 적응 postfiltering 과정의 거치게 된다. 적응 postfilter의 계수는 매 부프레임마다 갱신되며 단구간 postfilter, 장구간 postfilter, 단구간 잔여 postfilter, Tilt 보상 filter를 단계적으로 거치게 된다.

## 3. 실시간 구현

EVRC 음성코덱의 실시간 구현에 사용된 보드는 OakDSPCore®를 기반으로 설계된 C&S Technology사의 CSD17C00 칩을 탑재하고 있고, PC와 RISC 및 기타 MCU와의 host-interface가 용이하며, stand-alone 보드로도 사용 가능하다. CSD17C00 칩은 4K words의 데이터 RAM과 9K words의 데이터 ROM이 내장되어 있고  $\mu$ -law/A-law linear converter가 지원되는 2개의 직렬 포트와 소프트웨어 볼륨 조절 등의 기능이 내장되어 음성 압축/재생용으로 아주 적합하다.

DSP 프로그램의 개발은 준비 단계, 코딩 및 검증 단계, 시스템 연결 단계의 세가지 부분으로 구별되어 수행되었다. 우선 준비 단계에서는 알고리즘을 분석하고 C 소스를 검토하여 인코더 및 디코더에 필요한 모듈을 파악하고 call tree를 작성하였다. 효율적인 구현을 위해서는 알고리즘의 분석이 반드시 선행되어야 한다. 계산량을 줄이기 위해서는 dual operation이 필수적인데 이 부분에서 OakDSPCore®는 두개의 operand가 각각 X-memory와 Y-memory에 있어야 하는 제약이 있었다. 따라서 모든 메모리를 조사하여 알고리즘상의 메모리 특성에 맞게 전체적인 메모리의 위치를 배치하였다. 준비 단계에서 가장 중요한 것은 프로그램의 형식, 각 모듈의 depth에 따른 메모리의 위치, 지역 변수와 전역 변수의 지정 그리고 함수 호출 시 인자의 전달방법을 미리 정의하는 것이다.

EVRC의 코딩은 LPC 분석 및 양자화, 피치 탐색, 원신호의 warping, 고정 코드북 탐색, 이득값 양자화, 의 5가지로 구분 지을 수 있다. 각 모듈에 대하여 최적화가 이루어 졌는데, 특히 많은 복잡도를 요구하는 개회로 피치 탐색, 고정 코드북 탐색 등에 많은 비중을 두었다.

본 논문에서는 프로그램의 전 과정이 어셈블리어[5]로 이루어 졌으며 프로그램된 각각의 모듈은 TIA에서 제공된 test vectors를 이용한 검증을 통해 확정되었다. EVRC는 메모리 hangover와 update로 인해 한 개의 프레임에서 문제가 발생하면 그 후의 모든 프레임이 틀려 버리는 특성이 있으므로 특정 모듈에 대한 확실한 검증이 없이는 모듈 간 연결이 불가능하기 때문이다. 또한 최종적으로 모듈을 연결 시 메모리의 정확한 위치와 그것이 내포하는 알고리즘상의 의미, 그리고 전반적인 프로그램을 정확히 이해해야 한다. TIA에서 제공하는 test vectors는 총 19개로서, 인코더와 디코더에 공통으로 사용하는 15개의 벡터와 RDA(Rate Decision Algorithm)를 위한 4개의 벡터가 있다. 구현된 EVRC 어셈블리어 프로그램은 이러한 test vectors와 bit-exact하게 일치 하였다.

마지막으로 시스템 연결 단계에서는 시뮬레이션상에서 검증된 프로그램을 실시간 동작할 수 있도록 하드웨어와 소프트웨어

웨어를 연결하는 작업이 수행된다. 이때 항상 우선 고려되는 부분이 음성의 입출력과 관련된 A/D 및 D/A 로부터의 데이터 처리인데, 본 논문에서는 직렬 포트를 통한 더블 버퍼링을 사용하였다. 즉 현재 A 버퍼에 데이터가 차 있는 상태이고 B 버퍼는 비어있다고 가정하자. 이런 상황에서 인터럽트를 이용하는 경우, DSP의 CPU는 A 버퍼의 데이터에 대하여 음성 부호화작업을 수행하는 동시에 직렬포트로 들어오는 1/8000 초 마다의 인터럽트에 대하여 하던 일을 잠시 멈추고 직렬포트에 들어온 데이터를 B 버퍼로 이동시킨 후 다시 하던 일을 계속하게 된다. 따라서 B 버퍼가 가득차는 시간(20ms)안에 주어진 40K cycles 동안 인코딩 및 디코딩이 수행되어야 한다.

이러한 과정을 통해 구현된 EVRC 코덱의 각 전송률별 평균 복잡도를 표 3에 나타냈다. 표 4는 EVRC full rate 코덱의 각 모듈별 복잡도를 나타낸 것이다. 앞서 언급 하였듯이 DSP 어셈블리어 코딩시 이들 모듈에 대한 집중적인 최적화 작업을 통해 구현된 EVRC 코덱의 성능을 향상시키는데 중점을 두고 구현하였다. 구현된 EVRC 코덱의 메모리 사용량은 표 5에 나타내었다.

표 3. EVRC 코덱의 각 전송률별 MIPS

전송률	Func.	MIPS
Rate 1	Encoder	22.5
	Decoder	3.35
	Total	25.85
Rate 1/2	Encoder	16.7
	Decoder	3.29
	Total	19.99
Rate 1/8	Encoder	8.5
	Decoder	3.19
	Total	11.69

표 4. 구현된 EVRC full rate 코덱의 복잡도

인코더 (22.5 MIPS : 87.04%)				디코더 (3.35MIPS : 12.96%)	
LSP 및 피치 탐색 (19.3%)	고정 코드북 탐색 (11.9%)	적용 코드북 탐색 (28.1%)	기타 (27.8%)	Post- filter (6.1%)	기타 (6.8%)

표 5. EVRC의 메모리(Kwords)

Kwords	ROM		RAM	
	Program	Table	Static	Scratch
	10.8	6.72	1.7	1.23

실시간 구현 결과는 C 시뮬레이션의 결과와 비교하여 음질 저하가 없어야 하고, 실시간으로 음성 부호화기가 동작됨과 동시에 최소의 메모리를 사용하여야 한다. 실시간으로 음성 부호화기가 동작하기 위해서는 복잡도가 DSP 칩의 처리용량을 초과하지 않아야 하며 메모리는 가능한 on-chip 메모리만으로 동작하는 것이 좋다. OakDSPCore®의 최대 처리용량이 40MIPS 이

고 내부 메모리가 4kwords 임을 고려할 때 표 3 과 표 5에 나타난 복잡도와 메모리의 사용량에 대한 EVRC 코덱의 구현결과에 위의 조건을 모두 만족하는 우수한 성능을 가지고 있는 것을 알 수 있다.

## 결론

본 논문에서는 16 비트 고정 소수점 DSP 인 OakDSPCore®를 기반으로 설계된 C&S Technology사의 CSD17C00 칩을 이용하여 EVRC 음성 부호화기를 실시간 구현하였다. 본 연구를 통해 EVRC 음성 부호화기 및 복호화기가 실시간 동작되도록 구현되어, 직접 마이크로 폰으로 음성 신호를 입력하면 실시간 처리되어 스피커로 출력됨을 확인하였고, TIA에서 제공된 19개 test vectors의 검증을 통하여 C 시뮬레이션의 결과와 bit-exact 함을 확인 하였다. 또한 DSP 어셈블리어 코딩 과정에서 기존의 8kbps 음성부호화기에 비해 비교적 많은 계산량이 소요되는 EVRC 알고리즘에 대한 변경 없이 C 시뮬레이션에서와 같은 음질을 유지하면서, OakDSPCore®의 구조적인 장점과 명령어들의 효율적인 사용을 통하여 최소의 복잡도를 가지도록 DSP 어셈블리어 코딩 과정에 중점을 두었다.

구현된 EVRC 코덱은 최대의 계산량을 요구되는 8.55kbps 일 때 encoder 22.5MIPS, decoder 3.35MIPS으로 총평균 25.8MIPS의 복잡도를 나타낸다. OakDSPCore®의 최대 처리용량이 40MIPS인 점과, ASIC 공정에서 구현 가능한 코어 형태임을 고려할 때 EVRC 이외의 기능을 추가하여 IP phone이나 voice e-mail 같은 음성 압축/재생의 다양한 응용에 사용할 수 있으리라 본다.

## [참고 문헌]

- [1] TIA/EIA/IS-127 Enhanced Variable Rate Codec, Speech Service Option 3 for Wideband Spread Spectrum Digital Systems, Sept, 1996
- [2] W.B. Kleijn, P.Kroon, and D.Nahumi, "The RCELP Speech Coding Algorithm," European Transactions, Vol 5, No. 5, 573-582 Sept/Oct. 1994. pp.
- [3] W.B. Kleijn, R. P. Ramachandran, and P. Kroon, "Interpolation of the pitch-predictor parameters in analysis-by-synthesis speech coders," IEEE Trans Speech and Audio Proc, Vol. 2, No. 1, pp. 42-54, 1994.
- [4] B.S. Atal "Predictive coding of speech signals at low bit rates," IEEE Trans. Comm. Vol. 30, No. 4, pp. 5-6, 1982.
- [5] "SSP1820 OakDSPCore® Architecture Specification," Samsung Electronics Co. Ltd, Revision 3.0, March 12, 1998.