TMS320C6201을 이용한 2.4 kbps STC 음성 부호화기의 실시간 구현

유 승 형, 이 승 원, 배 건 성
경북대학교 전자 · 전기공학과

Implementation of 2.4 kbps STC Speech Codec on the TMS320C6201

Seung Hyung Yu, Seung Won Lee, Keun Sung Bae
School of Electronics & Electrical Engineering, Kyungpook National University
E-mail : yuwin@mir.knu.ac.kr

Abstract

In this paper, we implement a 2.4 kbps STC speech codec using the TMS320C6201 DSP. The main job for this work is twofold: one is to convert floating-point operation in the codec into fixed-point operation while maintaining the high resolution, and the other is to optimize the program to make it run in real time with memory size as small as possible. The implemented decoder uses 54.8 kbyte of program memory, 29.7 kbyte of data ROM and 55.2 kbyte of data RAM, respectively. It also uses about 45% of maximum computation capacity of TMS320C6201.

I. 서론

초고속통신망, ATM망, 인터넷, 이동통신망 등과 같은 다양한 형태의 디지털 통신망에서 디지털화된 음성을 더욱더 효율적으로 전송하고 저장할 수 있는 음성 압축 알고리즘, 즉 음성 부호화기가 요구된다. 특히, 이동통신과 함께 인터넷폰, 음성메일과 같은 컴퓨터 통신망에서의 음성통신 서비스를 위해서는 양호한 음질을 가지면서 보다 높은 압축률을 얻을 수 있는 저전송률 음성 부호화 알고리즘에 대한 연구가 필요하다.

본 논문에서는 부동 소수점 기반으로 개발된 2.4kbps STC 음성 부호화 알고리즘[1]을 TMS320C6201에서 동작 시키기 위해 고정 소수점으로 변환하는 작업을 수행하였으며, 실시간 처리를 위하여 최적화 작업을 수행하였다. 최적화 기법으로 intrinsic 함수, inline 함수 등이 사용되었으며, STC 알고리즘의 특성을 분석하여 음질에 영향을 미치지 않는 범위에서 계산순서 변경, loop unrolling, loop integration, loop separation 등의 방법이 사용되었다. 실험을 통하여 최적화 전후의 계산량과 사용한 메모리의 크기를 비교/분석 하였다.

본 논문의 구성은 다음과 같다. 먼저 2장에서 2.4 kbps STC 음성 부호화기 알고리즘에 대하여 간략히 설명하고, 3장에서는 TMS320C6201 DSP의 특징과 부동 소수점 연산을 고정 소수점 연산으로 바꾸는 과정, 실시간 동작을 위한 최적화 과정에 대해서 설명한다. 다음으로 4장에서 구현된 고정 소수점 코드의 성능 및 결과를 제시하고 5장에서 결론을 맺는다.

II. 2.4kbps STC 음성 부호화 알고리즘

STC 방식은 정현파 모델을 이용한 음성 부호화 방식으로서 주파수 영역에서 스펙트럼의 모든 피크치들을 정현파로 모델링하여 합성하는 방식을 말한다[1]. 그러나 정현파 모델에 기반한 저전송률 음성 부호화기에서는 전송되는 정보량을 줄이기 위해 스펙트럼의 모든 피크를 사용하는 대신에 스펙트럼 포락선 상에서 기본주파수와 고조파에 해당하는 크기 정보를 이용한

다. 즉, 음성신호가 유성음인 경우에는 기본주파수와 고조파에 해당하는 스펙트럼 포락선 성분들을 정현파로 표현하고, 무성음인 경우에는 주기성분이 일정하지 않으므로 스펙트럼 포락선에서 일정 간격으로 크기를 찾아서 이에 해당하는 정현파를 생성하게 된다[2,3]. 그림 1은 사용된 STC 알고리즘의 encoder와 decoder의 블럭도를 나타낸 것이다.

전송되는 파라미터로는 유성음일 경우에는 유성음 정보, 피치정보, 위상정보(n_o,β), 스펙트럼 포락선 정보(LPC 계수), 이득으로 구성된다. 그리고 무성음일 경우에는 무성음 정보, 스펙트럼 포락선 정보, 이득이 있으며 피치와 위상정보는 아무런 의미가 없는 0을 전송한다. 실제 파라미터는 25 ms 마다 60 bits로 양자화되어 전송되며, 표 1은 [1]에서 사용된 2.4 kbps STC의 비트할당을 나타낸 것이다.

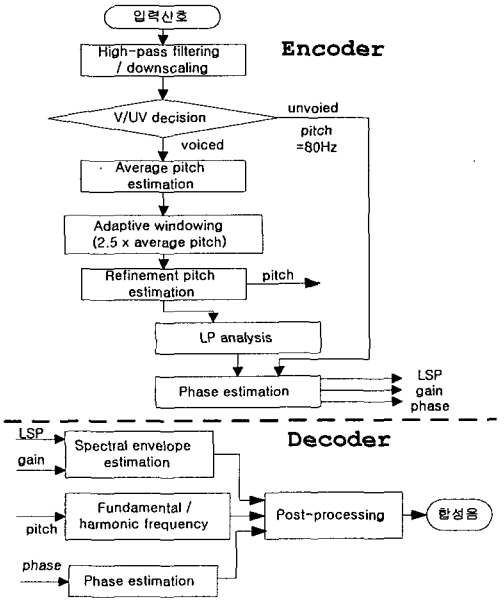

그림1. STC 알고리즘의 encoder/decoder 블럭도

표 1. 2.4kbps STC의 비트 할당

	subfame 1	subframe 2	Total
유/무성음	1	1	2
L S P		20	20
피 치	2	8	10
onset time	8	8	16
β	1	1	2
이 득	5	5	10
Total			60

Ⅲ. 고정 소수점 연산으로의 변환과 최적화 과정

3.1 TMS320C6201 DSP

TMS320C6201 DSP(Digital Signal Processor)는 Texas Instrument사의 TMS320C62xx 고정소수점 DSP 제품군의 하나로서, 기존의 DSP와 달리 VLIW(VelociTI advanced Very-Long Instruction Words) 구조로 된 최초의 DSP이다. 이것은 instruction packing을 통하여 프로그램 크기를 줄일 수 있고, 100% conditional instruction으로 보다 빠른 실행속도를 얻을 수 있다는 특징을 가진다. 그리고 RISC(Reduced Instruction Set Computing)와 유사한 instruction set을 가지기 때문에 사용하기에 용이하며 다채널 시스템의 실시간 구현에 효과적이다. CPU는 200 MHz의 clock rate에서 최대성능 1600 MIPS(Million Instructions Per Second)를 얻을 수 있다[4,5].

3.2 STC 알고리즘의 고정 소수점 변환 과정

고정 소수점으로 변환시 Q15 format을 기본으로 하였으며, Q15 format으로 나타내기 어려운 범위의 data는 Q31 format을 사용하여 좀 더 정확한 계산이 가능하도록 하였다. 고정 소수점을 위한 기본 연산자는 ETSImath operation을 따랐으며 DSP 코드 최적화 작업시 intrinsic 함수를 바로 적용할 수 있다. 각 단계별로 고정 소수점으로 전환하는 과정 중 사용한 방법들은 다음과 같다.

1) 가장 먼저 부동소수점 처리를 시작하는 부분은 음성 신호에 적응 윈도우를 적용하여 주파수 상에서의 스펙트럼 포락선을 찾는 부분이다. 실제 평균피치의 2.5배에 해당하는 길이의 해밍 윈도우를 사용하나, 윈도우의 길이는 140 에서 200 까지의 범위에서 10 샘플씩 증가시키면서 구한 값을 테이블로 만들어 사용하였다. 이렇게 하였을 때 스펙트럼의 분해능은 떨어지나 피치와 위상 추정에는 별다른 영향을 미치지 않음을 확인하였다.
2) 1)에서 구한 적응 윈도우를 취한 음성 신호의 스펙트럼의 포락선을 구하여 피크 정보를 이용하게 된다. 이 때 사용하는 FFT 함수는 TI사에서 제공하는 최적화된 FFT 함수를 사용하여 계산량을 감소시킬 수 있다.
3) STC 알고리즘의 특성상 삼각함수가 많이 사용되는데, 호출시 소요되는 cycle수를 줄이기 위해 sine 및 cosine값을 테이블로 작성하여 사용하였다. 실제 최

적화전 한 프레임 처리 시 sine 함수는 44292 번 호출되며 cosine 함수는 34786 번 호출이 되므로 이런 부분을 최적화 하는 것이 전체 실행시간을 단축하는 데 큰 영향을 주고 있다.

4) 고정 소수점 전환 시 사용되는 Log_e함수, Log_{10} 함수, Root함수 등은 ETSI에서 제공되는 Log_2함수와 *Pow2* 함수 등을 사용하여 오차를 최소화하도록 만들어서 사용하였다.

5) 전송하는 파라미터 중 LSF계수, 위상, 이득은 부동소수점으로 작성된 코드북을 사용하고 있다. 따라서 각각의 범위에 맞는 Q-format 을 사용하여 고정 소수점 기반으로 변환하여 사용하였다.

3.3 실시간 처리를 위한 코드 최적화 과정

최적화 과정의 첫 단계로 Code Generation Tool (ver 4.20) C-compiler 에서 제공하는 최적화 옵션을 최대로 사용하여 file 단위의 최적화작업을 한다[6]. STC의 경우에 무성음은 피치 추정과 위상 추정작업을 하지 않으므로 소스 코드에서 가장 긴 처리시간을 필요로 하는 유성음의 한 프레임을 처리하는 데 걸리는 시간을 표 2에 나타내었다.

표 2.-O3 option 사용 전후 실행시간 비교

	original source	-O3 option
Total(cycle)	176,443,865	64,390,419
Total(ms)	882	321

STC 알고리즘에서는 8 kHz로 샘플링되고 16 bits로 양자화 된 음성신호를 입력으로 하며, 200samples (25ms)길이의 프레임 단위로 처리된다. 목표로 하는 실시간 처리를 위해 먼저 C 레벨에서 최적화 작업을 수행한 후, 사용된 DSP에서 제공하는 컴파일러에서 최적화 과정을 수행하였다. 수행한 최적화 과정을 요약하면 다음과 같다.

1) 전체 알고리즘을 파악하여, 각 함수별로 최적화 방안을 세운다. 주로 피치와 위상을 추정하는 부분에서 많은 계산량을 사용하기 때문에 여기에 최적화 기법을 집중적으로 적용한다.

2) 알고리즘 파악을 바탕으로 계산 순서의 변경, 연산자 변경 등 결과에 영향을 주지 않는 작업을 위주로 최적화 작업을 수행한다.

3) 가장 일반적인 방법인 intrinsic 및 inline 키워드를 사용하여 연산 속도를 향상시킨다.

4) 반복되는 루프 중에서 가능한 경우 loop-unrolling 작업을 수행하되, profile 기능을 사용하여 가장 최적의 단계를 파악한다. 또한 루프의 반복횟수를 알 수 있거나 범위를 알 수 있다면 _nassert()를 사용하여 optimizer에게 최적화에 필요한 정보를 제공할 수 있다[7].

5) 알고리즘을 분석하여 다중 루프 내에서 불필요한 루프의 제거와 통합을 통해 계산량을 감소한다. 특히 원음성의 스펙트럼과 합성 피치를 사용한 스펙트럼의 차이가 최소가 되도록 하는 피치추정 과정의 3중 루프는 상당한 계산량을 소모하기 때문에 내부 루프의 범위 제한과 루프 통합을 통해 많은 계산량 감소를 얻을 수 있었다.

6) 입력값에 상관없이 항상 고정된 값이 입력되는 곳에는 미리 코드북으로 작성하여 파일 입출력 시간을 줄인다.

Ⅳ. 고정 소수점 코드의 성능 및 결과

3장에서 설명한 최적화 작업을 통해 한 프레임의 크기가 25 ms인 STC 알고리즘을 11.3 ms안에 처리하였다. 따라서 TMS320C6201 성능의 약 45%를 사용함으로써 실시간 처리가 가능하였다. 표 3은 이 결과를 intrinsic 함수만을 사용하였을 때와 비교한 것이며, 실제 intrinsic함수만을 사용하였을 때와 비교하였을 때 약 10.6%의 실행시간만을 필요로 한다는 것을 알 수 있었다.

표 4는 가장 많이 사용되는 함수별로 실행에 필요한 시간을 나타낸 것으로서 최적화 전과 비교한 향상도를 보여주고 있다. 코드 최적화는 전체 실행시간에 큰 영향을 주는 함수들에 대하여 수행하였다. 전체 실행시간에서 피치와 위상 추정과정에서 소요되는 시간이 상당히 감소된 것을 확인 할 수 있다.

표 5는 사용된 메모리의 크기를 보여주고 있다. 최적화 과정에서 상수값들에 대해서는 테이블로 작성하였기 때문에 데이터 ROM의 사용량이 증가하였으나 전체적인 실행시간은 단축시킬 수가 있었다.

표 3. 코드 최적화 전후의 실행시간 비교

	Intrinsic 함수 사용시	최적화 후
Total(cycle)	21,141,823	2,249,176
Total(ms)	105	11.3

표 4. 주요 함수별 실행시간 비교 [단위: cycle]

	Intrinsic함수만 사용시(A)	최적화 후 (B)	B/A
coder()	19,471,949	1,284,480	6.6%
Decoder()	8,512,128	1,236,279	14.5%
phase_estimation ()	9,533,578	485,596	14.5%
stc_pitch()	5,539,389	217,559	3.9%

표 5. 메모리 할당

	크기
프로그램 ROM	D620(54.8kbyte)
데이터 RAM	7444(29.7kbyte)
데이터 RAM	D7D4(55.2kbyte)

고정 소수점 기반으로 구현된 알고리즘을 통한 합성음성과 부동 소수점 기반의 합성음성을 샘플 단위로 비교한 결과 차이가 많이 관측되었는데, 이는 고정 소수점 전환시의 해상도와 실제 음성 합성 과정에서 중요한 역할을 하는 정확한 위상정보 복원의 어려움에 기인한다. 그러나 그림 2, 그림 3에 나타나듯이 출력음성을 부동 소수점 기반의 합성음과 비교하였을 때 합성한 파형 및 스펙트럼의 전체적인 포락선은 원 음성을 잘 따라가는 것을 확인하였으며, 주관적인 음질은 거의 동일함을 확인하였다.

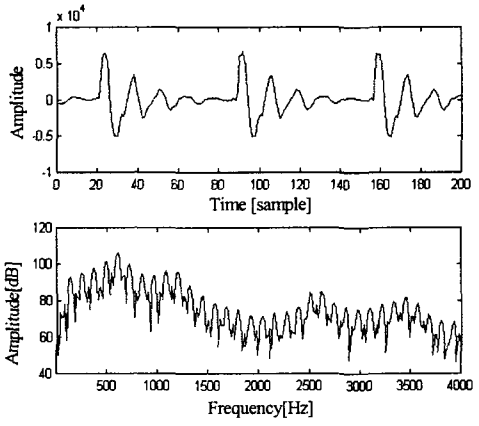

그림 2. 부동 소수점 기반의 합성 파형

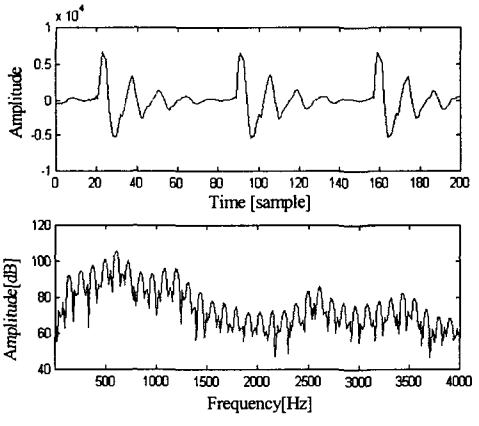

그림 3. 고정 소수점 기반의 합성 파형

V. 결론

본 논문에서는 TMS320C6201을 이용하여 STC 알고리즘을 실시간 구현하였다. 이를 위하여 부동 소수점 기반의 STC 코드를 고정 소수점 기반으로 전환하는 작업을 수행하였으며 주관적인 음질에서 동일한 결과를 얻을 수 있었다. TMS320C6201 DSP에서의 실시간 처리를 위하여 Intrinsic 함수 사용과 코드 최적화 작업을 통하여 실시간 구현을 위한 작업을 수행하였다. 그러나 피치와 위상 추정에서 많은 삼각함수의 사용으로 인한 계산량의 증가가 전체적으로 실시간 처리에 어려움을 주었다. 따라서 이러한 삼각함수의 assembly 작업을 통한 계산량 감소와 알고리즘 분석을 기반으로 위상추정과 합성과정을 수정하여 전체적인 음질 향상을 위한 연구가 계속되어야 할 것이다.

참고문헌

[1] 백성기, 배건성, " 정현파 모델을 이용한 2.4kbs 음성 부호화 알고리즘 ", 제13회 신호처리합동학술대회 , pp. 123-126, 2000.

[2] R. J. McAulay and T. F. Quatieri, " Pitch estimation and voicing detection based on a sinusoidal model ", IEEE proc. Int. Conf. Acoust Speech and Signal Proc. 249-252, 1990.

[3] A. El-Jaroudi and J. Makhoul, " Discrete all pole modeling ", IEEE Trans. on Acoustics, Speech and Signal Proc., Vol. 39(2), pp. 411-423, Feb. 1991.

[4] Texas Instruments Inc., TMS320C6000 Evaluation Module Users Guide

[5] Texas Instruments Inc. TMS320C6201: Fixed Point Digital Signal Processor., literature number SPRS051G, 2000.

[6] Texas Instruments Inc., Code Composer Studio Users Guide

[7] Texas Instruments Inc., TMS320C6000 Optimizing C Compiler Users Guide