

# 음성부호 처리에 적합한 RISC 프로세서의 설계 및 구현

김진, 이준용  
홍익대학교 컴퓨터공학과  
(jinkim, jlee)@cs.hongik.ac.kr

## Design and Implementation of RISC Processor for Speech Coding

Jin Kim, Jun Yong Lee  
Dept. of Computer Engineering, Hong Ik University

### 요약

디지털 음성통신을 위한 빠르고 쉬운 내장 프로세서(Embedded processor)가 요구되어짐에 따라 음성신호 압축 복원 알고리즘인 ADPCM과 LD-CELP의 구현에 가장 빈번히 사용되는 연산의 특성을 조사하였다. ARM6 processor core의 기본 구성요소들과 명령어 집합을 기반으로 하여 음성부호화 알고리즘의 연산의 특성을 효율적으로 처리하기 위한 명령어와 구조를 추가한 범용 프로세서의 구조를 제안하고 VHDL로 기술하여 동작을 검증하였다. ARM6의 ALU logic에 leading zero count를 위한 회로를 추가하였고 opcode를 변경하였으며, LPC 계수 연산을 위해 제안된 MAC을 도입하여 효율적인 구현이 가능하도록 설계하였다.

### 1. 서론

음성통신은 모든 정보통신의 기본 미디어이다. 음성통신을 위한 음성정보는 기존의 아날로그방식에서 디지털방식으로 변환되어 처리된다. 디지털 음성 신호의 통화 품질을 유지하면서 전송용량을 줄이기 위한 다양한 음성신호압축 복원 알고리즘들이 연구되었고, 구현을 위해 많은 하드웨어 Codec들이 개발되어 사용되고 있다.

시스템 디자인 시, 음성부호화(speech coding) 알고리즘을 구현한 전용 Codec(COder-DECoder)을 사용할 경우 chip의 효율성은 높일 수 있지만 다른 알고리즘의 응용에 유연(flexible)하지 못한 단점이 있다. 또한 사용자 인터페이스와 기능들을 컨트롤하기 위해 RISC microprocessor가 함께 사용될 때, 이것은 RISC microprocessor의 기능을 극히 적게 활용하는 것이다. DSP를 제거하면 시스템 가격의 이득을 얻을 수 있다[4]. 또는 범용 프로세서에 원하는 알고리즘을 프로그래밍 하여 소프트웨어 프로세서를 구현하는 방법이나 programmable application-specific processor를 사용할 수 있다. 범용 프로세서를 사용하는 경우 해당 프로세서의 구조가 알고리즘을 효율적으로 구현할 수 있어야 execution time constraint를 만족할 수 있다. [5]에서 나타낸 바와 같이 ADPCM을 구현하는데 ADSP2100에서는 encode/decode pair

of operation 하나를 수행하는데 900 clock cycle 미만을 필요로 하는 반면, AT&T DSP16은 최악의 경우 거의 2500 clock cycle이 필요하다. 이는 빈번히 사용되는 연산을 효율적으로 제공하느냐 그렇지 못하느냐에 따른 것이다.

본 연구에서는 음성부호화 알고리즘의 특성을 분석하여 real-time codec 구현을 제공하는 범용 RISC 프로세서 구조를 생성하였다. 32 kbps G.726 ADPCM 만을 고려하여 다양한 shift 연산과 leading zero count 연산만을 효율적으로 제공한 기존의 것에[5] 추가하여, 현재 가장 많이 쓰이는 압축 방식인 CELP의 변형 알고리즘들도 효율적으로 지원할 수 있도록 그 특성을 고려하여 ARM6 processor core를 기반으로 프로세서의 구조를 설계하였다. 그리고 VHDL을 이용하여 구현하였고 동작을 simulation하였다.

본 논문의 구성은 2장에서는 음성부호화 알고리즘에 대한 특성과 처리방법에 대하여 살펴보고, 3장에서는 구조와 명령어에 대하여 다룬다. 4장은 구현과정과 결과를 평가하며 마지막으로 5장에서는 결론을 논의한다.

### 2. 관련연구

음성코딩은 음성 파형을 직접 양자화하는 파형 코딩(Waveform coding)과 음성 신호로부터 특성계수를 분석하고

합성하는 음원 코딩(Source coding, Vocoder), 그리고 두 방법의 장점을 조합한 복합형 코딩(Hybrid coding)으로 분류된다. 이 중 파형 코딩인 ADPCM과 복합형 코딩인 CELP 방식에서 파생된 음성 압축 알고리즘이 많이 사용된다[9].

**2.1 ADPCM(Adaptive Differential Pulse Code Modulation)**

ITU-T에서 G.726 ADPCM 표준안으로 지정된 압축 방식이다. [1]에서 ADPCM Ecoder/Decoder의 block diagram을 볼 수 있다. 부호기에서는 uniform PCM 입력신호와 적응 예측기(Adaptive predictor) 출력신호의 차이를 다이내믹 록킹 양자화기(Dynamic locking quantizer)를 통하여 샘플 당 4bit로 양자화 하여 복호기 측에 보낸다. 복호기는 수신한 신호를 역 양자화기를 통하여 차이를 재생하고 이것에 예측기의 출력 신호를 더하여 재생 입력 신호를 얻는다. 부호기와 복호기의 예측 값을 같게 하기 위하여 부호기에도 복호화 조작을 한다[9].

ADPCM을 각 sub-block 별로 실행시간의 비율을 측정하면 fmult가 28%, power2\_exp가 13.8 %로 가장 빈번히 실행되는 부분임을 알 수 있다. sub-block들은 부동소수점연산, 지수연산, 로그연산 등으로 명시되어있는데, 알고리즘에 필요한 연산들은 몇 개의 부류로 나눌 수 있으며, 일반적인 논리연산, 산술연산 외에 수의 범위측정, 다양한 쉬프트 연산, 수의 형태변환 등이 빠르고 효과적으로 제공해야함을 알 수 있다[1][3][4].

**2.2 LD-CELP(Low-Delay Code Excited Linear Prediction)**

ITU-T G.728로 표준화된 16kb/s 알고리즘이다. 합성에 의한 분석(Analysis by synthesis) 원리에 입각하여 음성 파형의 왜곡을 작게 만든 부호화로서, 음성 입력 5샘플을 1 프레임으로 하여 프레임마다 여기신호 이득(excitation gain)을 전송하는 것으로 LPC(Linear Prediction Code) predictor와 여기신호 이득에 필요한 지연이 2ms 이내이다. 그림 1은 LD-CELP coder의 간단한 block diagram이다[10]. 부호기는 코드북(codebook)중의 여기 벡터(excitation vector)를 모두 조합하여 음성을 합성하며 그 결과를 원음성과 비교하여 정각적 하중거리가 최소가 되도록 선택하여 전송한다. 합성기는 여기 벡터에 후방 적응형 이득을 곱하고 다시 후방 예측(backward prediction)에서 구해진 선형 예측합성 필터를 거쳐 음성을 얻어낸다[9].

LD-CELP 알고리즘은 ADPCM보다 전송량이 적으나 많은 연산량을 필요로 한다. 전체 변수 계산 시간 중 70-80%를 피치(pitch)와 여기 코드북 매개변수 검색 과정이 차지하는데, Analysis by synthesis 검색법을 사용하므로 원음과 합성음과의 차이에 대한 에러를 최소화하는 과정에서 변수처리가 중요하다[7]. 이는 곱의 합으로 구성된 루프연산을 기본 연산 단위로 한다. 또한 메모리 접근의 부하가 크다. 효과적인 처리를 위한 곱셈 누적기(Multiplier & accumulator)를 제공하고 메모리 접근의 부하를 줄일 수 있는 구조가 제공되어야 한다.

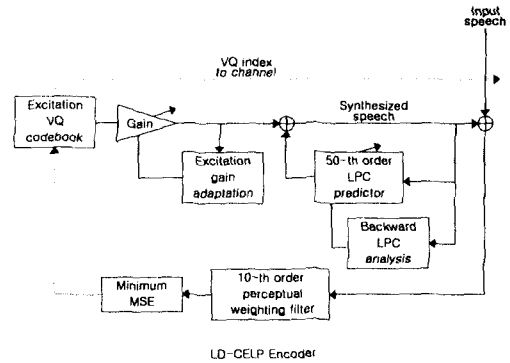


그림 1. LD-CELP의 구조

**3. Architecture**

모델링한 RISC 프로세서 구조는 ARM6 CPU core 구조를 기반으로 하였다[11]. pipeline 구조를 가지며 data buffer, 명령어 해석기, 레지스터뱅크, booth 곱셈기, barrel 쉬프트, ALU, 주소 레지스터와 로직의 주요 기능 블록들을 가진다.

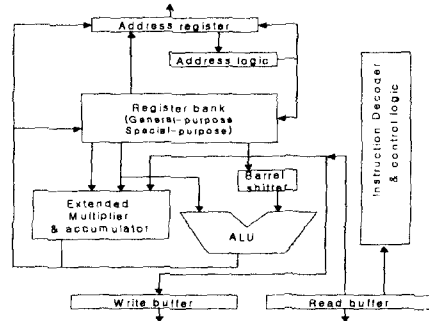


그림 2. 프로세서의 구조

Barrel 쉬프트로 다양한 쉬프트연산을 제한 없이 제공함으로써 ADPCM 알고리즘 구현 시 필요한 많은 쉬프트연산을 cycle의 추가 없이 효율적으로 수행할 수 있다.

ALU는 기본 산술연산과 논리연산, 비교연산 등의 기능을 가지고 있다. ARM의 ALU logic에 leading zero count를 위한 logic을 추가하였다. 이 연산을 지연 없이 한 cycle내에 수행하기 위해서 전체 bit의 비교연산 후 추가적인 shift 연산을 수행하는 방법대신 직접 gate level로 구현하였다. 각 bit에 비교기를 연결하는 것 보다 적은 수의 gate로 구현이 가능하다. 4 bit leading zero 회로를 확장하여 8 bit로 만들 수 있다. 앞의 4 bit가 모두 0일 때 즉, 출력이 100일 때만 뒤의 4 bit까지 고려하여 leading zero를 count한다. 마찬가지로 16 bit, 32 bit로 확장할 수 있다. 그림 3에 8 bit일 때의 회로와 부울식을 나타내었다. 그리고 연산의 decode를 위해서 ARM6 integer instruction set의 format에 따라 opcode를 새로 정의하였다. 그림 4에서 명령어의 형식을 볼 수 있다.

LD-CELP 알고리즘에 더 적합한 곱셈연산을 수행하기 위해 2의 보수곱셈을 수행하던 Booth 곱셈기를 [7]에

서 제안된 MACU(Multiply Accumulate Unit)을 도입하여 확장하였다. 곱셈과 곱셈누적연산을 수행하며 메모리 접근의 부하를 줄이기 위한 CSU(Compare Select Unit)이 포함되어있으며, 일반 곱셈 연산과 구별하기 위하여 instruction의 두 번째 필드의 값이 00이 아닌 값으로 정의하여 control 신호를 구별하였다.

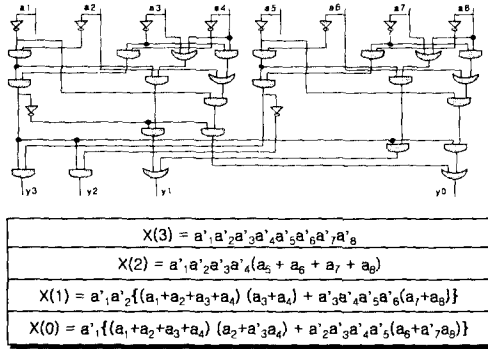


그림 3. leading zero count 회로와 부울식

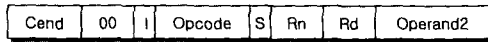


그림 4. ARM 기본 명령어 형태

4. 실험결과

본 연구에서 제안한 범용 프로세서의 구조의 동작을 확인하기 위해 VHDL 언어를 사용하여 동작레벨에서 기술하였으며 Altera의 MAX+plus2를 사용하여 시뮬레이션 하였다.

추가한 leading zero count회로와 MAC의 올바른 동작이 검증된 후 변경된 instruction decode와 control을 위해 전체적인 RISC 프로세서의 동작을 시뮬레이션 하였다.

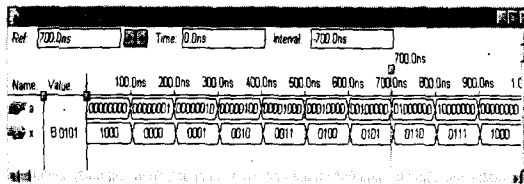


그림 5. simulation 결과

5. 결론 및 향후과제

본 연구에서는 음성부호화 알고리즘을 효율적으로 처리할 수 있는 범용 프로세서의 개선된 구조를 제안하였다. 가장 많이 쓰이는 ADPCM과 LD-CELP 부호화 표준 알고리즘의 연산의 특성을 효율적으로 처리하기 위해 새로운 명령어와 구조를 추가하였다.

ARM6 processor core의 기본 구성요소들을 기반으

로 하여 범용 프로세서의 구조를 제안하였고 동작을 검증하였다.

음성부호화의 구현에 범용프로세서를 사용함으로써 가격의 경쟁력을 살리고 프로그래밍이 가능하므로 여러 알고리즘에 유연성을 제공할 수 있다.

동작적으로 검증된 구조는 향후 구조적인 구현이 진행될 것이며 실제적인 음성부호화의 구현에 적용하여 직접적인 성능의 평가를 필요로 한다.

6. 참고 문헌

- [1] Recommendation G.726 : 40, 32, 24, 16kbit/s Adaptive Differential Pulse Code Modulation (ADPCM), CCITT, Geneva, 1990
- [2] Recommendation G.728 : Coding of Speech at 16kbit/s Using Low-Delay Code Excited Linear Prediction (LD-CELP), CCITT, Geneva, 1992
- [3] J. G. Bartkowiak, M. Nix, S. M. Asghar, "A Digital Signal Processing (DSP) Architecture for Adaptive Differential Pulse Code Modulation (ADPCM) Transcoders" *ICSPAT-Volume 1*, 1997
- [4] Anil Sharma, C. P. Ravikumar, "Efficient Implement of ADPCM Codec" *IEEE Computer Society*, 2000
- [5] Juhani Vehvilainen, Jari Nurmi, "A Process Core for 32kbit/s G.726 ADPCM Codecs" *ISCAS-Volume 3*, 1995
- [6] Alex van Someren, Carol Attack, "The ARM RISC Chip: A Programmer's guide" 1993
- [7] 박주현, 심재술, 김영민, "Dual MAC을 이용한 음성 부호화기용 피치 매개변수 검색 구조 설계" *전자공학회논문집, 제33권 A편 제5호*, pp. 172-179, 1996
- [8] D. P. Noel, T. A. Kwasniewski, S. Mahmoud, W. P. Leblanc, "A Low power, single chip realization of a low-speed, low-delay CELP coder/decoder for indoor wireless systems" *IEEE VTC-Volume 1*, 1997
- [9] 이문호, 이광재 편저. "멀티미디어/하이퍼미디어의 부호화", 대영사
- [10] Juin-Hwey Chen, Melvin J. Melchner, Richard V. Cox, Duane O. Bowker, "Real-Time implementation and performance of a 16kb/s low-delay CELP speech coder" *ICASSP'90*, 1990