

Adaptive Multi-Rate(AMR) 음성부호화 알고리즘

서 정욱, 배 건성

경북대학교 전자·전기공학부

Tel: +82-53-950-5527, Fax: +82-53-950-5505

Adaptive Multi-Rate(AMR) Speech Coding Algorithm

Jeong Wook Seo, Keun Sung Bae

School of the Electronic&Electrical Engineering, Kyungpook National University

jwseo@palgong.knu.ac.kr, ksbae@ee.knu.ac.kr

Abstract

An AMR(Adaptive Multi-Rate) speech coding algorithm has been adopted as a standard speech codec for IMT-2000. It is based on the algebraic CELP, and consists of eight speech coding modes having the bit rate from 4.75 kbit/s to 12.2 kbit/s. It also contains the VAD(Voice Activity Detector), SCR (Source Controlled Rate) operation, and error concealment scheme for robustness in a radio channel. The bit rate of AMR is changed on a frame basis depending on the channel condition.

In this paper, we introduced AMR speech coding algorithm and performed the real-time implementation using TMS320C6201, i.e., a Texas Instrument's fixed-point DSP. With the ANSI C source code released from ETSI and 3GPP, we convert and optimize the program to make it run in real time using the C compiler and assembly language. It is verified that the decoded result of the implemented speech codec on the DSP is identical with the PC simulation result using ANSI C code for test sequences. Also, actual sound input/output test using microphone and speaker demonstrates its proper real-time operation without distortions or delays.

I. 서론

현재의 유선통신망이 PSTN과 N-ISDN을 거쳐 B-ISDN으로 진화함에 따라 무선통신 분야도 디지털 셀룰러와 PCS를 거쳐 3세대 IMT-2000으로 발전하고 있다. IMT-2000에서는 기존의 디지털 셀룰러 및 PCS

에서 제공하는 음성서비스의 품질을 유선전화급으로 향상시키는 동시에 사용자의 선택적 가입을 통해 영상 정보를 주요한 요소로 하는 무선 멀티미디어 서비스의 제공을 목표로 하고 있다. 유럽의 ETSI 및 3GPP에서는 IMT-2000 통신서비스를 위한 표준화 작업을 주도하고 있는데, 음성부호화의 경우 무선채널의 환경변화에도 toll quality의 음성품질을 유지할 수 있도록 8개의 전송 모드를 갖는 AMR(Adaptive Multi-Rate) 음성부호화 방식을 표준안으로 선정하여 그에 관련된 작업을 1999년 12월에 완료하였다[1]. AMR 음성부호화기는 8개의 전송모드를 갖는 multi-rate 음성부호화기[1], 입력 신호의 음성/비음성 여부를 판단하는 VAD(Voice Activity Detector)[2], 비음성 구간에서 전송률을 낮추기 위해 comfort noise generation 시스템을 포함하는 SCR(Source Controlled Rate)[3] 부분, 그리고 전송 error나 lost packet의 영향을 극복하기 위한 error concealment mechanism[4] 등으로 구성된다. Multi-rate 음성부호화기는 4.75 kbits/s에서 12.2 kbits/s사이의 8가지 source rate와 low rate background noise 인코딩 모드(1.8 kbits/s)를 가진 하나의 통합된 음성부호화기로서, 한 프레임마다 무선채널 환경에 따라 codec 모드를 변환함으로써 bit-rate를 바꿀 수 있다. Codec 모드에 따른 bit-rate는 표 1과 같다.

본 논문에서는 IMT-2000에서 채택된 음성부호화 방식의 핵심기술인 AMR 음성부호화 알고리즘을 분석하여 개괄적으로 소개하고, fixed-point DSP 칩인 TMS320C6201[5]을 장착한 EVM Board에서 실시간 구현하였다. DSP로 구현된 시스템의 결과가 ETSI에서 공개된 ANSI C 소스 프로그램[6]의 수행결과와 일치함을 검증하였으며, 마이크 및 스피커를 장착한 2대의 부호화기를 LAN cable로 연결하여 full-duplex

표 1. AMR 음성부호화기의 codec 모드와 bit-rate
Table 1. Codec mode and bit-rate of AMR codec

Codec 모드	Source codec bit-rate
AMR_12.20	12.20 kbits/s (GSM EFR)
AMR_10.20	10.20 kbits/s
AMR_7.95	7.95 kbits/s
AMR_7.40	7.40 kbits/s (IS-641)
AMR_6.70	6.70 kbits/s (PDC-EFR)
AMR_5.90	5.90 kbits/s
AMR_5.15	5.15 kbits/s
AMR_4.75	4.75 kbits/s
AMR_SID*	1.80 kbits/s *

(*) SID 프레임이 연속적으로 전송된다고 가정

모드로 실시간 동작함을 확인하였다. 본 논문의 구성은 다음과 같다. 먼저 II장에서는 AMR 음성부호화 알고리즘에 대해 간략히 설명하고, 그리고, III장에서는 TMS320C6201 DSP를 이용하여 실시간 구현된 알고리즘의 최적화 과정과 실험결과를 제시하며, 마지막으로 IV장에서 결론을 맺는다.

II. AMR 알고리즘의 개요

2.1 AMR 인코더의 구성 [1]

AMR에서 사용되는 8가지 codec 모드의 음성부호화는 CELP(Code Excited Linear Predictive) 모델에 기본을 두고 있는데, 일반적으로 CELP 모델은 LP (Linear Prediction) 분석 과정 외에 adaptive codebook과 fixed (innovative) codebook으로 구성된 여기신호를 추정하는 과정을 포함하고 있다[7]. AMR 음성부호화기의 인코더 구조를 그림 1에 나타내었다. 8 kHz로 샘플링된 음성

프레임(20 ms/프레임)에 대해서 CELP 모델을 사용하여 세 가지 파라미터, 즉 LP 필터 계수, codebook gain과 인덱스를 추출하게 된다. 이 때, 8개의 codec 모드에 따라 각 파라미터에 할당되는 bit 수가 달라지게 되는데, 한 프레임당 244 bits(12.2 kbits/s 모드)에서 95 bits(4.75 kbits/s 모드) 사이로 양자화되어 전송된다.

AMR의 인코딩 과정을 각 단계별로 개괄적으로 살펴보면 다음과 같다. 먼저, 전처리 과정에서는 불필요한 저주파 성분을 제거하기 위한 고역통과 필터링과 fixed-point로 구현할 때 발생할 수 있는 overflow를 줄이기 위한 down-scaling 과정이 수행된다. LP 분석은 codec 모드에 따라서 수행 횟수가 달라지게 되는데, 12.2 kbits/s 모드에서는 한 프레임마다 두 개의 비대칭 창 함수를 이용하여 두 번 수행된다. 이때 lookahead는 이용되지 않고, Levinson-Durbin 알고리즘을 이용해서 LP 계수를 구한 후, 양자화와 interpolation을 쉽게 할 목적으로 LP 계수를 LSP(Line Spectrum Pair) 영역으로 변환시킨다. 나머지 7개의 모드에서는 LP 분석이 한번만 수행되며, 5 ms의 lookahead가 이용된다. 나머지 과정은 12.2 kbits/s 모드에서와 동일하다. AMR 음성부호화기에서는 LP 분석 과정을 거친 후에 한 프레임을 네 개의 서브프레임(5 ms)으로 나누어 분석하게 되는데, 이를 위해 현재 프레임과 인접한 프레임의 계수들을 interpolation하여 사용한다.

Open-loop pitch 분석은 pitch 분석을 간단하게 하고 closed-loop pitch search를 추정치 부근의 작은 범위 이내로 제한하기 위해서 한 프레임마다 두 번씩 수행된다. 12.2, 7.95, 7.40, 6.70, 5.90 kbits/s 모드에서는 {18, 35}, {36, 71}, {72, 143} 등의 세 가지 범위 내에서 식 (1)와 같이 correlation을 구해 각각의 최대치를 normalize시키고, 최대치에 해당하는 delay 중에서 가장 낮은 범위에 있는 값을 best open-loop delay T_{op} 로

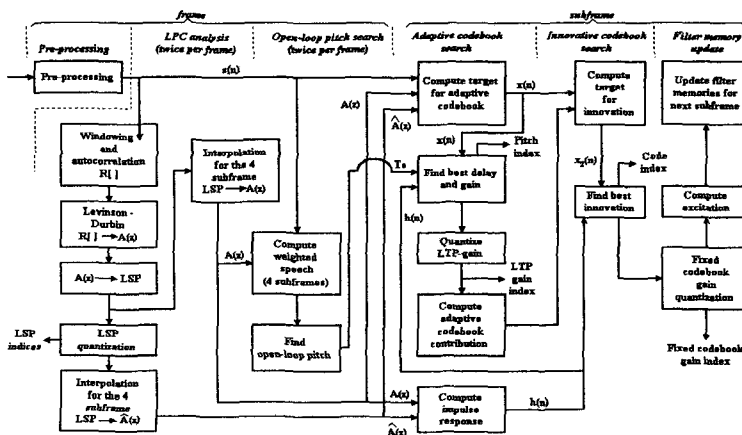


그림 1. AMR 음성부호화기의 인코더 구조
Figure 1. Block diagram of AMR encoder

선택한다. 단, 10.2 kbits/s 모드의 경우에는 correlation을 구할 때 따로 정의된 가중(weighting) 함수를 곱하여 계산한다. Delay 범위를 세 가지로 나누고 낮은 범위에 있는 값을 선택하는 것은 pitch의 multiples를 선택하는 것을 막기 위해서이다. 5.75, 4.9 kbit/s 모드에서는 open-loop 분석을 프레임당 한 번씩 수행하는 것을 제외하고는 위와 똑같은 과정을 수행한다.

$$O_k = \sum_{n=0}^{39} s_w(n) s_w(n-k) \quad (1)$$

여기서, $s_w(n)$ 는 가중 필터를 통과한 음성신호이다.

이런 과정을 거친 후, adaptive codebook과 fixed codebook 추정을 하기 위해서 서브프레임 단위로 가중 합성필터의 impulse response $h(n)$ 을 구한다. Adaptive codebook search는 서브프레임 단위로 수행되고, closed-loop pitch search와 adaptive codevector 계산과정이 수행된다. 모든 codec 모드에서 closed-loop pitch search는 원음성 신호와 합성된 음성 신호사이의 mean-squared weighted error를 최소화시키는 값을 선택한다.

Closed-loop pitch를 추정하는 과정은 각 codec 모드에 따라 달라지는데, 일반적으로 계산범위와 resolution에서 차이를 가진다. 즉, 12.2 kbits/s 모드의 경우 fractional pitch delay를 1/6 resolution 단위로 구하는 반면, 나머지 모드들은 1/3 resolution으로 구한다. 또, 12.2 kbits/s의 경우 pitch 계산범위가 넓은 반면, codec 모드가 낮아질수록 pitch 계산범위는 줄어들게 된다. 이런 과정을 거쳐 optimum integer pitch delay가 결정되면 1/6의 step으로 integer pitch의 (-3/6, +3/6)의 범위에서 fraction 값을 구하여 최종적인 fractional pitch lag값을 결정한다. 그리고, algebraic codebook을 찾는데 이용되는 adaptive codevector $v(n)$ 은 주어진 integer delay k 와 fraction t 에서의 여기신호 $u(n)$ 을 interpolation해서 구한다.

AMR 음성부호화기에서 algebraic codebook 구조는 ISPP(Interleaved Single-Pulse Permutation) design에 기반하고 있으며, 서브프레임 단위로 구성되어 있다. 각 codec 모드에 따라 codebook은 달라지는데, codebook 내의 각 innovation 벡터는 +1 혹은 -1의 크기를 가진 2~8개(codec 모드에 따라)의 non-zero 펄스를 가지고 있으며 하나의 서브프레임을 구성하는 40개의 펄스 위치는 2~5개(codec 모드에 따라)의 트랙으로 나누어진다. 각 트랙에 포함된 펄스는 codec 모드에 따라 2~10개가 선택되고, 그 펄스들의 위치와 크기를 부호화한다. Algebraic codebook search는 가중 필터를 거친 음성신호와 가중 필터와 합성 필터를 거친 음성신호사이의 mean square error를 최소화하여 수행한다. 즉, c_k 를 인덱스 k 에서의 algebraic codevector라고 할 때, closed-loop pitch search에 사용된 target 신호에서 식

(2)와 같이 adaptive codebook 기여도를 빼서 값을 갱신한 $x_2(n)$ 과 c_k 사이의 correlation이 최대가 되는 인덱스를 찾아서 수행하게 된다.

$$x_2(n) = x(n) - \hat{g}_p y(n), n=0, \dots, 39 \quad (2)$$

여기서, $y(n) = v(n) * h(n)$: filtered adaptive codebook
 \hat{g}_p : quantified adaptive codebook gain 이다.

Algebraic codebook gain의 양자화는 fixed coefficient를 가진 MA prediction을 이용해서 수행한다. 4차의 MA prediction을 수행하여 innovation energy와 predicted energy를 구한 후, energy의 차가 가장 작은 값을 찾아 각 codec 모드마다 정의된 factor들을 이용하여 양자화하게 된다.

한 프레임의 분석이 끝난 후, 다음 프레임에서의 target 신호를 구하기 위해서 합성 필터와 가중 필터의 상태를 갱신해 주어야 한다[1]. Adaptive codebook과 algebraic codebook의 gain들을 구하면 현재 서브프레임에서의 excitation 신호를 식 (3)과 같이 구한다. 가중 필터 상태는 입력신호에서 adaptive codebook search에서 구해진 adaptive codebook gain과 codevector의 곱, 그리고 algebraic codebook gain과 codevector의 곱을 각각 뺀 나머지를 계산함으로써 갱신된다.

$$u(n) = \hat{g}_p v(n) + \hat{g}_c c(n), n=0, \dots, 39 \quad (3)$$

여기서, \hat{g}_p : adaptive codebook gain,
 \hat{g}_c : fixed codebook gain,
 $v(n)$: adaptive codebook vector,
 $c(n)$: fixed codebook vector 이다.

2.2 AMR 디코더의 구성 [1]

AMR 음성부호화기의 디코더는 수신된 bits 정보에서 파라미터(LP 파라미터, adaptive codebook vector, adaptive codebook gain, fixed codebook vector, fixed codebook gain)를 디코딩하고, 합성음을 얻기 위해 합성 과정을 수행하게 된다. 이렇게 만들어진 합성음은 음질을 보상하기 위해 post-filtering과 up-scaling 과정을 거친 후 최종적인 합성음이 만들어지게 된다. 그림 2는 AMR 음성부호화기의 디코더 구조를 나타내고 있다.

2.3 SCR 및 VAD의 개요 [3,4]

SCR operation은 AMR 음성부호화기에서 음성의 비활성을 고려하여 가능하면 더 낮은 rate로 음성을 인코딩하는 메카니즘으로서, 사용자 단말기의 전력소모를 줄이고 network에서의 전체적인 load를 줄이는 역할을 한다. SCR의 전체적인 링크구조는 그림 3과 같다. TX(Transmit) 쪽의 SCR handler는 framing unit에

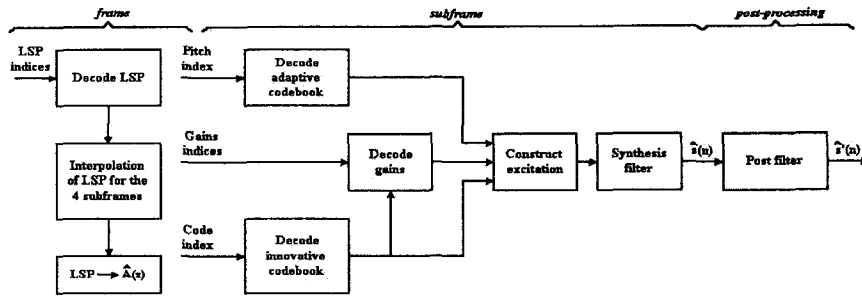


그림 2. AMR 음성부호화기의 디코더 구조
Figure 2. Block diagram of AMR Decoder

TX_TYPE(2 bits, 4개의 모드)에 의해서 표시된 트래픽 프레임이 전송하게 된다. 각각의 프레임은 정보를 포함한 bit field와 codec 모드, 그리고 TX_TYPE으로 구성되며 TX_TYPE에 따라 그 프레임의 contexts(특징 파라미터, comfort noise 혹은 none)가 조절된다. 또한, RX(Receive) 쪽에서는 수신된 트래픽 프레임에서 TX_TYPE을 RX_TYPE(3 bits: 8 개의 모드)으로 변환한 후, 그 값에 따라 AMR 디코더로 contexts를 전달하여 합성신호를 만들거나, error concealment procedure[4]를 수행하기도 하고, 혹은 comfort noise generation을 수행하게 된다.

VAD 알고리즘은 입력된 20 ms 길이의 프레임이 음성, 음악 등 전송해야 하는 신호인지의 여부를 결정하여 Boolean flag로 표시해 주는 기능을 한다[2]. VAD option 1 알고리즘은 그림 4에 나타나듯이 인코더의 파라미터를 이용하여 출력 Boolean VAD flag(VAD_flag)를 만든다. 입력 샘플들은 서브밴드별로 나누어지고, 각 밴드신호에 대한 level(level[n])이 계산되며, 인코더의 open-loop lags($T_{op}[n]$), autocorrelation 값들(t_0, t_1), 그리고 open-loop correlation vector 값을 이용하여 pitch 검출, tone 검출, 그리고 Complex 신호 검출 과정을 거쳐 현재 입력 프레임 신호의 음성 여부를 결정한다. 최종적인 VAD 판단은 구해진 flag 값과 hangover를 비교하여 결정한다[4].

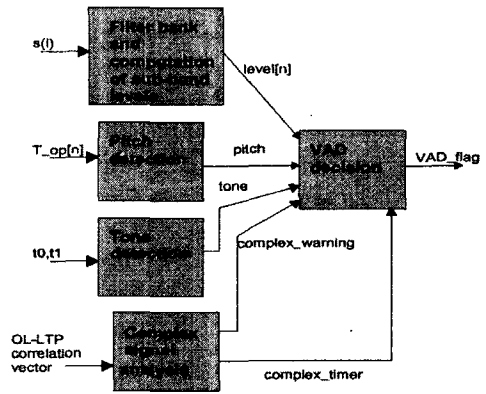


그림 4. VAD 알고리즘(option 1)의 전체 구조도
Figure 4. Block diagram of VAD option 1

III. TMS320C6201을 이용한 실시간 구현

3.1 최적화 과정

본 논문에서 AMR 음성부호화기의 실시간 구현을 위해 사용한 TMS320C6201 DSP는 Texas Instrument사의 TMS320C62xx 고정소수점 DSP 제품군의 하나로써, TMS320C6xx EVM 보드는 CPU, 메모리, 주변 장치 등으로 구성되며, 각각의 구성요소들은 program bus, data bus, peripheral bus 등을 통해 연결된다. 또한 CPU는 200 MHz의 clock rate에서 최대 1600 MIPS (Million Instructions Per Second)의 성능을 발휘할 수가 있다. TMS320C62xx EVM 보드는 내부 메모리로 프로그램 메모리와 데이터 메모리가 각각 64 kbytes씩 구성되며, 외부메모리는 동기식 메모리인 SDRAM 및 비동기식 메모리인 SRAM이 장착되어 있다. 본 논문에서는 마이크로부터 8 kHz로 샘플링된 음성 데이터를 입력받아서 AMR codec을 거친 후, 스피

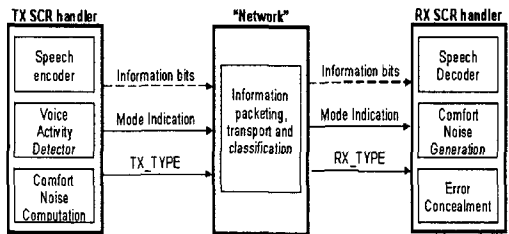


그림 3. One link SCR operation의 구조
Figure 3. Block diagram of one link SCR operation

커를 통해 실시간으로 출력하도록 구현하였다. 이때, 모든 수행동작은 DSP 보드에서 이루어지며, sound I/O와 host I/O를 구현하기 위해 TI에서 제공하는 sound 관련 라이브러리 함수를 사용하였다[9]. AMR 음성 부호화기의 최적화 작업은 ETSI에서 제공하는 ANSI C 소스 프로그램[6]을 기반으로 C compiler와 assembler를 이용하여 수행하였다. ETSI에서 제공하는 소스 프로그램에는 부호화 알고리즘과는 상관없는 count 함수들이나 print 함수들이 포함되어 있는데 이런 것들은 모두 제거한 후 최적화 작업을 수행하였는데, 그 과정을 간략히 요약하면 다음과 같다.

- AMR 음성 부호화 알고리즘을 충분히 분석한 후, 우선적으로 C 프로그램 상에서 가능한 최적화 작업을 먼저 수행하고 어셈블러 변환시 더 나은 최적화가 이루어질 수 있는 것은 어셈블러로 변환하여 작업을 수행한다.
- 약 32개 정도의 기본적인 연산을 수행하는 함수는 'C6x 컴파일러가 제공하는 intrinsic 함수로 변환한다.
- 반복 횟수가 같은 여러 개의 for 루프가 존재할 경우 알고리즘에 위배되지 않는 범위 내에서 적당한 개수만큼 하나로 묶는다.
- 유사한 구조를 가지는 함수의 경우 알고리즘에 위배되지 않는 범위 내에서 임의의 flag 값을 이용하여 하나의 함수로 묶는다.
- 16 bit 연산으로 이루어진 모든 함수들을 32 bit 연산으로 변환시킨다.
- 'C6x에서 제공하는 Software pipeline[10]을 이용하여 어셈블러로 함수를 구현한다.

3.2 실험 및 결과

위와 같은 최적화 과정을 통하여 AMR 음성부호화기를 TMS320C6201 EVM board에 porting하여 음성 신호의 실시간 처리를 구현하였다. 구현된 codec의 성능을 평가하기 위해서 프로그램 메모리와 데이터 메모리의 크기, 그리고 프레임당 수행 시간에 대한 clock을 측정하고 비교하였다. AMR 음성부호화기는 프로그램 크기가 너무 커서 전체를 내부 메모리에 넣을 수 없기 때문에, 호출되는 횟수가 많은 함수와 상수들을 실험적으로 선별하여 내부 메모리에 할당하고, 나머지는 외부 메모리인 SBSRAM에 할당하였다. 구현된 AMR codec의 성능평가는 수치적인 계산을 위해 파일 입출력을 통하여 측정하였다. 파일 입출력은 code composer studio에서 제공하는 파일 입출력 기능[11]을 이용하였으며, 실험에 사용된 입력 음성 파일은 ETSI에서 제공한 test sequence를 사용하였다. DSP로 구현된 시스템의 결과는 ETSI에서 공개된 ANSI C 소스 프로그램의 수행결과와 일치함을 검증하였으며, 표 2는 다음과 같이 정의된 음성부호화기에서 8 개의 codec 모드를

- 음성부호화기 A: ETSI에서 제공하는 original source로 구현된 음성부호화기
- 음성부호화기 B: 음성부호화기 A에 intrinsic을 적용하여 구현한 음성부호화기
- 음성부호화기 C: 음성부호화기 B에 최적화 과정을 적용하여 구현한 최종 음성부호화기

표 2. AMR 음성부호화기의 메모리 크기 및 수행속도 비교
Table 2. Comparison of memory size and processing clock(1 Word = 4 bytes)

음성부호화기		A	B	C
비교				
프로그램 메모리		52.42 kWords	38.48 kWords	31.06 kWords
데이터 RAM (stack 포함)		25.74 kWords	24.72 kWords	9.75 kWords
데이터 ROM (heap 포함)		23.31 kWords	21.12 kWords	19.89 kWords
1 프레임 (20msec) 처리속도 [clock 수]	Average	18,714,071	2,138,737	876,976
	Maximum	21,634,254	3,838,948	1,517,828
	Minimum	18,229,134	1,648,137	435,917

프레임마다 한번씩 차례로 변환시키면서 처리했을때(All modes)의 처리속도를 비교한 것이다. 표 2에서 볼 수 있듯이 최종적으로 최적화된 음성부호화기인 음성부호화기 C의 크기는 프로그램 메모리와 데이터 메모리 면에서 모두 ETSI에서 제공받은 음성부호화기 A에 비해 상당히 줄어든 것을 확인할 수 있다. 음성부호화기 C의 경우에 모든 데이터와 함수 호출을 far로 설정함으로써 데이터 RAM이 많이 줄어들게 되었다. TMS320C6201 DSP 칩은 한 clock이 5 ns이므로 한 프레임인 20 ms를 처리하는데 가용할 수 있는 clock 수는 4,000,000 clocks가 된다. 음성부호화기 A의 경우는 한 프레임을 처리하는데 걸리는 clock 수가 평균 18,714,071이므로 실시간 처리가 불가능하지만, 음성부호화기 C의 경우에는 clock 수가 평균 876,976이기 때문에 DSP 칩의 전체 가용한 clock의 21.94%만 사용해서도 충분히 실시간으로 동작 가능함을 알 수 있다. 즉, 20 ms인 한 프레임을 처리하는데 약 4.38 ms가 소요된다. 또한 이 결과는 DTX 모드[12]로 동작한 결과이므로 입력 음성에 비음성 구간이 얼마나 포함되어 있는가에 따라서 한 프레임을 처리하는데 소요되는 평균 clock 이 제시된 결과와 차이가 생길 수도 있다. 본 논문에서 사용한 test sequence의 경우 음성과 비음성 구간이 약 53%와 47% 정도 포함되어 있다.

표 3은 DTX 모드를 설정한 경우와 설정하지 않은 경우에 대하여 각 모드에 따른 clock 수를 측정하여 비교한 것이다. 6.7 kbits/s 모드인 경우에는 다른 모드에 비해서 상대적으로 많은 clock이 필요하며 5.15 kbits/s는 다른 모드에 비해 적은 clock이 필요함을 알

표 3. DTX 설정 유무와 모드별 프레임당 소요되는 clock 수 [clock]

Table 3. Comparison of processing clock between DTX mode and none

모드	DTX 설정			DTX 비설정		
	Average	Maximum	Minimum	Average	Maximum	Minimum
MR122	936,596.6	1,294,370	479,035	1,241,540.0	1,255,925	1,225,253
MR102	952,742.5	1,291,509	540,813	1,213,505.8	1,245,152	1,203,229
MR795	946,483.6	1,374,153	455,104	1,255,804.2	1,330,889	1,201,278
MR740	905,557.4	1,262,510	454,244	1,199,685.6	1,217,298	1,182,665
MR670	1,040,135.2	1,542,005	453,963	1,426,599.4	1,497,884	1,380,153
MR590	811,741.3	1,120,366	453,644	1,034,479.7	1,075,066	991,346
MR515	673,724.3	894,699	450,232	787,419.0	840,381	744,967
MR475	822,091.0	1,149,453	449,752	1,047,432.4	1,106,378	999,247
All modes	876,976.0	1,517,828	435,917	1,150,231.5	1,482,054	746,125

수 있다. 같은 모드간의 DTX 설정 유무에 따른 최대값의 차이는 음성/비음성 구간을 검출하기 위한 clock 수를 의미하는 것으로 전체 codec에 비해 많은 연산량을 필요로 하지는 않는다. 그러나, 최소값을 비교해 보면 DTX 모드로 동작하는 경우에는 상당히 많은 clock 이 줄어든 것을 알 수 있다. 따라서 DTX 모드를 사용함으로써 codec에 많은 부하를 주지 않으면서도 채널의 효율을 효과적으로 높일 수 있음을 확인하였다.

지금까지 분석한 결과를 토대로 구현된 AMR 음성부호화기를 Sound I/O 모듈과 결합하여 TMS320C6201 EVM board에 장착한 마이크로부터 음성을 입력받아 보드에서 AMR 알고리즘을 통해 인코딩과 디코딩을 거친 후, 다시 스피커를 통해 합성음을 출력해 본 결과, 어떠한 음질의 왜곡이나 지연 없이 실시간으로 충분히 동작함을 확인하였다.

IV. 결 론

본 논문에서는 IMT-2000에서 채택된 음성부호화 방식의 핵심기술인 AMR 음성부호화 알고리즘을 분석하여 개괄적으로 설명하고 TI사의 fixed-point DSP 칩인 TMS320C6201을 장착한 EVM Board에서 실시간 구현하였다. AMR 음성부호화기는 8개의 전송모드를 갖는 multi-rate 음성부호화기와 입력신호의 음성/비음성 여부를 판단하는 VAD, 그리고 비음성 구간에서 전송율을 낮추기 위해 구현된 SCR, 등으로 구성된다. TMS320C6201을 이용하여 실시간 구현된 codec은 한 프레임인 20 ms를 처리하는데 약 4.38 ms가 소요되어 충분히 실시간으로 동작 가능함을 확인하였으며, DSP로 구현된 시스템의 결과는 ETSI에서 공개된 ANSI C 소스 프로그램의 수행결과와 일치함을 검증하였다. 또한, 구현된 AMR 음성부호화기에 sound I/O, host I/O 모듈을 추가한 후, 마이크와 스피커를 장착한 2대의 TMS320C6201 EVM board를 LAN cable을 통해 full-duplex 모드로 연결하였을 때, 어떠한 음질의 왜곡

이나 지연없이 실시간으로 충분히 동작함을 확인하였다.

참 고 문 헌

- [1] ETSI Draft EN 301 704, Digital cellular telecommunication system(Phase 2+); Adaptive Multi-Rate(AMR) speech transcoding
- [2] ETSI Draft EN 301 708, Digital cellular telecommunication system(Phase 2+); Voice Activity Detector(VAD) for Adaptive Multi-Rate(AMR) speech traffic channels
- [3] 3GPP, 3G TS 26.093 Mandatory Speech Codec speech processing functions : AMR Speech Codec; Source Controlled Rate operation
- [4] ETSI Draft EN 301 705, Digital cellular telecommunication system(Phase 2+); Substitution and muting of lost frames for Adaptive Multi-Rate(AMR) speech traffic channels
- [5] Texas Instrument, TMS320C6201 : Fixed-Point Digital Signal Processor
- [6] ETSI Draft EN 301 712, Digital cellular telecommunication system(Phase 2+); Adaptive Multi-Rate(AMR) speech; ANSI-C code for the AMR speech codec
- [7] A.M.Kondoz, *Digital Speech*, John Wiley & Sons, 1994
- [8] Texas Instrument, TMS320C6201/6701 Evaluation Module User's Guide
- [9] Texas Instrument, TMS320C6201/6701 Evaluation Module Technical Reference
- [10] Texas Instrument, TMS320C62xx Programmer's Guide
- [11] Texas Instrument, Code Composer Studio User's Guide
- [12] ETSI Draft EN 301 707, Digital cellular telecommunication system(Phase 2+); Discontinuous Transmission(DTX) for Adaptive Multi-Rate(AMR) speech traffic channels