

# TMS320C542보드를 이용한 Adaptive Multi-Rate 음성부호화기의 실시간 구현

박세익, 전라온, 이인성  
충북대학교

## Real-Time DSP Implementation of Adaptive Multi-Rate with TMS320C542 board

Seik Park, Raon Jeon, Insung Lee  
Chungbuk National University  
seikpark@weppy.com

### 요약문

본 논문에서는 ETSI와 3GPP에서 차세대 이동통신 IMT-2000 서비스의 음성부호화기의 표준안으로 채택한 AMR(Adaptive Multi-Rate)에 대해 알고리즘을 분석하고 Texas Instrument 사에서 제공한 C 컴파일러와 어셈블리 언어를 이용하여 최적화 과정을 수행하였다. 인코더 약 28.2MIPS, 디코더 5.5MIPS로 40MIPS의 사양을 가지는 TMS320C542 보드의 82%를 사용하여 실시간 구현을 하였다. 또한 DSP보드상에서 구현한 결과가 ETSI에서 제공한 ANSI C 소스 프로그램의 결과와 일치됨을 확인하였으며, 마이크 입력과 증폭기를 이용한 스피커 출력의 시스템을 구성하여 지연과 왜곡이 없음을 확인하였다.

### ABSTRACT

3GPP and ETSI adopted AMR(Adaptive Multi-Rate) as a standard for next generation IMT-2000 service. In this paper, we analyzed algorithm about AMR and optimized ANSI C source on the C compiler and assembly language of Texas Instrument. The implemented AMR speech codec requires 28.2MIPS of complexity for encoder and 5.5MIPS for decoder. we performed real-time implementation of AMR speech codec using 82% of TMS320C542 with 40 MIPS specification. We give proof that the output speech of the implemented speech codec on DSP board is identical with result of C source program simulation. Also the reconstructed speech is verified in the real-time environment consisted of microphone and speaker.

### I. 서론

보코더는 12.2kbi/s에서 4.75kbit/s까지 8개의 전송모드로 전송되며, 유/무성음을 판단하는 VAD (Voice Activity Detector) 와 무성음 구간에서 전송률을 더 낮게 조정하는 SCR(Source Controlled Rate) 전송에서 생기는 에러와 lost packet를 조절하기 위한 error concealment mechanism 등으로 이루어져 있다. 본 논문에서는 광대역 이동통신 멀티 미디어 서비스를 위해 IMT-2000 의 보코더로 채택된 ETSI에서 제공한 AMR 알고리즘을 분석하고 고정 소수점 DSP 칩인 TMS320C542 EVM 보드를 이용하여 최적화 작업을 수행하였다.

### II. AMR(Adaptive Muti-Rate) 음성부호화기

ARM은 ACELP에 기반을 두고 있다. 또한 전송률이 여러개를 가짐으로써 MR-ACELP라고 하기도 한다. 기본 목적은 음성부호화의 무선 채널의 환경변화에 적응적으로 대처함에 있다. 그리하면서도 toll quality를 가진다는 장점이 있다. 이에 ETSI에서는 전송률을 4.75kbit/s에서 12.2kbit/s까지 8개로 나누고 하나의 low rate background noise 인코딩 모드를 가지는 하나의 통합된 음성부호화기를 구성했다. 프레임 사이즈는 8kHz로 샘플링된 5ms의 4개의 부 프레임을 가지는 20ms를 가진다.

AMR 음성부호화기는 비음성 구간에서 전송률을 조정하여 효과적인 전송과 전송용량을 증가시키기 위해서 SCR(Source controlled rate)부분과, 각각의 20ms 음성 코더에서 음성과 비음성에 관한 정보를 나타내는 VAD (Voice Activity Detector)부분과 비음성 구간에 삽입하는 comfort noise insertion 부분, 무선 환경에서 발생하는 전송에러나 손실프레임을 극복하기 위한 error concealment 부분을 가지고 있다. 일반적으로 CELP부호화 방식은 크게 LP (Linear Prediction)분석과 adaptive codebook과 fixed codebook으로 구성되어

있다.

LP 분석과 양자화 과정은 12.2kbit/s mode에서는 두 개의 LP 필터를 가지고 30ms 비대칭 윈도우를 이용해서 두 번 실행된다. 이때 lookahead는 사용되지 않고, Levinson-Durbin 알고리즘을 이용해 계수를 구한 후 양자화 한다. 나머지 모드에서는 비대칭 윈도우를 5ms lookahead를 사용한다. 계산된 LP filter 계수는 양자화와 보간을 위하여 LSP(Line Spectrum Pair)로 바뀐다. 12.2kbit/s의 경우 LPC의 2set은 각각 두 개의 1차 MA 예측을 사용해서 두 개의 residual vector를 구하게 되고 이는 SMQ(Split Matrix Quantization)을 사용해서 jointly 양자화 된다. 2x2의 5개의 submatrix로 나뉘게 되고 이는 7,8,8+1,8,6bit로 양자화 된다. 나머지 모드에서는 3,3,4의 subvector로 나뉘게 되고 7~9bit를 사용해서 양자화 한다.

일반적으로 fractional pitch lag에서 이전의 excitation을 보간하여 codevector를 구성한다. 모든 모드에서 closed-loop pitch는 원음성 신호와 합성된 음성 신호사이의 mean-squared weighted error를 최소화시키는 값을 선택하게 된다. 4.75kbit/s, 5.15kbit/s에서는 매프레임당 한번 OLM(open-loop method)를 실행하고 나머지 모드에서는 두 번 시행하고 4.75kbit/s, 5.15kbit/s는 첫 번째 subframe, 나머지 모드에서는 첫 번째와 세 번째 subframe에서 코드화 된다. CLM은 일반적으로 계산범위와 분해능에 차이를 보이는데, 12.2kbit/s에서는 1/6, 나머지 모드에서는 1/3의 분해능을 가지고 구한다. 그리고 subframe 1,3에서와 나머지 프레임에서의 피치 delay를 9bits, 6bit(12.2kbit/s)로 표현한다. algebraic codebook을 찾는데 이용되는 adaptive codevector는 주어진 delay와 fraction에서의 여기신호를 interpolation 해서 구하게 된다.

Algebraic 코드북 검색은 interleaved single-pulse permutation design(ISPP)에 기본을 두고 있다. 가중 필터를 통과한 음성신호화 가중필터와 합성필터를 통과한 음성신호 사이의 mean square error를 최소화 시키는 값을 찾아 수행한다. 결과와 인덱스의 algebraic codevector와의 상관이 최대가 되는 인덱스를 찾는다.

모드에 따라 codebook의 codebook의 차이를 보인다. 하나의 서브프레임을 구성하는 40개의 펄스의 위치는 2~5개의 트랙으로 구분되어진다. 각각의 트랙에 포함되는 펄스는 각 트랙마다 1~2 개씩 2~10개가 선택되고 그 펄스들의 위치와 크기를 부호화한다. 펄스의 크기는 +1~-1의 범위를 가지며 각각 부호 비트이고, 트랙이 전송 되어질 때 위치를 나타내게 된다.

모든 모드에 대하여 adaptive codebook gain을 위해 직접 양자화를 실행하고 algebraic codebook gain을 위하여 fixed 계수를 가진 MA-predictive 양자화를 실행한다. innovation energy와 predicted energy를 구하여 energy 차가 적은 값을 찾아 각 factor를

이용하여 양자화 하게 된다. 이렇게 구해진 두 개의 gains은 양자화 되어지고 12.2kbit/s 모드에서는 gain의 open-loop 양자화가 실행되고, 7.95kbit/s를 제외한 나머지 모드에서는 가중치 필터를 통과한 값에 closed-loop 양자화를 실행한다. 7.95kbit/s에서는 적용되는 closed-loop 양자화를 실행한다.

## 2.1 디코더 구조

LP 파라미터 adaptive codebook vector, adaptive codebook gain, fixed codebook vector, gain등을 수신하고 디코딩하여 합성음을 얻기위해 과정을 수행한다. excitation 신호는 adaptive와 innovative codevectors와 gains에 의해 구해진다. 그리고 음성은 LP 합성 필터를 통과하는 excitation의 필터링 과정을 통해 복원된다.

## III. 시스템 구성

TMS320C542 보드에서는 audio codec library를 제공하기에 이를 이용함으로써 Sound I/O나 PC와의 통신을 쉽게 구현할 수가 있다. 시스템 구성에 있어서는 외부 디바이스를 이용하여 (마이크 입력과 스피커 출력) 실험을 하였다.

## IV AMR 음성부호화기의 최적화

ETSI에서 제공하는 AMR C source을 Tera Instrument사에서 제공하는 C 컴파일러와 Mnemonic assembler 명령어를 사용하여 최적화 작업을 시행하였다.

### 4.1 최적화 구현

① TMS320C542보드는 곱셈연산을 하면 여분의 sign bit를 보상하기 위해 오른쪽으로 이동되어있다. 결과적으로 최적화 작업을 수행하기 위해서는 왼쪽으로 1bit 이동을 매번 해야한다. 상태레지스터1 (Status Register1)에서 6번 bit를 활성화 시킴으로써 다중의 multiplier와 mac(multiple and accumulator), mas(multiple and subtract), 사용시 프로그램을 간략화 시켰다.

② multiplier의 함수를 가질 경우 쉬프트를 사용하여 곱셈 연산을 처리하였다. 대부분의 경우 다량의 계산량을 요하는 부분은 for와 같은 loop 문에서 자주 발생하게 된다. 그러므로 반복문의 함수의 경우 곱셈이나 나눗셈 연산을 쉬프트 시켜 cycle 수를

절약하였다.

③ 앞에서 언급한 것 처럼 보드의 특성상 32bit 처리를 1cycle 에 할수 있기 때문에 막연히 32bits 연산을 처리하는 과정에서 Mpy\_32\_16() 이나 Mpy\_32() 함수처럼 data의 low와 high를 나눠서 연산을 하는 것보다 40 bits의 서로 독립적인 accumulator의 특성을 이용하여 32bit 연산을 처리하여 cycle 수를 절약하였다.

④ 불필요한 반복문 구조나 동일함수를 초기화 하는데 있어 분리시켜 놓은 경우를 많이 볼 수 있다. 이와 같은 우 C에서 최적화 과정을 하지 않고 assem 최적화 과정에서 직접 수정하여 cycle 수를 절약하였다.

⑤ 외부메모리를 사용하기보다 내부메모리를 사용하여 액세스 시간을 확보한다.

#### 4.2 최적화 결과

900MIPS가 넘는 코드를 보드의 최대 사양인 40MIPS에 실행할 수 없기에 인코더와 디코더를 함수별로 분리하여 작업을 시행하였다. stack의 크기는 0x2000, heap 사이즈는 0x1000으로 설정하였다. 그리고 작업중 최적화 작업을 한 파일과 원시소스 벡터와 비교하기 위해 file-processing 작업을 시행하였다. 표 6.은 최적화 작업을 수행한 결과와 도표를 나타내고 있다.

표 1. 최적화 작업결과와 원시소스 의 비교  
TABLE 1. comparison of optimized result and source code

	source code	optimization code
vocoder	AMR coder (Adaptive Muti-Rate)	AMR coder (Adaptive Muti-Rate)
chip	TMS320C542 EVM BOARD	TMS320C542 EVM BOARD
program memory	55.35 kWords	25.75kWord
data RAM +stack	25.42kWords	10.75kWord
data ROM +heap	24.02kWords	18.45kWords
MIPS	992.4MIPS	33.7MIPS

표2. allmodes 에서의 각 블록별 복잡도

TABLE 2. complexity of block in allmodes

인코더 (83.7%)			디코더(16.9%)
피치 검색	fixed code book 검색	adaptive code book 검색	5.5 MIPS
22.5%	41.6%	19.6%	

#### V. 결론

사용칩의 경우 40 MIPS의 사양을 가지며 인코더 28.2 MIPS와 디코더 5.5MIPS로 칩의 82%를 사용하여 실시간 구현을 하였다. 외부 디바이스 (마이크로폰 입력, 스피커 출력) 실시간 시스템을 구성하여 실험을 하였으며, allmodes로 최적화 작업을 하고 각 모드별로 데이터를 확인하였다. 또한 PC에서 시뮬레이션 한 결과와 최적화 작업을 한 결과가 일치함을 확인하였다.

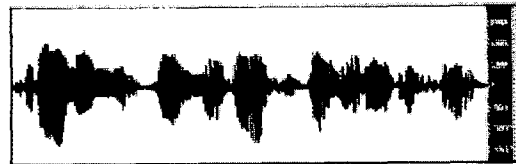


그림 1. 원음파형  
FIG 1. original signal



그림 2. 시뮬레이션 파형  
FIG 2. simulation signal



그림 3. 최적화된 파형  
FIG 3. optimized signal

#### VI 참고 문헌

- [1] F. Itakura, "Line spectral representation of linear predictive coefficients of speech signals," *J. Acoust. Soc. Amer.*, vol. 57, Supplement no. 1, S35, 1975.
- [2] 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.
- [3] K.K Paliwal and B.S. Atal, "Efficient vector quantization of LPC parameters at 24 bits/frame", *IEEE Trans. Speech and Audio Processing*, vol. 1, no 1, pp. 3-14, 1993.
- [4] P. Kabal and R.P. Ramachandran, "The computation of line spectral frequencies using Chebyshev polynomials", *IEEE Trans. on ASSP*, vol. 34, no. 6, pp. 1419-1426, Dec. 1986.
- [5] "TMS320C54X User's Guide", Texas Instruments Oct .1994