

Dual MAC을 이용한 음성 부호화기용 DSP Core 설계에 관한 연구

**박주현, *김영민, **최승호
*전남대학교 전자공학과
**동신대학교 정보통신공학과

Design of a dedicated DSP core for speech coder using dual MACs

**Park Ju Hyun, *Kim Young Min, **Choi Seung Ho
* Dept. of Electronic Eng., Chonnam Nat'l Univ.
** Dept. of Information and Telecommunication Eng., Dongshin Univ.

Abstract

In the paper, CDMA(Code Division Multiple Access)'s vocoder algorithm, QCELP(Qualcom Code Excited Linear Predictive), was analyzed. And, 16-bit programmable DSP(Digital Signal Processor) core for QCELP was designed. When it is used two MACs in DSP, we can implement low-power DSP and estimate decrease of parameter computation speed. Also, we implemented in FIFO(First-In First-Out) memory using register file to increase the access time of the data. This DSP was designed using logic synthesis tool, COMPASS, by top-down design methodology. Therefore, it is possible to cope with rapid change at mobile communication market.

I. 서론

현재 이동통신 시스템의 기술 발전은 가입자 수용량을 크게 하기 위해 디지털 통신 방식으로의 전환과 아울러 여러 가지 다원접속방식에 관한 연구가 활발히 이루어지고 있다. 이에 맞맞추어 최근 정보 전달의 용이성, 신속성, 편리성을 제공하는 휴대용 이동 통신에 대한 수요도 폭발적으로 일어나고 있으며, 이에 부응하여 세계 각국에서는 다양한 정보 처리, 고품질 음성 송수신, 다양한 기능 등을 사용자의 요구에 따라 제공하기 위해 기존의 아날로그 방식을 지체된, 다채널 용량, 잡음 면역성, 보안성 등의 구현이 쉬운 디지털 방식으로 전환해 가고 있다. 이동통신기가 디지털화 되어감수록 선전 외국업체들은 고품질, 고기능을 통제할 수 있는 일반용 CPU와 ASDSP(Application Specific Digital Signal Processor) 칩의 개발에 박차를 가하고 있으며, 또한 다변화하는 시장에 대처하기 위해 프로그래머블 DSP를 개발하고 주변 회로를 라이브러리화하여 공급하는 시장에 보다 능동적이고 신속하게 대처하고 있다. 이는 각종 통신기기 특히 휴대용 통신기기에 대한 소형화 경향과 저전력화 등의 신호 처리 요구사항에 유연하게 대처하기 위한 것이다.

본 논문에서는 고품질 음성신호를 처리하기 위해 현재 국제 표준 방식인 CDMA방식의 휴대용 단말기에서 사용하는 QCELP 부호화기 알고리즘 분석을 바탕으로 이것을 효과적으로 수행할 수 있는 프로그래머블 ASDSP의 코어 설계에 관한 연구이다.^[1] 계산 시간이 많이 소요되는 음성 부호화기는 Codebook, Pitch search, LPC Analysis-by-Synthesis, Frame Energy, Autocorrelation matrix, 다수의 디지털 필터와 모뎀으로 이루어져 있다. 특히 Analysis-by-Synthesis에서 매개 변수를 탐색하는 과정에서는 multiply-accumulate 연산이 대량으로 요구되며,

전체 연산의 대부분을 차지하고 있다. 따라서 변수의 고속 검색과 계산이 가능하도록 ASDSP로 구현하는 것이 중요하다.^[2]

따라서 본 논문은 휴대용 이동 통신 기기의 저전력성을 보장하기 위해 dual MAC을 사용하며, MAC 기능의 효율을 높이고, 데이터 검색 속도와 입출력 속도를 높이기 위한 방법으로 고속 FIFO(First-In First-Out)와 3개의 독립된 데이터 버스 구조를 채용한다. 또한 연산의 바이프라인 기능을 지원하기 위해 바이프라인이 가능한 레지스터 파일을 사용하며, 데이터패스 부분은 16 x 16 곱셈과 32비트 덧셈 수행, 16비트 BS(Barrel Shifter) 사용, 1개의 32비트 누산기(Accumulator) 등으로 구성된다.

본 논문의 구성은 II장에서 음성부호화기 알고리즘의 특징과 효과적인 DSP 구현을 위한 프로세싱에 대한 기술에 대해 살펴보고, III장에서는 음성 부호화기 알고리즘을 최적으로 구현해 낼 수 있는 어드레싱, 명령어와 DSP core 구조를 살펴본다. IV장에서는 본 논문에서 설계한 DSP의 실험 결과와 성능을 평가하며, 마지막으로 V장에서 결론을 내린다.

II. 알고리즘의 특징과 효과적인 구조화 프로세싱 방법

1. 음성 부호화기 알고리즘의 특징

이제까지 개발된 음성 부호화기는 파형 부호화(waveform coding), vocoder, 혼합형(hybrid coder)으로 분류할 수 있다. 이 중에서 혼합형은 LPC(linear prediction code) 코딩을 이용한 분석-합성 부호화기 방식(analysis by synthesis coding technique)으로 오차 신호가 최소가 되도록 exciter에 대한 정보를 파형 부호화 또는 대치 부호화하여 전송하는 방식이다. 그 중에서 대표적인 방식인 CELP(Code Excited Linear Prediction)는 N 개의 서로 다른 벡터 집합(코드 벡터)로 코드북을 구성하여, 매 부프레임마다 각각의 벡터 집합에 대해 합성 필터를 통과시켜 원음과의 오차가 최소가 되는 코드북의 code word와 gain을 구하여 전송함으로써 전송 정보의 양을 줄이고, 수신측에서는 이 코드 워드들을 이용하여 해당 벡터 집합을 읽는다. 그리고 이것을 구동 신호로 이용하여 합성을 하게 된다.^[3]

CELP의 코드북 검색에는 많은 계산량이 요구된다. 물론 코드북의 크기 N을 줄이면 계산량이 줄어든다. 음절이 지워진다. 따라서 계산량을 줄이기 위한 복수 알고리즘 개발이 필요하다 하겠다. 실제로 QCELP 알고리즘을 수행할 때 계산량의 70~80%는 베투프 방식의 피치 및 코드북 파라미터 검색에 사용된다.^[4]

CELP 알고리즘은 AT&T의 CELP와 유사한 구조로서 음성 부호화된 출력의 비트 속도가 8Kbps, 4Kbps, 2Kbps, 1Kbps로 가변되는 것이 가장 큰 특징이다. 즉 음성 신호가 있을 때는 8Kbps(160 샘플/20msec)의 비트 속도가 되며, 음성 신호가 존

1. 전체 구조 및 블록도

본 DSP는 이동 통신의 음성 코딩 환경에 적합하도록 설계된 DSP 프로세서이다. Execution part는 전체 시스템의 마이크로 인종 최종 단계로 source 값들이 중간 레지스터에 load되는 과정으로 마무리 된다. DSP core의 전체 블록도 그림 3과 같다.

음성 코딩 알고리즘을 수행하기 위한 본 DSP는 크게 DАLU(Data ALU)와 MAC 동작을 효율적으로 하기 위한 MACU(MAC Unit)으로 나누어진다.

DALU: 16비트 ALU와 IS, 레지스터 파일로 구성되어 있으며, MACU는 2개의 MAC과 3개의 FIFO로 구성되어 있다.

또한 기 상자를 제어할 수 있는 신호를 메모리는 제어기는 PLA로 구성되어 있다.

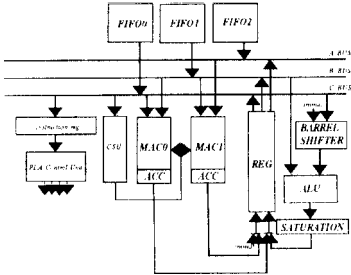


그림 3. DSP Core 블록도

데이터 ALU: 본 DSP의 MAC 연산과 논리 동작을 수행한다. 데이터 ALU의 연산은 2의 보수의 입력 데이터를 수행하고, 2의 보수 결과를 낸다. 데이터 ALU의 구성은 16비트 Barrel Shifter, 16 x 16비트 레지스터 파일, 16비트 논리 연산 장치로 구성된다. MACU(Multiplier-Accumulation Unit): 303 x 16비트 FIFO 1개, 160 x 16비트 FIFO 2개, MAC 2개, 32비트 마이크로파일 레지스터, 32비트 누적레지스터, CSU(Compare-Select Unit)로 구성된다.

데이터 FIFO, F0, F1, F2는 각각 160, 160, 303 개의 16비트 데이터 레지스터로 구성되어 있다. F0, F1, F2가 2개의 MAC 입력으로 사용되며, F0:F1, F1:F2를 결합하여 다시 FIFO에 저장하거나 레지스터 파일에 입력된다. 가변용 음성 코딩 알고리즘을 사용하기 때문에 입력 음성 신호를 8Khz로 샘플링 하여 이 음성을 한 프레임당 160 샘플씩 20ms음성 부호화기 프레임으로 나올 경우 최소 160개 FIFO가 필요하며, 피치 맞추기 검색을 위한 분석/합성(Analysis-by-synthesis) 과정에서 계층분와 계층복 과정의 변수를 찾을 때 143개의 계층복 값을 더 필요로 하기 때문에 303개의 FIFO가 하나 더 필요하다. 먼저 계산은 Up/Down 카운터를 이용한다.^[6] 한 클럭에 한 데이터 쌍 처리가 가능하고, 동시에 읽고 쓸 수 있기 때문에 변수 전체 계산 속도가 200% 빨라진다. 또한 모든 데이터가 동시에 이동하기 때문에 11/W 프로그램에서 내부적인 연이 많이 발생할 수 있으나 MAC을 두 개 사용함으로써 단일 장치의 이용도를 떨어뜨리기 때문에 시선력을 필요로 하는 이동통신용 DSP에서 사용가능하도록 한다. 또한 가변용 음성 코딩 방법을 쓰기 때문에 음성 신호의 에너지에 따라 부프레임의 크기가 160 이하가 되므로 전체 음성 코딩이 이루어질 경우 FIFO의 전체 이용도는 100%가 되지 않기 때문에 선력 문제는 극복되리라 본다.

MAC은 DSP의 중요한 산출 처리 장치이며, 모든 산출 연산

을 수행한다. 이 장치는 곱셈기, 덧셈기 블록으로 구성된다.^[7] COMPASS 라이브러리에서 제공하는 곱셈기와 덧셈기를 이용하여 설계하였다. 또한 곱셈기와 덧셈기 사이에 마이크로파일 레지스터를 두어 마이크로파일이 가능하도록 하였다. 이 곱셈기는 MAC 동작 이외에 일반 곱셈 연산도 가능하도록 다중화되어 있다.

음성 코딩 알고리즘에서는 많은 MAC 동작을 수행한다. 그러나 때때로 메모리에 접속하여 MAC 명령어를 가져온 경우 많은 시간 지연이 발생하여 음성 신호의 실시간 처리가 어렵게 된다. 따라서 한 번의 명령어로 여러 번의 동작을 반복해서 하기 위해 레지스터에 반복 횟수만을 저장하고 나서 그 값을 CSU 입력으로 한 경우 레지스터 값만큼 MAC 동작을 하게 한다. 카운터에서 클럭의 회수를 세고, 그 값이 메모리에서 입력된 16비트의 반복 회수와 비교되어 같아지면 때까지 MACU를 load하게 된다. 또한 MAC의 곱셈기를 일반 곱셈 연산으로 이용하기 위해 CSU에 다중화 신호를 입력하여 MAC과 곱셈을 구별하여 연산 장치로 동작하도록 한다.^[8]

제어기는 Execution part를 제어하는 신호들을 발생시키는 블록으로 PLA를 이용하여 설계하였다. 명령어 워드의 오퍼코드 필드를 해석하여 명령이 수행에 요구되는 신호들을 발생시킨다. 먼저 명령이 오퍼코드를 간략화 한 후 PLA 명령식을 이용하여 COMPASS에서 제공하는 합성 기능을 이용해 고속, 최적의 PLA 블록을 설계하였다. 명령이 레지스터에 명령어 워드가 load 되면서 5ns 이내에 제어 신호를 발생한다.

2. 어드레싱과 명령어 세트

본 논문에서는 음성 부호화기용 DSP를 효율적으로 구동시키기 위한 어드레싱과 명령어 세트를 제안하였다. 어드레싱은 최대한 동작을 단순화시키고, 레지스터를 통한 마이크로파일이 가능하도록 2개의 모드만을 채택하고 있으며, 더 다양한 모드에 따라 구조 조정이 용이하도록 비트 9를 보존해 놓았다. 명령어 세트는 일반적인 DSP 명령어에 MAC 동작과 FIFO 동작을 제어하기 위한 일부 명령어를 추가하였다.

이 어드레싱은 크게 register mode와 immediate mode로 나눌 수 있다. 레지스터 어드레싱은 오퍼코드가 직접 MDR (Memory Data Register)로 전달된다. CPU 레지스터는 일반 레지스터 12개와 Repeat Counter(RP), Stack pointer(SP), Program Counter(PC)등 16개 레지스터로 구성되어 있다. 그림 4는 일반 어드레싱 모드의 인코딩을 보여 주고 있다.

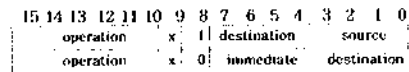


그림 4. 어드레싱 모드

명령어 세트의 목적은 각 상자가 매 명령 주기마다 바쁘게 동작하게 하는 것이다. 이는 최대한의 속도와 최적의 프로그램 메모리의 사용을 보장한다. 본 DSP에서는 Arithmetic, Logic, Move, FIFO 명령어로 나누어진다. 특히 데이터 FIFO가 RAM처럼 사용될 수 있는 구조인 점을 감안하여 FIFO의 입출력을 제어할 수 있는 FIFO 명령어가 필요하다. 이는 명령어는 각각 3개의 데이터 FIFO에 대해 입력 데이터와 출력 데이터의 출입을 가능하게 하기 때문에 FIFO에 대한 실질적인 제어 역할을 한다.

IV. 시뮬레이션

본 논문에서 제안한 dual MAC을 이용한 DSP의 구현의 성능은 평가하기 위하여 SUNsparc10 워크스테이션에서 ASIC 설계

제하지 않을 때는 1Kbps의 데이터율을 갖는다. 전이중(Full-duplex) 통신 방식의 시스템 전체 통화 시간의 60~70% 정도를 묶음 시간이 차지한다: 묶음 이용된 방식으로 음성 품질의 저하 없이 낮은 전송율로 부호화할 수 있는 장점이 있다. 이와 같이 데이터를 가변시키므로 용량을 증가시킬 수 있으며, 채널 오류에 강하고, Back ground 잡음은 부호화하여 자연스럽게 하는 효과를 얻을 수 있다.

인코딩 과정에는 매개 변수를 양자화하고 가장 최적의 변수를 찾아 데이터 패킷으로 보내며, 보낼 때는 데이터 전송율에 따라 패킹(Packing) 과정을 거쳐 수신단으로 보내게 된다. 이 알고리즘에서는 주로 분식/합성 방법을 사용하기 때문에 하드웨어로 구현할 때 이 부분을 최적으로 구현하는 것이 중요하다. 디코딩 과정은 인코딩 과정에 비해 실시간 처리 비용이 아주 작다. 왜냐하면 변수를 찾는 분식/합성 단계가 없기 때문이다. 이 과정에서는 데이터 패킷을 unpacking하고, 수신 매개 변수를 unquantizing 하며, LPC (Linear Predictive Coding) 필터라고 하는: formant 합성 필터로 필터링을 함으로써 음성 신호를 재생할 수 있다.

2. 효과적인 구조화 프로세싱

CELP 알고리즘의 효과적인 구조화 프로세싱에서 가장 중요한 부분은 전체 변수 계산 시간중 70~80%를 차지하는 피치와 여기(Excitation) 코드북 매개 변수 검색 과정이다. 이 과정에서 분식/합성 검색법을 사용하므로 원음과 합성음과의 차이에 대한 여러 웨이트를 최소화하는 과정에서 발생하는 변수의 처리가 음성 부호화기 구조를 결정하는 중요한 요소이다. 피치와 코드북검색 알고리즘은 매우 유사하며, 따라서 피치 변수 검색 과정에서 발생하는 MAC 동작에 대한 고려로부터 구조화 프로세싱에 대한 접근을 시도한다. 그림 2는 피치 변수 검색 과정을 보여주고 있다.

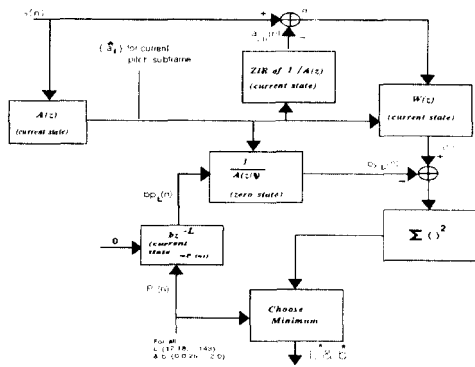


그림 1. 피치 변수 검색 과정

위 과정에서 알고리즘을 가장 잘 구현해 줄 수 있는 DSP를 설계하기 위해서는 먼저 원음과 합성을 정보 사이의 여러 웨이트를 최소화하고, 빠르게 변수를 검색할 수 있는 최적 알고리즘을 구현해야 한다. 먼저 웨이트된 필터를 통과한 원음 신호 $x(n)$ 과 피치 필터를 통과한 $by_L(n)$ 로 에너지 값을 구해보자.

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

$$E_{yL} = \sum_{n=0}^{L-1} y_L^2(n)$$

L 로 나타낸 최적 L 과 b 로 나타낸 최적 b 는 다음에서 구해진 최소값의 L 과 b 이다.

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

(2) 식의 최소값은 (3) 식의 최소값을 찾음으로써 구할 수 있다.

$$-2bE_{xyl} + b^2E_{yL} \quad (3)$$

(3) 식에서 b 와 E_{xyl} , E_{yL} 는 서로 독립이기 때문에 서로에 대한 편미분을 하면,

$$E_L = \frac{(E_{xyl})^2}{E_{yL}} \quad (4)$$

을 최대값으로 만족하는 L 값이 최적의 L 값이며, 그 때의 E_{xyl} , E_{yL} 을 (3) 식에 대입하게 되면 E_b 를 구할 수 있다.

E_b 는 (5)식과 같다.

$$E_b = \frac{E_{xyl}}{E_{yL}} \quad (5)$$

따라서 에너지 값, E_L 와 L 값에 따른 편미 출력, $y_L(n)$ 값을 가장 빠르게 찾는 것이 최적의 프로세싱을 하기 위한 조건이다. 그러나 에너지 값은 $x(n)$, $y_L(n)$ 값을 서로 곱해야 하므로 $x(n)$, $y_L(n)$ 와 에너지 값을 병렬로 처리할 수는 없다. 따라서 병렬 프로세싱이 가능하게 하기 위해서는 L 에 따른 $x_L(n)$ 의 에너지 항을 나누어 그림 3과 같이 병렬 프로세싱을 한다.^[6]

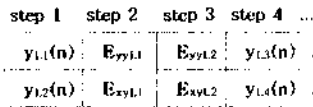


그림 2. MAC의 병렬 프로세싱

III. DSP 코어 모델

하드웨어로 구현할 때 가장 염두에 두어야 할 계산은 가중 합성 음성(weighted synthesized speech)인 $y(n)$ 과, 또한 최적의 가중치 에너지를 갖는 L , b 를 계산하기 위해 에너지 E_{yL} , E_{yyl} 을 빨리 계산하는 것이다.

따라서 dual MAC을 채용함으로써 많은 계산량을 두 블록으로 분산할 수 있기 때문에 MAC 블록의 이용도를 줄임으로써 지연력이 가능하게 할 수 있으며, 수학적으로 두 배의 시간 절약 을 가져올 수 있다. 또한 최적의 L , b 을 계산하는 과정에서 발생하는 다단계 과정은 하드웨어로 구현할 경우 칩 면적의 확대가 불가피하므로 어셈블리 프로그램을 통해 DSP Core에서 처리 하는 것이 좋다. 또한 L_1 , L_2 값이 동시에 계산이 가능하므로 계산된 값을 레지스터에 저장하여 또 다른 L_i 에 대한 계산이 MAC에서 이루어질 때 Core에서 두 값을 파이프라인으로 비교 하게 되면 최소오차 매개변수들 찾는데 50% 시간을 절약할 수 있다.

동인 COMPASS 내의 시뮬레이션 틀인 Quick SIM을 이용하였다. 선택된 명령어들은 Hand assembly하여 레지스터 내의 데이터가 올바르게 입증되었음을 근거로 프로세서 구조의 정확도를 점검할 수 있다. 먼저 dual MAC과 FIFO 메모리 동작을 확인하기 위해 MAC 명령어에 대한 실험을 하였다. 그림 5는 MAC & FIFO 동작의 실험 결과를 보여주고 있다.

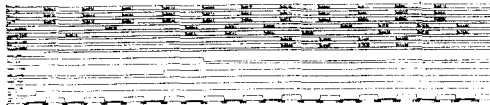


그림 5. FIFO&MAC 블록의 시뮬레이션 결과

첫 번째 클럭에서는 각 레지스터 내의 값을 reset시키기 위한 동작이 이루어지며, 두 번째 클럭부터 MAC 동작이 이루어진다. 파이프라인이 가능하도록 곱셈기와 덧셈기 사이에 파이프라인 레지스터가 있기 때문에 MAC 연산 한 동작은 두 클럭에 연산이 가능하다. 본 논문에서는 레지스터를 이용하여 FIFO를 설계하였으나 보다 더 낮은 전력 사용이 가능하게 하기 위해서는 속도는 다소 떨어지더라도 Dual port SRAM과 shift 기능을 동시에 갖는 FIFO를 사용한다면 고속과 저전력을 동시에 만족할 수 있다. 명령어 메모리 또한 SRAM을 응용한 FIFO를 사용한 경우 고속의 fetch 동작이 가능하기 때문에 파이프라인 단계를 더 늘릴 수 있는 이점이 있다.^[9]

전체 시뮬레이션은 MAC 동작과는 별개의 세스를 갖는 DALU 명령어에 대해서만 실험하였다. 음성 부호화기 알고리즘 특성상 MAC 동작과 일한 연산 동작은 구별되어도 상관이 없다. 서로 다른 클럭에서 동작하기 때문이다. 그림 6에 DALU 블록의 시뮬레이션 결과를 보여주고 있다.



그림 6. DALU 블록의 시뮬레이션 결과

본 논문 DSP의 critical path는 MAC 동작 클럭 중에서 16 X 16 곱셈기로서 COMPASS 라이브러리의 bit array unsigned 곱셈기이다. 전송 지연(Propagation delay)은 56ns로 약 18Mhz의 연산 속도를 갖는다. 파이프라인 프로세서에서는 core의 연산 속도가 곱셈기의 지연 시간에 가장 많은 영향을 받는다. 따라서 곱셈기의 곱셈기 설계로부터 전체 시스템의 처리 속도를 향상시킬 수 있다. 예로 16X16 비트 고속 병렬 곱셈기 구조를 이용한 MAC 설계를 할 경우 20ns이하로 지연 시간을 떨어뜨릴 수 있다.^[10] 따라서 전체 시스템 속도는 50Mhz까지 향상된다.

V. 결론

본 논문에서는 이동통신 기술중 하나인 CDMA 방식의 셀룰러

라 본의 음성 부호화 알고리즘의 병렬 특성에 기인해서 CELP 알고리즘을 효율적으로 구현할 수 있는 16비트 프로그래머블 DSP core를 설계하였다.

Core는 기본 연산을 수행하는 DALU와 전력 분산을 통한 사용 전력을 낮추기 위해 dual MAC을 사용하였으며, MAC의 임박이 되는 각종 매개 변수를 고속으로 MACU에 접속시키기 위해 레지스터를 이용한 FIFO 메모리를 사용하였다. 2개의 FIFO 메모리에서 출력은 한 클럭당 두 개의 데이터가 가능하며, 그것은 일반 워드 레지스터 속도인 2ns이하의 시간지연을 갖는다. 또한 전체 프로세서는 18Mhz 동작 속도를 가지며, 곱셈기를 full custom으로 설계한다면 50Mhz까지 동작 속도를 향상시킬 수 있다. 프로세서이다.

앞으로의 과제: QCELP의 분석을 통해 정의된 DSP 클럭 구조적인 보완을 통해 면적과 시간의 최소화를 통한 명령어 사이클을 가속화하고, 사용자와 개발자를 위한 어셈블리와 명령어 레벨 시뮬레이터와 같은 사용자 인터페이스를 개발하는 것이며, 본 논문은 동진기기의 소형화, 경량화, 저전력과 추세에 맞는 구조적인 조성과 보완을 통해 현재 국내에서 활발하게 진행중인 이동통신 기기의 개발에 밑거름이 될 수 있을 것이다.

참고 문헌

- [1] 휴대용 이동 전화를 위한 CPU 개발, 채신우, 1994
- [2] WE DSP16 and DSP16A Digital Signal Processor Information Manual, AT&T, Inc. 1989
- [3] 이인성, et al., "CDMA 디지털 셀룰러용 음성 부호화가 (QCELP)의 복잡도 감소알고리즘 및 성능 연구", 신호 처리합동학술회의 논문집, 제6권 1호, pp.20~23, 1993
- [4] P.Kroon, K.Swaminathan, "A High-Quality Multirate Real-Time CELP Coder", IEEE J.Select. Areas Commun., Vol.10 No.5, pp.854~856, June 1992
- [5] J.H. Park, Y.M.Kim, "Analysis and Optimization of Speech Coder Algorithm for CDMA Digital Cellular", IEEE Int'l Conf. on Communication Systems, Singapore, pp.870~874, 1994
- [6] S.G.Shiva, Computer Design&Architecture, H.C.P., New York, pp.145~148, 1991
- [7] K.Ueda, et al., "A 16b Low-Power-Consumption Digital Signal Processor", ISSCC, Session 2, WP2.2, pp.28~30, 1993
- [8] Kai Hwang, Computer Architecture, John Wiley & Sons, New York, pp.43~47, 1979
- [9] 박주현, 김영민, "Instruction FIFO memory를 이용한 범용 DSP 구조", 전자공학회의논문집, 제32권 B편 제3호, pp.31~37, 1995
- [10] 조진호, 김영민, "32X32 비트 고속 병렬 곱셈기 구조", 전자공학회의논문집, 제31권 B편 제10호, pp.67~72, 1994