

# 고성능 DSP를 이용한 G.723.1 Annex A 구현

## Implementation of a G.723.1 Annex A Using a High Performance DSP

최 용 수\*, 강 태 익\*  
(Yong-Soo Choi\*, Tae-Ik Kang\*)

\*LG전자 기업통신연구소

(접수일자: 2002년 6월 18일; 채택일자: 2002년 8월 22일)

본 논문에서는 고성능 범용 DSP (Digital Signal Processor)를 이용하여 멀티미디어 통신 등에 널리 사용되고 있는 G.723.1 Annex A (G.723.1A)의 다채널 구현에 관해 기술한다. 다채널 구현을 위해서 G.723.1의 알고리즘과 ITU-T (International Telecommunication Union-Telecommunication)에서 제공된 정수 연산 C 코드의 모듈별 계산량을 분석한 후, 이를 기준으로 C코드를 최적화한다. 각 모듈의 최적화 과정은 ITU-T에서 제공되는 시험 벡터를 이용한 검증과 병행한다. 최적화된 코드의 성능을 측정한 결과, 200 MHz TMS320C62x에서 내부 메모리만을 사용하여 5.3/6.3 kbps 두 가지 전송률에 대해서 부호화기와 복호화기를 동시에 17 채널을 수용하였다. 또한 비트-호환 버전의 경우에는 고속 코드북 검색 알고리즘을 도입하여 음질을 유지하면서 22 채널을 수용하였다.

**핵심용어:** G.723.1, 음성 부호화기, DSP, 최적화, 다채널 구현

**투고분야:** 음성처리 분야 (2,2)

This paper describes implementation of a multi-channel G.723.1 Annex A (G.723.1A) focused on code optimization using a high performance general purpose Digital Signal Processor (DSP). To implement a multi-channel G.723.1A, functional complexities of the ITU-T G.723.1A fixed-point C-code are measured and analyzed. Then we sort and optimize C functions in complexity order. In parallel with optimization, we verify the bit-exactness of the optimized code using the ITU-T test vectors. Using only internal memory, the optimized code can perform full-duplex 17 channel processing. In addition, we further increase the number of available channels per DSP into 22 using fast codebook search algorithms, referred to as bit-compatible optimization.

**Keywords:** G.723.1, Speech coder, DSP, Optimization, Multi-channel implementation

**ASK subject classification:** Speech signal processing (2,2)

## I. 서론

최근 인터넷 사용의 저변 확대에 의해 VoIP (Voice over Internet Protocol)[1]와 같은 저렴한 패킷 음성 통신이 각광을 받고 있다. 이러한 네트워크를 통한 음성 통신 분야에서 G.723.1 (6.3 kbps Multi Pulse Maximum Likelihood Quantization: MP-MLQ / 5.3 kbps Algebraic Code-Excited Linear Prediction: ACELP)[2]은 G.729 (8kbps Conjugate-Structure ACELP: CS-ACELP)[3]와 더불어

필수적인 모듈이 되었다. G.723.1은 5.3/6.3 kbps의 두 가지 전송률을 갖는 멀티미디어 화상 회의 시스템이나 음성 저장용 음성 부호화기 국제 표준이다.

본 논문에서는 G.723.1 시리즈 중에서 정수 연산 C 코드로 구성되어 있는 Annex A[4]를 16비트 고정 소수점 연산을 갖는 고성능 범용 DSP를 사용한 다채널 구현을 기술한다. DSP로 사용된 Texas Instruments의 TMS320C62x[5] (이하 C62x)는 8개의 독립적인 연산 모듈을 가지고 있어서 한 사이클당 최대 8개의 연산을 동시에 수행할 수 있어서 다채널 시스템의 실시간 구현에 효과적이다. 다채널 구현을 위해서는 G.723.1의 알고리즘과 ITU-T (International Telecommunication Union-Telecom-

책임저자: 최용수 (cando@lge.com)  
43-080 경기도 안양시 동안구 호계동 533  
LG전자 기업통신연구소 미디어통신 Gr  
(전화: 031-450-4893; 팩스: 031-450-4943)

munication)에서 제공된 고정 소수점 연산C 코드의 모듈 별 계산량을 분석한 후, 이를 기준으로 C코드를 C62x 기반으로 최적화한다. 최적화는 C62x의 특성을 효율적으로 이용하도록 계산량이 많은 모듈부터 집중적으로 수행하며, 각 모듈의 최적화 과정은 ITU-T에서 제공되는 검증 벡터를 이용한 비트-동일 검사와 병행한다. 최적화 기법으로는 주로 intrinsic 함수, inline 함수, word-wide, loop-unrolling, 루프 합병, 루프 분리, 계산 순서 변경 방법 등이 사용된다. 또한 최적화 이전과 이후의 메모리 사용량과 연산량을 비교 분석한다. 200 MHz C62x에서 최적화 전에는 한 채널도 수용할 수 없지만 최적화를 수행한 결과, 5.3/6.3 kbps 두 가지 전송률에서 부호화기와 복호화기를 동시에 17채널을 수용할 수 있다. 최적화된 C62x 코드는 ITU-T에서 제공되는 검증 벡터를 이용한 비트-동일 시험을 통과하였다. 채널 수를 더 늘리기 위해서 부호화기 모듈 중에서 가장 많은 연산량이 소요되는 고정 코드북 검색 과정에 고속 알고리즘을 도입한다. 실험 결과, 200 MHz C62x에서 22채널을 수용하였고 음질 차이는 구별할 수 없었다.

본 논문의 구성은 다음과 같다. II장에서는 G.723.1의 구조와 특징에 대해서 간단히 설명하고, III장에서는 코드 최적화 방법을 설명하고, IV장에서는 성능을 비교 분석한다. 마지막으로 V장에서 결론을 맺는다.

## II. G.723.1 (6.3/5.3 kbps MP-MLQ /ACELP)

G.723.1은 그림 1에서와 같이 대개의 표준 음성 부호화기와 마찬가지로 선형 예측 (linear prediction)과 분석-합성 (analysis-by-synthesis) 방식을 사용하는 CELP[6]에 바탕을 두고 있다. 다중 템 피치 예측기와 다중 펄스로 구성된 고정 코드북 검색을 주요한 특징으로 하며, 두 가지의 전송률에 따라서 비트 할당, 코드북 구성과 검색 범위가 다르며 이에 따라 전체적인 복잡도와 음질이 다소 차이가 다르다. G.723.1은 응용 목적에 따라 세 가지 시리즈를 갖는다. Annex A는 16비트 고정 소수점 연산을 갖는 DSP 구현을 목적으로 한 정수 연산 C 코드로 구성되어 있으며 목음 구간 압축을 위한 VAD (Voice Activity Detection)/ CNG (Comfort Noise Generator) 기능이 포함되어 있다. Annex B는 Annex A와 동일한 알고리즘을 부동 소수점 연산으로 구성하고 있으며 비트 스트림도 동일하며, Annex C는 이동통신 응용을 위해서 0.7~14.3 kbps의 다양한 채널 코덱 비트율을 지원한다. 표.1은 전송률에 따른 G.723.1의 비트 할당을 보여준다.

G.723.1 음성 부호화기의 알고리즘 특징을 요약하면 다음과 같다.

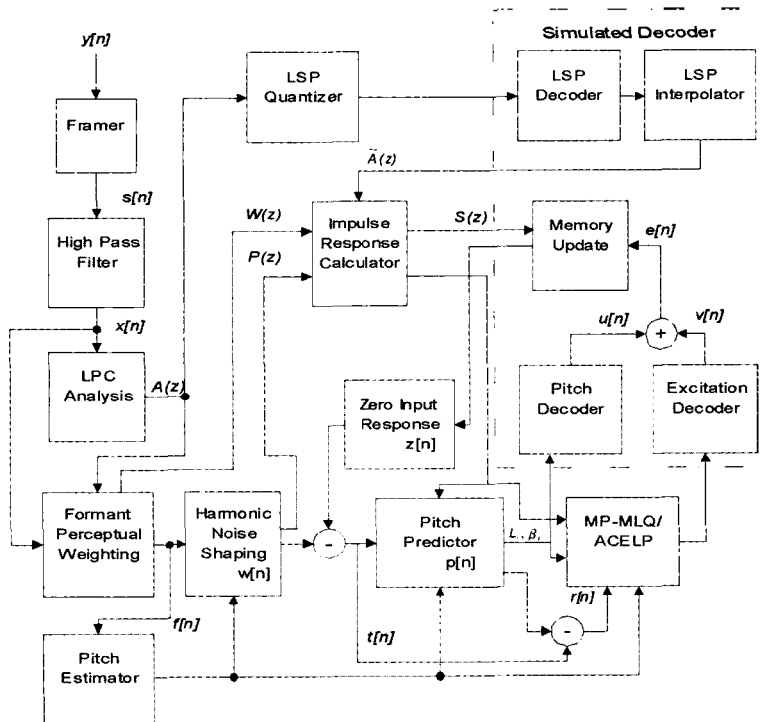


그림 1. G.723.1 음성 부호화기 구조  
Fig. 1. Block diagram of the G.723.1 speech coder.

표 1. G.723.1A 비트 할당  
Table 1. Bit Allocation of G.723.1A.

Parameter	6.3 kbps	5.3 kbps
LPC indices	24	24
ACB lags	18	18
ACB & FCB gains	48	48
Pulse positions	73	48
Pulse signs	22	16
Grid index	4	4
Total	189	158

- 8 kHz 샘플링, 16 비트 linear PCM 입력 신호.
- 30 ms frame + 7.5 ms lookahead = 37.5 ms algorithmic delay.
- Voice Activity Detection / Comfort Noise Generator
- 7.5 ms마다 10차 LPC 분석 및 LSP변환
- 15 ms마다 피치 검색 및 갱신
- 7.5 ms마다 적응 및 고정 코드북
- 검색, 5.3 kbps와 6.3 kbps의 두 가지 비트 할당 지원.
- 전송률에 따라 다른 고정 코드북 구성  
5.3 kbps: 대수 (algebraic) 코드북, 6.3 kbps: 다중 펄스 코드북
- Harmonic noise shaping.
- Adaptive post filtering

5.3 kbps ACELP에서는 표 2와 같이 대수 코드북이라고 불리는 특수한 코드북 구조를 가지고 있어서 계산량 면에서 효율적이며 우수한 음질을 재생해 낼 수 있다. 4개의 펄스로 모델링된 여기 신호는 +1 혹은 -1의 크기를 갖고 미리 정해진 펄스 위치 중에서 최적의 위치를 갖도록 구해진다.

6.3 kbps MP-MLQ의 다중 펄스 코드북은 펄스 수, 펄스 위치, 검색 방법이 다르나 기본적인 원리나 효과는 대수 코드북과 유사하다고 볼 수 있다. ACELP 알고리즘에 비해서 MP-MLQ 알고리즘은 표준안 문서나 논문에서 자세하게 기술되어 있지 않다. 그러나 MP-MLQ 알고리즘을 이

표 2. G.723.1의 5.3 kbps 대수 코드북  
Table 2. 5.3 kbps algebraic codebook of G.723.1.

Pulse	Sign	Positions
$i_0$	+/- 1	0, 8, 16, 24, 32, 40, 48, 56
$i_1$	+/- 1	2, 10, 18, 26, 34, 42, 50, 58
$i_2$	+/- 1	4, 12, 20, 28, 36, 44, 52, (60)
$i_3$	+/- 1	6, 14, 22, 30, 38, 46, 54, (62)

해하는 것은 III장에서 언급할 최적화 과정에 매우 필요하다. 이러한 이유로 본 논문에서는 MP-MLQ 알고리즘에 대해서 자세하게 기술한다.

MP-MLQ 여기 신호는 +1 혹은 -1의 크기를 갖는 다중 펄스로 구성된다. k번째 펄스 검색에 대해서 목적 신호와 여기 신호와의 오차는 다음과 같다.

$$e(n) = r(n) - G a_k \delta(n - m_k) * h(n), \quad (1)$$

여기서  $r(n)$ ,  $G$ ,  $a_k$ ,  $m_k$ ,  $h(n)$ 은 각각 목적 신호, 코드북 이득, k번째 펄스 부호, k번째 펄스 위치, 그리고 인지가 중합성 필터의 충격파 응답이다. 식 (1)은 벡터 형태로 다시 표현하면 식 (2)와 같다.

$$e = r - GHX, \quad (2)$$

여기서  $X$ 는 여기 벡터이다. 일반적인 CELP[6]에서처럼 펄스의 부호와 위치는 정규방정식 형태의 기준치 (criterion)를 최소화하는 값으로 결정된다.

$$crt = \frac{((HX)^T r)^2}{((HX)^T HX)}. \quad (3)$$

편의상 식 (3)에서  $HX \Rightarrow h(n - m_k)$ ,  $H^T r \Rightarrow d(n)$ 와 같이 벡터 형태를 샘플 형태로 대체하면 식 (3)은 식 (4)로 표현된다.

$$crt = \frac{(\sum_n a_k \delta(n - m_k) d(n))^2}{\sum_n h(n - m_k) h(n - m_k)}. \quad (4)$$

다시 식 (4)는 다음과 같이 간략화될 수 있다.

$$crt = \frac{(a_k d(n - m_k))^2}{R_{hh}(0)}, \quad (5)$$

여기서  $R_{hh}(0)$ 는  $h(n)$ 의 에너지이다. 식 (5)에서 분모는 펄스 검색에 대해서 상수이므로 분자만을 최소화하면 된다. 결과적으로 최적의 펄스 위치와 부호는 다음의 두 식을 이용하여 쉽게 결정된다.

$$m_k = \arg \max_n |d(n)|, \quad (6)$$

$$a_k = \text{sign}(d(m_k)). \quad (7)$$

이전 펄스의 성분을 제거하기 위하여  $e(n)$ 은 다음과 같이 갱신된다.

$$e_{new}(n) = e_{old}(n) - Ga_{k-1}\delta(n - m_{k-1}) * h(n). \quad (8)$$

다시,  $d_{new}(n)$ 은 식 (9)처럼  $e_{new}(n)$ 을  $h(n)$ 으로 역방향 필터링 (backward filtering)하여 얻어진다.

$$e_{new}(n) * h(-n) = e_{old}(n) * h(-n) - Ga_{k-1}\delta(n - m_{k-1}) * h(n) * h(-n). \quad (9)$$

또한 식 (9)는 식 (10)과 같이 반복식 (recursive) 형태로 다시 표현되며, 식 (10)은 식 (6)과 (7)에 적용된다.

$$d_{new}(n) = d_{old}(n) - Ga_{k-1}\delta(n - m_{k-1}) * R_{hh}(n) = d_{old}(n) - Ga_{k-1}R_{hh}(m_{k-1}). \quad (10)$$

### III. 코드 최적화

#### 3.1. C62x DSP 특징

Texas Instruments의 TMS320C62x[5]는 VelociTI, 즉, 진보된 VLIW (Very-Long-Instruction-Word) 구조를 갖는 DSP로서 다음과 같은 특징을 가지고 있어서 다채널 시스템 구현에 효과적이다.

- 32개의 32 비트 범용 레지스터.
- 8개의 독립적인 연산 모듈을 가지고 있어서 한 사이클 당 최대 8개의 32비트 명령을 동시에 수행 가능.
- 8/16/32 비트 데이터를 지원하고 있어서 효율적인 메모리 관리가 가능.
- 40비트 산술 연산 옵션이 가능하여 상당한 정밀도를 요구하는 응용에 적합.
- 다채널 DMA (Direct Memory Access), 다채널 BSP (Buffered Serial Port).
- 대용량 내부 RAM (Random Access Memory)
- 고효율의 C 컴파일러, assembly optimizer 제공으로 인한 개발 기간 단축

DSP 솔루션 개발자의 입장에서 보면, C62x의 큰 장점 중의 하나는 어셈블리어로 직접 프로그래밍을 하지 않고 C 코드만을 최적화하여도 원하는 성능을 얻을 수 있어서 개발 기간을 단축시킬 수 있다는 것이다.

#### 3.2. 비트-동일 (Bit-exact) 최적화

G.723.1A는 매우 복잡한 알고리즘으로 구성되어 있어서 ITU-T에서 제공된 원시 C코드를 TI의 C 컴파일러를 사용하여 C62x로 구현할 경우 최적화 옵션을 최대로 하여도 한 채널을 수용하지 못한다. 따라서 보다 많은 채널을 수용하기 위해서는 C 코드를 C62x특성에 맞도록 직접 C 코드를 수정하는 최적화를 수행해야 한다. 최적화 방법으로 TI에서 권고하고 있는 일반적인 방법들[7,8], G.723.1A에 적용 가능한 기존의 음성 부호화기에 사용된 방법들[8]을 우선적으로 적용한다. 다음으로 G.723.1A 알고리즘 특성에 맞는 새로운 최적화 방법들을 적용한다. G.723.1A의 C62x 최적화에 대한 전체 과정을 정리하면 그림 2와 같다.

단, 최적화 단계에서의 연산량 측정은 Texas Instruments의 Code Composer Studio의 Profile기능을 사용할 수도 있지만 이 경우는 디버그 (-g) 옵션을 주어야 하므

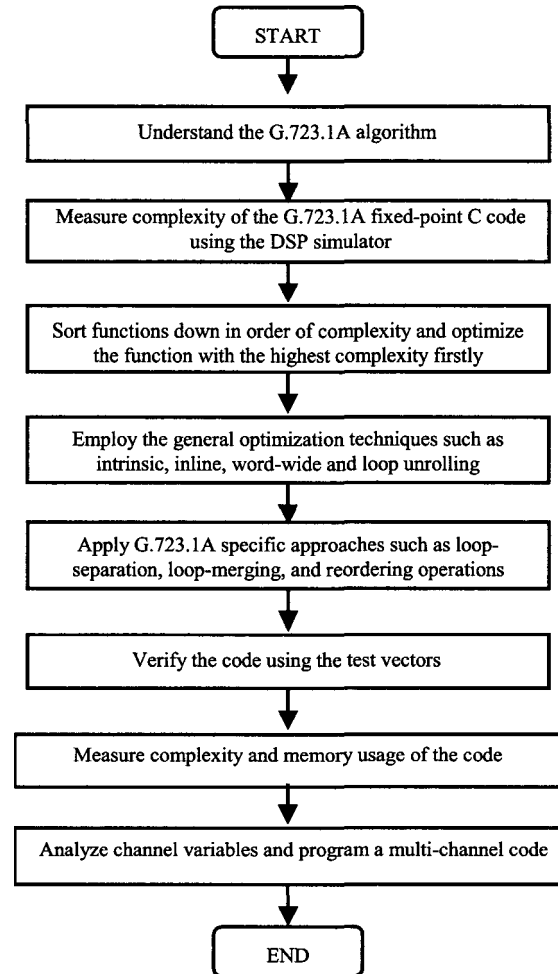
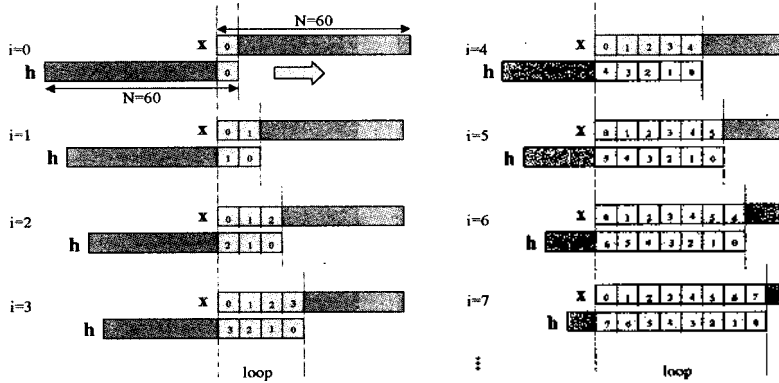


그림 2. G.723.1 C코드 최적화 과정  
Fig. 2. Flowchart of G.723.1 C code optimization.

Convolution,  $y(n) = x(n)*h(n)$ : conv(x, h, y, N)  
Conventional: 3050 clocks



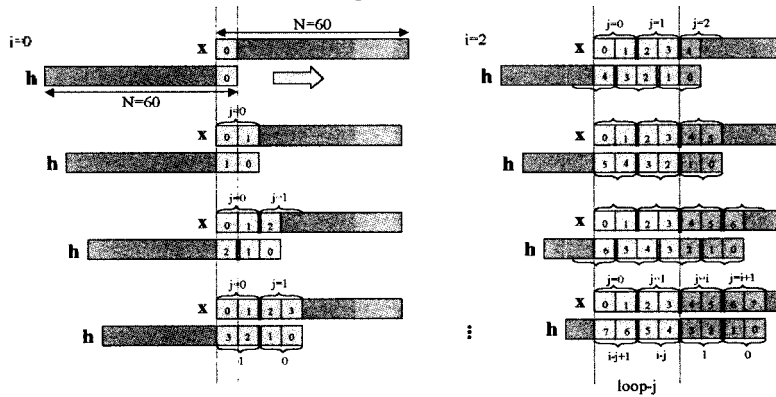
```

/* loop */
for ( i = 0; i < N; i ++ ) {
    Acc0 = (int) 0 ;
    for ( j = 0; j <= i; j ++ ) {
        Acc0 = _sadd( Acc0, _smpy(x[j], h[i-j]) );
        y[i] = _sadd(Acc0, 0x8000)>>16;
    }
}

```

그림 3. 비최적화된 컨벌루션  
Fig. 3. Non-optimized convolution.

Convolution,  $y(n) = x(n)*h(n)$ , conv3(X, H, y, N)  
4 unrolling, LDW: 1344 clocks



```

y[0]=...; y[1]=...; y[2]=...; y[3]=...;
/*loop */
for (i=2; i<N2; i+=2) {
    Acc0=0; Acc1=0; Acc2=0; Acc3=0;
    Acc00=0; Acc11=0; Acc22=0; Acc33=0;
    for(j=0; j<i; j++) {
        Acc0= _sadd(Acc0, _smpy(X[j], H[i-j]));
        Acc00=_sadd(Acc00, _smpyh(X[j], H[i-j-1]));
        Acc1= _sadd(Acc1, _smpyh(X[j], H[i-j]));
        Acc11=_sadd(Acc11, _smpyh(X[j], H[i-j]));
        Acc2= _sadd(Acc2, _smpy(X[j], H[i-j+1]));
        Acc22=_sadd(Acc22, _smpyh(X[j], H[i-j]));
        Acc3= _sadd(Acc3, _smpyh(X[j], H[i-j+1]));
        Acc33=_sadd(Acc33, _smpyh(X[j], H[i-j+1]));
    }
    y[2*i]=...; y[2*i+1]=...; y[2*i+2]=...; y[2*i+3]=...;
}

```

그림 4. 최적화된 컨벌루션  
Fig. 4. Optimized convolution.

로 상대적인 연산량 비교만 의미가 있다. 따라서 최적화 (-o3) 옵션과 clock() 함수를 사용하여 절대적인 연산량을 측정하는 것이 필요하다.

일반적인 최적화 기법을 좀 더 상세히 설명하면 다음과 같다.

- basop.c와 basop.h에 함수로 선언되어 있는 기본 연산자들을 C62x intrinsic 함수와 shift 연산자를 사용하여 다시 정의한다[8,9]. 그렇지 못한 경우에는 예를 들어 div\_s(), div\_l()같은 경우는 inline 함수로 선언한다.
- LPC 분석, LSP 양자화, 피치 추정, 적응 코드북 검색, 고정 코드북 검색 등에 많이 사용되는 dot product 형태의 자기상관, 상호상관, 에너지, 컨벌루션 계산 과정에서 word-wide 연산과 loop-unrolling을 적절히 혼용하여 최적화한다. 대개 이러한 dot product 형태는 중복 루프 안에 존재하여 중간 값에 대한 Q 포맷 이동이 있는 경우가 많아서 함수화하기 어려운 경우가 많지만 그렇지 않은 경우는 모듈화하여 재사용한다. 특히 컨벌루션은 G.723.1내에서도 자주 사용되고 G.729와 같은 이중의 CELP 부호화기에도 적용이 가능하고 correlation 연산과도 유사하기 때문에 그 중요도는 매우 크다. 그림 3은 비최적화된 컨벌루션 연산 과정과 코드를 보여 주며 그림 4는 최적화된 컨벌루션 과정과 코드를 도식적으로 보여준다.

일반적인 최적화 기법은 연산량 감소에 한계가 있으므로 G.723.1A 알고리즘과 코드 특성에 맞는 최적화를 수행할 필요가 있다. G.723.1A 특화된 최적화 방법 중에서 연산량 감소에 큰 영향을 주는 사항을 언급을 하면 다음과 같다.

- -mw -ss 옵션을 사용하여 software pipelined information을 보고 임시 배열, 루프 합병, 루프 분리 등을 사용하여 dependency를 개선한다. 예를 들어 D4i64\_LBC()에서 mean, max를 구하는 부분과 상관 값을 수정하는 부분에서는 독립된 루프들을 하나의 루프로 통합하는 루프 합병 방법을 사용하고, 4개의 다중(nested) 루프로 구성된 펄스 검색 부분은 3중 루프와 4중 루프로 분리하는 루프 분리 방법을 사용한다.
- 모든 반복 루프에서는 가장 내부에 있는 루프 또는 함수만 -o3 옵션으로 자동 최적화되므로 루프 안에서는 가능한 함수 호출이나 루프를 사용하지 않는다.

대표적인 예로 Find\_Acbk()에서 루프 안에서 사용되는 Get\_Rez()의 경우에 비교적 간단한 함수이므로 이를 루프 안에서 전개하여 inline화 한다. 이 때 모듈로(%) 연산을 피치 구간에 따른 조건문으로 처리한다.

- 루프에서 반복 횟수가 충분히 작으면 루프를 제거하고 반복 횟수만큼 코드를 직접 전개한다. 또한 다중 루프의 경우에도 내부 루프의 반복 횟수가 작으면 루프를 순차적으로 전개한다. 후자의 경우에 계산량 감소 효과는 매우 크다. 대표적인 예로 Lsp\_Svg()의 경우에 3중 루프로 구성되어 있는데 가장 내부의 3번째 루프의 반복 횟수가 아주 작다. 이 경우에는 3개의 3번째 내부 루프들을 하나로 합치는 루프 합병을 한 후, 3번째 루프를 전개한다.
- 다중 루프에서 외부 루프의 반복 횟수가 작은 경우에는 외부 루프를 제거하고 반복 횟수만큼 내부 루프를 순차적으로 반복한다.
- 루프의 반복 횟수가 크지만 그 값을 알 수 있거나 범위를 알 수 있다면 \_nassert()를 사용한다.

### 3.3. 비트-호환 (Bit-compatible) 최적화

앞 절에서 언급한 비트-동일 최적화와는 조금 다른 접근 방식도 가능하다. 실제 음성 처리 시스템 응용에 있어서는 주관적인 음질이 유지된다면 계산량을 줄여서 채널 수를 더 늘리는 것이 더 효과적이라고 할 수 있다. 이러한 목적을 위해서 주관적 음질을 유지하는 범위 내에서 알고리즘을 수정하여 계산량을 원천적으로 줄이는 방법이 사용될 수 있다. 본 논문에서는 부호화기 모듈 중에서 가장 많은 연산량이 소요되는 고정 코드북 검색 과정에 고속 알고리즘을 도입하여 계산량을 크게 줄인다.

6.3 kbps MP-MLQ에서는 각 서브프레임에서 충격파 응답의 수정없이 펄스 검색 Find\_Best()를 수행하고, 만약 개회로 피치가 서브프레임 길이보다 2만큼 더 작으면 충격파 응답을 피치 주기로 수정하여 펄스 검색 Find\_Best()을 한 번 더 한다. 이것은 무성음에서의 음질 향상을 위한 것으로 추정되는데 본 논문에서는 충격파 응답 수정을 포함한 펄스 검색 Find\_Best()을 한 번만 하도록 알고리즘을 수정한다. 또한 5.3 kbps ACELP에서는 [10]에서 처럼 depth-first tree 검색[11]을 적용하여 1개의 4중 루프를 2개의 2중 루프로 분리하여 검색하도록 알고리즘을 수정한다. 실험 결과, 계산량을 크게 줄이면서도 음질 차이는 구별할 수 없었다.

## IV. 성능 비교

우선 최적화 결과를 고찰하기 전에 간단히 C62x C 컴파일러의 성능을 살펴본다. 표 3은 컴파일러 옵션, intrinsic 함수의 사용 유무에 따른 G.723.1A 부호화기가 소요하는 클럭 수를 예로 보여 준다.

이거서 200 MHz C62x를 기준으로 보면 30 ms 프레임 당 이용 가능한 클럭 수는 6,000,000이다. 따라서 원시 C 코드는 컴파일러 옵션을 최대로 하여도 한 채널도 수용할 수 없지만, 최적화의 첫 단계인 intrinsic만 사용하면 부호화기 소요 클럭이 82,645임을 감안하더라도 4채널 까지 수용할 수 있다는 것을 알 수 있다. 표 4는 앞장에서 설명한 기본 전략을 가지고 최적화를 하였을 경우 비트-동일 G.723.1A가 소요하는 클럭 수와 채널 수를 보여준

표 3. ITU-T G.723.1A 코드의 계산량  
Table 3. Complexity of the ITU-T G.723.1A code.

Compiler option	Optimization	# of clocks / 30 ms frame	# of channels @ 200 MHz C62x
-g		48,144,532	0
-o3		16,000,535	0
-o3	intrinsic	1,141,370	4.9

표 4. 최적화된 G.723.1A 코드의 계산량: 비트-동일 버전  
Table 4. Complexity of the bit-exact optimized G.723.1A code.

	6.3 kbps	5.3 kbps
Encoder	306,222	299,423
Decoder	30,926	28,753
Total	337,148	328,176
# of channels @ 200 MHz C62x	17.8	18.3

표 5. 최적화된 G.723.1A 코드의 계산량: 비트-호환 버전  
Table 5. Complexity of the bit-compatible optimized G.723.1A code.

	6.3 kbps	5.3 kbps
Encoder	237,198	242,195
Decoder	30,926	28,753
Total	268,124	270,948
# of channels @ 200 MHz C62x	22.4	22.1

표 6. 선호도 평가  
Table 6. Preference test.

	Bit-exact version	Bit-compatible version	No preference
Percent (%)	26	19	43

다. 여기서 클럭 수는 모든 시험 벡터들을 대상으로 얻어진 최대 값이다. 표 4에서 C 코드 최적화를 수행한 결과, 5.3/6.3 kbps 두 가지 전송률에서 부호화기와 복호화기를 동시에 17 채널을 수용함을 알 수 있다. 최적화된 C62x 코드는 ITU-T에서 제공되는 검증 벡터를 이용한 비트-동일 시험을 통과하였다. 표 5는 비트-호환 최적화 버전의 경우 소요되는 클럭 수와 수용 가능한 채널 수를 보여 준다.

표 5에서 고속 알고리즘 도입으로 인하여 채널 수를 크게 증가시킬 수 있다.

비트 호환 버전의 음질 평가를 위해 선호도 평가를 비공식적으로 수행하였다. 평가 결과, 표 6에서와 같이 비트 호환 버전과 차이를 구별할 수 없다는 응답이 전체 청취자의 62%를 차지하였다. 결과적으로 고속 알고리즘을 적용한 비트-호환 버전의 경우 만족할 만한 음질을 가짐을 확인할 수 있었다.

한 채널 코드에 필요한 메모리는 프로그램 영역에 60 KB, 데이터 영역에 51 KB였다. 64 KB 프로그램 메모리, 64 KB 데이터 메모리를 가지고 있는 C6201을 기준으로 할 때 다채널 구현에 필요한 추가적인 메모리를 포함하여도 순수한 프로그램과 데이터는 내부 램에 다 들어갈 수 있고, 비교적 많은 메모리를 필요로 하는 입출력 버퍼만 외부 메모리에 두면 되므로 전체 채널 수에는 큰 영향이 없다. 다채널 구현을 위해서 채널 변수를 분석하고 이를 채널 별로 할당하고[12], 다중 채널 신호 입출력은 DMA와 McBSP를 사용하여 효과적으로 구현한다[13]. 또한 패킷 입출력은 HPI를 통해 구현한다[14].

## V. 결론

본 논문에서는 C62x를 이용하여 G.723.1 Annex A를 구현을 하였다. 다채널 구현을 위해서 G.723.1의 알고리즘과 ITU-T에서 제공된 정수 연산 C 코드의 모듈별 계산량을 분석한 후, 이를 기준으로 C 코드를 C62x 기반으로 최적화하였다. 최적화 방법은 일반적인 방법과 G.723.1 알고리즘 특성에 맞는 특수한 방법을 사용하였다. 각 모듈의 최적화 과정은 시험 벡터 검증과 병행하였다. 비트-동일 버전의 경우는 200MHz C62x에서, 5.3/6.3 kbps 두 가지 전송률에 대해서 부호화기와 복호화기를 동시에 17 채널을 수용하였으며 고속 알고리즘을 적용한 비트-호환 버전의 경우는 만족할 만한 음질을 유지하면서 22 채널을 수용하였다. 또한 제안된 최적화 방법들을 G.729A

와 같은 다른 CELP부호화기에도 적용하여 구현하였다. 현재 톤 검출기/발생기, 반향 제거기, 이득 조절기 등을 통합하는 VoIP 게이트웨이용 DSP 솔루션에 관한 연구를 진행 중에 있다.

### 참고 문헌

1. G. Held, *Voice Over Data Network*, McGraw-Hill, 1998.
2. ITU-T Recommendation G.723.1, *General Aspects of Digital Transmission Systems: Dual Rate Speech Coder For Multimedia Communications Transmission at 5.3 and 6.3 kbit/s*, 1996.
3. ITU-T Recommendation G.729, *Coding of Speech at 8 kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP)*, November 1996.
4. ITU-T Recommendation G.723.1-Annex A, *General Aspects of Digital Transmission Systems: Dual Rate Speech Coder For Multimedia Communications Transmission at 5.3 and 6.3 kbit/s Annex A: Silence Compression Scheme*, 1996.
5. Texas Instruments, *TMS320C62x/C67x CPU and Instruction Set Reference Guide*, 1998.
6. M. R. Schroeder and B. S. Atal, "Code-Excited Linear Prediction (CELP): High Quality Speech at Very Low Bit Rates," *IEEE Proc. Int. Conf. Acoust. Speech and Signal Proc.*, 25.1.1-25.1.4, 1985.
7. Texas Instruments, *TMS320C6000 Optimizing C Compiler User's Guide*, 1999.
8. Texas Instruments, *TMS320C62x/C67x Programmer's Guide*, 1998.
9. Texas Instruments Application Report SPRA 617, *ETSI Math Operation in C for the TMS320C62xx*, December 1999.
10. 정성고, 박영철, 윤성완, 차일환, 윤대희, 효율적인 다채널 구현을 위한 ITU-T G.723.1 음성 부호화기 고속 알고리즘 제안, 한

- 국통신학회 하계학술발표 대회 논문집, 19 (1(s)), 67-70, 2000.
11. ITU-T Recommendation G.729-Annex A, *Coding of Speech at 8 kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP), Annex A: Reduced Complexity 8 kbit/s CS-ACELP Speech Codec*, November 1996.
12. Texas Instruments Application Report SPRA552, *G.723.1 Dual Rate Speech Coder: Multichannel TMS320C6000 Implementation*, 1999.
13. Texas Instruments Application Report SPRA559, *A Multichannel Serial Port Driver Using DMA on the TMS320C6000 DSP*, 1999.
14. Texas Instruments, *TMS320C6000 Peripherals Reference Guide*, 1999.

### 저자 약력

● 최 용 수 (Yong-Soo Choi)



1993년: 연세대학교 전자공학과 (공학사)  
 1995년: 연세대학교 전자공학과 (공학석사)  
 1999년: 연세대학교 전기컴퓨터공학과 (공학박사)  
 1999년 ~ 현재: LG전자 기업통신연구소 선임연구원  
 \* 주관심분야: 음성신호처리, 음성부호화, 음질평가, VoIP, 실시간 DSP 응용

● 강 태 익 (Tae-Ik Kang)



1979년: 연세대학교 전자공학과 (공학사)  
 1981년: 연세대학교 전자공학과 (공학석사)  
 1990년: Polytechnic University 전기공학과 (공학박사)  
 1981년 ~ 1984년: 육군사관학교 조교수  
 1990년 ~ 현재: LG전자 기업통신연구소 책임연구원  
 \* 주관심분야: 디지털신호처리, 영상/음성 신호처리, VoIP, ATM