

QCELP 보코더의 최적화 및 실시간 구현

Optimization and Real-time Implementation of QCELP Vocoder

변 경 진*, 한 민 수**, 김 경 수*

(Kyung Jin Byun*, Minsoo Hahn**, Kyung Soo Kim*)

요 약

디지털 이동통신 단말기에서 사용되는 보코더는 통화 품질을 높이기 위하여 새로운 알고리즘들을 채택함에 따라 서로 다른 알고리즘을 사용하는 단말기 간의 통신이 문제가 된다. 본 논문에서는 8kbps QCELP 및 13kbps QCELP 보코더 알고리즘을 하나의 DSP 칩에 효율적으로 구현함으로써 이러한 단말기 사이의 통신 문제를 해결하였다. 또한 QCELP 보코더 알고리즘을 실시간 구현하는데 있어 계산량 감소를 위한 단계별 최적화 방법에 대하여 기술하였다. 알고리즘 레벨, 수식 레벨, 코딩 레벨 등의 단계별 최적화 과정을 통하여 계산량이 가장 큰 부분인 코드북 검색 루프에서 약 50% 정도의 계산량을 감소하여 8kbps QCELP는 25 MIPS, 13kbps QCELP는 33 MIPS에 각각 구현하였다. 실시간 구현에 사용한 DSP는 자체 설계한 16 비트 고정소수점 DSP로써 보코더 알고리즘의 구현에 적합하도록 설계되었고, 저전력을 실현하기 위하여 범용의 DSP에 비해 매우 간단한 구조를 가지고 있다.

ABSTRACT

Vocoders used in digital mobile phone adopt new improved algorithm to achieve better communication quality. Therefore the communication problem occurs between mobile phones using different vocoder algorithms. In this paper, the efficient implementation of 8kbps and 13kbps QCELP into one DSP chip to solve this problem is presented. We also describe the optimization method at each level, that is, algorithm-level, equation-level, and coding-level, to reduce the complexity for the QCELP vocoder algorithm implementation. The complexity in the codebook search-loop that is the main part for the QCELP algorithm complexity can be reduced about 50% by using these optimizations. The QCELP implementation with our DSP requires only 25 MIPS of computation for the 8kbps and 33 MIPS for the 13kbps ones. The DSP for our real-time implementation is a 16-bit fixed-point one specifically designed for vocoder applications and has a simple architecture compared to general-purpose ones in order to reduce the power consumption.

1. 서 론

디지털 이동통신 시스템에서는 전송채널의 대역폭을 효율적으로 사용하고, 무선채널 환경에서 고음질의 통화를 위하여 여러가지 보코더 알고리즘들을 사용하고 있다. 현재 국내의 이동통신 단말기에서는 8kbps QCELP, 13kbps QCELP, EVRC의 세 가지 방식의 보코더 알고리즘이 사용되고 있다. 그러나 보코더 알고리즘의 특성상 각기 다른 방식의 알고리즘을 사용하는 단말기 사이의 통신은 불가능하다. 이러한 문제는 기지국에서의 이중 보코딩 과정을 수행하는 방법으로 해결할 수 있다. 이중 보코딩 과정을 수행하기 위해서는 두개의 보코더가 필요하다는 문제점이 있고, 더욱이 이중 보코딩 과정을 거쳐도 음질의 저하

및 지연시간의 증가가 심각하므로 좋은 해결책은 되지 못하고 있다. 그러므로 본 논문에서는 8kbps QCELP 및 13kbps QCELP 알고리즘을 하나의 DSP에 구현함으로써 하나의 단말기에서 두 가지 알고리즘을 모두 사용할 수 있도록 하여 이러한 문제를 해결하였다.

8kbps QCELP 및 13kbps QCELP 알고리즘은 모두 CELP 알고리즘을 기반으로 하는 가변 전송률 음성 압축/복원 알고리즘이다. CELP 알고리즘은 4 - 16kbps의 낮은 비트율에서 고음질을 얻을 수 있는 우수한 알고리즘 중의 하나이다. CELP 알고리즘은 합성에 의한 분석 방법을 사용하므로 낮은 전송률에서도 우수한 음질을 얻을 수 있지만 페루프 구조의 음성을 합성하는 과정에서 많은 계산량이 요구되어 실시간 구현에 어려움이 따른다. 특히 13kbps QCELP 알고리즘은 20ms의 프레임마다 고정 코드북 검색을 하는 서브프레임이 16개나 되므로 엄청난 계산량이 요구된다. 따라서 본 논문에서는 이러한 QCELP 알고리즘의 계산량을 최소화하여 실시간 구동을 실현하고자

* 한국전자통신연구원

** 한국정보통신대학원대학교 공학부

접수일자: 1999년 11월 15일

알고리즘 레벨, 수식 레벨, 어셈블리 코딩 레벨 등 각 단계에서의 최적화 방법을 제안하였다. 한편 8kbps QCELP와 13kbps QCELP 알고리즘은 기본적으로 같은 구조를 가지고 있기 때문에 이러한 최적화 방법 들은 양쪽에 함께 적용할 수 있으며, 어셈블리 프로그램으로 구현 시 공통으로 사용할 수 있는 부분이 많아 프로그램 ROM 및 데이터 ROM을 효율적으로 사용할 수 있어 DSP H/W의 칩 크기는 하나의 보코더를 구현한 경우와 큰 차이가 없으므로 H/W의 사용을 최소화하면서 두 가지 알고리즘을 함께 구현할 수 있었다.

제 II절에서는 8kbps 및 13kbps QCELP 보코더 알고리즘의 기본적인 구조 및 특성에 대하여 살펴보고, 제III절에서 효율적인 실시간 구현을 위한 최적화 방법에 대하여 논하고, 제 IV절에서는 하나의 DSP를 이용하여 8kbps QCELP 및 13kbps QCELP 보코더를 실시간 구현한 결과에 대하여 논하고, 마지막으로 V절에서 결론을 맺는다.

II. QCELP 보코더 알고리즘

QCELP 알고리즘은 13kbps와 8kbps 두 가지가 있으며 두 가지 모두 입력음성의 에너지에 따라 4가지의 가변 전송률을 갖는 보코더 알고리즘이다. 표 1에 각 전송률에 따른 비트 할당을 나타내었다.

표 1. QCELP의 비트할당
Table 1. Bit allocation of QCELP.

8kbps	Rate 1	Rate 1/2	Rate 1/4	Rate 1/8
LPC	40	20	10	10
pitch	40	20	10	0
codebook	80	40	20	6
전송률	8kbps	4kbps	2kbps	1kbps
13kbps	Rate 1	Rate 1/2	Rate 1/4	Rate 1/8
LPC	32	32	32	10
pitch	44	44	0	0
codebook	188	48	20	6
전송률	13.3kbps	6.2kbps	2.7kbps	1.0kbps

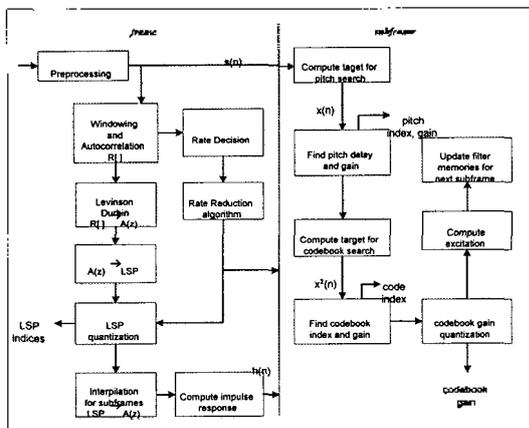


그림 1. QCELP 인코더 블럭도
Fig. 1. QCELP encoder block.

그림 1에 QCELP 인코더 블록을 나타내었다. 13kbps QCELP와 8kbps QCELP보코더는 알고리즘의 기본적인 구조는 같고, LSP 파라미터의 양자화, 전송률 결정논리, 코드북 파라미터의 양자화 방법 등에서 차이가 있다.

2.1. 8kbps QCELP 알고리즘

8kbps QCELP 보코더의 인코더 블록에서는 8kHz로 샘플링된 음성신호 160 샘플(20msec)을 한 프레임으로 하여 각 파라미터를 추출하게 된다. 그림 1의 전처리 과정에서 입력음성의 DC 값을 제거한 후 포먼트 성분을 추출하기 위해 10차의 LPC 계수를 구한다. LPC 계수는 양자화 왜곡을 최소화하기 위하여 10개의 LSP 계수로 변환한 후 각 LSP 계수를 스칼라 양자화기를 사용하여 양자화 한다. 전송률의 결정은 음성의 에너지를 3개의 문턱값과 비교하여 rate 1, 1/2, 1/4, 1/8로 구분한다. 피치 검색에서는 입력음성과 합성 음성을 비교하여 그 오차가 최소가 되는 피치 지연 값과 이득을 찾는다. 피치 검색과정에서 사용되는 피치 합성 필터는 식 1과 같이 표현된다.

$$\frac{1}{F(z)} = \frac{1}{1 - bz^{-L}} \quad (1)$$

여기서 L은 피치 지연을 나타내며 17에서 143까지의 값을 갖는다. 그리고 b는 피치이득을 나타내며 0에서 2사이의 값을 갖는다. 입력음성과 합성음성의 오차는 다음의 식 2와 같으며 합성음성은 매 지연 값에 대해 여기신호를 피치 합성필터와 포먼트 합성필터를 거쳐서 만들어진다.

$$\sum_{n=0}^{L-1} \{x(n) - by_L(n)\}^2 \quad (2)$$

여기서 x(n)은 입력음성을 나타내고, y_L(n)은 L만큼 지연된 과거의 여기신호 u_L(n)가 h(n)과 컨볼루션 과정을 거쳐 만들어진 합성음성으로 식 3과 같다.

$$y_L(n) = \sum_{i=0}^{L-1} h(i)u_L(n-i), 16 < L \leq 143 \quad (3)$$

앞의 식 2에서의 오차 신호값은 $-2bE_{xy} + b^2E_{yy}$ 을 계산함으로써 구할 수 있으며, 여기서 E_{xy} 와 E_{yy} 는 다음의 식 4와 같다.

$$E_{xy} = \sum_{n=0}^{L-1} x(n)y_L(n), \quad (4)$$

$$E_{yy} = \sum_{n=0}^{L-1} y_L(n)y_L(n)$$

코드북 검색과정도 앞의 피치 검색과 마찬가지로 입력음성과 합성음성의 최소오차를 구하는 과정이므로 앞의 수식과 같은 형태의 계산이 이루어진다. 단지 피치 이득 b와 피치지연 L 대신에 코드북 이득 G와 코드북 인덱스

1로만 대치하면 앞의 수식들은 코드북 검색에서의 수식으로 대치된다. 코드북 검색과정에서는 검색과정의 계산량을 줄이고 코드북의 저장 메모리를 줄이기 위하여 코드북의 값 중 약 80%가 0으로 채워진 128개의 순환 코드북을 사용한다. 코드북 검색 시 코드북의 값이 0인 경우는 합성음성을 계산하는 과정이 생략되어 계산량을 줄일 수 있다.

2.2. 13kbps QCELP 알고리즘

13kbps QCELP의 인코딩에서 LSP 계수를 구하는 과정까지는 8kbps의 과정과 같고, LSP를 양자화 할 때 rate1, 1/2, 1/4의 경우는 32비트의 백터 양자화기를, rate 1/8의 경우는 스칼라 양자화기를 사용한다. 전송률 결정블록은 2단으로 구성되어 있으며, 첫째 단에서는 음성의 에너지에 따라 전송률을 rate1, 1/2, 1/4로 구분한다. 그리고 둘째 단에서는 앞 단의 rate가 1, 1/2 이고, RRM(Rate Reduced Mode)인 경우 ZCR(Zero Crossing Rate)나 NACF(Normalized Autocorrelation Function) 등의 7가지 feature를 이용하여 전송률을 rate1, 1/2, 1/4로 다시 분류한다. 피치 검색과 코드북 검색과정은 8kbps의 경우와 거의 유사하지만 피치 검색의 경우 지연값 17-139 사이에서 1/2 샘플 지연값에 대한 검색이 추가적으로 이루어 진다. 그리고 코드북 검색은 rate1, 1/2의 경우에만 수행되며, rate1/4, 1/8의 경우는 검색과정 없이 잔차 신호의 에너지 값만을 계산하게 된다. Rate 1/2일 때 사용하는 코드북은 8kbps의 rate1에서 사용하는 코드북과 같으며, rate1일 때 사용하는 코드북은 0으로 채워진 부분이 거의 없고, 특히 8kbps에 비해 서브프레임 구간이 절반이므로 검색과정이 두 배로 늘어나 엄청난 계산량이 요구된다.

III. 실시간 구현 및 최적화 방법

3.1. 디지털 신호처리 프로세서

QCELP 보코더 알고리즘을 구현하기 위하여 자체 설계된 DSP는 16비트 고정소수점 DSP로써 프로그램 블록, 메모리 블록, ALU 블록, I/O 블록의 4개의 블록으로 구성되어 있으며 다음과 같은 특징을 갖고 있다.

- 25 nsec instruction cycle (40 MIPS)
- MAC operation and 32 bit data load in one cycle
- 2 x 36 bit accumulator
- 16K x 24 bit internally and can use 64 x 24 bit external memory
- Internal 3K x 16 bit data RAM and 4K x 16 bit data ROM
- Single cycle exponent evaluation
- One serial port and one parallel port

보코더 응용분야에 사용하기 적합하게 설계된 DSP는 저전력과 빠른 계산능력을 위하여 RISC 타입의 명령어, 손실 없는 repeat, dual bank memory 구조 등을 채택하였

다. 일반적으로 범용의 DSP들은 다양하고 복잡한 명령어 세트와 다양한 어드레싱 모드 들을 지원하고 있다. 그러나 RISC 타입의 명령어와 간단한 어드레싱 모드, 그리고 콘트롤 로직을 최소화 함으로써 칩의 소형화와 저전력을 실현하였다. 더욱이 24비트 명령어의 채택은 immediate addressing, direct addressing을 한 워드의 명령어로 처리할 수 있으므로 모든 명령어를 한 사이클에 실행하는 것이 가능하다.

3.2. 알고리즘 상의 최적화

QCELP 알고리즘의 피치검색 및 코드북 검색 과정에서는 패루프 구조의 검색을 수행하기 때문에 인코딩의 전체 계산량의 80% 이상을 차지할 정도로 많은 계산량을 필요로 한다. 특히 13kbps QCELP의 경우 검색해야 할 서브 프레임이 16개나 되므로 더욱 심각하다.

그러므로 본 논문에서는 코드북 검색 과정에서 입력음성과 합성음성 사이의 최소 오차를 검색하는 루프의 횟수를 제한함으로써 계산량을 대폭 줄였다. 코드북 검색에서 검색해야 할 코드벡터의 개수가 총 128개이므로 128번의 반복과정을 거쳐야 검색이 종료된다. 그러나 여기서는 최소 오차 값을 찾은 횟수를 카운트하여 6회 이상 최소 값을 찾은 경우에는 더 이상 검색과정을 반복을 하지 않고 검색을 종료하도록 하였다. 표 2에 여성 음성에 대해 13kbps QCELP에서 최소오차 카운터의 값에 따른 검색율과 SNR의 변화를 나타내었다.

표 2. 코드북 검색율에 따른 SNR
Table 2. SNR for percentages of codebook search.

최소오차 카운터	검색율 (%)	SNR (dB)	segSNR (dB)
Not use	100	22.61	12.27
9	97.2	22.54	12.27
8	92.8	22.35	12.22
7	84.4	22.27	12.20
6	68.6	22.14	11.98
5	50.4	21.31	11.60
4	31.2	20.58	11.10
3	14.7	18.90	10.19

표 2의 결과를 보면 최소 오차 카운터의 값을 줄임에 따라 검색에 필요한 계산량은 급격히 줄어드는 것을 알 수 있다. 여기서 최소 오차 카운터의 값을 6으로 하였을 때를 살펴보면 계산량은 전체 검색에 비해 68.6% 수준으로 감소하지만 음질저하는 0.5dB 밖에 안되어 매우 미미한 정도이다.

3.3. 수식 상의 최적화

수식 상의 최적화는 수식을 변화시켜 계산량을 줄이는 것이므로, 즉 원래의 수식과 같으면서 계산량을 줄일 수 있는 형태로 변환한 것이므로 이론적으로는 알고리즘 상의

최적화 과정에서와 같은 음질의 저하는 없다. 그러나 고정 소수점형 DSP에서의 계산 수행이므로 수식의 변화는 전체 계산 결과의 정확도에 영향을 미칠 수 있기 때문에 완전히 같은 결과라고 할 수는 없지만 음질에 미치는 영향은 거의 없다고 할 수 있다.

수식 상의 최적화에서도 계산량이 많은 코드북 검색과정의 검색횟수를 줄일 수 있는 방법을 수식을 통하여 구현하였다. 코드북 검색 시 최소 오차 값은 다음의 식 5와 같이 계산하여 얻을 수 있었다.

$$Min_err = -2GE_{xy} + G^2 E_{yy} \quad (5)$$

위 식의 계산은 E_{xy} 와 E_{yy} 를 계산한 후 양자화 된 4개의 이득 값 G를 대입하고 식 5의 계산을 4번 수행하여 그 중의 최소값을 찾게 된다. 그러나 여기서는 E_{xy} 와 E_{yy} 값을 구한 후 한번의 계산을 통하여 4번의 계산을 계속 수행할 것인지 안 할 것인지를 판별할 수 있도록 하여 계산량을 줄이고자 하였다. 이것을 판별할 수 있는 판별식 9는 다음과 같이 유도 된다.

$$G = \frac{E_{xy}}{E_{yy}} \quad (6)$$

$$Min_err = -\frac{E_{xy}^2}{E_{yy}} \quad (7)$$

$$Min_err_{previous} < -\frac{E_{xy}^2}{E_{yy}} \quad (8)$$

$$E_{xy}^2 + Min_err_{previous} \times E_{yy} < 0 \quad (9)$$

식 5를 최소화 하는 G는 식 6과 같이 식 5를 G에 대해 미분하여 구해진다. 식 6을 식 5에 대입하면 min_err가 식 7과 같이 구해지며 결과적으로 현재 루프에서의 최소값이 이전 루프에서 구해진 최소값 보다 적을 경우에만 G 값을 대입하여 검색을 수행한다. 다시 말하면 식 9의 판별식이 만족되는 경우에만 4개의 G 값을 대입하여 식 5를 계산하므로 전체 검색의 약 64%의 계산량으로 검색이 가능하다. 만일 알고리즘 상의 최적화에서 최소오차 카운터를 6으로 하고 수식 상의 최적화 과정을 함께 적용하면 전체 검색에 비해 약 48%의 계산량으로 코드북 검색을 수행할 수 있다.

3.4. 어셈블리 코딩 상의 최적화

어셈블리 코딩 상의 최적화에서는 피치검색이나 코드북 검색에서 최소 에러를 구하기 위해 음성을 합성하는 과정의 콘볼루션 계산을 구현하는데 있어 DSP의 파이프라인 구조 상의 특징을 이용하여 코드를 최적화함으로써 계산량을 줄일 수 있었다.

합성음성을 계산할 때 검색구간 중 첫번째 지연 값에 대해서만 콘볼루션 계산을 수행하고 나머지 지연 값들에 대해서는 식 10과 같이 재귀적인 계산에 의해 콘볼루션 과정의 계산을 간단히 할 수 있다.

$$y_L(n) = y_{L-1}(n-1) + u(-L)h(n) \quad (10)$$

이러한 재귀적인 특성에 의해 콘볼루션 과정의 계산량을 대폭 줄일 수는 있지만 이러한 특성을 이용하여 구현하여도 DSP에서 제공되는 일반적인 명령어를 가지고 구현하는 데는 여전히 많은 계산량이 요구된다. 위의 식 10에 의해 일반적인 DSP 명령어를 가지고 콘볼루션을 구현하는 데는 다음과 같은 동작이 필요하게 된다.

```

RY2 = *(AY0+);
RX2 = *(AX0);
REPEAT n |
    R0 = R0 + RX2 * RY2;
    RY2 = *(AY0+);
    *(AX0+) = RX0;
    RY0 = 0;
    RX0 = *(AX0);
    |
    
```

위와 같은 과정을 반복하게 되면 재귀적인 콘볼루션을 구현할 수 있게 된다. 위의 구현 예에서는 REPEAT 블록 내에 5개의 명령어를 사용하여 구현하였으나 실제로 최근의 DSP들은 대부분 병렬처리 동작을 지원하므로 위의 과정을 좀 더 줄일 수는 있다. 하지만 그렇다 하더라도 DSP에서 특수한 명령어를 제공하지 않는다면 최소한 3개의 명령어를 사용하여야 가능하다. 본 논문에서는 이러한 콘볼루션 과정을 구현하는데 있어서 특별한 명령어의 추가 없이 자체 개발한 DSP의 파이프라인 특성을 이용하여 다음과 같이 MAC 명령어 2개 만으로 구성하여 계산량을 감소 시켰다.

```

R1 = R1 R1 || RX2 = *(AX0), RY2 = *(AY0+);
RX0 = *(AX1);
REPEAT n |
    R0 = R1 + RX2 * RY2 || *(AX1+) = RX0;
    R0 = R0 + RX2 * RY1 || RX1 = *(AX0+), RY2
    = *(AY0+);
    |
    
```

이러한 콘볼루션의 구현에서 반복구간 내의 한 사이클의 증가는 보코더 전체 계산량에서 약 1 MIPS 이상의 증가를 가져올 정도로 심각하다.

IV. 결과 및 검토

8kbps QCELP와 13kbps QCELP알고리즘을 구현한 결과 소요된 자원은 표 3에, 13kbps QCELP의 주요 블록 들에

대한 자세한 계산량은 표 4에 나타내었다.

표 3. QCELP 구현 결과

Table 3. Result of implementation of QCELP.

	8kbps	13kbps	8k/13kbps
PROM	7.3k words	11.7k words	15.6k words
DRAM	2.0k words	2.9k words	2.9k words
DROM	2.2k words	2.9k words	4.0k words
MIPS	25.6 MIPS	33.5 MIPS	33.5 MIPS

표 3에서 보면 8kbps와 13bps를 따로 구현 하였을 때와 두 가지를 함께 구현 하였을 때 사용한 자원들을 비교해보면 두 가지를 함께 구현 한 것이 각각을 따로 구현하는 것보다 보다 효율적임을 알 수 있다.

표 4에 나타낸 계산량들은 앞 절에서 설명된 최적화 과정 중에서 알고리즘 상의 최적화 부분은 제외한 계산량이다. 실제 실시간 구현에서는 표준 알고리즘을 그대로 구현하기 위하여 알고리즘 레벨에서의 계산량 감소를 위한 최적화를 적용하지 않았다. 그러나 알고리즘 상의 최적화 과정이 계산량 감소의 효과가 가장 크기 때문에 앞으로 좀 더 세밀한 유질평가를 수행하여 실시간 구현에 적용한다면 매우 큰 효과를 가져올 수 있다. 그리고 표 4에서 확인할 수 있듯이 피치검색과 코드북 검색이 전체 계산량의 80% 이상을 차지하므로 계산량 감소를 위한 노력을 이곳에 집중하였다.

표 4. 13kbps QCELP의 계산량

Table 4. Complexity of 13Kbps QCELP.

	Function block	MIPS	
Encoder	Pre-processing (HPF)	0.43	
	Compute LPC	1.08	
	LPC to LSP	1.36	
	Rate decision	0.03	
	LSP vector quantization	1.59	
	LSP to LPC	0.65	
	Compute residual	0.20	
	Reduced rate mode	0.64	
	Pitch search (1 subframe)	3.02	
	Codebook search (1 subframe)	0.73	
	Reconstruction	0.05	
	Pitch*3+(codebk+reconst)*15	19.96	
	Data packing	0.27	
	Sub total	Encoder total	30.02
		Decoder total	3.47
Total	Total	33.49	

표 5는 13kbps QCELP의 음질 평가를 위해 5개의 음성 시료에 대하여 SNR과 segSNR을 측정하여 비교한 것이다.

표 5. SNR 및 segSNR의 측정결과

Table 5. Measurement of SNR and segSNR (dB).

Speech sample	Floating-point C Code		Assembly code	
	SNR	segSNR	SNR	segSNR
Male1	18.53	10.75	19.06	11.13
Male2	21.80	18.01	22.35	18.40
Female1	20.88	16.29	21.08	16.41
Female2	19.45	18.12	19.85	18.31
Female3	22.61	12.27	22.64	12.47

실시간으로 구현한 어셈블리 코드의 시뮬레이션 결과와 C 코드 시뮬레이션의 결과를 비교해보면 유질 상의 성능이 거의 동일함을 알 수 있다. 오히려 어셈블리 코드의 결과가 C 코드 결과 보다 더 좋은 것으로 나타났는데 이것은 각각의 시뮬레이션 입력 및 출력을 가지고 측정하였기 때문이다. 즉 SNR 및 segSNR의 측정에 사용한 입력은 인코더 블록의 전처리를 거친 신호이고, 출력신호는 인코더 내에서 포맷트 합성필터를 거쳐 복원된 음성 신호로써 디코더 블록에서의 후처리 과정을 수행하기 전의 신호와 같은 신호이다.

V. 결론

디지털 이동통신 시스템에서 서로 다른 보코더 알고리즘을 사용하는 단말기 사이에는 통신이 불가능하다는 문제가 있다. 본 논문에서는 8kbps QCELP 및 13kbps QCELP 보코더 알고리즘을 하나의 DSP 칩에 구현함으로써 이러한 단말기 사이의 통신 문제를 해결하였다. 이 두 가지 보코더는 알고리즘의 기본적인 구조가 같고, 추출되는 파라미터의 종류도 같으므로 어셈블리 프로그램으로 구현 시 많은 부분을 공통으로 사용할 수 있다. 그러므로 프로그램 구조를 효율적으로 설계할 수 있고, 각각을 따로 구현 한 것과 비교하여 H/W 자원을 약 80% 이하로 사용하여 하나의 DSP에 두 가지 알고리즘을 구현함으로써 H/W의 사용 및 개발 노력을 최소화하였다.

휴대형 단말기에서의 저전력화는 사용 시간을 늘릴 수 있으므로 최적화 과정을 통하여 계산량을 줄임으로써 전력 감소를 실현하는 것이 중요하다. 그러므로 QCELP 보코더 알고리즘을 구현하는데 있어 계산량의 대부분을 차지하고 있는 피치검색 및 코드북 검색에 대하여 알고리즘 레벨, 수식 레벨, 어셈블리 코딩 레벨 등의 각 단계에서의 최적화 과정을 통해 계산량을 줄임으로써 8kbps QCELP는 25MIPS, 13kbps QCELP는 33MIPS의 계산량으로 실시간 구현하였다. 특히 어셈블리 코딩 단계의 최적화에서는 자체 설계한 DSP의 구조적 특성을 이용한 방법이기 때문에 보코더 구현에 적합한 DSP를 함께 개발하는 장점을 최대한 이용한 것이라 할 수 있다.

참고 문헌

1. TIA/EIA IS-96, "Speech service option standard for wideband spread spectrum digital cellular system, April 1994.
2. Qualcomm Inc., "High rate speech service option for wideband spread spectrum communications systems, Feb. 22, 1995.
3. K.J. Byun, MS. Hahn, K.S. Kim, "Implementation of 13kbps QCELP Vocoder ASIC", AP-ASIC '99, pp.258-261, Aug. 23, 1999.
4. H.Y. Yoo, J.J. Kim, K.J. Byun, K.C. Han, D.K. Kim, J.S. Kim, H.B. Lee, M.J. Bae, "Efficient DSP Design for Vocoder Application", CICC '95, pp. 189-192, May 2, 1995.
5. SekChin Chang, Eung-Bae Kim, Ki-Cheol Han, "An Improved 13 Kb/s Speech Coder for PCS", ICICS '97, pp. 532-536, Sep. 9, 1997.
6. 현진일, 한기천, 변경진, 김종재, 유하영, 김경수, 배명진, 보코더용 DSP의 설계, 대한전자공학회, DSP 설계 및 응용 워크숍 논문집, pp. 201-209, Mar. 25, 1995.
7. 한국전자통신연구소 반도체연구단, "ES-C2340 DSP2 Digital Signal Processor Users Guide", 1995.

※ 주관심분야: 디지털 신호 처리, 음성 분석, 합성, 인식 및 코딩, 적응 처리, 3-D 음향 등

▲김 경 수(Kyung Soo Kim)

1977년 2월: 서강대학교 전자공학과 (공학사)
 1977년 2월~1985년: 한국전자기술연구소
 1986년~현재: 한국전자통신연구원 책임연구원

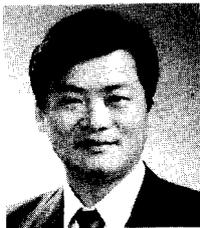
▲변 경 진(Kyung Jin Byun)



1987년 2월: 국민 대학교 전자공학과 (공학사)
 1987년 3월~현재: 한국전자통신연구원 통신디지털회로팀 선임연구원
 1998년 3월~현재: 한국정보통신대학원대학교 공학부 석사과정 재학

※ 주관심분야: DSP 설계, 음성신호 코딩

▲한 민 수(Minsoo Hahn)



1979년 2월: 서울대학교 전기공학과 (공학사)
 1981년 2월: 서울대학교 대학원 (공학석사)
 1989년 12월: University of Florida (공학박사)
 1982년 4월~1985년 8월: 한국표준과학연구원 연구원

1990년 2월~1997년 12월: 한국전자통신연구원 책임연구원
 1998년 1월~현재: 한국정보통신대학원대학교 공학부 부교수