

OakDSPCore®를 이용한 AMR 음성 부호화기의 실시간 구현

*이남일, *손창용, **홍성훈, *이동원, *강상원
(*한양대학교 전자전기제어계측공학과, **C&S Technology)

Real-time implementation of the AMR Speech Coder using OakDSPCore®

Nam-Il Lee, Chang-Yong Son, Seong-Hoon Hong, Dong-Won Lee, Sangwon Kang
Dept. of Control & Instrumentation Eng., Hanyang Univ.
E-mail) swkang@selab.hanyang.ac.kr

요 약

본 논문에서는 AMR 음성 부호화 알고리즘을 분석하고 C프로그램 최적화 과정을 거친후 OakDSPCore®를 기반으로 설계된 C&S Technology사의 CSD17C00A 칩을 이용하여 전과정을 어셈블리어로 실시간 구현 하였다. 구현된 코덱은 최대의 계산량을 요구하는 6.7kbps 모드일때, 인코더부분이 최대 20.6MIPS 이며 디코더부분은 약 2.7MIPS 의 복잡도를 나타낸다. 사용된 프로그램 메모리는 약 21.97kwords, 데이터 RAM 메모리는 약 6.64kwords를 가지며 데이터 ROM 메모리는 약 15.1kwords 이다. 구현된 코덱은 최대 약 23.29MIPS의 복잡도를 가지고 있으므로 40MIPS의 처리용량을 가지는 CSD17C00A 를 이용한 보드상에서 실시간 동작이 가능함을 확인하였다. 구현된 프로그램은 3GPP에서 제공하는 21개의 test 벡터들을 통하여 bit-exact 함을 확인하였다.

1. 서 론

디지털 셀룰러 통신 시스템에서 중요한 하나의 과제는 채널상태의 변화에도 좋은 음질을 제공하는 음성 부호화기를 설계하는 것이다. 그러나 현재 서비스 되고 있는 디지털 셀룰러 통신 시스템은 채널 상태에 상관없이 소스 및 채널코딩에 고정된 비트를 할당하는 단일

코딩 모드를 사용함으로써 가변적인 채널상태에 적합한 부호화 solution 을 제공하지 못한다. 채널상태의 넓은 변화 범위에 대해 고르게 좋은 음질 성능을 얻기 위한 하나의 방법은 통신 네트워크가 통신채널의 상태를 감시하고 그 결과에 따라 소스코딩과 채널코딩간의 비트 할당을 조절하는 기능을 갖는 것이다. 이것은 네트워크가 미리 정의된 다수의 유효한 음성 코더 및 채널 코더 들 중의 한쌍을 채널상태에 맞게 최적으로 선택하는 적응 알고리즘을 통하여 수행된다. 이러한 개념의 부호화 방식을 Adaptive Multi-Rate(AMR) 코딩이라 하며, 이는 네트워크에 의해 조절되는 다중모드 음성 부호화기의 일종이다. 3GPP에서는 12.2kbps에서 4.75kbps까지 8개의 전송모드를 가지고 있는 AMR코덱을 W-CDMA 용 음성 부호화 방식 표준안으로 채택하였다.

본 연구에서 AMR 알고리즘을 분석 및 최적화 하고 C&S Technology 사의 CSD17C00A 칩을 이용하여 전과정을 어셈블리어로 실시간 구현하였다. 본 논문에서 사용한 CSD17C00A 칩은 DSP 그룹사에서 개발된 저전력 소비형 16비트 고정 소수점 DSPcore인 OakDSPCore®를 기반으로 miscellaneous logic, serial port, compander, host interface 및 timer의 5가지 peripheral과 범용 I/O ports로 설계되었고, 최대 40MIPS의 처리 용량을 갖는다.

본 논문의 2장에서는 AMR의 기본 구조와 특성에 대해 설명하고, 3장에서는 실시간 구현에 관해 논하며,

마지막으로 4장에서 결론을 맺도록 하겠다.

2. AMR 음성 코덱

AMR 음성부호화기는 multi-rate 음성코덱, 입력 신호의 음성/비음성 여부를 판단하기 위한 voice activity detector (VAD), 비음성 구간에서 전송률을 낮추기 위한 comfort noise generation (CNG) 시스템을 포함하는 source controlled rate (SCR), 그리고 무선 환경에서 발생하는 전송 에러나 손실 프레임에 극복하기 위한 error concealment 부분 등으로 구성된다. AMR 코덱은 12.2, 10.2, 7.95, 7.40, 6.70, 5.90, 5.15, 4.75 kbps의 8개 전송모드와 1.8kbps에서 동작하는 저속 배경잡음 인코딩 모드를 갖는 다중 모드의 음성 부호화기로서 무선 채널 에러 환경에 따라 매 프레임마다 전송 모드를 변화시킨다.

2.1 AMR 음성 부호화기

AMR 음성 부호화기는 code-excited linear predictive (CELP)구조에 기반을 두고 있다. CELP 구조는 LP 분석과정 외에 적응 코드북과 고정 코드북으로 구성된 여기신호를 추정하는 과정을 포함하고 있다. 전체 부호화 과정의 구조도는 그림 1과 같다.

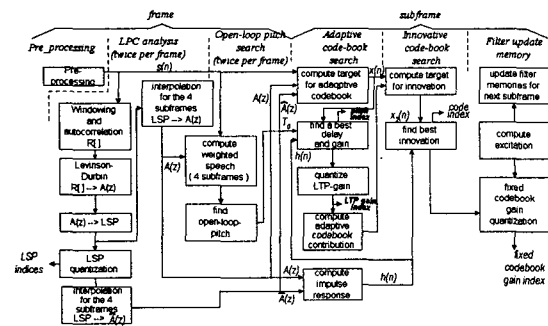


그림 1. AMR 부호화기의 전체 구조도

입력된 신호는 8 kHz로 샘플링되며, 20ms를 한 프레임으로 하고 5ms의 네개 부프레임으로 구성된다. 저주파 성분 제거와 fixed-point 구현시 overflow방지를 위한 전처리과정이 수행된다. LP 분석은 12.2kbps 모드에서는 두개의 비대칭창 함수를 이용하여 2번 수행되며 LP 분석에 필요한 30ms중 10ms는 이전의 프레

임에서 취하고 20ms는 현재 프레임을 이용하므로써 전체 알고리즘 지연은 20ms이다. 나머지 모드에서의 LP 분석은 비대칭창을 이용하여 1번 수행되며 미래 프레임에서 5ms를 취하므로 전체 알고리즘 지연은 25ms이다. 12.2kbps의 경우 LP파라미터의 두 set은 line spectrum pairs(LSP)계수로 변환되고 38비트가 할당되는 split matrix quantization(SMQ) 방식으로 양자화된다. 나머지 모드에서는 변환된 LP파라미터의 set이 split vector quantization(SVQ) 방식으로 벡터 양자화된다.

개루프 피치 분석은 피치분석을 간단하게 하고 페루프 검색을 추정치 부근의 작은 범위 이내로 제한하기 위해 5.15 및 4.75kbps모드에서는 매 프레임마다 한번, 나머지 모드에서는 두번씩 수행된다. 페루프 피치 분석은 피치 지연값과 이득값을 찾기위해 대상 신호 $x(n)$ 과 임펄스 응답 $h(n)$ 을 사용하여 개루프 피치 지연값의 주변을 검색하는 과정이 수행된다. 모드에 따라 미리 결정된 개루프 피치 지연값을 기준으로 1/6 또는 1/3의 정밀도를 갖는 fractional 피치가 이용된다. 즉 12.2kbps모드는 1/6의 피치 정밀도가 사용되고 나머지 모드는 1/3의 피치 정밀도가 이용된다. 대상 신호 $x(n)$ 은 적응코드북 기여분을 제거함으로써 갱신되며 이렇게 얻어진 새로운 대상 신호 $x_s(n)$ 은 고정 대수 코드북 검색에 사용된다. 각 부프레임을 구성하는 40개 펄스의 위치는 모드에 따라 2~5개의 트랙으로 나뉘어지며 각 트랙은 모드에 따라 2~10개의 +1 또는 -1의 펄스를 가지고 있다. 대수 코드북 검색은 가중필터를 거친 음성 신호와 합성필터 및 가중필터를 통과한 신호와의 mean squared error를 최소화 시키는 값을 찾게 된다. 적응 코드북과 고정 코드북의 이득값은 모드에 따라 각각 4비트와 5비트로 스칼라 양자화되거나 6~7비트로 벡터 양자화되며 고정 코드북 이득값은 moving average(MA) 예측기를 사용한다. 여기서 12.2kbps와 7.95kbps모드는 스칼라 양자화하고 나머지 모드는 벡터 양자화를 수행한다. 한 프레임의 분석이 끝나후 다음 프레임에서의 대상신호를 구하기 위해서 합성 필터와 가중 필터의 상태를 갱신해 주어야 한다.

2.2 AMR 음성 복호화기

AMR 음성 복호화기의 전체 구조는 그림 2와 같다.

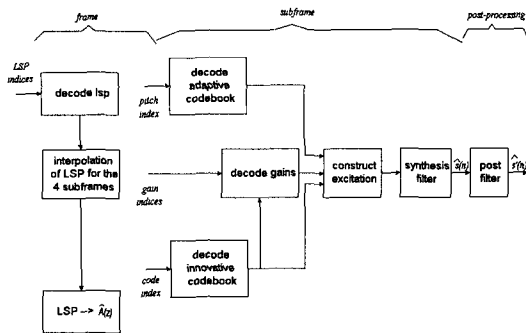


그림 2. AMR 복호화기의 전체 구조도

AMR 복호화기의 원리를 살펴보면, 선택된 모드를 기반으로 부호화기에서 전송받은 비트스트림으로부터 LSP 벡터, 적응 코드북 벡터, 고정코드북 이득, 고정 코드북 벡터, 고정 코드북 이득 등의 인덱스가 추출되며 추출된 인덱스들은 복호화되어 각 전송 프레임에서의 부호화 계수를 결정한다. LSP 벡터는 LP 필터 계수로 변환되어지고 프레임내 각 부프레임에 대한 LP 필터 계수들을 얻기 위한 보간 처리를 한다. 여기신호는 적응 코드벡터와 고정 코드 벡터에 각각의 이득값을 곱한후 더하여 얻어지며, 음성은 LP 합성필터를 통해 여기신호를 필터링하여 재합성 된다. 마지막으로, 재합성된 음성신호는 보다 나은 음질을 위해 적응 post-filter로 처리된다.

3. 실시간 구현

AMR 음성코덱의 실시간 구현에 사용된 보드는 OakDSPCore®를 기반으로 설계된 C&S Technology사의 CSD17C00A칩을 탑재하고 있다. CSD17C00A칩은 4Kwords의 데이터 RAM과 9Kwords의 데이터 ROM이 내장되어 있고 μ -law/A-law linear converter가 지원되는 2개의 직렬 포트와 소프트웨어 볼륨 조절 등의 기능이 내장되어 음성 압축/재생용으로 아주 적합하다. DSP프로그램의 개발은 개발환경 구축, C프로그램 최적화, 어셈블리 코딩 및 최적화, bit-exactness 검증의 4 단계로 구별되어 수행되었다. 실시간 구현의 단계는 그림 3과 같다. 개발 환경 구축 단계에서는 AMR 알고리즘을 분석하고 C 소스를 검토하여 call tree를 작성하고 부호화기 및 복호화기에 필요한 메모리를 조사한다. 실제 구현 이전의 알고리즘 분석을 통하여 계

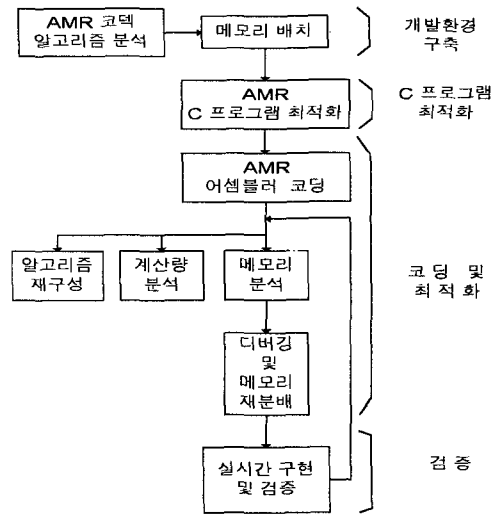


그림 3. 실시간 구현의 단계

산량 감소 방안을 모색하였다. 계산량 감소를 위해서 dual-operation을 사용하였는데 이를 위해서는 연산자가 각각 x-memory와 y-memory에 위치하여야 한다. 이러한 조건을 만족시키고 각 모듈별의 특성에 맞도록 메모리를 최적으로 배치하므로써 구현시 계산량을 감소시킬 수 있었다. 또한 C 프로그램의 불필요한 부분 및 어셈블러의 특성과 맞지 않는 구조는 계산량의 증가를 가져오게 되는데, 이러한 부분을 사전에 제거하여 어셈블러 구현에 적합하도록 C 프로그램을 최적화하였다.

어셈블러 코딩은 크게 LP 분석, 피치 분석, 고정 대수 코드북 탐색, 이득 양자화, 복호화 부분으로 나눌 수 있다. 여기서 계산량의 많은 부분을 차지하는 고정 코드북 탐색 및 피치 분석 부분에 많은 비중을 두었다. 어셈블러 코딩과정에서 각 모듈의 계산량 및 메모리를 분석하고 알고리즘을 재구성하여 최적화를 수행하였다. 어셈블러 코딩후 overflow, 프로그램 작성 과정의 오류 등을 찾아내어 수정하였으며 초기에 설정된 메모리를 전체 코덱에 적합하게 재배치하였다. 구현된 프로그램을 검증하기 위해 ETSI와 3GPP에서 제공하는 21개의 테스트 벡터를 이용하였으며 모두 bit-exact하게 통과하였다. 최종적으로 DSP 보드상에서 음성을 실시간 입력하여 인코딩 및 디코딩의 절차를 거친후 실시간 출력음을 확인함으로써 실시간 구현 여부와 C 프로그램의 출력음과 동등한 음질을 가지는 것을 확인할 수 있었다. 이러한 과정을 통해 구현된 AMR 코덱의 각 전송률별 복잡도를 표 1에 나타냈다. 표 2는 최대복잡도를 갖는

6.7 kbps 모드에서의 각 모듈별 복잡도를 나타낸 것이다. 구현된 AMR 코덱의 메모리 사용량을 표 3에 나타내었다.

표 1. 각 전송률별 복잡도 (MIPS)

MIPS 모드	Max.	Avg.	Min.
12.2 kbps	23.08	18.99	7.16
10.2 kbps	21.66	17.7	6.34
7.95 kbps	22.67	18.22	6.06
7.4 kbps	21.37	17.44	6.06
6.7 kbps	23.29	18.76	6.05
5.9 kbps	18.33	15.24	6.05
5.15 kbps	15.48	13.17	5.93
4.75 kbps	19.34	15.95	5.93

표 2. 6.7kbps 모드의 모듈별 복잡도 분포

인코더 (20.6 MIPS :88.4%)				디코더 (2.7 MIPS:11.6%)	
LSP 및 피치탐색 (21.8%)	고정 코드북 탐색 (39.8%)	작은 코드북 탐색 (13.6%)	기타 (13.2%)	Post- filter (3.1%)	기타 (8.5%)

표 3. 구현된 AMR 코덱의 메모리

	ROM		RAM	
	Program	Table	Static	Scratch
keywords	21.97	15.1	2.89	3.75

위의 실험에서 사용된 데이터는 총 328개의 프레임으로 구성되어 있으며 그 중 DTX 모드에 해당되는 프레임은 92개 이다. 실시간 구현 결과는 C 시뮬레이션의 결과와 비교하여 음질저하가 없어야 하고, 실시간으로 음성 부호화기가 동작됨과 동시에 최소의 메모리를 사용하여야 한다. 실시간으로 음성 부호화기가 동작하기 위해서는 복잡도가 DSP 칩의 처리용량을 초과하지 않아야 한다. OakDSPCore®의 최대 처리용량이 40MIPS 이므로 표 1에 나타난 구현결과는 실시간 동작 조건을 만족하는 우수한 성능을 나타냄을 알 수 있다.

본 논문에서는 W-CDMA용 음성부호화 방식 표준인 AMR 음성 코덱을 고정 소수점 DSP인 OakDSPCore®를 기반으로 설계된 C&S Technology사의 CSD17C00A 칩을 이용하여 실시간 구현하였다.

구현된 어셈블리 프로그램은 최대의 계산량을 갖는 6.7kbps 일때, 인코더부분이 최대 20.6MIPS, 디코더부분이 최대 2.7MIPS으로 최대 23.3MIPS의 복잡도를 나타낸다. ETSI와 3GPP에서 제공된 21개 test 벡터들을 이용한 검증을 통하여 C 시뮬레이션의 결과와 bit-exact함을 확인 하였다. 또한 DSP 보드에 장착된 마이크로를 통해 얻어진 음성 입력이 부호화 및 복호화의 과정을 거쳐 지연 및 왜곡 없이 실시간으로 처리됨을 확인하였다. OakDSPCore®의 최대 처리용량이 40MIPS 이고 ASIC 공정에서 구현이 가능한 코어 형태이므로 IMT-2000용 이동 단말기의 모뎀칩에 활용이 가능할 것으로 생각된다.

[참고 문헌]

- [1] ETSI Draft EN 301 704, Digital cellular telecommunication system (Phase 2+); Adaptive Multi-Rate(AMR) speech transcoding.
- [2] ETSI Draft EN 301 712, Digital cellular telecommunication system (Phase 2+); Adaptive Multi-Rate(AMR) speech; ANSI-C code for the AMR speech codec.
- [3] F.K. Soong and B.H. Juang, "Line spectrum pair (LSP) and speech data compression," in *Proc.ICASSP '84*, pp. 1.10.1-1.10.4.
- [4] M.R. Schoroeder and B.S. Atal, "Code-Excited Linear Prediction (CELP) : High quality speech at very low bit rates," in *Proc.ICASSP '85*, pp.937-940,1985.
- [5] L.R. Rabiner and R.W. Schafer. *Digital processing of speech signals*. Prentice-Hall Int., 1978.
- [6] "SSP1820 OakDSPCore® Architecture Specification," Samsung Electronics Co. Ltd, Revision 3.0, March 12, 1998.

4. 결 론