

Vincent6 DSP 코어를 이용한 G.728 음성 부호화기의 실시간 구현

성호상

한국전자통신연구원

Real-time implementation of the G.728 speech codec using the Vincent6 DSP core

Hosang Sung

Electronics and Telecommunications Research Institute

E-mail : hssung@etri.re.kr

요 약 문

본 논문에서는 고성능 고정 소수점 DSP (Digital Signal Processor) 코어인 Vincent6 코어[1]를 이용하여 ITU-T G.728 음성 부호화기를 실시간으로 구현하였다. G.728 은 16 kb/s 전송률의 ITU-T 표준 음성 부호화기이며, 입력신호는 8 kHz로 샘플링되며 샘플 당 16 bit 로 양자화된 PCM 신호이다. G.728 은 LD-CELP(Low Delay Code Excited Linear Prediction) 라고도 하며, 알고리즘 delay 는 0.625ms 이다. Vincent6 DSP core 는 VLIW (Very-Long Instruction Word) 특성을 가지므로 다중 명령 (multiple instruction)을 수행할 수 있다. 이를 위해서 G.728 annex G 를 이용하여 고정 소수점 연산으로 코드를 작성한 후, 이를 Vincent6 어셈블리 코드로 구현하였다. 최종적으로 구현된 코드는 ITU-T 의 test vector 에 대해 bit exact 한 결과를 보이며 34 MCPS (Million Cycles Per Second)의 계산량을 가지며 사용 메모리크기는 데이터 메모리가 약 9 KByte, 프로그램 메모리가 약 57 KByte 이다.

1. 서 론

과거에 전화 대역폭을 가지는 음성 부호화기는 1960 년대에 표준화가 된 64 kb/s A-law, μ -law PCM 과 1984 년에 표준화가 된 32 kb/s ADPCM 이 있으며 이들은 샘플 단위로 부호화를 하므로 낮은 지연과 toll 음질을 제공했다. 위의 두 음성 부호화기는 CCITT (International Telegraph and Telephone Consultative Committee)에 의해 각각 G.711 과 G.721 로 명명되었다. 그 이후 저속 음성 부호화기에 대한 요구로 LPC-10E, FS-1016, VSELP, IS-54, RPE-LTP 등 다양한 음성부호화기가 출현하였다. 하지만 여기서 언급된 대부분의 음성 부호화기가 낮은 음질 그리고 긴 지연을 가지고 있었기 때문에 사용에 많은 단점을 가지고 있었다. 그래서 1988 년부터 CCITT 는 알고리즘 지연, 프로세싱 지연, 전송 지연을 합한 단방향 지연이 5 ms 이하이며 toll 음질을 제공하며, 비디오 폰, 무선전화기, DCME (Digital Circuit Multiplication Equipment), PSTN (Public Switched Telephone Network), ISDN (Integrated Service Digital Network), 음성 저장 및 전송장치, 팩스 음성 등의 용도에 사용할 목적으로 16 kb/s 음성 부호화기에 대한 표준화 작업을 시작했다. 그 후 1992 년 고음질을 갖으며 약 2 ms 의 단방향 지연을 갖는 LD-CELP (Low Delay

Code Excited Linear Prediction)가 표준이 되었으며 이는 G.728 로 명명되었다[2][3]. 그리고 CCITT 가 ITU-T 로 명칭이 바뀐 후 G.729, G.723.1 등의 새로운 음성 부호화기가 개발되어 그 사용처가 많이 줄어들었지만 아직도 H.320, H.321, H.322 를 이용한 영상회의 시스템, ISDN 전화 등에서는 낮은 delay 와 고음질로 인해 G.728 음성 부호화기가 많이 사용되고 있다.

본 논문에서는 32 비트 고정 소수점 연산을 갖는 멀티미디어용 DSP 코어인 Vincent6 DSP 코어를 이용하여 G.728 을 실시간으로 구현을 하였다. Vincent6 DSP 코어는 100MHz 로 동작하며 VLIW 특성을 가지므로 한 사이클에 여러 명령어를 수행할 수 있다. 본 논문의 구성을 다음과 같다. 2 장에서는 G.728 의 알고리즘에 대해서 설명하고, 3 장에서는 G.728 의 실시간 구현에 대해 살펴보고 마지막을 4 장에서 결론을 맺는다.

2. ITU-T G.728 음성 부호화기의 개요

2.1 G.728 음성 부호화기

G.728 의 입력신호는 8 kHz 로 샘플링 된 후 샘플 당 16 비트로 양자화된 PCM 신호이다. G.728 의 알고리즘 지연은 0.625ms (5 샘플)이며 이 5 샘플을 블록이라고 한다. G.728 은 블록 단위로 부호화를 하며 4 개의 블록이 하나의 변경 주기 (adaptation cycle)가 되는데 이는 한 변경 주기 내에서 각 블록이 수행하는 내용이 다르기 때문이다. 하지만 각 블록은 독립적으로 수행되므로 알고리즘 지연은 0.625 ms 가 된다. G.728 은 거의 MOS 4.0 의 toll 수준의 음질 성능을 제공하며 CELP 방식이며, 합성을 통한 분석을 하는 AbS (Analysis-by-Synthesis) 구조를 가진다[4]. 일반적인 CELP 에서는 5 개의 정보가 부호화되어서 복호화기로 전송된다. 이 5 개의 정보는 1) LPC (Linear Predictive Coding) 파라미터; 2)피치 예측기 (pitch period); 3)피치 예측기 탭수 (pitch predictor tap); 4) 여기 신호 이득 (excitation gain); 5)여기신호 셰이프 코드벡터 이다. 그러나 G.728 에서는 50 차 LPC 분석을 이용하므로 피치 예측기가 필요 없게 된다. 그리고 합성필터의 계수와 이득 정보를 backward adaptation 을 사용하여 주기적으로 예측기의 계수를 재구성 하는데, 이전의 양자화된 신호와 이득이 증폭된 여기 신호 (Gain-scaled excitation)를 이용한다. 이 방

식은 다음과 같은 특징이 있다. 사이드 인포메이션(Side information)이 없으므로 낮은 전송률이 가능하지만, 버퍼가 많이 필요하며 계산량이 많은 단점도 있다. 각 블록에 대한 전송 파라미터는 단지 최적의 코드북 벡터 (10bit)이며, 이는 7 비트의 셰이프 코드북 (shape codebook)과 3 비트의 이득 코드북 (부호 : 1 비트, 크기 : 2 비트)으로 이루어져 있으며, 이는 CELP 방식의 음성 부호화기에서 여기 신호로 볼 수 있다.

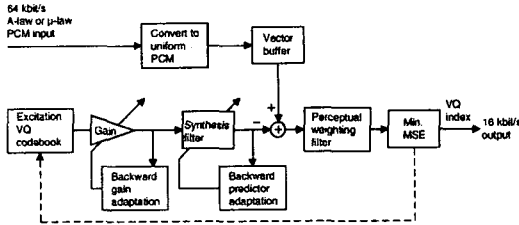


그림 1. G.728 음성 부호화기의 부호화 부분의 블록도

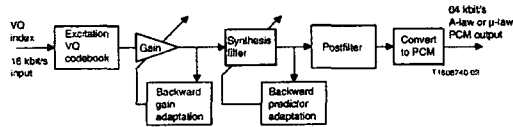


그림 2. G.728 음성 부호화기의 복호화 부분의 블록도

복호화기에서도 전송된 비트 스트림을 블록단위로 복호화를 하며 gain 정보와 합성필터 계수는 backward adaptation을 이용하여 자체적으로 구성하며 전송된 코드북 벡터를 이용하여 음성을 합성한다. 합성된 음성은 포스트필터링(postfiltering)과정을 거치는데 포스트필터는 장구간 포스트필터(Long-term postfilter)와 단구간 포스트필터(Short-term postfilter) 그리고 출력 이득 증폭(Output gain scaling) 과정으로 이루어져 있다.

2.2 고정 소수점 C 소스 코드 구현

음성이나 영상 같은 디지털 신호를 고속으로 처리하기 위해서는 DSP chip을 사용하여야 한다. 일반적으로 DSP chip을 사용하는 이유는 다음과 같다. 첫째로 MAC (Multiplication and Accumulation) 유닛을 하드웨어로 내장하고 있으므로 디지털로 신호를 처리할 때 자주 사용되는 MAC 연산을 한 cycle에 처리할 수 있다. 그래서 행렬 연산이나 sum-of-product 연산을 고속으로 처리할 수 있다. 두 번째로 데이터와 인스트럭션을 동시에 가져다 처리하기 위해 데이터 메모리와 인스트럭션 메모리(instruction memory)를 따로 가지는 하바드 구조를 채용하여 두개 이상의 operand를 동시에 가져올 수 있는 다중 버스구조를 가진다. 그리고 반복적인 연산을 빠르게 처리하기 위해 loop control, circular buffer control 기능을 하드웨어적으로 처리한다. Vincent6는

고정 소수점 DSP 코어이며 고정 소수점 C 소스 코드 구현의 목적은 고정 소수점 DSP에 원하는 알고리즘을 구현하기 위해서이며, 고정 소수점 DSP는 부동 소수점 DSP에 비해 저가이므로 같은 성능을 가진 시스템을 저렴하게 구성할 수 있다. 고정 소수점 DSP에 실시간으로 구현을 하기 위해서는 원하는 DSP 구조를 고려한 integer 시뮬레이션이 선행되어야 한다. 이를 위해서 ITU-T의 G.728 Annex G[5]를 이용하였다. ITU-T G.728 Annex G는 94년 11월에 발표되었고 고정 소수점 G.728에 대한 스펙이며 수도 코드(pseudo code)를 포함하고 있다. G.728의 고정 소수점 C 소스 코드와 부동 소수점 C 소스 코드는 다음과 같은 차이가 있다.

- (1) 변수의 표현과 연산을 Q 포맷을 이용한다.
- (2) 수학적으로 동일한 backward vector gain adaptation을 사용한다.
- (3) Levinson-Durbin recursion 모듈에서 예측기의 계수를 고정 소수점 구조에 맞게 변경한다.

G.728 Annex G의 테스트 벡터는 G.728 Appendix I에 자세히 나타나 있다.

3. ITU-T G.728 음성 부호화기의 실시간 구현

3.1 VINCENT6 DSP의 기능 분석

Vincent6 코어는 VLIW 멀티미디어 DSP 기능에 범용 마이크로 컨트롤러 기능을 결합한 구조로, 고성능 / 작은 코드크기 / 고급 언어 지원 / RTOS (Real Time Operating System) 지원 / ASIC을 위한 유연한 확장구조를 만족시키도록 구성된다.

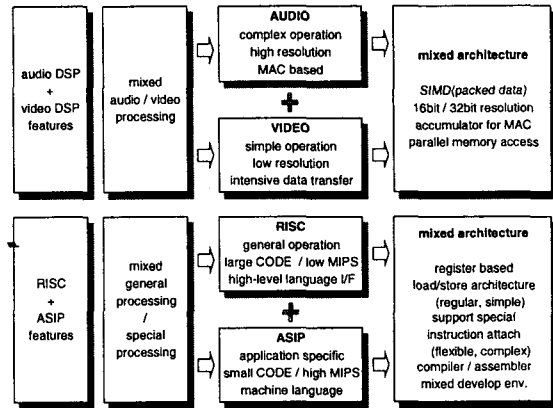


그림 3. Vincent6 DSP 코어의 특징

동시에 2개의 명령이 수행될 수 있으며, 두 개의 명령의 수행여부는 컴파일 시에 결정되며, 진보된 명령어 프리픽스 포맷 (prefix format)을 채용하여 32 비트, 48 비트, 64 비트, 80 비트 그리고 96 비트의 가변 길이 명령어 포맷을 지니며, 최

대로 한 사이클에 2개의 비 메모리 명령과 4개의 메모리 명령을 수행할 수 있다. 그리고 총 32개의 32비트 레지스터를 가지고 있으며, 2개의 64비트 어큐뮬레이터를 제공하고, 8/16/32/64비트 크기의 데이터 처리가 가능하다. 효과적인 코딩을 위한 6스테이지(stage) 파이프라인을 제공하며, delayed branch 기능이 있고 SIMD (Single Instruction Multiple Data) 명령을 이용해서 코드를 줄일 수도 있다.

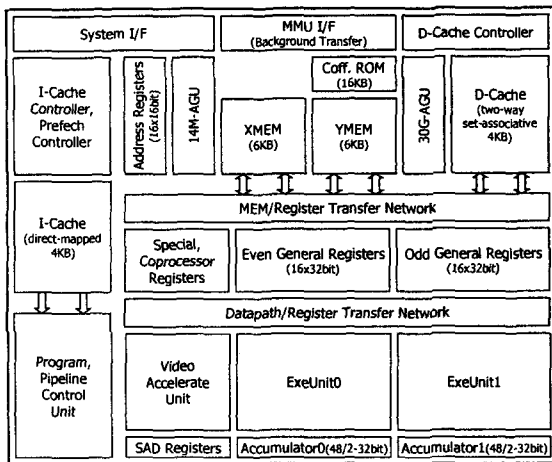


그림 4. Vincent6 DSP 코어의 구조

Vincent6에서는 2가지 형태의 메모리를 사용한다. 첫번째는 일반적인 마이크로프로세서 구조에서 사용되는 데이터 메모리로, SDRAM 영역을 어드레스 공간으로 하며, 내부적으로는 4KByte의 direct mapped cache를 사용한다. 캐쉬관리는 하드웨어에서 자동적으로 처리되므로, 데이터의 지역성이 보장되는 일반적이 프로그램에 적합하다. 두번째 타입의 메모리는 일반적인 DSP 구조[6]에서 주로 사용되는 내부 메모리 (internal memory)에 해당한다. Vincent6에서는 DSP에서 발생하는 데이터 병목현상을 효과적으로 처리하기 위해, 두개의 내부 메모리 (X, Y 메모리)를 사용한다. 각각은 2개씩의 read/write 포트를 제공하기 때문에, 동시에 4개의 메모리 access가 가능하다. X, Y 메모리는 사용자의 프로그램에 의해 처리되므로, 소프트웨어 캐쉬와 같이 동작시킬 수 있어 cache의 성능을 극대화할 수 있다. 그리고, X, Y 메모리는 Vincent6에서 제공되는 병렬 메모리 명령어로 접근할 수 있기 때문에, 빈번한 접근을 요구하는 DSP 커널을 매우 효과적으로 처리할 수 있다. Vincent6에서는 데이터 메모리와 X, Y 메모리에 대한 어드레싱 방식이 다르다. 기본적으로 데이터 메모리에 대해서는 일반적인 RISC 구조에서 지원되는 linear 어드레싱만을 제공한다. 주소계산은 general 레지스터에 의해 이루어지고, base 레지스터와 offset 레지스터나 offset immediate의 연산으로 계산된다. 그리고, base 레지스터의 값이 자동적으로 갱신되는 것을 지원한다. 그리고 데이터 메모리에 대한 접근은 30G load/store 명령만으로 처리된다. X, Y 메모리에 대한 어드레싱은 주로 DSP 기능에서 사용되기 때

문에, 복잡한 어드레싱 방식을 지원한다. 크게 30G load/store 명령을 사용하는 경우와 14M 메모리 명령을 사용하는 경우로 나눌 수 있다. 30G load/store 명령은 앞의 데이터 메모리와 접근하는 방식이 동일하며, 단 정렬 방식에서 차이가 있다. 데이터 메모리는 byte addressing이 가능하지만, word로 접근할 경우에는 word 정렬에 맞추어야 한다. 그러나, X, Y 메모리는 byte addressing은 불가능하지만, word로 접근할 경우에는 half-word 정렬을 지원한다.

3.2 G.728 음성 부호화기의 실시간 구현

G.728의 실시간 구현은 효율적인 코딩을 위해서 C 코드에서 계산량이 많은 부분만 어셈블리 코드로 작성한 후 이 부분을 C에서 호출하는 방식을 사용한다. 먼저 VCC (Vincent6 C Compiler)를 이용해서 크로스 컴파일을 한 후 시뮬레이터 (simulator)를 이용하여 각 모듈별로 instruction level의 계산량을 구할 수 있다. 이 결과를 보고 계산량이 많은 모듈을 중점적으로 코딩을 해주면 짧은 시간에 많은 최적화를 할 수 있으며 코드개발 기간을 단축시킬 수 있다. 어셈블리 코딩을 할 때 사용한 최적화 기법은 다음과 같다. 첫째로 자주 사용되는 데이터를 X, Y 메모리에 올리는 방법이다. 이렇게 데이터를 X, Y 메모리에 올리면 한 사이클에 4개의 메모리 명령어를 사용할 수 있는데 16비트로 정렬된 데이터를 불러오면서 SIMD 명령어를 이용하면 동시에 8개의 데이터를 불러오면서 4개의 MAC 연산을 수행할 수 있다. 그러므로 음성 코덱에서 사용하는 프레임의 샘플 개수가 8의 배수인 경우 최대의 성능을 제공할 수 있다. 하지만 G.728은 프레임의 크기가 5 샘플이므로 많은 제약이 있다. 그러므로 강력한 데이터의 load/store 기능을 최대한 활용하기 위해서는 결과값이 변하지 않는 범위내에서 C 소스코드의 변경이 요구된다. 아래 그림은 가장 많은 계산량을 가지는 여기신호 코드북의 탐색 과정이다.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

(a) Original codebook table

1	6	11	16
2	7	12	17
3	8	13	18
4	9	14	19
5	10	15	20

(b) Modified codebook table

그림 5. codebook table의 구조 변경

원래의 코드북은 그림 5에서 (a)와 같이 5개의 디멘션 (dimension)을 가지는 128개의 코드북이다. 그중에서 제일 상단의 4개의 코드북을 그림 5의 (a)로 나타냈는데 여기서 순

서대로 데이터를 읽어오는 경우에는 4개의 데이터를 읽고 나면 하나의 데이터가 남으므로 X,Y 메모리에 올려도 많은 계산량 감소를 기대할 수 없다. 그러나 이를 그림 5의 (b)와 같이 구성하면서 파이프라인을 이용하면 데이터를 8의 배수로 읽을 수 있으므로 4개의 벡터를 고속으로 탐색할 수 있게 된다. 두번째로는 다양한 DSP 인트린직 명령어를 C 코드에서 직접 사용하는 것이다. 이를 이용하면 overflow나 underflow가 발생하는 basic operation을 C 코드에서 간단하게 처리할 수 있다. 세번째는 loop unrolling 기법을 이용하는 것이다. 복잡한 loop를 unrolling한 후 코드의 패턴을 파악하고 Vincent6의 장점이라고 할 수 있는 풍부한 레지스터를 효과적으로 사용해서 파이프라인을 구성하면 계산량을 많이 줄일 수 있다.

최종적으로 구현된 코드는 ITU-T에서 제공하는 테스트 벡터를 적용하여 검증하였다. 적용결과 모든 벡터에 대해 비트 단위로 동일한 결과를 얻을 수 있었다. 표 1은 G.728을 VCC를 이용하여 크로스 컴파일한 결과와 C 언어와 어셈블리어를 동시에 이용하여 구현한 결과의 계산량을 비교하였다. 구현결과 부호화기와 복호화기를 합해서 총 34 MCPS (Million Instruction Per Second)의 성능을 가지는데 이는 C 언어만을 이용한 결과보다 약 1/13의 복잡도를 가지는 것을 알 수 있다. 표 2는 G.728에 사용된 메모리의 크기를 나타내었다. X,Y 메모리와 데이터 메모리를 합한 전체 데이터에 사용되는 메모리가 약 9 KByte 사용되었으며 프로그램 메모리가 약 57 KByte 사용되었다.

표 1. G.728의 MCPS 성능

	부호화기	복호화기	계
C	244.5	197.1	441.6
C+Assembly	19.5	14.5	34

표 2. G.728에 사용된 메모리(Byte)

	X memory	Y memory	Data memory	Program Memory
Encoder	1592	584	160	
Decoder	1960	998	32	
Common	0	3224	316	
계(Byte)	3552	4806	508	57300

4. 결 론

본 논문에서는 32비트 고정 소수점 DSP 코어인 Vincent6를 이용하여 ITU-T G.728 음성 부호화기를 실시간으로 구현을 하였다. 구현된 음성 부호화기는 부호화기가 19.5 MCPS, 복호화기가 14.5 MCPS의 성능을 가지며 필요한 메모리크기는 데이터 메모리가 약 9 KByte, 프로그램 메모리가 약 57 KByte이다. C 언어와 어셈블리어로 구현된 코드는 ITU-T의

테스트 벡터를 완벽하게 통과하였다. 개발된 음성 부호화기는 영상 코덱과 함께 H.320 영상회의 응용으로 사용될 수 있으며, 낮은 지연을 가지는 ISDN용 전화나 H.323 프로토콜을 이용한 VoIP (Voice over Internet Protocol)에도 사용가능할 것으로 예상된다.

감사의 글

본 연구는 정보통신부 출연 "저가형 멀티미디어 통신단말 핵심기술 개발"과제의 일환으로 수행되었습니다.

참고 문헌

- [1] J.-M. Kim, Y.-S. Shin, I.-G. Hwang, K.-M. Lee, K.S. Oh, K.-S. Lee, S.-I.Han, and S.-I. Chae, "High-performance videophone chip with dual multimedia VLIW processor cores," in *Proc. COOL Chips III*, pp.75-88, Apr.2000.
- [2] CCITT Recommendation G.728, "Coding of speech at 16kb/s using low-delay code excited linear prediction," Sep. 1992.
- [3] Chen J-H, Cox RV, Lin Y-C, Jayant N, Melchner MJ, "A low-delay CELP coder for the CCITT 16 kb/s speech coding standard," *IEEE Journal on Selected Areas in Communications*, Vol.10,no.5, June 1992, pp.830-49.USA
- [4] B.S. Atal "Predictive coding of speech signals at low bit rates," *IEEE Trans. Comm.*, Vol. 30, No.4, 1982, pp. 600-614.
- [5] ITU-T Recommendation G.728-Annex G, "Coding of speech at 16kb/s using low-delay code excited linear prediction Annex G:16kbit/s fixed point specification," Nov. 1994.
- [6] J. Eyre, and J. Bier, "The evolution of DSP processors," *IEEE Siganal Processing Mag.*, pp.230-231, Feb. 2000.