

ADSP-2181 DSP 를 이용한 G.723.1 음성부호화기 개발

박정재, 정익주
강원대학교 전자공학과 DSP & Multimedia Lab.

Development of G.723.1 Speech Codec
Using a Fixed-point DSP(ADSP-2181)

J.J. Park, I.J. Chung
DSP & Multimedia Lab. Dept. of Electronics Eng., Kangwon National University
jjchung@cc.kangwon.ac.kr

요 약

본 논문은 고정소수점 DSP 인 Analog Devices 사의 ADSP-2181 을 이용하여 실시간 G.723.1 음성부호화기를 개발한 사례이다. G.723.1 은 ITU 에서 개발한 세계 표준 음성부호화기로 낮은 전송율에서 고음질을 얻을 수 있다. 본 논문에서는 고정소수점 DSP 를 이용하여 부호화기를 개발하는데 필요한 사항들을 제시하였다. 먼저 1 절에서는 음성부호화기의 특성에 대한 개괄을 설명하고, 2 절에서는 G.723.1 부호화기의 특징을, 3 절에서는 고정소수점 DSP 를 이용하여 개발하는 과정을, 4 절에서는 구현결과를 분석하였으며, 마지막으로 5 절에서 결론을 맺는다.

1. 서 론

약 60 년 전 2 차 세계대전 때 보안유지를 위해 군사용으로 Voice Coder 가 처음 사용된 이래로 다양한 부호화기가 개발되었다. 특히 20 년 전부터 디지털 통신이 널리 퍼지기 시작했으며 최근 10 년 사이에 다양한 부호화기가 표준화 되었다. 80 년대에 들어서면서 PC 와 Cellular Phone 의 시대가 열리면서 디지털화를 촉진한 것도 디지털 음성부호화기의 요구를 증가시킨 요인 중 하나이다.

현재 표준화된 부호화기 외에도 다양한 부호화기가 개발되어 사용되고 있다. 이 음성부호화기들은 현재 다양한 분야에서 사용되고 있다. 예로 셀룰러폰, 비디오폰, 음성메시지, 멀티미디어 문서, 인터넷폰, 디지털 음성저장 등 다양하다. 이러한 응용을 위하여 부호화기를 선택하기 위해 부호화기의 특성을 파악하는 것이 중요한데, 부호화기의 특성을 결정짓는 요

인들로 비트율, 지연, 복잡도, 음질 등이 있다. 이 밖에도 전송오류에 대한 강인성, 다중코딩(두번 이상의 코딩을 거치는 경우)성능, signal tone(DTMF 등)의 전송능력, 입력신호 레벨 특성, 다양한 언어와 목소리, 잡음과 음향이 섞인 음성의 코딩능력 등 고려할 사항이 많다. 이들 중 가장 중요한 네가지인 비트율, 지연, 복잡도, 음질을 가지고 부호화기의 특성을 파악한다. 이 요소들 중 어떤 것이 응용분야에 가장 중요한지 결정해야 한다.

먼저 비트율(bit rate)은 응용분야의 채널용량에 의해 가장 먼저 제약을 받는다. 적은 비용을 위해 가능한 한 낮을수록 좋다. 그러나 비트율이 낮은 경우 대체로 특성이 나빠지므로 응용분야에 따라 적절히 선택해야 한다. 낮은 비트율의 부호화기로는 음질보다 음을 알아듣는데 목표를 두는 secur telephony (2~4 Kbps)를 들 수 있다. 중간비트율의 부호화기로는 cellular phone (6.7~13 Kbps)용을 들 수 있다. 이것으로 인해 아날로그통신보다 3~5 배 가량 채널 수를 증가시키게 되었다. 높은 비트율의 부호화기로는 초기 ITU(International Telecommunication Union) 표준 (16~64 Kbps)을 들 수 있다. 비트율은 가변형태로 설계될 수 있는데, 채널이나 신호의 특성에 따라 부호화 알고리즘과 전송율이 바뀌도록 설계될 수 있다. 특히, 일반적인 대화에서는 실제로 40%의 시간에만 음성이 존재한다. VAD(Voice Activity Detection) 알고리즘을 이용하여 무음 구간을 판별하여 부호화할 수 있다. 고정 비트율의 부호화기에 이 알고리즘을 적용하여 가변 비트율 부호화기를 만들 수 있다.

지연(delay)은 실시간 대화에서 중요한 요소이다. 심리학적으로 단방향 지연이 300 ms 를 초과하면 대화가 상당히 부자연스럽게 느껴지는 것으로 알려져

있다. 지연요인은 프레임크기, lookahead(현 프레임 처리에 이후에 올 신호를 미리 이용), 연산처리, 전송, multiplexing 등이 있다. 알고리즘은 프레임 크기와 lookahead 지연을 결정하며, 복잡도(complexity)는 processing 지연에 영향을 주며, 시스템은 multiplexing 및 전송지연에 영향을 준다. 특히 프레임 크기에 의한 지연은 필수적이므로 가장 주목되는 요인이다.

복잡도(Complexity)는 구현에 필요한 시스템 성능을 좌우하며 그만큼 비용과 관련된다. 복잡도는 연산량과 메모리량으로 평가하는데, 연산량은 프로세서에 따라 다르므로 구현에 필요한 연산의 수로 표시하며, 메모리량은 보통 DSP에서 프로그램과 데이터가 모두 고속의 램에 탑재되어 처리되므로 두 가지 모두를 합한 크기로 평가한다.

음질(Quality)은 보통 부호화기의 최고 목표이다. 가능한 한 낮은 비트율과 다양한 환경에서 고유질을 낼 수 있는 부호화기일수록 우수한 성능의 부호화기이다. 고전적으로 음질은 SNR로 평가되었으나 음성 생성모델을 기반으로 한 부호화기에서는 이것이 음질을 대변하지 못하기 때문에 주관적 음질 평가방법인 MOS (Mean Opinion Score)를 사용한다. 이 밖에 음질과 관련된 요인으로 다양한 신호의 부호화 능력(잡음, 신호레벨, 음향, 언어, 목소리)과 채널오류, signaling tone(DTMF, voice band modem, fax signal 등)전송 능력, 다중인코딩(여러 부호화기를 거치는 경우)시의 음질 등이 있다. 이러한 여러 환경에서의 음질평가가 이루어져야 목표하는 응용분야에 가장 합당한 부호화기를 선택할 수 있다. ITU에는 음질의 측정과 성능이 응용분야에 합당하기를 평가하는 전문 그룹(Speech Quality Experts Group)이 있으며, 이 그룹의 활동에 의해 ITU 부호화기 성능이 가장 널리 알려지게 되었다.

구현된 부호화기의 성능이 표준 사양과 부합되는지를 평가하는 것도 중요하다. 따라서 음성부호화기의 표준 스펙에는 이것과 관련된 내용도 포함되어 있다. 이 방법으로 부호화 결과의 bitstream 사양만을 지정하여 원하는 알고리즘과 구현방법으로 구현하도록 지정된 것과, 알고리즘은 물론 각 연산 단계의 연산 정밀도까지도 지정(bit-exact)되어 있는 것이 있다. 전자의 예는 Secure telephony standard 인 FS-1016(DoD CELP: 미국방성 표준)가 있고, 후자의 예는 G.723.1 이 있다. Bit-exact 사양은 구현된 부호화기의 성능이 표준과 동일함을 보증한다. 이 사양의 테스트 방법으로, test vector 를 이용하여 입력 test vector 를 부호화한 결과가 결과 test vector 와 일치하는가를 평가하는 방

법과, 컴퓨터 시뮬레이션 코드를 이용하여 동일한 입력에 대해 구현 부호화기의 결과와 시뮬레이션 프로그램의 결과가 일치하는지를 평가하는 방법이 있다. 물론 테스트 벡터에 대해서도 동일한 결과가 나와야 한다. 전자는 부호화기 성능을 100% 테스트할 수 있는 벡터를 만드는 것이 어렵기 때문에 최근의 부호화기들은 후자를 주로 이용한다. 시뮬레이션 프로그램 언어로는 ANSI C-언어가 일반적으로 사용된다.

이상에서 부호화기의 성능 평가 척도로 주로 활용되는 요인들과 구현된 부호화기를 평가하는 방법에 대하여 살펴보았다. 본 논문에서는 ITU에서 비교적 최근에 표준화한 저비트율의 부호화기 중 하나인 G.723.1을 고정소수점 DSP 인 Analog Devices 사의 ADSP-2181 을 이용하여 구현하였다. 먼저 G.723.1 부호화기의 특징을 분석하고, 다음으로 고정소수점 DSP 로 구현하는 방법을 소개하며, 마지막으로 구현 결과를 검토한다.

2. G.723.1 음성부호화기

2.1 G.723.1 의 특징

G.723.1 부호화기는 1992년 비디오폰의 상용화에 자극을 받아 이에 사용할 음성부호화기의 표준화 요구에 의해 ITU에서 촉박하게 표준화를 진행하여 95년 말에 표준이 완성되었다. 표준화 과정에서 다음과 같은 성능이 요구되었다. Telephone bandwidth modem 을 통해 비디오 신호와 더불어 전송되기 위해 가능한 낮은 전송율(상한 8 Kbps로 표준화 진행)을 갖도록 요구되었고, 음질은 모든 저비트율 부호화기에 대해 ITU가 요구하는 G.726(32 Kbps ADPCM)과 유사한 성능을 요구했다. 지연과 복잡도는 비트율의 중요성 때문에 다소 제약이 적었다.

이상의 목표에 대하여 표준화된 결과는 [표 1]과 같다.

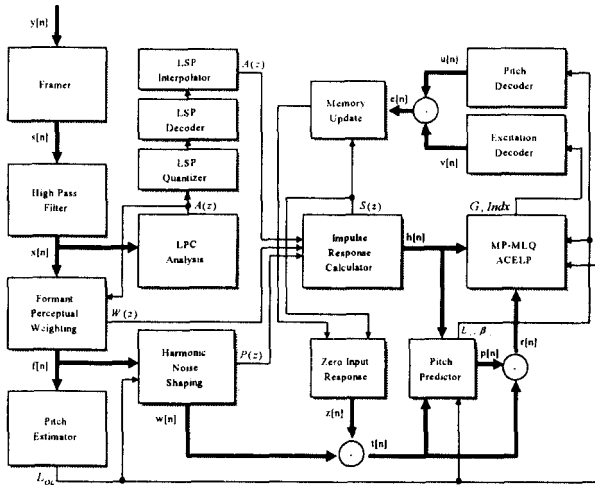
[표 1] G.723.1 특성

비트율	5.3/6.3 Kbps (반드시 둘 모두 지원할 것)
지연	Total: 37.5 ms Frame size: 30 ms Lookahead: 7.5 ms
복잡도	연산량: 16 MIPS 메모리: PM 16 Kword DM 2.2 Kword
음질	G.726 과 유사 (clean signal: 3.98 MOS)

또, G.723.1 은 부록 A, B, C 가 있다. A 는 VAD 와 관련된 것으로, G.723.1 이 이것을 포함한 최초의 ITU 음성부호화기이다. 부록 B 는 부동소수점 C 프로그램으로 PC 등과 같은 호스트 컴퓨터에서 사용할 수 있다. 이것을 이용하여 IBM 에서 인터넷 폰을 개발하기도 하였다. 부록 C 는 scalable channel coder 에 관한 것으로 잡음이 심한 채널에서 사용가능하도록 하기 위해 표준화 되었다. 에러에 민감한 비트에 비중을 두어 보호비트를 할당하는 방식이다. 이것도 부록 A 와 마찬가지로 채널 에러 보호 방식(channel error protection scheme)을 지정한 최초의 ITU 음성부호화기이다.

2.2 G.723.1 부호화 알고리즘

입출력 신호는 8 KHz, 16-bit 로 샘플링된 PCM 신호이다. 입력된 신호는 240 샘플(30 ms)의 프레임 단위로 분석이 이루어진다. 한 프레임의 신호가 입력되면 부호화 과정에 들어간다. 부호화 과정은 [그림 1] 과 같다. 그림에서 굵은 화살표는 신호의 흐름을, 가는 실선은 파라미터의 흐름을 의미한다.



[그림 1] G.723.1 부호화기의 구조

개괄적인 부호화 과정은, 먼저 LPC 분석을 하고 피치분석을 하며, 피치예측과 잔여신호 부호화를 실시한다. 구체적으로 살펴보면, 먼저 입력 프레임을 High-Pass Filtering 하여 직류(DC)성분을 제거하고, 이 신호를 가지고 이후의 분석을 실시한다.

다음으로 LPC 분석을 하는데, 프레임은 60 샘플

(7.5 ms)의 4 개 서브프레임으로 나누어 4 회 분석한다. 분석하고자 하는 서브프레임을 중심으로 이전, 이후 서브프레임을 묶어 180 샘플을 Hamming 윈도우를 취한 후 분석하여 10 자 LP 계수를 얻는다. 마지막 서브프레임은 이후의 한 서브프레임, 즉 다음 프레임의 첫 서브프레임을 필요로하게 되는데 이것을 Look-ahead 라고 부른다. 이것에 의해 지연이 증가하게 된다. 분석된 LP 계수는 다음의 분석에도 쓰이며, 마지막 서브프레임의 계수는 LSP 로 변환, 벡터 양자화되어 채널로 전송된다. LSP 양자화에는 PSVQ (Predictive Split Vector Quantization) 방식이 사용되는데, 256 개의 엔트리를 갖는 세개의 코드북을 이용하여 24 비트로 양자화 된다.

다음 각 서브프레임에 대한 LP 계수를 이용하여 Formant Perceptual Weighting Filter 를 구성하고 신호를 필터링한다.

그리고 피치분석을 하게 되는데 피치 분석은 프레임을 120 샘플(15 ms)씩 두개의 프레임으로 나누어 분석한다. 분석구간은 18~145 이다. 이 값은 채널로 전송되지 않고 다음의 분석을 위해 사용된다.

분석된 피치 정보를 가지고 Harmonic Noise Shaping Filter 를 구성하여 신호를 필터링한다. 이 필터와 위의 Weighting 필터는 Abs(Analysis-by-Synthesis) 방식으로 분석할 때 청각 특성에 맞게 비교할 수 있도록 신호를 변형하는 것이다.

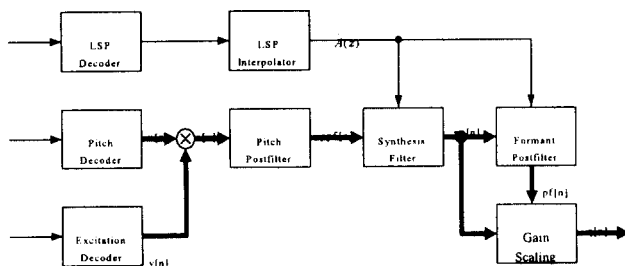
이 필터링된 신호를 가지고 피치예측과 잔여신호의 부호화를 하게 되는데, 이 때 LP 합성필터의 지연 메모리에 이전프레임의 합성에서 남겨진 값이 존재하게 된다. 그러나 분석에서는 이 값을 이용하지 않기 때문에 그 효과만큼 제거해 주어야 한다. 따라서 합성필터의 Zero Input Response 를 구하여 신호에서 감한다.

이 감해진 신호를 가지고 서브프레임 단위로 피치예측을 한다. 피치예측기는 5 차 예측필터이다. 예측은 피치분석 결과를 이용하는데 이전 1 샘플과 이후 2 샘플 사이에서만 예측한다. 짝수프레임은 피치분석결과에 이 결과를 추가하여 7 비트로 부호화하고, 홀수 프레임은 이 예측기의 결과만을 2 비트로 부호화한다. 이득값은 벡터양자화 하는데 85 개(6.3Kbps 부호화기에서 피치지연이 60 보다 작은 경우에만 사용)나 170 개의 엔트리를 갖는 코드북을 이용하여 벡터양자화 된다. 분석된 파라미터를 이용하여 피치예측값을 구하고 분석신호에서 이 값을 감하여 다음 분석에 들어간다. 여기까지는 5.3 Kbps 와 6.3 Kbps 부호화기에 대하여 동일하다.

다음으로 잔여신호의 부호화를 행한다. 5.3 Kbps 에 대하여는 ACELP(Algebraic Code Excited Linear Prediction)방식을 이용하고, 6.3 Kbps 에 대하여는 MP-MLQ(Multi-Pulse Maximum Likelihood Quantization)방식을 이용한다. ACELP 방식은 각 서브프레임을 4 개의 펄스로 근사화 한다. 펄스는 한 프레임에 대하여 동일한 이득값을 가지며, 부호는 개별적으로 갖는다. 또한 펄스는 한 서브프레임 내에서 모두 짝수이거나 모두 홀수라고 가정한다. 이것을 표시하기 위해 서브프레임 당 1 비트를 할당한다. MP-MLQ 방식은 짝수 서브프레임에 대하여 6 개 펄스, 홀수 서브프레임에 대하여 5 개의 펄스로 근사화한다. 펄스는 한 프레임에 대하여 동일한 이득값을 가지며, 부호는 개별적으로 갖는다. 또한 펄스는 한 서브프레임 내에서 모두 짝수이거나 모두 홀수라고 가정한다. 이것을 표시하기 위해 서브프레임 당 1 비트를 할당한다. 이득값은 24 개의 엔트리를 가지고 있는 코드북을 이용하여 벡터 양자화 된다.

마지막으로 다음 분석에서 사용하기 위해 각 필터들의 지연 메모리를 갱신해 주어야 한다. 즉, 원래 신호와 분석을 위해 합성한 신호를 비교하기 위해서는 합성필터의 지연메모리를 고려해야 하기 때문이다. 비교를 위해 이 지연메모리의 효과(Zero Input Response)만큼을 원래신호에서 제거한다. 이렇게 하여 한 프레임의 분석이 끝났다.

복호화기는 간단하며 [그림 2]와 같다. 잔류신호를 복원하고 여기에 예측된 피치성분을 합하며 합성필터를 거쳐 신호가 복원된다. 합성필터는 LSP를 각 서브프레임에 대하여 선형보간하여 LP 계수로 복원하여 구성한다. 분석에서 Formant 와 Harmonics 에 Noise Shaping 한 것에 대응하여 Postfiltering 을 실시한다. Gain Scaling 은 Postfilter 의 이득인자로 필터가 시간에 따라 Adaptive 하게 변하기 때문에 Postfilter 의 입력신호와 같은 레벨의 이득을 만들어 주기 위해 사용한다.



[그림 2] G.723.1 복호화기의 구조

3. 고정소수점 DSP 를 이용한 구현

고정소수점 DSP 를 이용하여 부호화기를 제작하는 것은 상당히 어려운 작업이다. 부호화기를 구현하는 것뿐만 아니라 구현된 부호화기의 성능을 검증하는 것도 어렵기 때문이다. DSP 는 대부분 제한된 개발환경에서 작업을 진행하게 되므로 구현 알고리즘의 검증이 어렵고 시간이 많이 걸린다. 따라서 먼저 알고리즘을 호스트 컴퓨터를 이용하여 부동소수점 연산으로 구현한 후 고정소수점 DSP 의 연산들을 시뮬레이션한 라이브러리를 제작하여 이 연산 라이브러리로 부동소수점으로 구현된 프로그램을 이 라이브러리 연산으로 대체한다. 마지막으로 이 DSP 연산 시뮬레이션 코드를 bit-exact 방법으로 DSP 프로그램으로 포팅한다.

ITU 는 G.723.1 부호화기에 대해 bit-exact fixed-point ANSI-C source 를 제공한다. 이것은 PC 용으로 작성되었으며, 정수연산으로 DSP 연산을 시뮬레이션한 라이브러리를 통해 제작되었다. 따라서 고정소수점 DSP 로 용이하게 구현이 가능하다. 본 논문에서는 알고리즘을 직접 구현하지 않고 ITU 에서 제공하는 고정소수점 연산을 시뮬레이션한 코드를 이용하였다. 따라서 DSP 프로그램으로 변환하는 작업만이 필요했다. 이 변환 작업은 단순한 작업으로 생각되지만 실제의 구현에서는 복잡한 과정이다.

먼저 고정소수점 연산에 대한 지식이 필요하다. 시뮬레이션 연산이 실제 DSP 의 연산과 차이가 있기 때문에 충분한 지식을 가져야 한다.

둘째로 제작에 사용할 프로세서를 선택하고 프로세서에 대한 충분한 지식을 습득해야 한다. 프로세서의 선택 요인은 여러 가지가 있지만, 일단 구현성능을 발휘할 수 있는가를 판단하는 것이 중요하다. 이것을 위해 부호화기의 예상 복잡도(연산량, 메모리 요구량)를 파악해야 한다. 다행히도 G.723.1 부호화기는 ITU 에 의해 그 특성이 많이 알려져 있고, 구현 사례도 소개되어 있다. 본 논문에서 선택한 ADSP-2181 프로세서는 33 MIPS 의 연산속도와 프로그램 메모리 16 Kword, 데이터 메모리 16 Kword 를 가지고 있다. 따라서 충분히 실시간 구현이 가능하다. 프로세서가 선택되었으면, 프로세서에 대하여 가능한 한 충분히 알아야 한다. 가능한 한 작은 연산량과 적은 메모리를 사용하여 구현하기 위해서다.

세번째로 개발도구에 대하여 잘 알아야 한다. 개발도구를 잘 아는 만큼 시간이 절약되고 보다 효율적인 코드를 개발할 수 있기 때문이다.

네번째로 부호화기를 충분히 분석해야 한다. 구현에 앞서 구현을 위한 프로그램의 구조를 설계해야 하는데, 이 구조에 의해 코드에 대한 검증노력과 코드의 모듈화 및 메모리 사용량이 결정되기 때문이다.

이상의 것이 준비 되었으면 실제의 구현에 들어간다. 먼저 세부 알고리즘 모듈을 구현하고 연산 단위로 bit-exact 검증을 실시한다. 즉, 각 연산 단계의 결과가 시뮬레이션 코드와 일치하는지 확인하는 것이다. 다음 이 세부 모듈들을 조합하여 상위 알고리즘 모듈을 구현한다. 마찬가지로 다시 검증을 실시한다. 세부 알고리즘 모듈이 충분히 검증되었다면 이 과정에서는 각 세부 모듈들이 제대로 조합되었는가를 확인하는 것으로 충분하다. 다양한 입출력 벡터를 만들고 검증을 실시한다. 입출력 벡터는 시뮬레이션 코드를 통해 만들 수 있다. 마지막으로 이 중간 모듈들을 이용하여 부호화기의 전체루틴을 구성한다. 여기서의 검증은 입력에 대하여 각 알고리즘 모듈들이 정상적인 결과를 내는지 단계적으로 검증한다. 세부 모듈 검증에서 충분한 검증에는 시간이 많이 걸리겠지만 전체적으로 본다면 오히려 충분히 검증을 할수록 적은 노력과 시간이 든다. 이 경우 전체 부호화기의 검증에서 쉽게 문제점을 찾을 수 있으며 각 알고리즘 모듈 단계에서 정상적인 결과를 내는 것이 확인되었다면 반복 검증 없이 테스트 벡터를 이용하여 검증하면 된다. 이 검증은 비 실시간으로 처리한다.

이것으로 부호화기의 구현과 성능 검증이 완료되었다. 다음의 목표는 최적화를 수행하는 것이다. 최적화 대상은 연산량과 데이터량 두 가지가 있다. 먼저 연산량의 최적화는 실시간화를 목표로 둔다. 만약 목표로 하는 연산 성능이 발휘된다면 프로세서가 결정된 시점에서는 불필요할 수도 있으나 부가적인 기능의 추가를 위해 연산능력의 여분을 남기도록 최적화를 수행해야 한다. Abs 구조의 부호화기는 대부분의 연산이 합성을 통한 비교와 벡터 양자화 하는데 집중된다. 따라서 이 부분을 집중적으로 최적화 한다. 메모리량의 최적화는 부호화기 제작을 위한 코드구조 설계에 가장 많은 영향을 받는다. 처음 설계에서 충분한 고려를 하는 것이 이 노력과 시간을 절약하는 비결이다. 특히 지역변수 사용과 알고리즘 사이의 데이터 전달 방식을 주목하여 설계해야 한다. 그리고 속도 최적화를 위해 어쩔 수 없이 메모리 사용량을 늘려야 하는 경우도 있다. 프로세서를 잘 이해하여 효율적인 연산을 이용하는 것도 메모리 사용량을 줄이는 비결중의 하나이다.

비용의 문제가 구현에서 가장 중요한 핵심이다. 비용은 프로세서 비용 외에 개발기간과 노력, 연산량, 메모리량의 세 가지가 가장 중요한 요인이다. 프로세서는 정수연산기가 저렴하며, 전력소모량도 적으므로 음성부호화기의 구현에 주로 이용된다. 개발기간과 노력을 줄이기 위해서는 좋은 성능의 개발장비를 준비하는 것과 개발 장비에 대한 충분한 숙지 및 위에서 설명한 요인들을 고려하는 것 등이 필요하다. 개발기간과 노력은 개발능력의 척도이다. 또 연산량과 메모리량은 직접비용과 관련된다.

4. 구현결과

부호화기의 구현결과는 구현성능과 비용가치(비용 문제에서 설명된 내용)를 이용하여 평가될 수 있다. 구현 성능은 bit-exact 방법에 의해 검증된다면 이것으로 만족된 것이다. 본 논문에서는 bit-exact 방법으로 구현, 검증하였으며 전송오류에 대한 처리 부분을 제외한 모든 부분이 구현되었다. 그리고 구현 결과는 각 연산단위로 bit-exact 검증되었다. 즉, 표준부호화기 성능과 동일하다.

개발기간과 노력은 평가지수가 없다. 본 부호화기의 구현에 든 기간은 1년 가량이었으며, 준비기간과 개발기간이 반반가량 걸렸다. 특히 개발물의 성능이 미약하여 개발에 시간이 더 걸렸다.

연산량은 프로세서 내부의 타이머를 이용하여 연산에 걸린 시간을 측정하여 구할 수 있다. 하지만 본 논문에서는 ADSP-2181용 시뮬레이터에 연산량 측정 기능이 있어 이것을 이용하였다. 메모리 사용량은 링커의 MAP 파일을 통하여 얻어졌다. 그 결과는 [표 2]와 [표 3]에 각각 나타나 있다. 연산량의 최적화는 연산부하가 높은 피치예측과 잔류신호 부호화 부분에 대해서만 행하여 졌다.

아직 전반적인 최적화가 이루어지지 않은 상태이다. 실시간화를 위한 일부의 최적화 만을 실시하였다. 연산량이 집중된 피치예측 부분과 잔류신호 부분에 대해서만 실시했는데 약 30% 가량 연산량을 감소시켰다. 물론 코드 최적화로 bit-exact에는 영향을 주지 않았다. 연산량이 [표 1]의 특성에 비해 큰 이유는 표에 나타난 수치는 순수하게 연산의 수를 표시한 값이기 때문에 실제 구현과는 거리가 있다. 실제 다른 구현 예를 보면 20~25 MIPS 가량 되는 것으로 알려져 있다. 메모리 사용량은 데이터 메모리 부분에서 [표 1]의 특성에 비해 상당히 많은 양을 사용하였다. 이것은 로컬 메모리를 제사용하지 않고 각 모듈에서

ADSP-2181 DSP를 이용한 G.723.1 음성부호화기 개발

개발 할당하여 사용했기 때문이다. 또한 속도향상을 위해 메모리의 위치변경 용으로 버퍼를 할당하는데 사용한 경우도 있다.

[표 2] DSP 구현 결과의 연산량

DSP 구현		연산량(MIPS)	
		최적화전	최적화후
알고리즘			
LPC 분석		0.91	-
LSP 양자화		0.87	-
피치검출		2.51	-
피치예측		8.21	4.81
간류신호 부호화	MP-MLQ	16.83	8.85
	ACELP	6.76	-
부호화기 전체	MP-MLQ	36.47	24.79
	ACELP	26.40	23.00
복호화기 전체		2.18	2.18
전 체	MP-MLQ	38.65	26.97
	ACELP	28.58	25.18

[표 3] DSP 구현 결과의 메모리 사용량

메모리 종류	메모리 사용량(word)
프로그램 메모리	7208
데이터 메모리	5753
테이블	9160

전반적으로, 목표했던 bit-exact 한 부호화 성능과 실시간화, 그리고 메모리 사용량을 실현했다. 차후 최적화를 실시하여 필요한 응용에 사용 가능할 것으로 생각된다.

5. 결 론

고정소수점 DSP로 음성부호화기를 구현한 처음 경험이어서 구현 전에 많은 준비를 했음에도 불구하고 구현에 어려움이 많았다. 생각보다 많은 작업량과 디버깅 문제, 순수 어셈블리만으로 고정소수점 연산을 행해야 하는 지루함, 비효율적인 개발장비 등으로 특히 어려움을 겪었다. 하지만 적은 인력으로 짧은 기간 동안에 개발하는데 성공한 것은 다행한 일이었다.

G.723.1 부호화기는 비교적 최근에 표준화된 것으로 다양한 알고리즘이 도입되어 있어 음성부호화기에 대한 안목을 넓히는데 좋은 경험이 되었다. 구현 결과 낮은 비트율이면서도 고음질 임을 확인했고, 다

소 긴 준비기간을 지치면서 구현에 필요한 고려 사항들을 정리할 수 있었다. 앞으로 고정소수점 DSP를 이용한 개발에 참조가 될 것으로 생각된다.

6. 참고문헌

- [1] 손중서, 감사현, 장지양, 성원영, "FS-CELP 음성부호화기의 고정소수점 성능 분석 및 구현", 한국통신학회논문지, Vol.21 No.2, pp.365-374, 1996년 2월
- [2] Jun-Hwey Chen & Allen Gersho, "Adaptive Postfiltering for Quality Enhancement of Coded Speech", IEEE Trans. On Speech and Audio Processing Vol.3 No.1, pp.59-71, Jan. 1995.
- [3] Peter Kabal & Ravi & Prakash & Ramachandran, "The Computation of Line Spectral Frequencies using Chebyshev Polynomials", IEEE Trans. On Acoustics, Speech and Signal Processing Vol. ASSP-34 No.6, Dec. 1986.
- [4] S.Cucchi & M. Fratti & M. Ronchi, "On Improving Performance of Analysis by Synthesis Speech Coders", IEEE Trans. On Speech and Audio Processing Vol.4 No.3, May 1996.
- [5] Ravi P. Ramachandran & Peter Kabal, "Pitch Prediction Filters in Speech Coding", IEEE Trans. On Acoustics, Speech and Signal Processing Vol.37 No.4, Apr. 1989.
- [6] W.B. Kleijn, "On the Periodicity of Speech Coded with Linear-Prediction Based Analysis by Synthesis Coders", IEEE Trans. On Speech and Audio Processing Vol.2 No.4, pp.539-542, Oct. 1994.
- [7] ANALOG DEVICES, "Digital Signal Processing Applications Using the ADSP-2100 Family", Vol.1, pp.13-50, ANALOG DEVICES, 1992
- [8] A.M. Kondoz, "Digital Speech Coding for Low Bit Rate Communication Systems", pp.160-174, WILEY, 1994.
- [9] W.B. Kleijn & K.K. Paliwal, "Speech Coding and Synthesis", pp.51-77, ELSEVIER, 1995