

IMT-2000 비동기식 단말기용 ASIC을 위한 적응형 다중 비트율 (AMR) 보코더의 구현

Implementation of Adaptive Multi Rate (AMR) Vocoder for the Asynchronous IMT-2000 Mobile ASIC

변 경 진*, 최 민 석*, 한 민 수**, 김 경 수*

(Kyung Jin Byun*, Min Seok Choi*, Min Soo Hahn**, Kyung Soo Kim)

*한국전자통신연구원, **한국정보통신대학원대학교 공학부

(접수일자: 2000년 11월 7일; 채택일자: 2000년 12월 20일)

본 논문은 비동기 방식의 IMT-2000 단말기용 ASIC (주분형 집적회로)에 포함되는 음성부호화기 알고리즘인 AMR (Adaptive Multi Rate) 보코더의 실시간 구현에 관한 것이다. 구현된 AMR 보코더는 12.2kbps에서 4.75kbps까지 8가지의 다중 비트율을 가지고 있으며, 인코더와 디코더 기능 외에 VAD (Voice Activity Detection) 블록과 SCR (Source Controlled Rate operation) 블록 등의 부가기능 및 시스템과의 접속 처리를 위한 프레임 구성 기능도 구현되어 있다. AMR 보코더를 구현하기 위하여 설계된 DSP (디지털 신호처리기)는 TeakLite 코어를 기반으로 하여 메모리 블록, 직렬 접속 블록, CPU와의 접속을 위한 레지스터 파일 블록, 인터럽트 제어회로 등으로 구성된 16비트 고정 소수점형 DSP이다. 실시간 구현 방법에서는 메모리의 효율적인 관리를 통하여 계산량을 최적화하여 최대 동작 계산량을 약 24MIPS로 줄였으며, 구현된 AMR 보코더는 3GPP의 표준 시험 벡터를 모두 통과하여 검증에 완료하고, 실시간 보드 시험에서도 안정적으로 동작하는 것이 확인되었다.

핵심용어: 적응형 다중 비트율 (AMR), 보코더, 음성 부호화기, 디지털 신호 처리기, IMT-2000

투고분야: 음성처리 분야 (2.2)

This paper presents the real-time implementation of an AMR (Adaptive Multi Rate) vocoder which is included in the asynchronous International Mobile Telecommunication (IMT)-2000 mobile ASIC. The implemented AMR vocoder is a multi-rate coder with 8 modes operating at bit rates from 12.2kbps down to 4.75kbps. Not only the encoder and the decoder as basic functions of the vocoder are implemented, but VAD (Voice Activity Detection), SCR (Source Controlled Rate) operation and frame structuring blocks for the system interface are also implemented in this vocoder. The DSP for AMR vocoder implementation is a 16bit fixed-point DSP which is based on the TeakLite core and consists of memory block, serial interface block, register files for the parallel interface with CPU, and interrupt control logic. Through the implementation, we reduce the maximum operating complexity to 24MIPS by efficiently managing the memory structure. The AMR vocoder is verified throughout all the test vectors provided by 3GPP, and stable operation in the real-time testing board is also proved.

Key words: Adaptive multi rate (AMR), Vocoder, Speech coder, Digital signal processor (DSP), IMT-2000

Subject classification: Speech signal processing (2.2)

I. 서 론

디지털 이동통신 시스템에서는 전송 채널의 대역폭을 효율적으로 사용하고, 다양한 조건의 무선 채널 환경에서도

고음질의 통화를 하기 위하여 다양한 보코더 알고리즘들을 채택하고 있다. 현재 국내에서 상용 서비스를 하는 셀룰러 PCS (Personal Communication System) 시스템에서 사용되는 보코더 알고리즘으로는 QCELP (Qualcomm Code Excited Linear Prediction), EVRC (Enhanced Variable Rate Coder) 등이 있다. 이러한 QCELP나 EVRC 알고리즘은 가변 비트율을 사용하는 특징을 가지고 있다. 이러한 가변 비트율을 갖는 보코더는 입력음성의 변화에

책임저자: 변경진 (kjbyun@etri.re.kr)

305-350 대전광역시 유성구 가정동 161 한국전자통신연구원
통신디지털 회로팀

(전화: 042-860-5831; 팩스: 042-860-6108)

따라 전송 비트율을 조절할 수 있으므로 전송채널을 효율적으로 사용할 수 있다는 장점은 있지만, 통신 중에 수시로 변화하는 무선채널의 조건에 적용할 수 없기 때문에 채널 상황이 악화되면 음질이 급격히 저하된다. 또한 고정된 비트율을 갖는 코코더들도 앞에서 언급된 코코더와 마찬가지로 무선채널의 조건이 악화되면 양질의 통화상태 유지가 어려워진다.

차세대 이동통신 시스템인 IMT-2000에서는 이러한 문제점을 해결하고자 AMR (Adaptive Multi Rate) 코코더를 표준안으로 채택하게 되었다[1]. AMR 코코더 알고리즘은 주어진 채널환경에서 가장 최적의 음질을 유지할 수 있도록 구성되어 있다. 즉 전체 전송 비트율은 일정하게 고정되어 있으나 채널환경이 좋은 상태에서는 오류 방지를 위한 비트 할당을 적게하는 대신 높은 비트율의 부호화 모드를 선택하고, 채널환경이 열악해지면 낮은 비트율의 부호화 모드를 선택하는 대신에 오류 경정을 위한 비트 할당을 많이 함으로써 채널 변화에 따라 최적의 음질을 유지할 수 있게 된다. 그림 1에 채널 상태에 따른 음질의 변화를 나타내었다[2].

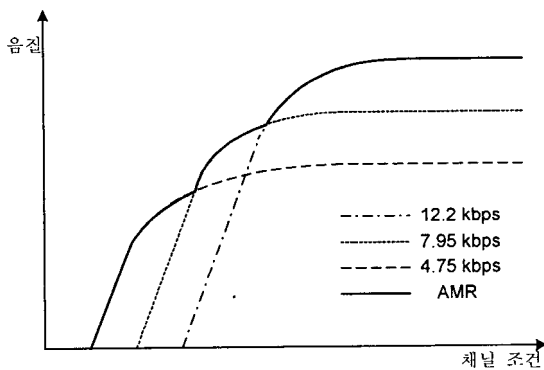


그림 1. 채널 조건에 따른 음질의 변화
Fig. 1. Quality variation for the channel condition.

그림 1에서는 AMR 음성 부호화기의 8가지 모드 중에서 3가지 모드를 예로 들었다. 그림에서 12.2kbps 모드는 채널오류가 없는 경우 가장 높은 음질을 얻을 수 있으나 낮은 비트율인 7.95kbps, 4.75kbps 모드에 비해 상대적으로 낮은 수준의 오류 방지를 위한 비트가 할당되므로 채널오류가 많아지면 7.95kbps, 4.75kbps 모드보다 오히려 음질이 더 떨어지게 된다. 하지만 AMR 코코더의 기본적인 개념은 현재의 채널 조건을 측정하여 주어진 조건 하에서 최적의 음질을 얻을 수 있는 모드로 부호화를 수행하기 때문에 그림 1의 굵은 실선과 같이 항상 최적의 음질을 유지할 수 있게 된다.

본 논문에서는 IMT-2000 단말기용 기저 대역용 ASIC에 포함되는 AMR 코코더를 구현한 내용에 대하여 기술하였다. AMR 코코더의 구현에는 3GPP의 표준으로 지정된 고정 소수점형 C 소스 코드를 기준으로 사용하였다[3]. 그리고 구현을 위해 설계된 DSP는 TeakLite DSP 코

어를 기반으로 하여 메모리 블록, 직렬 및 병렬 접속부 등의 블록으로 구성되어 있다. 제 II절에서는 AMR 음성 부호화기 알고리즘에 대한 개요를 간략하게 설명하고, III절에서는 AMR 코코더의 실시간 구현 및 DSP의 설계, 그리고 구현 결과에 대하여 설명하고 IV절에서 결론을 맺었다.

II. AMR 코코더 알고리즘

AMR 음성부호화기는 8개의 비트율 (12.2, 10.2, 7.95, 7.4, 6.7, 5.9, 5.15, 4.75kbps)로 동작할 수 있는 다중 비트율을 갖는 부호화 알고리즘으로 구성되어 있다. 각 비트율에 따른 변수에 대한 비트 할당은 다음의 표 1과 같다.

표 1. AMR 코코더의 각 모드별 비트 할당
Table 1. Bit allocation of the AMR vocoder.

Mode	Parameter	subframe				total
		1	2	3	4	
12.2 kbit/s (244)	LSP	38				38
	Pitch lag	9	6	9	6	30
	Pitch gain	4	4	4	4	16
	Alg CB	35	35	35	35	140
	Alg gain	5	5	5	5	20
10.2 kbit/s (204)	LSP	26				26
	Pitch lag	8	5	8	5	26
	Alg CB	31	31	31	31	124
	Gains	7	7	7	7	28
7.95 kbit/s (159)	LSP	27				27
	Pitch lag	8	6	8	6	28
	Pitch gain	4	4	4	4	16
	Alg CB	17	17	17	17	68
	Alg gain	5	5	5	5	20
7.4 kbit/s (148)	LSP	26				26
	Pitch lag	8	5	8	5	26
	Alg CB	17	17	17	17	68
	Gains	7	7	7	7	28
6.7 kbit/s (134)	LSP	26				26
	Pitch lag	8	4	8	4	24
	Alg CB	14	14	14	14	56
	Gains	7	7	7	7	28
5.9 kbit/s (118)	LSP	26				26
	Pitch lag	8	4	8	4	24
	Alg CB	11	11	11	11	44
	Gains	6	6	6	6	24
5.15 kbit/s (103)	LSP	23				23
	Pitch lag	8	4	4	4	20
	Alg CB	9	9	9	9	36
	Gains	6	6	6	6	24
4.75 kbit/s (95)	LSP	23				23
	Pitch lag	8	4	4	4	20
	Alg CB	9	9	9	9	36
	Gains	8		8		16

$$A_k = \frac{(C_k)^2}{E_{Dk}} = \frac{(d^T C_k)^2}{C_k^T \Phi C_k}$$

여기서 $d = H^T x_2$ 은 목표신호 $x_2(n)$ 과 임펄스 응답 $h(n)$ 사이의 상관도를 나타낸다. 그리고 $\Phi = H^T H$ (H 는 Toeplitz convolution matrix)는 $h(n)$ 의 상관관계 매트릭스이다. 그리고 벡터 d 와 Φ 는 검색과정의 계산량을 줄이기 위하여 코드북 검색 전에 미리 계산되어진다.

피치 및 코드북의 이득 양자화에서는 모든 모드에 대하여 피치이득 값은 직접 양자화 되고, 코드북 이득 값은 예측 오차 값에 대하여 양자화를 수행한다. 12.2kbps와 7.95kbps의 경우는 피치 이득과 코드북 이득 값을 분리해서 양자화하지만 나머지 모드の場合は 두개의 이득 값을 묶어서 벡터 양자화를 수행한다.

III. 실시간 구현 및 최적화 방법

3.1. 디지털 신호 프로세서의 구조

AMR 코코더를 구현하기 위하여 사용한 DSP는 16비트 고정 소수점형 DSP로써 그림 3와 같이 TeakLite 코어를 기반으로 하여 프로그램 메모리 및 데이터 메모리 블록, 음성 코덱과 데이터를 주고받기 위한 직렬 접속부, CPU와의 패킷 데이터 송수신을 위한 병렬 접속부 등으로 설계되어 있다.

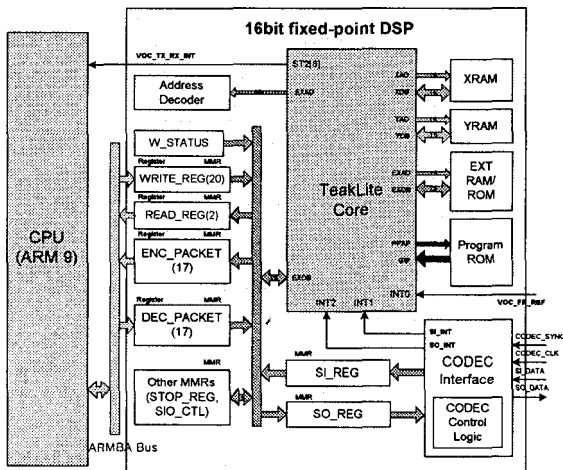


그림 3. 디지털 신호 프로세서의 구조
Fig. 3. DSP architecture.

그림 3에서 TeakLite 코어 부분은 연산 블록 (computation unit: CU), 비트 조작 블록 (bit manipulation unit: BMU), 데이터 주소생성 블록 (data addressing unit: DAU), 프로그램 제어 블록 (program control unit: PCU) 등의 4개의 주요 블록으로 구성되어 있다. 메모리 블록은 크게 프로그램 ROM 블록과 데이터 메모리 (RAM, ROM) 블록으로 구성되어 있으며, 데이터 메모리 블록은 2개의 뱅크를 사용하는 구조로 설계되어 두개의 데이터를 동시에 연산

블록으로 가져올 수 있다. 메모리 블록의 크기는 최종적인 코코더 프로그램의 크기에 맞추어 최적으로 조정할 수 있다. 직렬 접속 블록에서는 음성코덱으로부터 직렬 데이터를 입력으로 받아 병렬 데이터 (16bit or 8bit)로 만들어 DSP 코어 쪽으로 넘겨주고, 음성코덱으로 출력할 때는 반대의 동작을 하는 기능을 가지고 있다. 시스템과의 접속을 위한 병렬 접속 블록은 주로 레지스터 파일을 사용하여 CPU와의 패킷 데이터 전송을 주고 받도록 설계하였다. 그리고 병렬 접속부를 통하여 CPU에서 코코더의 동작을 조정하는 명령들을 사용할 수 있는 기능이 있다. 결과적으로 코어를 이용한 DSP의 설계는 메모리를 효과적으로 사용할 수 있고 AMR 코코더 구현에 적합하도록 접속부 및 제어 블록들을 설계할 수 있는 장점이 있다. 설계된 DSP의 주요 특징은 다음과 같다.

- 60MHz 동작 주파수 (60MIPS)
- 한 사이클에 곱셈 및 덧셈 연산 동시 수행
- 2 개의 36비트 누산기
- 36비트의 ALU와 배럴 시프터 (barrel shifter)
- Max/Min 명령을 한 사이클에 처리
- 두개의 메모리 뱅크 구조
- 데이터 RAM (4k words x 2), 데이터 ROM (8k words x 2), 프로그램 ROM (20k words)
- 한 개의 직렬포트와 한 개의 16비트 병렬포트

3.2. 실시간 구현 및 최적화

AMR 코코더의 실시간 구현은 3GPP 표준안인 TS 26.090과 함께 제공되는 고정 소수점 C 프로그램의 표준안인 TS 26.073을 바탕으로 어셈블리 언어로 작성하였다. AMR 코코더에는 인코더, 디코더의 기본적인 기능 외에 DTX (Discontinuous Transmission) 기능을 위한 VAD (Voice Activity Detection) 블록과 SCR (Source Controlled Rate) 블록이 구현되어 있다. 그리고 전송 시의 오류에 대비한 오류 은폐 (error concealment) 블록 및 매우 적은 크기를 갖는 잡음을 생성 (comfort noise generation) 하는 기능 등이 구현되어 있으며, 시스템과의 접속 규격을 맞추기 위한 프레임 구성 (frame structuring) 기능도 구현되어 있다.

AMR 코코더의 실시간 구현은 C 코드의 각 기능블록들을 어셈블리 코드로 작성하고, 사람 벡터를 각각 인가하여 C의 결과와 어셈블리 코드의 결과가 비트 단위에서 같도록 하였다. AMR 코코더를 실시간 구현함에 있어서 표준 알고리즘을 그대로 구현하기 위해서 알고리즘 상의 최적화는 고려되지 않았으며, 어셈블리 코딩 레벨에서 최적화 작업을 수행하였다. AMR 코코더의 C코드 자체가 이미 최적화 단계를 거쳐서 표준화된 것이기 때문에 어셈블리 코딩 레벨에서의 최적화 작업은 주로 메모리 관리 및 반복문 내에서의 메모리 접근시의 연산량을 줄이는데 집중되었다.

효율적인 메모리의 관리를 위해서 AMR 코코더에서 사용되는 변수들을 기능 별로는 인코더와 디코더에서 사용되는 변수들에 대해 독립적으로 메모리를 할당하여 인코더와 디코더 사이의 메모리 충돌없이 동작하도록 하였

다. 변수의 성질에 따라 크게 정적변수와 동적변수들로 나누고 정적변수들을 우선적으로 DSP의 메모리로 할당하였다. DSP의 메모리 구조는 X, Y 두개의 뱅크를 가지고 있으므로 정적변수들을 할당할 때, 프로그램에서 루프 내의 MAC (Multiply and Accumulate) 명령어 동작 시 두개의 데이터 값을 동시에 가져올 수 있도록 구성하여 반복문 내에서 메모리 접근 시 연산량을 최소화하도록 하였다. 그리고 서브루틴 내에서 사용되는 변수들에 대해서는 전체 프로그램 트리어에서 각 서브루틴의 레벨에 따라서 인코더의 경우 5개의 단으로, 디코더의 경우 4개의 단으로 메모리를 구성하여 서브루틴 내에서 사용되는 변수들에 대해서는 페이지 레지스터를 통하여 직접주소 (direct addressing)를 사용함으로써 한 사이클에 메모리 접근을 할 수 있도록 하여 계산량을 최소화하였다. 서브루틴 내에서 사용하는 변수가 정적변수인 경우에는 서브루틴에 들어가기 전에 해당 서브루틴의 레벨에 맞는 메모리에 복사를 하고 서브루틴을 수행한 후 결과를 다시 정적변수로 할당된 메모리에 저장하는 방법을 사용함으로써 서브루틴 내에서는 직접주소를 사용하여 메모리 접근 사이클 손실을 최소화하였다.

앞에서 설명된 방법과 같이 서브루틴의 레벨에 따라 메모리를 계층적인 구조로 관리함으로써 각 서브루틴에서 사용되는 변수들 간의 메모리 충돌을 피할 수 있고, 서브루틴에서 사용되는 변수들의 전달이 용이하기 때문에 코딩 시 공동 작업이 용이하며 오류 검출 작업의 효율도 높일 수 있는 장점이 있다.

3.3. 검증 및 실시간 시험

어셈블리 언어로 구현된 AMR 보코더의 인코더와 디코더의 기능 검증에서는 3GPP에서 제공하는 모든 시험 벡터 23개 (7,907 프레임)에 대해서 8개의 모드별로 전부 수행하여 C 프로그램의 수행 결과와 완전히 일치된 결과를 보였다. 그리고 DTX 기능의 검증을 위하여 DTX용 시험 벡터 4개 (4,416 프레임)에 대하여 모드별로 수행하여 C 프로그램의 수행결과와 일치함을 확인하였다[7].

구현 및 검증 과정에서 주의를 요하는 부분 중의 하나는 C 코드와 DSP간의 데이터 정확성을 맞추는 일이다. 즉 C 코드에서는 모든 데이터 값들을 32비트 이내의 값으로 제한하도록 되어 있으나, DSP의 누산기는 오버플로우의 가능성을 줄이기 위하여 36비트까지 확장되어 있으므로 어셈블리 코딩 시에 C 코드와의 호환성을 위하여 누산기의 데이터 값을 32비트 이내로 제한하였다. 실제로 시험 벡터를 사용하여 오류를 검증하는 과정에서도 이러한 부분에서의 오류가 가장 많이 검출되었다.

시뮬레이션을 통한 기능검증을 완료하고, 실시간 시험 및 주문형 집적회로 (ASIC: application specific integration circuit)으로 제작 시의 오류를 방지하기 위하여 TeakLite DSP 칩을 이용하여 앞의 그림 3과 같은 접속 블록을 갖는 보드를 설계 제작하여 실시간 시험 환경을 구축하고 자체의 음성 데이터 베이스를 이용하여 실시간 시험을 수행하였다.

실시간 시험에서는 8개의 각 모드별로 동작하는 상태와,

DTX 기능의 시험, 그리고 임의의 프레임 별로 모드를 변화시켜 가면서 동작하는 기능 등을 시험하였다. 주관적인 음질평가 결과로는 가장 높은 비트율을 갖는 12.2kbps 모드와 가장 낮은 비트율을 갖는 4.75kbps 모드와의 음질 차이가 거의 구별할 수 없을 정도였으며, 실제로 프레임 별로 모드를 바꿔 가면서 동작하는 경우에도 모드가 바뀌는 것을 감지하기 어려웠다. 12.2kbps 모드의 음질은 EVRC 보코더에 비해 매우 우수함을 확인할 수 있었으며, 나머지 모드에서도 EVRC 보코더와 거의 같은 수준의 음질상태를 보여 주었다.

3.4. 구현 결과

표 2에 구현된 AMR 보코더의 모드별 계산량을 나타내었다.

표 2. AMR 보코더의 각 모드별 계산량
Table 2. Complexity of AMR.

모드 (비트율)	계산량	최적화 후
12.2 kbps	30.88 MIPS	23.92 MIPS
10.2 kbps	28.61 MIPS	22.29 MIPS
7.95 kbps	27.83 MIPS	23.71 MIPS
7.4 kbps	26.63 MIPS	21.49 MIPS
6.7 kbps	29.13 MIPS	23.41 MIPS
5.9 kbps	23.70 MIPS	18.84 MIPS
5.15 kbps	19.99 MIPS	15.76 MIPS
4.75 kbps	23.83 MIPS	19.09 MIPS

AMR 보코더를 실시간 구현한 결과 12.2kbps 모드에서의 계산량이 가장 크고, 5.15kbps 모드에서의 계산량이 가장 적게 소요 되었다. 표 2에서 최적화 후의 결과는 계산량이 가장 큰 12.2kbps 모드를 중점적으로 최적화 하였고, 나머지 모드는 1차적인 최적화 결과이다.

표 3은 가장 계산량을 많은 12.2kbps 모드의 주요 기능별 세부 계산량을 표시한 것으로, 최적화 과정을 수행한 후의 것을 기준으로 한 것이다.

표 3. 12.2kbps AMR 보코더의 계산량
Table 3. Complexity of AMR at 12.2kbps.

	Function block (12.2kbps)	MIPS
Encoder	Pre-processing	0.39
	Compute LPC	1.10
	LSP 변화 및 양자화	4.22
	open-loop pitch search	1.79
	closed-loop pitch search	2.90
	codebook search	8.64
	Gain 양자화	0.31
	Memory update, 기타	1.52
Sub total	Encoder total	20.87
	Decoder total	3.05
Total	Enc.+ Dec.	23.92

표 3에서 확인할 수 있듯이 피치검색과 코드북 검색 모듈이 전체 인코더 계산량의 대부분을 차지하고 있다. 따라서, 본 논문에서는 우선적으로 8가지 모드 중에서 계산량을 가장 많이 차지하는 12.2kbps 모드의 피치검색과 코드북 검색 블록에 대하여 어셈블리 레벨에서 계산량을 최적화한 결과 표 2과 같이 약 7MIPS 정도의 계산량을 줄일 수 있었다. 향후 나머지 모드에 대해서도 추가적인 최적화 작업을 수행하면 모든 모드에서 실시간 동작 시의 최대 계산량을 24MIPS 이하로 감소시킬 수 있을 것으로 예측하고 있다. AMR 코코더의 실시간 구현에 사용된 프로그램 메모리 (ROM)는 19,395words이고, 데이터 메모리는 RAM 6,972words, ROM 15,051words를 사용하였다. 메모리에 대한 최적화 작업은 아직 수행하지 않았지만, 프로그램 ROM의 크기와 데이터 RAM의 크기는 약 10% 정도 줄일 수 있을 것으로 예상된다.

IV. 결 론

AMR 코코더는 이동기식 방식의 IMT-2000 시스템의 표준 코코더 알고리즘으로써 채널의 조건에 따라 최적의 음질을 유지할 수 있도록 8가지 모드의 비트율을 가지고 있다. 본 논문에서는 이러한 AMR 코코더를 실시간 구현하기 위하여 TeakLite 코어를 기반으로 하여 16비트 고정 소수점형 DSP를 설계하고, 계층적인 구조의 메모리 구성을 통하여 적은 계산량으로 효율적인 어셈블리 프로그램을 작성하였다. 구현된 AMR 코코더는 3GPP에서 지정한 모든 시험 벡터를 통과시켜 검증하였으며, 12.2kbps 모드에서 약 24MIPS의 계산량으로 동작된다. 앞으로 추가적인 메모리 최적화 및 계산량 최적화 작업을 거친 후 IMT-2000 단말기용 ASIC에 포함시켜 하나의 칩으로 집적화 될 예정이다.

참 고 문 헌

1. 3GPP (3rd Generation Partnership Project) TS 26.090, "AMR Speech Codec; Tanscoding functions," December 1999.
2. S. Bruhn, P. Blocher, K. Hellwig, J. Sjoberg, "Concepts and solutions for link adaptation and inband signaling for the GSM AMR speech coding standard," Vehicular Tech. Conference, vol. 3, pp. 2451-2455, 1999.
3. 3GPP TS 26.073, "ANSI-C code for the Adaptive Multi Rate speech codec," December 1999.
4. R. Salami, C. Laflamme, B. Bessette, J-P. Adoul, "Description of GSM Enhanced Full Rate Speech Codec," ICC '97, vol. 2, pp. 725-729, 1997.
5. T. Honkanen, J. Vainio, K. Jarvinen, P. Haavisto, "Enhanced Full Rate Speech Codec for IS-136 Digital Cellular System," ICASSP, vol. 2, pp. 731-734, 1997.
6. T. Ohya, H. Suda, T. Miki, "5.6kb/s PSI-CELP of the half-rate PDC speech coding standard," Vehicular Tech. Conference, vol. 3, pp. 1680-1684, 1994.
7. 3GPP TS 26.074, "AMR Speech Codec Test Sequences,"

December 1999.

8. E. Ekudden, R. Hagent, I. Johansson, J. Svedberg, "The Adaptive Multi-rate Speech Coder," Speech Coding Proceedings, IEEE Workshop, pp. 117-119, 1999.
9. 안도건, 유승규, 최용수, 이재성, 강태익, 박성현, "16비트 고정 소수점 DSP를 이용한 다채널 G.729A 음성 부호화기의 실시간 구현," 한국음향학회지, 19권 4호, pp. 45-51, 2000.

▲ 변 경 진 (Kyung Jin Byun)

1987년 2월: 국민 대학교 전자공학과 (공학사)
 2000년 2월: 한국정보통신대학원대학교 공학부 (공학석사)
 2000년 3월~현재: 한국정보통신 대학원대학교 공학부 박사과정 재학중
 1987년 3월~현재: 한국전자통신연구원 통신디지털회로팀 선임연구원
 ※ 주관심분야: DSP 설계, 음성신호 코딩

▲ 최 민 석 (Min Seok Choi)

1997년 8월: 한국과학기술원 전기 및 전자공학과 (공학사)
 1999년 8월: 한국정보통신대학원대학교 공학부 (공학석사)
 1999년 8월~현재: 한국전자통신연구원 통신디지털회로팀 연구원
 ※ 주관심분야: 음성 신호처리, 음성인식 및 합성, DSP 설계

▲ 한 민 수 (Min Soo Hahn)

1979년 2월: 서울대학교 전기공학과 (공학사)
 1981년 2월: 서울대학교 대학원 (공학석사)
 1989년 12월: University of Florida (공학박사)
 1982년 4월~1985년 8월: 한국표준 과학연구원 연구원
 1990년 2월~1997년 12월: 한국전자통신연구원 책임연구원
 1998년 1월~현재: 한국정보통신대학원대학교 공학부 부교수
 ※ 주관심분야: 디지털 신호 처리, 음성 분석, 합성, 인식 및 코딩, 적응 신호 처리, 3-D 음향 등

▲ 김 경 수 (Kyung Soo Kim)

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