

HCI PROCESSOR
Human and Computer Interaction Processor

HCI 프로세서 칩

HCI Processor One Chip ASIC

삼성전자 종합기술원

과 학 기 술 부

최종보고서 초록

과제관리번호	00-J-HI-01-B-05	총연구기간	'98. 12 - '02. 12 (4년 1개월)		
연구과제명	중 과 제 명	HCI PROCESSOR			
	단 위 과 제 명 ¹⁾	HCI 프로세서 칩			
연구책임자	장 영 훈	총참여 연구원수	총 : 5 명 내부 : 5 명 외부 : 0 명	연구비	정부 : 14,100 만원 기업 : 21,800 만원 계 : 35,900 만원
연구기관명 및 소속부서명	삼성전자 종합기술원 Micro Architecture Center	참여기업명	삼성전자		
요약					면수
<p>- 1차 년도는 프로세서 설계를 위한 방법론 설정, 상위수준에서 하위수준까지의 프로세서 설계 환경 구축, 프로세서 설계의 설계 연습, 시제품 보드 제작, 그리고 선진 프로세서 설계 기법의 연구를 통하여, 설계 개발기간을 단축하고 시스템 개발에 대한 신뢰성 확보를 위한 기반 기술이 선행되어 졌으며, 또한 이를 바탕으로 실제 설계가 진행됨과 동시에 설계환경 및 설계 방법론을 지속적으로 보완해 나가면서 향후 고성능 프로세서 제작에 신속히 대응할 수 있는 바탕을 마련하였음</p> <p>- 2차 년도는 음성 인식 부분의 알고리즘을 분석, 하드웨어 알고리즘인 고정 소수 연산 동작 모델링 및 검증, 시제품 보드상의 구현을 통한 HCI 프로세서의 일부를 구현하였음</p> <p>- 더 나아가 현실적인 관점에서 사업화를 위해 시장 조사 결과를 실시하고, 이를 사양 결정에 반영하였고, 또한 아키텍처 결정에도 참조 후 반영하여 3차 년도 계획 분을 선행하였으며, 효과적인 산학연 협동을 위한 글로벌 사이버 연구 센터를 인터넷상에 구축하였음</p>					
색 인 어	한 글	프로세서, 보드, 설계, 알고리즘, 모델링, 시제품, 아키텍처, 사이버			
	영 어	HCI			

제 출 문

과학기술부 장관 귀하

본 보고서를 “HCI PROCESSOR”과제 (세부과제 “HCI 프로세서 칩”) 의 보고서로 제출합니다.

2001. 03. 22

주관연구기관명 : 삼성전자 종합기술원

주관연구책임자 : 장 영 훈

연 구 원 : 박 현 우

“ : 김 승 현

“ : 장 호 랑

“ : 김 성 재

“ : 김 종 호

요 약 문

I. 제 목 : HCI 프로세서 칩 개발

II. 연구개발의 목적 및 필요성

1. 연구개발의 목적

- 가. HCI 프로세서 개발
- 나. SOC (System On Chip) 설계 방법론 확보

2. 연구개발의 필요성

가. 기술적 측면

- (1) HCI (Human & Computer Interaction) 기술 구현은 많은 데이터를 신속히 처리하는 것이 요구되므로 많은 데이터를 처리 할 수 있는 전용 H/W가 탑재된 HCI전용 단일 칩 솔루션이 절실히 필요함

나. 경제·산업적 측면

- (1) 금년은 HCI Chip Solution의 원년이라고 할 만큼 이와 관련된 시장이 급속히 성장하고 있다. 이에, HCI 시장을 선점/Lead 하기 위해서는 가격 경쟁력과 응용 범위가 넓은 Digital Signal Processor구조의 단일 전용 Chip Solution이 시장 선점에 유리함

다. 사회·문화적 측면

- (1) 정보단말의 개인화, 휴대화가 가속화됨에 따라 고집적, 저소비 전력의 제품 창출을 통해 언제, 어디서나 정보를 공유하고 이용하기에 편리해야 함

III. 연구개발의 내용 및 범위

구분	연구개발목표	연구개발내용 및 범위
1차년도	Multi-processor Design Methodology Setup	<ul style="list-style-type: none"> • Multi-processor Co-design/Sim./Verification Design Methodology 개발
2차년도	Fixed-point System Modeling	<ul style="list-style-type: none"> • Fixed-point System Modeling <ul style="list-style-type: none"> - 알고리즘분석 - 데이터 분석 - 데이터 분석 검증
3차년도	Micro Architecture 개발	<ul style="list-style-type: none"> • 화자독립/어휘독립 50단어 인식을 위한 ISA(Instruction Set Architectures) 설정 • Assembler 개발 • Assembly Coding • 12만 Gates급 Processor의 H/W 설계 • JTAG을 이용한 Test Logic 설계
	설계 및 검증	<ul style="list-style-type: none"> • 1차 Chip제작 (Processor Type => Performance : 20 Mips , 전력소모 : 80 mW, 제작공정 : 0.35um) • Test Board 제작 • Board Level의 Chip Debugging & Analysis • 2차 Chip제작 (Processor Type => Performance : 20 Mips , 전력소모 : 50 mW, 제작공정 : 0.25um) • 2차 제작 Chip의 Test 및 Evaluation

구분	연구개발목표	연구개발내용 및 범위
4차년도	Proto System 개발	<ul style="list-style-type: none"> • 음성합성압축 / 휴먼인식 알고리즘 분석 및 Fixed-point Modeling • 음성인식, 음성합성압축, 휴먼인식을 위한 ISA(Instruction Set Architectures)설정 • Assembler 개발 • Assembly Coding • 100만 Gates급 Processor의 Full Custom H/W 설계 • JTAG을 이용한 Test Logic 설계
	Chip Integration	<ul style="list-style-type: none"> • 1차 Chip제작(Processor Type=> Performance : 800 Mips , 전력소모 : 170 mW, 제작공정 : 0.18um) • Test Board 제작 • Board Level의 Chip Debugging Analysis • 2차 Chip제작(Processor Type=> Performance : 800 Mips , 전력소모 : 100 mW, 제작공정 : 0.15um) • 2차 제작 Chip의 Test 및 Evaluation

IV. 연구개발결과

구분	세부연구목표	달성내용	달성도 (%)
1차년도	멀티 프로세서 설계 방법론 개발	<ul style="list-style-type: none"> • 상용 프로세서 및 DSP 에 대한 분석 • 선진 음성 인식용 프로세서 아키텍처 비교 • 프로세서 최적화 설계를 위한 방법론에 대한 기술 분석 및 추적 • 설계 틀 환경 구축 	100
	멀티 프로세서 실험 방법론 개발	<ul style="list-style-type: none"> • 상/하위 수준에서의 시뮬레이션 환경 구축 • 시제품 보드 설계를 위한 환경 구축 	100
	멀티 프로세서 검증 방법론 개발	<ul style="list-style-type: none"> • 테스트를 고려한 최적화 설계 방법론 분석 • 알고리즘 및 아키텍처/로직 수준에서의 설계 검증 환경 구축 (HW/-S/W 통합설계에 대한 기술 이해) 	100
2차년도	알고리즘 분석	<ul style="list-style-type: none"> • 음성인식 알고리즘 중 음성특성치 비교검색을 위한 부분 DTW/HMM 분석 완료 • 하드웨어 기본 알고리즘 분석 및 방안 연구 • 50단어 화자독립/어휘독립 인식기 알고리즘에 대한 Floating-point Operation 및 연산량 분석 	100
	데이터 분석	<ul style="list-style-type: none"> • 음성 인식 알고리즘의 Fixed-point System Modeling 및 검증 • 50단어 화자독립/어휘독립 인식기의 데이터 분석을 통한 기본 프로세서 아키텍처 개발 	100
	데이터 분석 검증	<ul style="list-style-type: none"> • 시제품 보드를 이용한 Fixed-point System Model의 설계 및 검증 • HCI 응용 소프트웨어 솔루션 마련 	100
	마이크로 아키텍처 연구	<ul style="list-style-type: none"> • 3차년도 계획분인 마이크로 아키텍처 기반 기술 선행 연구 • 글로벌 사이버 연구센터 구축 	20

V. 연구개발결과의 기대효과 및 활용계획

1. 기대효과

- 가. 프로세서 설계 환경 구축을 통한 개발 기간 단축 및 설계의 재활용
- 나. 프로세서 설계에 대한 기반 기술과 Know-how 확보를 통한 개인별 저전력 /고성능 설계 기술 향상
- 다. 시제품 보드를 이용한 신속한 하드웨어 검증
- 라. 음성 인식 알고리즘 분석을 통한 하드웨어 분석 능력 증가
- 마. 음성 인식 알고리즘 고정 소수 모델링을 통한 하드웨어 모델링 능력 증가
- 바. 향후 HCI 프로세서 설계에 대한 기반 기술 확보

2. 활용계획

- 가. HCI 칩내의 일부 기능 블록(필터, 명령어 등)에 대한 부분별 선행 설계 및 검증
- 나. System On a Chip 개발 환경에 활용
- 다. 음성 인식 전용 프로세서의 경우 사업화 부분에 기여할 수 있음
- 라. 음성 인식 전용 프로세서의 Preprocessing부분의 경우 음성 관련 전처리 프로세서로 활용할 수 있음
- 마. 시제품 보드는 다양한 음성이나 오디오 처리 응용 제품들의 시제품 보드로도 활용 가능

S U M M A R Y

The purpose of the HCI project is obtaining both the chip solution based on the digital signal processor (DSP) architecture for the implementation of the HCI specific ASIC and the methodology for the development and the design for the HCI DSP. Doing the design and development of the HCI DSP, a lot of data to be processed and the real-time operation have to be considered in the technological aspect so that the HCI system can obtain the high performance and it can be used conveniently in the practical environment. Furthermore, one chip solution, which is called SoC (System on a Chip) that contains processing unit, memory unit, I/O unit, and the specific intellectual property, is emerging to satisfy the needs of the end-users. In this project the HCI DSP based on the state-of-the-art implementation such as HMM engine, FFT engine, and low power architecture is developing. Specific engines are to gain the high performance and the low power technology is to save the power consumption of the HCI system so that it can be used with the small battery.

The range of the research and development for the HCI DSP is presented in the section 3 of the first chapter. The second chapter presents the state of the national and international R&D related to the HCI DSP. In the third chapter, the results of R&D for HCI DSP are represented in the R&D stage aspect. The R&D of the multi-processor co-design methodology such as the analyses of both the commercial micro processor and digital signal processor, how to design the optimized low power architecture for the specific application field, and etc. was executed in the first stage. In the second stage, the HCI DSP architecture was set up exploiting the analyses of the HCI algorithm and the data structure for the independent speakers, the fixed-point digital signal modeling, and the industrial-educational cooperation system. The achievement of the R&D for the HCI DSP and the contribution to the related technological fields are presented in the fourth chapter. Finally, using the results of HCI R&D, that one can apply a new process in industry and its effect were mentioned in the fifth chapter.

C O N T E N T S

Chapter 1 Introduction -----	10
Section 1. The object of the research and development -----	10
Section 2. The necessity of the research and development -----	10
Section 3. The range of the research and development -----	12
Chapter 2 The current state of the national/international technology development -----	14
Section 1. The current state of the national/international technology development related to the HCI DSP -----	14
Section 2. The place of R&D results in the state of the national/international technology development -----	15
Chapter 3 The contents and results of the research and development-----	16
Section 1. The contents of the research and development -----	16
Section 2. The results of the research and development -----	26
Chapter 4 The achievement and the contribution to the related technological fields -----	43
Chapter 5 Plan of the practical approach exploiting results of R&D -	45
Section 1. Application field and expectation effect of annual results of the HCI R&D -----	45
Chapter 6 References -----	47

목 차

제 1 장 서 론.....	10
제 1 절 연구개발의 목적.....	10
제 2 절 연구개발의 필요성.....	10
제 3 절 연구개발의 범위.....	12
제 2 장 국내외 기술개발 현황.....	14
제 1 절 국.내외 관련분야에 대한 기술개발현황.....	14
제 2 절 연구결과가 국.내외 기술개발현황에서 차지하는 위치.....	15
제 3 장 연구개발수행 내용 및 결과.....	16
제 1 절 연구개발 수행 내용.....	16
제 2 절 연구개발 수행 결과.....	26
제 4 장 연구개발목표 달성도 및 대외기여도.....	43
제 5 장 연구개발결과의 활용계획.....	45
제 1 절 연차별 연구결과 활용분야 및 기대효과.....	45
제 6 장 참고문헌.....	47

제 1 장 서 론

제 1 절 연구개발의 목적

1. HCI 프로세서 개발

가. One Chip (800Mips급)

나. 저전력 (100mW)

2. SOC (System On Chip) 설계 방법론 확보

제 2 절 연구개발의 필요성

1. 연구개발의 과학기술, 사회경제적 중요성

가. 기술적 측면

(1) 전 세계적으로 HCI Algorithm을 산업체와 학계에서 꾸준히 연구/개발해 왔다. HCI의 H/W Implementation을 위해서는 대량의 Data Processing과 Real Time Operation등을 고려해야 함

(2) 이 고려 사항들을 충족시키기 위해서는 Digital Signal Processor의 Architecture 기술과 HCI 전용Engine 설계 기술의 지속적인 연구/개발을 통해 HCI 단일 Chip Solution을 제시해야 함

나. 경제·산업적 측면

(1) 금년은 HCI Chip Solution의 원년이라고 할 만큼 이와 관련된 시장이 급속히 성장하고 있다. 이에, HCI 시장을 선점/Lead 하기 위해서는 가격 경쟁력과 응용 범위가 넓은 Digital Signal Processor구조의 단일 전용 Chip Solution이 시장 선점에 유리함

다. 사회·문화적 측면

(1) 정보통신과 반도체 산업의 발달로 개인 뿐만 아니라 사회도 많은 변화가 있으며, 이로 인한 새로운 형태의 문화가 도래하고 있다. 정보 단말의 개인화, 소형화, 휴대화와 반도체의 Low Power Consumption, High Performance등을 통한 시간과 공간의 제약 없이 정보를 제공/습득/공유 함에 있어 편리성이 절실히 요구되고 있음

2. 앞으로의 전망

가. 휴대용 전자분야의 중요 기술 전망

- (1) 휴대용 전자분야의 구성은 중요 기술인 핵심부품 개발을 근간으로 이루어지고 있다.
- (2) 핵심 부품 개발은 미세 가공기술, 저전력기술, 고성능 구조기술을 이용하여 이루어진다.
- (3) SOC(System On a Chip)의 구현은 핵심기술을 이용한 One Chip화를 통해 휴대용 기기의 필수 조건인 경박단소화 및 저전력소비의 요건을 만족시킬 수 있다.

나. System On a Chip(SOC)

반도체의 고집적화, 기능의 복합화에 따라 향후 System은 단일 Chip에 거의 모든 기능을 집적시키는 방향으로 전개되며, 이를 구현 할 수 있는 기술력이 경쟁력의 가장 중요한 쟁점이 되고 있다.

- (1) SOC의 정의 : 시스템을 구성하는 모든 전자 부품들을 하나의 반도체 칩으로 집적한 결과 및 기술

System-on-a-chip의 정의: A highly integrated monolithic chip that contains processing(DSP, MPU, etc.), memory, and application specific intellectual property. This device will define the basic function of the electronic system.

* source: Dataquest

- (2) SOC의 필요성

- (가) 반도체 공정의 발달 및 제품 기술의 발달에 따른 제품의 성능과 기능 향상
- (나) 새로운 기능과 다양한 기능에 대한 소비자들의 요구
- (다) 휴대화 및 소형화에 대한 소비자들의 욕구
⇒ 하나의 칩 안에 제품 구성에 필요한 모든 전자 부품들의 집속화 필요

- (3) SOC 추진 방향

- (가) 설계자의 원하는 사양에 맞춰 System On a Chip화 가능
- (나) 소비자들의 원하는 형태의 SOC 가능

제 3 절 연구개발의 범위

구분	연구개발목표	연구개발내용 및 범위
1차년도	Multi-processor or Design Methodology Setup	<ul style="list-style-type: none"> • Multi-processor Co-design/Sim./Verification Design Methodology 개발
2차년도	Fixed-point System Modeling	<ul style="list-style-type: none"> • Fixed-point System Modeling <ul style="list-style-type: none"> - 알고리즘분석 - 데이터 분석 - 데이터 분석 검증
3차년도	Micro Architecture 개발	<ul style="list-style-type: none"> • 화자독립/어휘독립 50단어 인식을 위한 ISA(Instruction Set Architectures) 설정 • 64개 ISA를 Compile/Debugging하기 위한 Assembler 개발 • 22.4KB Assembly Coding(1Word의 Instruction Length : 24 bits) • 12만 Gates급 Processor의 H/W 설계 • JTAG을 이용한 Test Logic 설계
	설계 및 검증	<ul style="list-style-type: none"> • 3차년도 1차 Chip제작 (화자독립/어휘독립 50단어 인식을 수행하는 Processor => Performance : 20 Mips , 전력소모 : 80 mW, 제작공정 : 0.35um) • Chip을 Test하기 위한 Test Board 제작 • Board Level의 Chip Debugging & Analysis • 3차년도 2차 Chip제작 (화자독립/어휘독립 50단어 인식을 수행하는 Processor => Performance : 20 Mips , 전력소모 : 50 mW, 제작공정 : 0.25um) • 3차년도 2차 제작 Chip의 Test 및 Evaluation

구분	연구개발목표	연구개발내용 및 범위
4차년도	Proto System 개발	<ul style="list-style-type: none"> • 음성합성압축 / 휴먼인식 알고리즘 분석 및 Fixed-point Modeling • v음성인식, 음성합성압축, 휴먼인식을 위한 ISA(Instruction Set Architectures)설정 • 64개 ISA를 Compile/Debugging하기 위한 Assembler 개발 • 50.4KB Assembly Coding(1Word의 Instruction Length : 24 bits) • 100만 Gates급 Processor의 Full Custom H/W 설계 • JTAG을 이용한 Test Logic 설계
	Chip Integration	<ul style="list-style-type: none"> • 4차년도 1차 Chip제작(음성인식, 음성합성압축, 휴먼인식을 수행하는 Processor => Performance : 800 Mips , 전력소모 : 170 mW, 제작공정 : 0.18um) • Chip을 Test하기 위한 Test Board 제작 • Board Level의 Chip Debugging & Analysis • 4차년도 2차 Chip제작(음성인식, 음성합성압축, 휴먼인식을 수행하는 Processor => Performance : 800 Mips , 전력소모 : 100 mW, 제작공정 : 0.15um) • 4차년도 2차 제작 Chip의 Test 및 Evaluation

제 2 장 국내외 기술개발 현황

제 1 절 국·내외 관련분야에 대한 기술개발현황

1. 외국의 경우

가. 자료 출처: ISSCC'00

나. 관련기술군: RISC Processor 설계기술/Low Power 설계기술

다. 연구 기관: Hitachi Ltd.(일본)

라. 연구 테마: A 450MHz 64b RISC Processor using Multiple Threshold Voltage CMOS

마. 적용시스템: 저전력 시스템

바. 주요 특징: (1) 공정: 0.25 um, Vdd : 1.8 V

(2) Multiple-threshold-voltage design with minimum standby current

(3) Multiple-threshold-voltage technique can Apply static circuits, clock-distribution drivers, register files and dynamic circuits in RAM macros

(4) Chip Size : 8.3 M Logic-gate Transistors, 20 M RAM Transistors

사. 주요 효과: (1) 저전력 시스템 구현(Multiple-threshold-voltage)

(2) 64bit RISC Processor 구현

2. 국내의 경우

가. 자료 출처: DAC'00

나. 연구 기관: 서울대학교 전자공학과

다. 연구 테마: Bus Encoding for Low-Power High-Performance Memory Systems

라. 적용시스템: 빠른 속도이고, Burst-mode로 Data를 전달하는 Memory System에 적용 가능

마. 주요 특징: New Bus-inversion Decision 기법 개발

바. 주요 효과: 저전력 시스템 구현(Random Data : 평균 14%까지 소비전력 절감, Trace Data : 평균 67.5%까지 소비전력 절감)

3. 조사 연구개발사례에 대한 평가

가. 외국의 경우: 미국을 비롯한 유럽 및 일본에서는 시스템의 One-Chip화 설계기술과 시스템 레벨의 저전력화 방법 연구 및 이를 실용화하는 단계에 이르기까지 유관 기술들이 급진전된 것으로 판단됨.

나. 국내의 경우: 학계 및 일부 기업의 연구소를 중심으로 관련 기술들이 연구되고 있으나 현재 선진국과의 기술격차는, 기술분야에 따라 약간의 차이는 있지만, 적지 않은 차이를 나타내고 있는 실정임. 그러나 이미 확보되어 있는 차별화된 Low Power 설계기술 및 우수한 ASIC 설계기술을 활용하면 이러한 기술격차는 쉽게 극복되리라 판단됨.

제 2 절 연구결과가 국·내외 기술개발현황에서 차지하는 위치

제품·기술명	선진국 수준	국내 수준	현재 기술격차
Multi-embedded Core 내장 기술	DSP, RISC등 복수개의 Embedded Core 내장 실용화 확대	DSP 또는 RISC 단일 Core 내장 실용화 단계	1년
DSP/RISC Programming 기술	DSP/RISC 응용 기술 일반화	C/C++, Assembly Coding을 통한 응용 기술 확보	0년
H/W 설계 기술	100만 Gates급 이상의 ASIC 개발 확대	50-100만 Gates 규모의 ASIC 개발 기술 확보	0.5년
Low Power 설계 기술	Low Power 설계 기술 실용화 적용 단계	Low Power 설계기술 실용화 시도 단계	1년
ASIC 공정 기술	0.2 μ m이하 실용화 단계	0.25 μ m 실용화 단계	1년

제 3 장 연구개발수행 내용 및 결과

제 1 절 연구개발 수행 내용

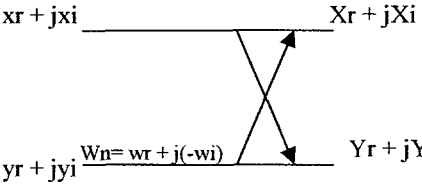
연구 범위	연구수행방법	연구 내용
프로세서 설계 기반 연구 (1차년도)	<ul style="list-style-type: none"> • 상용 프로세서 및 DSP 에 대한 분석 	<ul style="list-style-type: none"> • 상용 프로세서 분석 기존 선진회사의 프로세서에 대한 조사와 분석을 하였다. Intel사의 마이크로 프로세서들과 ARM사에서 제공되는 프로세서에 대한 조사와 분석이 이루어졌다. 이러한 조사와 분석을 통해 앞으로의 프로세서 설계 기술을 전망하였다. • DSP 분석 DSP 프로세서의 선두기업인 TI사와 다른 선진회사인 Motorola사, DSP Group의 각 DSP를 분석하였다. 또한 이들의 문제점을 파악하였다.
	<ul style="list-style-type: none"> • 저전력 DSP 프로세서 설계기반기술 습득을 위한 자체 기술 세미나 	<ul style="list-style-type: none"> • 프로세서 설계에 대한 지식과 설계기술을 향상시키기 위하여 프로세서 설계에 대한 기본 연구가 진행되었다. 프로세서 설계의 세부적인 기술 내용과 설계관련 Tool에 대해 세미나가 이루어졌다. 응용테마의 해석을 통하여 상위시스템 사양 분석에서부터 하드웨어 구현에 대한 내용이 진행되었고, 응용테마의 알고리즘을 검증하는 단계별 방법에 대한 기술세미나가 이루어졌다. 이렇게 검증된 응용테마의 하드웨어설계 방법 및 실험 방법에 대한 내용이 발표되었다.
	<ul style="list-style-type: none"> • 응용테마를 이용한 설계 	<ul style="list-style-type: none"> • QMF (Quadrature Mirror Filter) 음성처리방법에 있어서 QMF (Quadrature Mirror Filter) 방법은 주어진 대역을 low, high 두 개의 대역으로 분할한 후, 분할된 두 개의 대역을 다시 각각 low, high 두 개의 대역으로 나누는 과정을 필요로 하는 대역 수가 이루어질 때까지 반복한다. QMF 를 특정용도 프로세서 구조로 개발하였다. 그러한 접근방법으로 하드웨어 자원을 보다 효율적으로 활용하여 저전력화를 할 수 있다. 별첨1. QMF 전체 블록도

연구 범위	연구수행방법	연구 내용
<p>프로세서 설계 기반 연구 (1차년도)</p>	<p>• 응용테마를 이용한 설계</p>	<p>• 선형예상 분석프로세서 음성신호 전달 시, 적은 정보량으로 정확한 데이터를 송신하기 위하여 기법을 사용한다. 이 방법은 송신된 신호를 왜곡 없이 복원해 낼 수 있는 음성파라미터 즉, 예상계수를 정확하게 추출해낼 수 있는 능력과 빠른 연산능력을 지닌 방법이다. 이를 수행할 수 있는 분석 프로세서를 구현하였다.</p> <p>별첨2. 선형예상 분석 프로세서 설계</p> <p>• FIR Low Pass Filter Passband 주파수 보다 낮은 주파수를 갖는 입력신호를 출력하고, stopband 주파수보다 높은 주파수를 갖는 입력신호는 통과시키지 않는 특성을 가지는 Finite-impulse Response (FIR) 필터를 설계하였다.</p> <p>별첨3. FIR LP 필터 설계 별첨4. FIR LP 필터 설계 칩 구조 별첨5. FIR LP 필터 전체 블록도</p>

연구 범위	연구수행방법	연구 내용
Tool Path Setup (1차년도)	<ul style="list-style-type: none"> 알고리즘 수준에서의 설계 및 검증 	<ul style="list-style-type: none"> 알고리즘 수준에서 시스템의 기능을 검증하고 성능을 예측함으로써, 효과적으로 동작하는 시스템을 설계, 검증할 수 있다. 또한 설계의 비용과 시간도 단축할 수 있다. 따라서 시스템 모델링과 해석이 용이한 알고리즘 수준에서의 설계검증방법을 확립하는 것은 중요하다. 기존에 나와있는 알고리즘 수준에서의 설계, 검증 Tool을 분석하여, 효율적인 설계/검증 환경을 구축하였다.
	<ul style="list-style-type: none"> Architecture/Logic 수준에서의 설계 및 검증 	<ul style="list-style-type: none"> Architecture 수준에서의 설계, 검증을 위한 환경구축의 일환으로 이 수준에서의 설계환경을 제공하는 Tool들을 분석하였다. RTL, Schematic, Layout, PCB 설계 관련 Tool들과 트랜지스터 수준에서의 회로 검증, 설계된 회로의 전력을 예측하는 Tool들에 관해 조사하고, Architecture 수준에서의 설계를 위한 Tool Path를 구축하였다.
	<ul style="list-style-type: none"> Proto-typing Tool Setup 	<ul style="list-style-type: none"> 프로세서 Proto-type 프로세서의 기능 설계 검증 및 실험을 위한 Tool들을 조사하였다. 멀티프로세서 Architecture 개발은 성능뿐만 아니라 저전력 구동성도 견비되어야 한다. 이러한 이유를 구현하는데 모델이 될 수 있는 기존에 상품화된 프로세서 Architecture 중 ARM 프로세서를 선정하였다. DSP(Digital Signal Processing) Proto-type DSP 분석을 위한 도구로는 TEAK, TMS320C 등이 있다. 이들에 대한 자료를 분석한 후, 사용에 적합한 분석도구를 결정하였다. FPGA를 이용한 Proto-type FPGA를 이용한 Proto-type 설계를 위한 프로그램으로 Xilinx 의 M1, Altera 의 MAXPLUSII 등이 있다. 각 프로그램에 대한 비교 검토가 이루어졌다.
	<ul style="list-style-type: none"> Circuit 및 저전력 Custom Design 	<ul style="list-style-type: none"> 회로의 설계 방법은 Static 회로 설계, 다이내믹 회로설계 등 다양하다. 저전력화, 고성능화를 적용시키기 위해서는 목적에 적합한 설계방법을 선택하여 사용해야 한다. 또한 저전력 회로 설계 시 설계의 최적화를 위해서는 Custom 설계가 필요하며, Layout 과 같은 하위수준에서의 설계를 수행하게 된다. 이와 관련한 기술세미나와 교육이 진행되었다.

연구 범위	연구수행방법	연구 내용
<p>최적화 설계를 위한 방법론에 대한 기술 분석 (1차년도)</p>	<p>• 테스트 및 디버깅이 용이한 설계를 위한 설계방법론에 대한 기술 분석</p>	<p>• 테스트가 용이한 설계 방법론 테스트가 용이한 설계 방법론에 대한 기술조사를 의뢰하였다. 상용 VLSI 칩들에 사용된 테스트가 용이한 설계기법들에 대한 문헌조사, 기능 블록들을 테스트 기법의 차이에 따른 분류에 의한 테스트 기법 적용 전략에 대하여 조사되었다. 당 랩에서 조사된 내용을 분석하여 실제 칩 설계 시 테스트를 고려하여 설계하고자 한다.</p> <p>• 디버깅이 용이한 설계 방법론 디버깅이 용이한 설계 방법론은 개발 중인 VLSI 칩을 위한 소프트웨어 및 응용보드를 검증하고 개발하는데 필요한 환경이 된다. 일반적으로 양산용 VLSI 칩은 Fab.을 하기 전에 Emulation 과정을 통해 검증하며, 이 과정에서 디버깅이 설계를 통하여 VLSI 칩들을 쉽게 검증 할 수 있게 된다. 또한 이러한 VLSI 칩은 Fab. Out이 되기 전에 디버깅이 설계는 Source-level Debugging 시스템 소프트웨어를 하드웨어 설계와 동시에 수행 할 수 있게 해 준다. 설계 시에도 테스트를 고려하여 설계하여 디버깅 시간과 비용을 절감할 예정이다.</p>
	<p>• 하드웨어-소프트웨어 통합설계에 대한 기술 이해</p>	<p>기술협력으로 서울대와 연구가 진행 되었고 향후, 본 과제가 진행됨에 따라 필요한 기술임으로 선행하여 준비중</p> <p>• 시스템 설계 시 효율적인 설계 및 검증을 위하여 하드웨어-소프트웨어(H/W-S/W) 통합설계환경을 구축하는 것은 매우 중요하다. 하드웨어-소프트웨어 통합설계를 위해서는 시스템의 각 부분을 여러 계산 모델을 사용하여 표현 할 수 있어야 한다. 이렇게 표현된 전체 시스템은 설계의 각 단계에서 수행가능 해야 한다. 또한 하드웨어-소프트웨어를 설계의 각 단계에서 통합실험을 할 수 있어야 하며, 시스템의 Prototyping이 빠른 시간 내에 가능해야 한다. 이러한 하드웨어-소프트웨어 통합설계 환경을 이해, 적용함으로써 앞으로의 설계시간을 단축할 수 있을 것이다.</p>

연구범위	연구수행방법	연구 내용
알고리즘 분석 (2차년도)	<ul style="list-style-type: none"> • 선택 배경 <ul style="list-style-type: none"> - HCI Processor 기능중 1단계 사양으로 적당한 규모 - 시장규모의 고성장 예상 - 음성인식 알고리즘의 확보 - 확장성을 고려한 기본 아키텍처 개발 	<ul style="list-style-type: none"> • 예상 Chip Size의 규모 HCI Processor의 기능중 음성인식을 H/W로 구현할 경우 20만 Gates 전후의 크기로 구현가능. (참조 : 출시된 음성인식 전용칩(Sensory RSC-364)을 분석한 결과) 음성인식용 칩의 H/W 구현이 One Chip Solution의 HCI Processor 구현을 위한 1단계 H/W 사양으로 적당함 • 음성인식 칩의 시장 규모 고성장 예상 가장 시장성이 높으며 세계 음성인식 시장이 1999년 이후 2005년까지 매년 2배 이상의 고성장 예상 • 음성인식 알고리즘의 확보 HCI 프로세서 사업단 내의 구어이해기술 과제에서 제공된 화자독립/어휘독립 50단어 인식기 알고리즘 확보 • 확장성을 고려한 기본 아키텍처 개발 음성인식을 위해 개발된 H/W 아키텍처는 다른 HCI기술에도 적용 가능하다
	<ul style="list-style-type: none"> • 음성 인식 기본 알고리즘 분석 	<ul style="list-style-type: none"> • 단어의 끝점 검출을 포함한 Preprocessing 분석 및 연구 화자독립/어휘독립50단어 알고리즘에서 Preprocessing 알고리즘을 분석하였다. Preprocessing의 분석은 데이터 처리를 위한 연산기의 종류와 각 수행단계마다 연산의 과정을 분석하였다. 그리고 입력되는 데이터 중 음성만을 검출하는 끝점 검출은 어떤 방식으로 단어의 끝점 추출이 이루어지는지 분석하였고, 이것이 현재 개발될 제품에 적합한 방식인지, 개량이 필요하면 어떻게 해야 할 지에 관한 분석도 수행하였다. Preprocessing의 동작에 대한 분석 후 Fixed-point 개발용 Tool인 SPW(Signal Processing Workbench : Cadence)를 이용하여 재 Modeling하여 Data Bus Width에 따른 SNR(Signal to Noise Ratio)을 분석, 검증하였다. 그 결과 Preprocessing 단계에서 수행되는 동작과 이를 위해 H/W 구현시 필요한 연산기 및 그에 필요한 Data Bit폭을 파악 할 수 있었다.

연구범위	연구수행 방법	연구 내용
<p>알고리즘 분석 (2차년도)</p>	<p>• 하드웨어 기본 알고리즘 분석</p>	<p>• Preprocessing의 ASIC화 음성인식 알고리즘 분석에 의해 Preprocessing에서 수행되는 연산수가 72,500 / 10 msec(80 Samples) 임을 알 수 있다. 이중 FFT(Fast Fourier Transform)가 차지하는 연산량이 99%이상을 차지한다. 따라서 FFT 연산의 기본 구조인 Butterfly 연산기를 하드웨어적으로 제작, 탑재하면, 그 결과 Preprocessing 연산을 더욱 빠르게 수행하고, 소프트웨어적인 연산량을 줄임으로써 CPU의 동작 주파수를 줄여 칩을 저전력화에 기여하도록 하였다.</p> <p>- 버터플라이 연산</p> $X_r = x_r + w_r \cdot y_r \quad w_i \cdot y_i$ $X_i = x_i + w_i \cdot y_r + w_r \cdot y_i$ $Y_r = x_r - w_r \cdot y_r + w_i \cdot y_i = 2x_r - X_i$ $Y_i = x_i - w_i \cdot y_r - w_r \cdot y_i = 2x_i - X_i$ <div style="text-align: center;">  </div> <p>그림 1. 버터플라이 연산 구조</p> <p>• HMM Searching Engine 음성 인식 알고리즘에서 전체 연산량의 3/4을 차지하는 곳이 HMM부분이다. HMM은 기본적으로 음소단위로 각 음소의 관측확률과 상태전이 확률을 계산하며, 이때 사용되는 연산량은 약 380,000 / 10msec(80 Samples)이다. 이런 HMM의 주요 연산 순서는 다음과 같다. 관측확률 계산은 곱셈 -> 덧셈 -> 뺄셈 순이고, 상태전이 확률 계산은 덧셈 -> 비교 순이다. 이와 같은 HMM을 전용 하드웨어로 구현하면, 전체 시스템의 연산량을 줄여 저전력화 및 연산 속도를 향상시킬 수 있었다.</p>

연구범위	연구수행 방법	연구 내용
데이터 분석 (2차년도)	<ul style="list-style-type: none"> • 화자독립/어휘독립 음성 인식 알고리즘 연산량 분석 	<ul style="list-style-type: none"> • 계산량 분석 화자독립/어휘독립 50단어 음성인식 알고리즘에서 데이터 처리에 필요한 중요 연산기로는 곱셈, 나눗셈, 덧셈, 제곱근, 로그연산으로 구분하였다. 또한 연산을 간소화 위해 상수 나눗셈을 상수 곱셈으로 치환 될 수 있는 것에 대한 분석도 하였다. 그리고 효율적인 H/W 아키텍처를 구현하기 위하여 알고리즘의 흐름별로 연산량을 분석하였다. 그 결과 System의 복잡도를 예측 할 수 있었고, H/W 아키텍처를 구상하는 기초자료로 활용하였다. • 메모리 사용량 분석 화자독립/어휘독립 50단어 음성인식 알고리즘의 수행 흐름별로 메모리 필요량을 분석하였다. 프로그램은 최적화 했다고 가정했을 경우 필요한 고정 및 변동 메모리량 계산하였다. 또한 사용된 메모리의 목적을 분석하여 ROM으로 구현할지 혹은 RAM으로 구현할지에 대한 기초 분석을 행하였다. 그 결과 H/W 구현시 필요한 종류별 메모리량을 예측 할 수 있었고, 메모리 Map 작성시 기초자료로 활용하였다. <ul style="list-style-type: none"> - 메모리 사용량/Word 고정 : 118.4 KB 변동 : 10 KB
	<ul style="list-style-type: none"> • 음성 인식 알고리즘 Fixed-point System Modeling 전환 	<ul style="list-style-type: none"> • SPW를 이용한 Fixed-point System Modeling 전환 음성인식 알고리즘을 SPW의 Floating-point Library를 이용하여 구성된 후 동작을 검증하였다. 이 Design을 SPW 상에서 다시 Fixed-point Library를 이용하여 재구성하였다. 이렇게 재구성된 Fixed-point Model의 Data Width를 변화 시켜가면서 시뮬레이션을 반복하였다. 그 결과 각 부분에서 Floating-point Model과 Fixed-point Model간에 어느 정도의 신호차이가 있는지, 또는 이 신호차가 음성 인식률에 어떠한 영향을 미치는 가를 예측할 수 있었다.

연구범위	연구수행 방법	연구 내용
데이터 분석 (2차년도)	• 기본 아키텍처 개발	<ul style="list-style-type: none"> • 인식기를 위한 고정소수 연산 동작에 필요한 아키텍처 & ISA(Instruction Set Architectures) 개발 <p>음성인식 알고리즘을 데이터 전송, 연산 단위 명령으로 나누고, 1사이클로 실행이 가능한 연산은 하나의 명령으로 추출하였다. 추출한 명령 종류는 데이터 전송, 산술연산, 논리연산, 제어, HMM의 특수명령으로 나누었다. 이런 ISA를 실행할 수 있도록 레지스터-레지스터 연산을 기본으로 레지스터-메모리 연산을 가능하게 하고, 메모리 전송이 필요한 부분에서 메모리-메모리 전송이 가능한 하버드 아키텍처를 기본으로 하는 3버스 구조의 아키텍처를 개발하였다. 이 아키텍처에서는 데이터 분석에 따라 블록 반복기능을 2중 루프가 가능하도록 하였고, 데이터가 C언어의 구조체형식을 사용하여, 2차원의 배열과 구조체를 지원할 수 있는 1쌍의 인덱스 레지스터에 의한 데이터 어드레싱 기능을 가능하게 하였다. 그 결과 음성인식을 위한 ASSP(Application Specific Signal Processor) 기본 아키텍처를 개발할 수 있었다.</p>

연구범위	연구수행 방법	연구 내용
<ul style="list-style-type: none"> • 데이터 분석 검증 (2차년도) 	<ul style="list-style-type: none"> • 시제품 보드 구현/검증 	<ul style="list-style-type: none"> • Preprocessing(Pre-emphasis) Fixed-point System Model구현 및 시제품 보드 구현 <p>음성인식을 위한 Preprocessing중 Pre-emphasis는 입력된 음성 Data의 고주파 영역을 보다 강조하는 기능을 하는 것으로 High Pass Filtering한 결과와 비슷한 결과가 예상된다. 이러한 동작을 하는 Preprocessing(Pre-emphasis)을 FPGA를 이용하여 Processor 구조를 가진 Prototype H/W로 구현/검증을 하였다. 그 결과 Preprocessing에 적합한 H/W 아키텍처를 검증할 수 있었다.</p>
	<ul style="list-style-type: none"> • HCI 응용 소프트웨어 구현을 위한 모델링 (서울대 기술협력) 	<ul style="list-style-type: none"> • HCI 관련 신호처리 알고리즘 모델링 및 일반 마이크로프로세서의 HCI 응용 구현할 때의 방법론 <p>HCI(Human Computer Interface) 응용은 저전력(Low Power), 실시간 처리(Real-time Processing), 그리고 높은 계산요구(High Computational Demand)라는 특징을 갖는다. 이러한 HCI 응용에 있어 범용 마이크로 프로세서 기반 아키텍처에서 HCI 응용 소프트웨어를 빠르게 개발하는 방법론을 개발하여 Assembly Coding 작업 효율을 높일 수 있었다.</p>
<ul style="list-style-type: none"> • 3차 년도 마이크로 아키텍처 기반 기술 선행 연구 (2차년도) 	<ul style="list-style-type: none"> • 마이크로 아키텍처 사양 결정을 위한 선행연구 	<ul style="list-style-type: none"> • 세계 음성인식 시장조사 <ul style="list-style-type: none"> - 세계 음성인식 관련 시장 현황 및 전망 조사 <ul style="list-style-type: none"> ① 목적 : 마이크로 아키텍처 사양 결정을 위한 시장조사 ② 자료 조사 방법 : 전문 조사기관 Computer Economics사(URL: http://www.computereconomics.com)에서 자료 발취 - 국내/외 Toy업체 방문 조사 <ul style="list-style-type: none"> ① 목적 : 음성인식 전용 Processor의 국내/외 수요 예측 및 개발사양 수립시 고려 ② 참여 박람회 : <ul style="list-style-type: none"> The Salon International du Jouet, Paris,/The British International Toy & Hobby Fair, London ③ 방문 업체 : 손오공, 영실업, 대도실업 ④ 상담 대상 : 마케팅 담당자 및 개발 담당자 ⑤ 조사 방법 : 사전 준비된 설문항목 질의 및 협의

연구범위	연구수행 방법	연구 내용
3차 년도 마이크로 아키텍처 기반 기술 선행 연구 (2차년도)	<ul style="list-style-type: none"> • 마이크로 아키텍처 사양 결정을 위한 선행연구 	<ul style="list-style-type: none"> • 세계 음성인식 시장 조사 결과 <ul style="list-style-type: none"> - 단기전략 (시장진입) : 저가형 음성인식 제품 (Toy, Remote Control, TAD 등) - 중기전략 (시장확대) : 고부가 가치형 음성인식 제품 (Hand-Held Device, Intelligent System 등)
	<ul style="list-style-type: none"> • 소비전력 최소화를 위한 마이크로 아키텍처 선행연구 	<ul style="list-style-type: none"> • 음성신호의 소비전력 최대 소모 부분인 메모리와 버스 구조의 개선 <p>VLSI 설계기술과 공정기술의 발달로 인하여, 고집적화로 인한 한 칩내의 Component수가 증가하게 되고, 이들 Component간을 연결하는 Net수가 증가 될 수 밖에 없게된다. 따라서 H/W적으로 한정된 Net수에 Data의 Transition을 적게 하는 Coding기법을 사용하여, Data가 진행되는 Bus Line에서의 Power소모를 줄이려고 하였다. 그 결과 HIHR(Half Identity Half Reverse) Bus Encoding 방식을 제안하였다</p>
	<ul style="list-style-type: none"> • 산학연 협동을 위한 글로벌 사이버 연구 센터 구축 	<ul style="list-style-type: none"> • 글로벌 사이버 연구 센터 구축 <p>글로벌 사이버 연구 센터는 당원의 Mobile System Lab.을 중심으로 미국의 University of Utah, 서울대학교 및 성균관 대학교의 핵심 연구원들이 HCI Processor의 원활한 공동 개발을 위하여 운영되는 사이버 협력회이다.</p> <p>주 관심연구 대상은 System On a Chip, Low Power Design, HW/SW Co-design and Simulation 기술들이다.</p> <ul style="list-style-type: none"> - H/W-S/W 통합 설계 방법론 기술 자문 (서울대) - 고성능 프로세서 설계 기술 자문 (미, Utah대) - 저전력 관점에서의 레이아웃 설계 기술 자문 (성균관대)

제 2 절 연구개발 수행 결과

연구 방법	연구 결과
<ul style="list-style-type: none"> • 상용 프로세서 및 DSP 에 대한 분석 (1차년도) 	<ul style="list-style-type: none"> • 상용 프로세서를 생산하는 회사 (Intel, IBM, ARM, etc.)가 앞으로는 Multi-processor 환경을 제공하는 것으로 예정하고 있으며, 이를 위한 연구가 각 대학과 연구기관에서 활발히 진행되고 있다. 그리고, 연구 동향 및 시장 분석을 한 결과, 앞으로는 프로세서의 전망은 다음과 같은 특징을 가져야 할 것으로 판단된다. <ul style="list-style-type: none"> - Low Power - Reconfigurable - Scalable - Networked - Multiprocessor - Intelligent & Reliable • DSP는 TI, Analog Device, Motorola사와 같은 선진사의 DSP를 조사, 분석하였다. TI사의 TMS320C54x, Motorola의 DSP56000, DSP Group 의 Oak, Teak 등에 대한 조사, 분석이 이루어졌다. • DSP 에 대한 분석 결과는 현재 Embedded System에서는 Oak, Teak 계열이 많이 사용되며, 많은 회사에서 IP 로 제공되기까지 하였다. 특히 저전력용으로 새로운 DSP 가 많이 출시되었으며, 통신부분에 많이 이용되고 있다. VLIW 를 근간으로 한 프로세서의 경우, 필요 없는 연산들을 많이 수행하게 되므로 VLIW에서는 저전력이 적합하지 않다. 이를 기반으로 프로세서 개발에 활용하고자 한다.
<ul style="list-style-type: none"> • 저전력 DSP 프로세서 설계기반기술 습득을 위한 자체 기술 세미나 (1차년도) 	<ul style="list-style-type: none"> • 프로세서 설계기술/설계흐름을 습득하기 위하여 자체 기술 세미나가 4개월에 걸쳐 매주 진행되었다. 프로세서 설계 시 진행되는 각 단계에 관한 기술 세미나가 진행되었다. 또한, 각 단계에서 필요로 되는 프로그램 및 하드웨어 사용법에 대한 세미나도 함께 설명되었고, 고성능을 위해 사용되는 파이프라인 구조에 대한 내용도 설명되었다. 프로세서를 설계하기 위해서 알고리즘을 분석하고, C/C++언어, 명령어 (Instruction Set)와 마이크로 Architecture 설계를 통해 알고리즘을 검증하는 방법에 대한 기술적 설명이 이루어졌고, 검증된 알고리즘을 하드웨어로 설계하는 방법 및 구현된 하드웨어의 동작을 실험하고, 결과를 분석하는 방법에 대해서도 세미나가 이루어졌다.

연구 내용	연구 결과
<ul style="list-style-type: none"> 저전력 DSP 프로세서 설계기반기술 습득을 위한 자체 기술 세미나 (1차년도) 	<ul style="list-style-type: none"> 프로세서 설계 기반기술 습득 자체 세미나에서 진행된 내용은 다음과 같다. <ul style="list-style-type: none"> DSP 설계에 관한 소개 응용테마에 대한 해석 및 C/C++ 언어를 이용한 상위수준에서의 알고리즘 검증 및 구현 명령어 설계 어셈블러 설계 및 어셈블러를 이용한 알고리즘 검증 마이크로 Architecture 설계 및 파이프라인 구현 Architecture VHDL 를 이용한 데이터 Path 설계 VHDL 를 이용한 Control 설계 입/출력 핀 설계 설계된 특정 응용프로세서와 오디오 CODEC과의 인터페이스 설계 고장 검출 향상을 위한 테스트 로직 설계
<ul style="list-style-type: none"> 응용테마를 이용한 설계 (1차년도) 	<ul style="list-style-type: none"> 응용테마를 이용한 설계 <ul style="list-style-type: none"> 프로세서 설계 기반기술 습득을 위한 자체 기술 세미나 진행과 동일한 각 과정을 실제적으로 구현하였다. QMF 구현 결과 <ol style="list-style-type: none"> 2 대역 필터(0KHz~4KHz, 4KHz~8KHz) 각각 16-tap을 가지는 LPF (Low Pass Filter) 구조 구현 정수형 덧셈과 곱셈을 이용한 연산기 설계 12MHz의 동작주파수, 16KHz의 샘플링주파수 4단계(Instruction Fetch, Instruction Decode, Execution, Write Back) 를 가지는 파이프라인 구조 설계 삼성 라이브러리를 이용하여 설계한 경우, 약 54,000개의 게이트를 사용 <ul style="list-style-type: none"> 별첨6. 설계된 ALU관련 신호도 별첨7. 입력된 클럭 관련 신호도 별첨8. RAM 관련 신호도

연구 내용	연구 결과
<p>• 응용테마를 이용한 설계 (1차년도)</p>	<ul style="list-style-type: none"> • 선형예측 기법을 이용한 신호처리 프로세서 <ul style="list-style-type: none"> ① 13개의 음성예측 파라미터 추출 별첨 9. 추출된 Coefficient 값 ② 음성신호 파라미터(Pitch, Formants, Spectra, Vocal Tract Area Function)를 예측 ③ 곱하기는 MAC 구조를 사용 ④ 16 bits 나누기를 32 bits 빼기와 쉬프트를 사용하여 소프트웨어적으로 구현 ⑤ 3개의 Formants 주파수 특성을 갖는 신호를 테스트용 입력으로 사용 별첨10. 입력 신호 스펙트럼 별첨11. 출력 신호 스펙트럼 • LP Filter를 위한 특정응용 프로세서 개발 <ul style="list-style-type: none"> ① Altera 라이브러리를 사용하여 1,044개의 Logic 셀로 구현하였으며, Altera FPGA의 Logic부분은 20 %, 메모리부분은 33 %을 사용 ② 삼성 라이브러리를 이용한 경우, 53,630개 게이트로 설계 ③ 동작주파수는 12 MHz, 샘플링 주파수는 16 KHz, 29 Tap을 가지는 필터 구현 ④ Passband 주파수는 5 KHz이고, Stopband 주파수는 6 KHz ⑤ Passband Ripple은 3 dB, Stopband Attenuation은 6 dB ⑥ 4단계의 파이프라인을 가지는 프로세서 구조 ⑦ 기존에 보유하고 있던 FPGA Board을 이용하여 설계된 프로세서의 기능 검증 별첨12. FIR LP 필터의 입/출력 관련 타이밍도 별첨13. FIR LP 필터의 파이프라인 관련 타이밍도

연구 내용	연구 결과
<p>• 알고리즘 수준에서의 설계 및 검증 (1차년도)</p>	<ul style="list-style-type: none"> • 기존의 알고리즘 수준에서의 설계, 검증을 위한 프로그램의 특징 조사 및 비교 <ul style="list-style-type: none"> - COSSAP : <ul style="list-style-type: none"> ① Multi-rate, 비동기시스템에서의 표현 용이 ② Data-driven 방식으로 운용되므로 검증 속도가 빠름 ③ 상위수준 (Abstraction)에서 하드웨어/소프트웨어 모두 구현 가능 • SPW : <ul style="list-style-type: none"> ① Block-oriented Design ② Cycle-based, Synchronous-dataflow, Dynamic Dataflow의 방식 ③ 알고리즘, 하드웨어, 소프트웨어 등의 다중(Multiple) 상위수준에서 구현가능 • Ptolemy (Berkeley Univ.) <ul style="list-style-type: none"> ① 이종간 (Heterogeneous)의 모델링과 Con-current 시스템 설계용어 예로 아날로그와 디지털, 하드웨어와 소프트웨어 등과 같은 것을 포함하는 혼합기술, Embedded Systems에 사용 • SPW를 이용한 알고리즘 검증을 위한 환경구축 <ul style="list-style-type: none"> ① SPW를 이용하여 알고리즘을 부동 소수점으로 검증하고 이로부터 VHDL 이나 C++ 과 같은 상위수준의 언어로 변환한다. 변환된 내용으로 통합실험을 해 봄으로써 알고리즘 수준에서부터 RT 수준까지 일관되게 검증 할 수 있다. ② 구축된 환경을 검증하기 위하여 Voice Signal Processor를 부동소수점으로 설계하고 성능을 검증하였다. 또한, SPW CGS(Code Generation System)로 알고리즘 수준의 블록 모델로부터 C 코드 생성 하였다. 생성된 코드를 이용하여 동작을 검증하는 실험으로 구축된 설계환경을 검증했다. <p>별첨14. SPW로 설계된 Voice Signal Processor 블록도 별첨15. SPW로 설계된 Voice Signal Processor 신호도</p> • Ptolemy를 이용한 알고리즘 검증을 위한 환경구축 <ul style="list-style-type: none"> ① Ptolemy는 UC. Berkeley에서 개발했고, 현재 개발 중인 비상업용 프로그램이다. 알고리즘 설계 및 검증을 위하여 사용되는 tool이다. 부동소수점으로 구현하고 Ptolemy로 설계하여 VHDL, C++, DSP 어셈블러 등의 언어로 코드 생성이 가능하다. 이렇게 생성된 코드를 이용하여 기존의 설계된 블록들과 함께 통합실험을 한다.

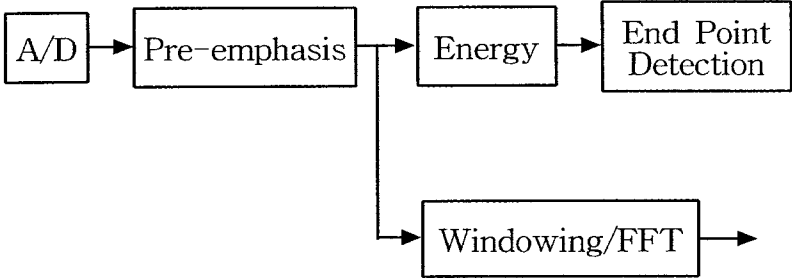
연구 내용	연구 결과
<p>기존의 알고리즘 수준에서의 설계, 검증을 위한 프로그램의 특징 비교 (1차년도)</p>	<p>② 도플러효과, Sonar, Ultra-sonic과 같은 시스템들을 상위수준에서 하드웨어-소프트웨어 통합설계, 검증을 하는 것은 시간과 비용 면에서 많은 장점을 가진다. 위의 Sonic 시스템을 Ptolemy 을 이용하여 하드웨어-소프트웨어 통합설계, 검증하였다. Ptolemy의 SDF Domain을 이용하여 부동소수점방식으로 설계하고 시뮬레이션을 수행하였다. CG Domain에서 C코드를, VHDL Domain에서 VHDL을 생성하고 이렇게 생성된 블록을 기존에 설계되어 있던 블록들과 같이 Co-simulation을 수행하였다</p> <p>별첨16. Ptolemy로 구현된 Sonic 블록도 별첨17. Ptolemy로 구현된 Sonic 동작신호</p> <ul style="list-style-type: none"> • 기존의 알고리즘 수준에서의 설계, 검증을 위한 프로그램의 특징 비교 <ul style="list-style-type: none"> - COSSAP과 SPW는 기능상에서 유사한 특성을 보임 - COSSAP과 SPW는 성능적인 면에서 커다란 차이가 없음 - 기존에 보유하고 있는 SPW & Ptolemy를 사용하여 알고리즘 설계 환경 구축 - 지속적인 시스템 수준의 설계 Tool 조사 및 확보
<p>Architecture/Logic 수준에서의 설계 및 검증 (1차년도)</p> <p>PCB 관련 설계 환경 구축 (1차년도)</p>	<ul style="list-style-type: none"> • RT 수준에서의 설계환경 구축 <ul style="list-style-type: none"> - Synopsys Tool <ul style="list-style-type: none"> ① VHDL을 사용한 설계에 대하여 다른 프로그램에 비해 합성능력 (Synthesis)이 뛰어남 ② Design Compiler, FPGA Compiler, VHDL Simulation System(VSS)를 사용하여 게이트 수준에서의 합성 및 실험에 사용 ③ 전력해석 Tools <ul style="list-style-type: none"> PowerMill, AMPS, RailMill: 트랜지스터 수준 PowerCompiler: RTL, 게이트 수준 DesignPower: 게이트 수준 PowerArc: 셀 라이브러리의 전력특징 (Power Characterization)을 제공 ④ RTL, 게이트 수준에서 주로 설계하는 당 랩의 경우에는 Design Power가 적합 • PCB 관련 설계환경 구축 <ul style="list-style-type: none"> - PCB 관련 설계를 위하여 Mentor Tool이 보편적으로 널리 사용됨 - Mentor Tool 중 PCB 관련 board Station을 사용하고자 함

연구 내용	연구 결과
<ul style="list-style-type: none"> • 프로세서 Proto-type Tool (1차년도) - 프로세서 Proto-type Tool - DSP Proto-type Tool - FPG A Proto-type Tool 	<ul style="list-style-type: none"> • 프로세서 Proto-type Tool <ul style="list-style-type: none"> - Strong ARM SA-1100의 평가보드(이하Brutus Board)의 테스트내용 <ol style="list-style-type: none"> ① Brutus보드를 구성하는 메모리, 주변장치: DRAM(16MB), SRAM(512kB), ROM(256kB), Flash(256kB), LCD ,RS-232C, IrDA, PCMCIA 등 <ul style="list-style-type: none"> 별첨18. StrongARM 개발도구 ② Brutus board 상태 분석 <p>보드의 정상동작을 체크하기 위해서는 보드에 부착되는 전용 테스트 프로그램이 필요하지만 메이커에서 제공해주지 않는 관계로 프로그램을 개발하기 위한 필수적인 동작부분으로 생각되어지는 주변장치에 대하여 간단한 보드테스트 일부 프로그램을 작성함</p> <ol style="list-style-type: none"> ③ Brutus Board 테스트 내용 <ul style="list-style-type: none"> . DRAM 전영역의 Read/Write, General I/O의 동작, RS-232C 송수신 체크 별첨19. ROM 동작 확인(장착된 프로그램 R/W) 별첨20. RS-232C 송수신 체크(PC와 송수신) 별첨21. LCD 그래픽 표시 테스트 . SRAM 전영역의 Read/Write, Timer와 인터럽트에 의한 인터럽트체크 • DSP Proto-type Tool <ul style="list-style-type: none"> - TSM320C54x의 테스트 프로그램 동작확인 • FPGA Proto-type Tool <ul style="list-style-type: none"> - Altera의 MAXPLUSII에 대한 분석 <ol style="list-style-type: none"> ① 입력: AHDL(Altera HDL), VHDL, EDIF netlist, Schematic Entry ② Simulation: Function & Timing Simulator ③ 장점: 다양한 입력형식 지원, 자체적인 Simulator 보유, 사용이 편리한 GUI 환경 ④ 단점: 실험은 항상 처음단계부터 시작하여야 함, Constraint를 정량적으로 줄 수 없음 - Xilinx의 M1에 대한 분석 <ol style="list-style-type: none"> ① 입력: XNF, SXNF(synopsys XNF), EDIF ② Simulation: 단지 SDF 파일만 생성됨 ③ 장점: Synopsys에서 SXNF 형식을 출력 할 수 있음, Synopsys에서 사용한 Constraint를 바로 적용가능, 중간단계에서 다시 실험을 시작 할 수 있음 ④ 단점: 자체 Simulator가 없음, 항상 다른 합성/검증 Tool 필요 <ul style="list-style-type: none"> 별첨22. FPGA Placement & Routing 별첨23. FPGA Device Programming

연구 내용	연구 결과
<p>Circuit 및 저전력 Custom Design (1차년도)</p>	<ul style="list-style-type: none"> • Circuit 및 저전력 Custom Design <ul style="list-style-type: none"> - Full-custom Design 환경 구축 저전력 회로의 Full-custom 설계를 위한 최적화된 Methodology Setup에 관한 이해 및 기술적 접근을 위해 관련 세미나와 교육을 실시하였다. Intel, HP, SUN에서 Full-custom Processor Design 및 Design Methodology 수립에 기여한 Univ. of Utah의 김용빈 교수에게 세미나 및 기술교육을 받았다. ① 1998년 12월 22일 삼성종합기술원에서 김용빈 교수가 "High Performance VLSI Circuit Design Methodology"와 "Clock Distribution for High Performance μ-processors"에 대한 세미나가 진행되었다. 세미나의 내용을 바탕으로 Circuit 수준에서의 설계, 검증 프로그램에 관하여 조사, 분석하였다. ② Full-custom ASIC 설계와 고성능 VLSI 회로 설계 방법론에 관한 기술교육이 진행되었다. - Circuit 및 저전력 Custom Design을 위한 설계 프로그램 <ul style="list-style-type: none"> ① Cadence 프로그램: Schematic & Layout, LVS(Layout vs. Schematic Verification)를 포함(다른 프로그램에 비해 성능이 뛰어남) ② Star-Hspice 프로그램: 아날로그 및 디지털 회로의 트랜지스터 수준에서 동작을 검증하는 프로그램 - Full-custom ASIC 설계와 고성능 VLSI 회로 설계 방법론 <ul style="list-style-type: none"> ① Semi-custom 설계: CAD tool을 주로 사용하여 설계기간이 비교적 짧으나, 성능을 극대화하는 데는 한계가 있음 ② Full-custom 설계: 설계자가 CAD tool로 할 수 없는 부분의 성능 개선을 도모 가능 ③ Full-custom 설계 시 중요 고려사항 <ul style="list-style-type: none"> . 저전력 회로설계: 공급전압을 줄이는 것이 부수적인 문제를 발생 시킴에도 불구하고 가장 효과적이지만, 보통은 회로의 스위칭 Activity를 줄이는데 가장 많은 노력을 기울여야 함 . 아날로그 부분과 디지털부분의 분리는 필수 적임 . 공정이 미세해 질수록 Wire, Capacitor 등의 파라미터에 대한 비율 축소기술이 필요 ④ 고성능 VLSI 회로 설계 방법론 <ul style="list-style-type: none"> . 칩 제작 후, 칩의 정상동작을 위해서는 전체 칩을 구성하는 각 설계 블록들의 설계성능(Design Quality)의 일관성 유지 필요 . 설계성능의 일관성을 유지하기 위해 설계에 공통적으로 적용되는 설계방법론과 검증방법 등이 필수적으로 필요

연구 내용	연구 결과
<p>테스트 및 디버깅 용이한 설계를 위한 설계 방법론에 대한 기술 분석 (1차년도)</p> <p>- 중앙대학교 송오영 교수에게 기술조사 의뢰</p> <p>- 테스트 용이 설계에 대한 기술조사</p>	<ul style="list-style-type: none"> • 테스트 및 디버깅 용이한 설계를 위한 설계방법론에 대한 기술 분석 <ul style="list-style-type: none"> - 테스트가 용이한 설계에 대한 기술조사 <ol style="list-style-type: none"> ① Scan Design: VLSI 칩 내부에 존재하는 Flip-Flop이나 Latch등의 Controllability 및 Observability를 확보하기 위해 사용하는 방법 ② Built-in-Self-Test: VLSI 칩 내부에 내장되는 SRAM/DRAM/Flash-Memory를 테스트하기 위해 사용 ③ Boundary Scan Design: VLSI 칩이 사용될 응용보드 및 시스템의 고장검출 및 고장진단 등을 위해 필수적인 기법 ④ Ad hoc D.F.T: Tristate buffer 테스트 기법, scan chain ordering, pin 공유기법 등과 같은 기법 - 디버깅이 용이한 설계에 대한 기술조사 <ol style="list-style-type: none"> ① ICE(In-Circuit Emulator) ② RM(ROM Monitor) ③ BDM(Background Debug Mode) ④ JTAG-based Emulator - 설계와 테스트의 협동연구추진체계 <ol style="list-style-type: none"> ① RTL수준의 회로설계 완성(설계팀): 테스트의 용이도를 높이기 위한 회로를 JTAG Boundary Scan부분을 설계하고, 메모리가 내장 된다면 메모리 테스트 회로도 설계 ② 게이트 설계완료(설계팀): 테스트 용이도를 높이기 위한 테스트 설계 수행 ③ Fabrication(설계팀과 테스트팀의 설계완성): 높은 고장 검출율을 갖는 테스트 패턴 공급 <p>별첨24. 설계와 테스트의 협동연구 추진 체계도</p> <ul style="list-style-type: none"> - 조사 분석 결과 <ol style="list-style-type: none"> ① 상위수준에서의 Testability를 높이기 위한 테스트 설계는 아웃소싱이 효과적이라는 판단

연구 내용	연구 결과
<ul style="list-style-type: none"> • 하드웨어-소프트웨어 통합설계에 대한 기술 이해 (1차년도) 	<ul style="list-style-type: none"> • 하드웨어-소프트웨어 통합설계에 대한 세미나가 진행되었음 (1999년4월2일 서울대학교 최기영 교수) <ul style="list-style-type: none"> - 하드웨어-소프트웨어 통합설계에 대한 기술 이해 • 하드웨어-소프트웨어 통합설계에 대한 세미나를 통한 기술이해 <ul style="list-style-type: none"> - 하드웨어-소프트웨어 통합설계를 위한 환경의 조건 분석 <ol style="list-style-type: none"> ① 시스템의 수행가능(Executable) 표현(Re-presentation) 가능 ② 통합시뮬레이션을 통한 사양검증 ③ 빠른 Prototyping 가능 ④ 국, 내외의 통합시뮬레이션 관련 프로그램 - Ptolemy를 기반으로 한 통합설계 환경개발 <ol style="list-style-type: none"> ① Ptolemy를 이용한 설계(Design) Flow 구축(소프트웨어-C 언어, 하드웨어-VHDL) ② 통합시뮬레이션을 위한 추가 기능(Untimed-하드웨어/소프트웨어 인터페이스생성, Timed-기존에 제작된 Prototyping Board와의 통합실험을 위한 인터페이스생성) - 통신시스템 설계를 위한 통합설계환경 이해 <ol style="list-style-type: none"> ① Ptolemy 환경에서의 통신시스템 설계 ② 하드웨어-소프트웨어 분할 및 합성, 통신부분합성 수행 ③ 통합시뮬레이션으로 혼합시스템 동작 검증 및 시스템의 프로토타입 • Multimode 시스템의 성능분석 및 결과를 설계 반영할 수 있는 기법에 대한 기술이해 <ul style="list-style-type: none"> - Multimode 시스템 모델링 : 시간제약조건의 특성에 따라 태스크 등을 분류, 종료시한 내에 수행을 끝마칠 수 있는 확률(Quality of Service : QoS) - 태스크 형태에 따른 Multi-mode 시스템의 분석 <ol style="list-style-type: none"> ① Hard real-time 태스크만으로 된 Multimode 시스템 분석방법 ② Hard Real-time 태스크와 Soft Real-time 태스크가 혼합된 Multimode 시스템 분석방법 - 모든 태스크의 QoS를 만족하기 위한 최소 성능비 검출

연구 내용	연구 결과
<p>알고리즘 분석 (2차년도)</p>	<ul style="list-style-type: none"> • 음성인식 알고리즘 기반기술 확보 <ul style="list-style-type: none"> - 단어의 끝점 검출, 에너지 계산 및 Preprocessing 부분 분석 및 연구 <ul style="list-style-type: none"> ① 동작 설명 : Preprocessing 부분은 Pre-emphasis Block, FFT Block, Energy Block 및 End Point Detection Block으로 구성되어 있다. Pre-emphasis Block은 음성신호 내의 저주파 성분을 억제하고 고주파 성분을 강조함으로써 고주파 성분에 포함된 특성치를 이용할 수 있게 한다. FFT Block은 시간영역의 음성신호를 주파수 영역의 신호로 바꾸는 256-point FFT를 수행한다. 이렇게 얻어진 주파수 스펙트럼은 좌우 대칭의 특성이 있어, 절반만이 추후 연산에 사용된다. Energy Block에서는 끝점 검출에 사용할 음성신호의 에너지를 계산하고 End Point Detection Block에서는 이를 이용하여 끝점검출을 하여 특성치를 추출할 음성만을 선별하여 낸다. <div style="text-align: center;">  <pre> graph LR A/D --> Pre-emphasis Pre-emphasis --> Energy Pre-emphasis --> Windowing/FFT Energy --> End Point Detection </pre> </div> <p style="text-align: center;">그림 2. Preprocessing 부분 Block Diagram</p> <p style="text-align: center;">별첨 25. Preprocessing 부분 계산량/메모리 소요량</p> <ul style="list-style-type: none"> - 음성 특성치 추출을 위한 Cepstrum 기법 분석 및 연구 <ul style="list-style-type: none"> ① 동작 설명 : 특성치 추출을 위한 Function Block으로는 Melfilter Block, IDCT Block, Scale Block, Cepstral Window Block, Normalize Block, Dynamic Feature block이 있다. Melfilter Block에서는 FFT에 의해서 주파수 영역으로 바뀌어진 음성신호를 32개 밴드의 필터뱅크를 이용하여 주파수 대역별로 음성신호값을 구한다. IDCT Block에서는 32개로 나누어진 음성의 주파수 특성을 다시 9개의 Cepstrum 값으로 변환한다. Scale Block에서는 이 변환된 값에 일정 상수를 곱해 준다. Cepstral Window Block에서는 Cepstrum에 Hamming Window를 씌우는 연산을 수행하고 Dynamic Feature Block에서는 특성치와 특성치간의 변화 성분(Δ Cepstrum)으로써 또 하나의 특성치를 추출하는 연산을 수행한다.

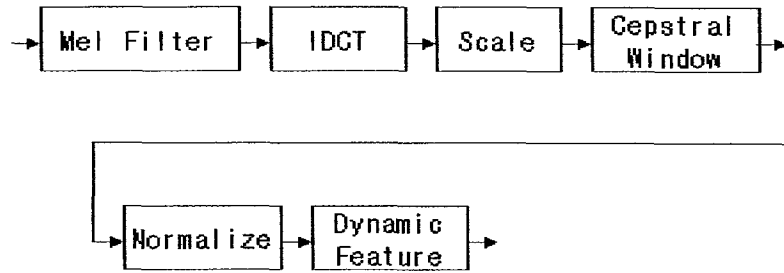


그림 3. 음성 특성치 추출을 위한 Block Diagram

별첨 26. 음성 특성치 추출을 위한 계산량/메모리 소요량

별첨 27. 화자독립/어휘독립 50단어 음성인식 알고리즘의 전체 Block Diagram

알고리즘 분석
(2차년도)

• DTW(Dynamic Time Warping)기법과 HMM(Hidden Markov Model) 기법 연구

DTW는 100단어 이하의 인식율이 상대적으로 낮아도 되는 화자 종속 시스템에 적당한 단어 단위의 검색 기법이고, HMM은 인식율이 높은 화자 독립 시스템에 적당한 음소 단위의 검색 기법임을 확인

100단어 이하의 음성 인식 시스템 경우에는 DTW 기법을 채택하여도 무방하지만, 향후 단어 확장을 고려하여 HMM기법을 음성 인식을 위한 Searching 알고리즘으로 채택하였다.

	속성	인식률
Dynamic Time Warping	화자종속	중간
Hidden Markov Model	화자독립	높음

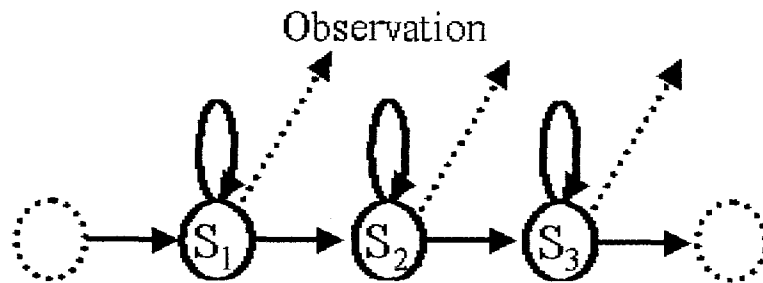


그림 4. 음성인식 알고리즘용 3 state left-to-right HMM model

연구 내용	연구 결과
<p>알고리즘 분석 (2차년도)</p>	<ul style="list-style-type: none"> • 하드웨어 기본 알고리즘 분석 <ul style="list-style-type: none"> - Preprocessing의 ASIC화 <p>Preprocessing에 필요한 명령은 기본적인 Load, Store, Add, Sub, Mul, Shift 명령어 외에 음성 에너지 계산의 제곱 연산을 가능케 하는 Square와 누적합을 구하는 Addac가 있어 각각의 연산기를 동작하게 한다. 이 두 명령어의 구현 방법은 다음과 같다. Square 명령어는 곱셈기의 양 오퍼랜드를 메모리상 같은 데이터 값을 읽어 곱함으로 구현할 수 있고, Addac 명령어는 ALU가 덧셈기능을 수행할 때에 출력 레지스터의 값을 다시 입력에 환원하여 다시 덧셈을 함으로 구현된다.</p> <ul style="list-style-type: none"> - HMM Searching Engine <p>HMM의 주요 연산은 관측확률과 상태전이 확률을 계산할 때 발생한다. 그러므로 HMM Searching Engine은 연산기(Adder, Subtract, Multiplier, Comparator)와 로컬 메모리 어드레스 제어 부분, 내부버스 인터페이스 부분에 의해 구성되어지고, 각 연산기를 5가지 명령에 의해 제어된다. HMM_INIT는 HMM 연산기를 초기화하는 기능을 가지고 있다. HMM_RD는 HMM 내부레지스터 값을 데이터 레지스터로 읽는데 사용된다. HMM_WRD는 HMM 내부레지스터 값에 데이터 레지스터의 값을 써줄 때 사용한다. 컨트롤 레지스터에 의해 제어되는 HMM 연산기는 바로 이 명령에 의해 제어된다. 로컬 메모리와 데이터 메모리간의 전송은 HMM_MDH, HMM_MHD에 의해서 이루어진다.</p>

연구 내용	연구 결과															
<p>데이터 분석 (2차년도)</p>	<p>• 화자독립/어휘독립 인식기 알고리즘 확보</p> <p>- HCI 프로세서 사업단 내의 "구어이해기술"과제에서 제공된 화자독립/어휘독립 인식기 알고리즘에 대한 계산량 분석</p> <table border="1" data-bbox="469 389 1137 551"> <thead> <tr> <th rowspan="2"></th> <th colspan="3">계산량 / Sample</th> </tr> <tr> <th>곱셈</th> <th>덧셈</th> <th>MIPS</th> </tr> </thead> <tbody> <tr> <td>상용 DSP</td> <td>18,063</td> <td>16,525</td> <td>25</td> </tr> <tr> <td>ASSP</td> <td>12,200</td> <td>10,525</td> <td>15</td> </tr> </tbody> </table> <p>*상용 DSP : TI TMS320C54X *ASSP : Application Specific Signal Processor</p> <p>- 알고리즘 수행 시 필요한 메모리 사용량 분석 <가정></p> <ul style="list-style-type: none"> ◦ Fixed-point Modeling (모든 변수는 2byte) ◦ N: # of states (3) ◦ M: # of mixtures/state (10) ◦ P: # of sub-unit (40) ◦ D: Feature dimension (18: cepstrum+delta_cepstrum) <p>. 고정 메모리 (HMM parameter + Program)</p> <p>- HMM parameter</p> <ul style="list-style-type: none"> ① mean : $N * M * P * D * 2B = 43KB$ ② var : $N * M * P * D * 2B = 43KB$ ③ mixture weight : $N * M * P * 2B = 2.4KB$ <p>계: 88.4KB</p> <p>만일 M을 4로 떨어뜨리면, 35.4KB면 됨.</p> <p>. 인식 프로그램: 30KB</p> <p>그러므로,</p> <p>if M=10 118.4KB if M=4 65.4KB</p> <p>. 변동 메모리 (Dynamic memory allocation)</p> <p>- Dynamic memory allocation (6K: frame synchronous 가정)</p> <ul style="list-style-type: none"> ① 전처리: 5KB ② Search: 1KB <p>그러므로 10KB 이하로 처리 가능</p>		계산량 / Sample			곱셈	덧셈	MIPS	상용 DSP	18,063	16,525	25	ASSP	12,200	10,525	15
	계산량 / Sample															
	곱셈	덧셈	MIPS													
상용 DSP	18,063	16,525	25													
ASSP	12,200	10,525	15													

연구 내용	연구 결과																						
데이터 분석 (2차년도)	<ul style="list-style-type: none"> • 고정소수 연산동작 모델링 <ul style="list-style-type: none"> - 알고리즘 수준의 검증 틀인 SPW로 설계 및 검증 결과 화자독립/어휘독립 50 단어 인식 알고리즘을 SPW를 이용한 Fixed-point System Model을 시뮬레이션 결과 16 bits Data Bus인 경우에는 HMM Block에서 Quantization Error가 39.2 dB이고, 12 bits Data Bus인 경우에는 25.4 dB이다. 그러므로 Fixed-point 변환시 Truncation Error에 대한 충분한 보상이 되지 않으면 음성 인식이 많이 저하 될 것으로 예상된다. <table border="1" data-bbox="470 685 1219 969" style="margin: 10px auto; border-collapse: collapse; text-align: center;"> <thead> <tr> <th colspan="2">Data Bit 수</th> <th>12 bits</th> <th>16 bits</th> </tr> </thead> <tbody> <tr> <td colspan="2">Area(Adder + Multiplier)</td> <td>67.29mm² (10.49mm²+56.80mm²)</td> <td>99.18mm² (13.96mm²+85.22mm²)</td> </tr> <tr> <td colspan="2">Power(Adder + Multiplier)</td> <td>159.6μW/MHz 13.60μW/MHz + 146.0μW/MHz</td> <td>225.51μW/MHz 18.61μW/MHz + 206.9μW/MHz</td> </tr> <tr> <td rowspan="3">Quantization Error(dB)</td> <td>Preprocessing</td> <td>45.3 dB</td> <td>64.8dB</td> </tr> <tr> <td>MFOC</td> <td>31.5dB</td> <td>50.2dB</td> </tr> <tr> <td>HMM</td> <td>25.4dB</td> <td>39.2dB</td> </tr> </tbody> </table>	Data Bit 수		12 bits	16 bits	Area(Adder + Multiplier)		67.29mm ² (10.49mm ² +56.80mm ²)	99.18mm ² (13.96mm ² +85.22mm ²)	Power(Adder + Multiplier)		159.6μW/MHz 13.60μW/MHz + 146.0μW/MHz	225.51μW/MHz 18.61μW/MHz + 206.9μW/MHz	Quantization Error(dB)	Preprocessing	45.3 dB	64.8dB	MFOC	31.5dB	50.2dB	HMM	25.4dB	39.2dB
	Data Bit 수		12 bits	16 bits																			
Area(Adder + Multiplier)		67.29mm ² (10.49mm ² +56.80mm ²)	99.18mm ² (13.96mm ² +85.22mm ²)																				
Power(Adder + Multiplier)		159.6μW/MHz 13.60μW/MHz + 146.0μW/MHz	225.51μW/MHz 18.61μW/MHz + 206.9μW/MHz																				
Quantization Error(dB)	Preprocessing	45.3 dB	64.8dB																				
	MFOC	31.5dB	50.2dB																				
	HMM	25.4dB	39.2dB																				
	<ul style="list-style-type: none"> • 기본 아키텍처 개발 <ul style="list-style-type: none"> - 음성 인식 프로세서를 위한 고정소수 연산 동작에 필요한 아키텍처 & ISA(Instruction Set Architecture) 개발 별첨 28. 음성인식용 ASSP의 기본 아키텍처 별첨 29. 음성인식용 ASSP의 기본 ISA 																						

연구 내용	연구 결과
데이터 분석 검증 (2차년도)	<ul style="list-style-type: none"> • 시제품 보드 구현/검증 <ul style="list-style-type: none"> - Preprocessing(Pre-emphasis) 고정소수 연산 동작 시뮬레이션 및 시제품 보드 구현 <p>음성인식을 위한 전처리 단계의 하나인 Pre-emphasis를 FPGA를 이용한 Processor 구조의 H/W로 구현하여, 동작을 검증하기 위해서 다음과 같이 실험을 진행하였다.</p> <ol style="list-style-type: none"> ① 음성알고리즘(Preprocessing)에 대한 분석 및 Fixed-point Simulation(C-언어를 이용)을 이용한 알고리즘 검증 ② 명령어 설계 ③ 어셈블러 설계 및 어셈블러를 이용한 알고리즘 검증 ④ 마이크로 Architecture 설계 및 파이프라인 구현 Architecture ⑤ VHDL를 이용한 데이터 Path 설계 ⑥ VHDL 를 이용한 Control부분 설계 ⑦ 설계된 특정응용 프로세서와 오디오 Codec과의 인터페이스 설계 <p>이러한 실험방법 및 순서에 따라 만들어진 Processor 구조의 Pre-emphasis는 Filter Bank를 이용하여 8Khz Sampling, 16bits 크기의 음성 Data를 각 주파수 대역으로 나눈 후 크기(Magnitude)를 줄이기 원하는 저주파수 대역을 크기를 줄인 후 DAC(16bits)를 통과하여 음성으로 출력한다.</p> <p>별첨 30. Preprocessing을 위한 ASSP의 아키텍처 별첨 31. Preprocessing(Pre-emphasis) 동작 Block Diagram</p> - 음성 인식 프로세서 전체 시제품 보드 구현 진행
	<ul style="list-style-type: none"> • HCI응용 소프트웨어 구현을 위한 모델링 (서울대 최기영 교수 기술자문) <ul style="list-style-type: none"> - HCI 관련 신호처리 알고리즘 모델링 - 일반 마이크로프로세서 HCI 응용을 구현할 때에 발생하는 실시간 시스템 설계 문제, 저전력 설계 문제, 소프트웨어 코드 최적화 문제 만족시키는 방법론 확보 <ol style="list-style-type: none"> ① Schedulability-driven Performance 해석 ② 저전력 고정 우선 순위 스케줄링 ③ Loop 파이프라이닝을 이용한 소프트웨어 코드 최적화 ④ Ptolemy를 이용한 소프트웨어 모델링 분석 ⑤ 저전력을 고려한 컴파일러 최적화 기법

연구 내용	연구 결과																									
<p>3차 년도 마이크로 아키텍처 기반 기술 확보 (2차년도)</p>	<p>• 마이크로 아키텍처 사양 결정을 위한 선행연구 - 세계 음성인식 관련 시장 현황 및 전망</p> <p>Source : Computer Economics, October 27 1999 (URL: http://www.computereconomics.com)</p> <table border="1" data-bbox="587 421 1417 694"> <thead> <tr> <th></th> <th>1999 (\$M)</th> <th>2001 (\$M)</th> <th>2003 (\$M)</th> <th>2005 (\$M)</th> </tr> </thead> <tbody> <tr> <td>Voice Recognition (음성인식 기능 솔루션 시장)</td> <td>50</td> <td>130</td> <td>365</td> <td>785</td> </tr> <tr> <td>Voice Generation (음성재생 솔루션 시장)</td> <td>300</td> <td>575</td> <td>840</td> <td>1,200</td> </tr> <tr> <td>Interactive Voice Response (대화형 응답 솔루션 시장)</td> <td>900</td> <td>1,300</td> <td>1,800</td> <td>2,300</td> </tr> <tr> <td>합계</td> <td>1,250</td> <td>2,005</td> <td>3,005</td> <td>4,285</td> </tr> </tbody> </table> <p>① 음성관련 제품시장은 1999년 12억 달러 규모에서 2005년 40억 달러 규모로 약 340%이상의 신장률을 보일 것으로 예측되므로 시장크기의 성장률은 매우 크다. (전문 조사기관 Computer Economics 자료 발췌)</p> <p>- 음성인식 전용 프로세서의 사업화를 고려한 마이크로 아키텍처 사양 결정을 위한 시장조사</p> <p>① 대상 품목 : 사무실 기기(복사기, 프린터), 가정용 기기 (리모컨, 자동응답기), 장난감 등등</p> <p>② 국내/외 Toy업체 방문 조사 결과 → 사양 문의 결과 : 10~20 단어 인식, \$2 이하의 음성인식 칩 . 상담 결과 ◦ 음성인식용 제품에 대한 관심 및 수요 증대 ◦ 30~50단어 미만의 인식단어 수 ◦ \$5 미만의 가격대 요구 ◦ 높은 인식률 및 인식률에 대한 신뢰성 문제 . 분석 결과 ◦ 음성인식 필요성 공감 ◦ 본격적인 개발 계획 보유 업체 - 소수업체 ◦ 시장전망 : 급속한 성장세 예상 (20%이상/년)</p> <p>③ 국제 장난감 메이커 Hasbro와 시스템 기능 정의에 대한 기술 협력상호 양해 각서 체결</p> <p>별첨 32. 장난감 및 TAD(Tapeless Answering Device)에서 필요한 음성인식 칩 세계 시장 전망 예측</p>		1999 (\$M)	2001 (\$M)	2003 (\$M)	2005 (\$M)	Voice Recognition (음성인식 기능 솔루션 시장)	50	130	365	785	Voice Generation (음성재생 솔루션 시장)	300	575	840	1,200	Interactive Voice Response (대화형 응답 솔루션 시장)	900	1,300	1,800	2,300	합계	1,250	2,005	3,005	4,285
	1999 (\$M)	2001 (\$M)	2003 (\$M)	2005 (\$M)																						
Voice Recognition (음성인식 기능 솔루션 시장)	50	130	365	785																						
Voice Generation (음성재생 솔루션 시장)	300	575	840	1,200																						
Interactive Voice Response (대화형 응답 솔루션 시장)	900	1,300	1,800	2,300																						
합계	1,250	2,005	3,005	4,285																						

연구 내용	연구 결과										
<p>3차 년도 마이크로 아키텍처 기반 기술 확보 (2차년도)</p>	<p>• 소비전력 최적화를 위한 마이크로 아키텍처 선행연구 음성신호의 소비전력 최대 소모원인이 메모리와 버스의 접속에서 유발되는 것을 확인하여 메모리와 버스의 구조 개선이 필요함을 파악 기존 방법으로는 BI(Bus-Invert) 방식과 BITS(Bus-Invert with transition signaling)방식이 있는데, 이 방식들과 우리가 제안한 HIHR(Half Identity Half Reverse)방식과의 성능을 비교 하였다. 결과적으로 기존의 BI방식과 BITS방식은 data를 인코딩할 때 사용되는 여분의 회로와 여분의 선에 의해서 우리가 제시한 HIHR방식이 Power, Area, Timing Delay면에서 우수한 효과를 나타냄을 알 수 있었다.</p> <p>별첨 33.IRSIM이라는 버클리 Simulator를 사용하여 Power를 비교표(Time Delay는 HSPICE를 사용)</p> <div data-bbox="654 840 1101 1041" data-label="Figure"> <table border="1"> <caption>그림 5. RISC 프로세서의 소비전력 분포도</caption> <thead> <tr> <th>Category</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>Memory</td> <td>35%</td> </tr> <tr> <td>Clock</td> <td>30%</td> </tr> <tr> <td>Interconnect</td> <td>20%</td> </tr> <tr> <td>Others</td> <td>15%</td> </tr> </tbody> </table> </div>	Category	Percentage	Memory	35%	Clock	30%	Interconnect	20%	Others	15%
Category	Percentage										
Memory	35%										
Clock	30%										
Interconnect	20%										
Others	15%										
	<p>• 효과적인 산학연 협동을 위한 글로벌 사이버 연구센터 삼성종합기술원 Site에 구축 - www.sait.samsung.co.kr/newsait/global/g.html 별첨 34.글로벌 사이버 연구센터의 형태</p> <p>• 글로벌 사이버 연구센터구축에 따른 기대효과 - 원활한 Communication을 통한 공동개발을 활성화 - 기술자료의 Data Base화 - 기술자료의 공유</p>										

제 4 장 연구개발목표 달성도 및 대외기여도

	연구목표	평가착안점	달성도 (%)	관련분야기술 대외기여내용
1차년도	Multi-processor Co-design Methodology 개발	- Multi-processor Co-design Methodology 및 설계 툴 환경 구축 여부	100%	<ul style="list-style-type: none"> 프로세서의 최적화 설계 방법론에 대한 기술 분석 및 축적 설계 툴 환경 구축
	Multi-processor Simulation Methodology 개발	- Multi-processor Simulation Tool 및 시제품 보드 설계 환경 구축 여부	100%	<ul style="list-style-type: none"> 상하위 수준의 설계 및 시뮬레이션 환경 구축 시제품 보드 설계를 위한 환경 구축
	Multi-processor Verification Methodology 개발	- Multi-processor Verification 환경 구축 및 최적화 설계 방법론 구축 여부	100%	<ul style="list-style-type: none"> 테스트를 고려한 최적화 설계 방법론 분석 알고리즘 및 아키텍처/로직 수준에서의 설계검증 환경 구축 (H/W-S/W 통합설계에 대한 기술 이해)
2차년도	알고리즘 분석	- 화자독립, 어휘독립 음성 인식 기본 알고리즘 및 하드웨어 기본 알고리즘 분석 여부	100%	<ul style="list-style-type: none"> 음성인식 알고리즘 중 음성특성치 비교검색을 위한 DTW/HMM 분석 완료 하드웨어 기본알고리즘 분석 및 방안 연구 50단어 화자독립/어휘독립 인식기 알고리즘에 대한 Floating-point Operation 및 연산량 분석
	데이터 분석	<ul style="list-style-type: none"> 화자독립, 어휘독립 음성 인식 알고리즘 연산량 분석 여부 음성 인식 알고리즘 Fixed-point System Modeling 전환 여부 데이터 분석을 통한 프로세서 기본 아키텍처 개발 여부 	100%	<ul style="list-style-type: none"> 음성인식 알고리즘의 Fixed-point System Modeling 및 검증 50단어 화자독립/어휘독립 인식기의 데이터 분석을 통한 기본 프로세서 아키텍처 개발

	연구목표	평가착안점	달성도 (%)	관련분야기술 대외기여내용
2차년도	데이터 분석 검증	- 시제품 보드 구현을 통한 Fixed-point System Model의 검증 여부	100%	<ul style="list-style-type: none"> • 시제품 보드를 이용한 Fixed-point System Model의 설계 및 검증 • HCI 응용 소프트웨어 솔루션 마련
	마이크로 아키텍처 연구	<ul style="list-style-type: none"> - 마이크로 아키텍처 개발 여부 - 글로벌 사이버 연구센터 구축 여부 	20%	<ul style="list-style-type: none"> • 3차년도 계획분인 마이크로 아키텍처 기반 기술 선행 연구 • 글로벌 사이버 연구센터 구축

제 5 장 연구개발결과의 활용계획

제 1 절 연차별 연구결과 활용분야 및 기대효과

1. 1 년차 : Multi-processor Co-design/Sim./Verification Design Methodology 개발

가. 활용분야

- (1) System On a Chip 개발 환경에 활용
- (2) 프로세서 설계 환경 구축을 통한 개발 기간 단축에 활용
- (3) 설계의 효율성을 위한 테스트/디버깅을 고려한 설계에 활용
- (4) H/W-S/W 통합 설계 방법론을 적용 설계의 효율성을 높이는데 활용

나. 기대효과

- (1) 프로세서 설계 환경 구축을 통한 개발 기간 단축 및 설계의 재활용
- (2) 향후 HCI 프로세서 설계에 대한 기반 기술 확보
- (3) HCI 칩내의 일부 기능 블록(필터, 명령어 등)에 대한 부분별 선행 설계 및 검증
- (4) 시제품 보드 설계 환경 구축 통한 보드 설계 기간 단축

2. 2 년차 : Fixed-point System Modeling

가. 활용분야

- (1) 하드웨어 사양 결정시 활용
 - (가) Fixed-point System Modeling에 따른 메모리 크기의 분석
 - (나) 아키텍처 구조 및 데이터 폭 결정시 활용
- (2) 하드웨어 구현에 따른 시스템의 성능 변화 판단의 기본 데이터
 - (가) Floating-point System Model의 실제 데이터 값 (Fixed-point) 추출
 - (나) 음성 인식률 저하 요인 분석
 - (다) 인식률 개선 방안 모색을 위한 기초 자료로 활용
- (3) 제조 원가 계산 및 절감 방법 연구에 활용
 - (가) 제조 비용 계산 (칩 제조 비용 및 패키징 비용 유추) 가능
 - (나) 원가 절감 요인 분석 및 검증 자료로 활용

나. 기대효과

(1) 연구개발시 효과

(가) 상위 수준에서 하드웨어를 고려한 성능 검증 가능

(나) 단기간내 최적의 하드웨어 구조 도출

(다) 초기단계에서의 검증으로 인한 전반적인 개발기간 단축

(2) 사업화시 효과

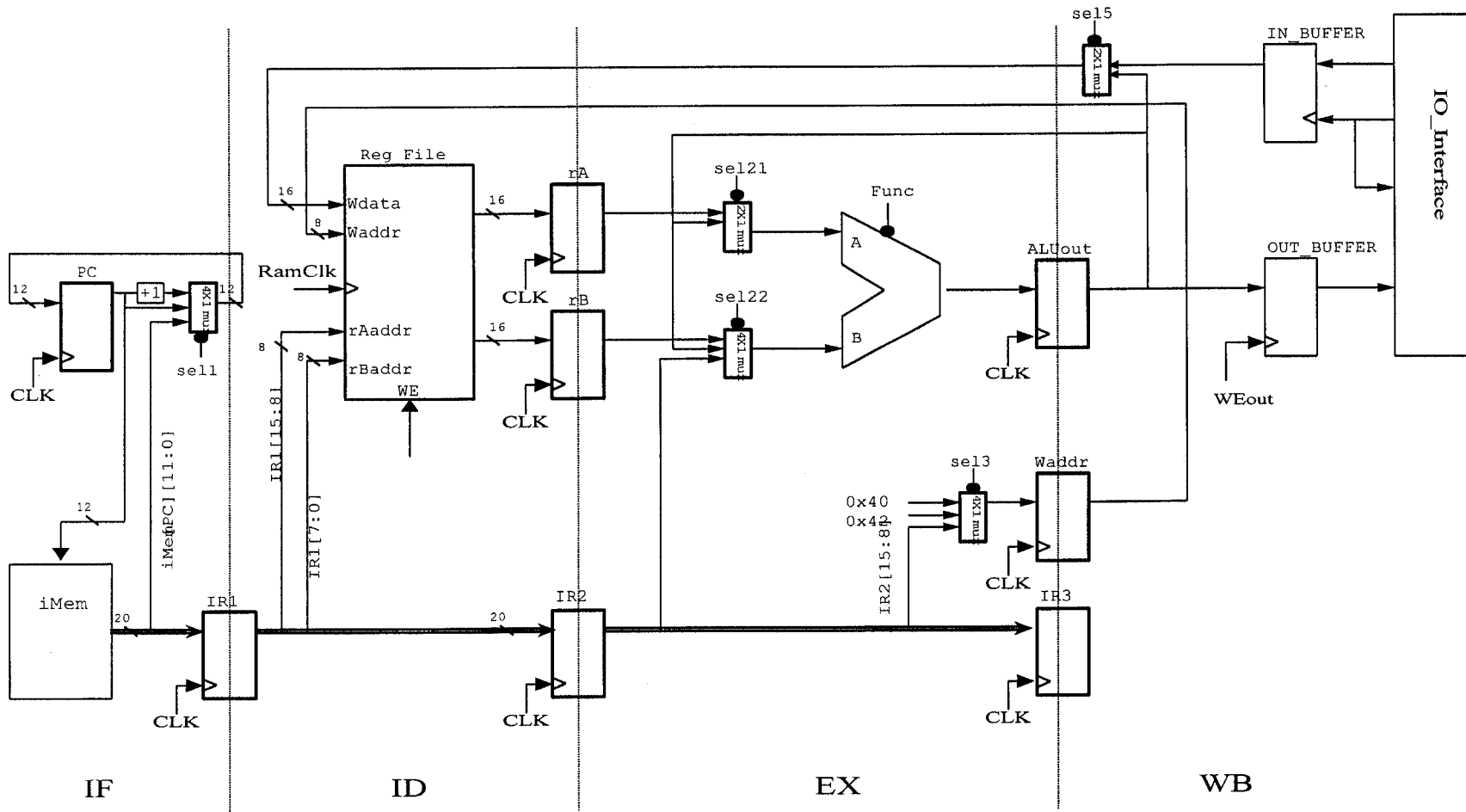
(가) 상위 수준에서의 시행 착오를 통해 하드웨어 제작상의 오류 감소 (비용절감)

(나) 최적의 구조 검증을 통해 최소의 제조비용 및 원가 수립 가능

제 6 장 참고문헌

- [1] H.R. Jang, S.H. Kim, and Y.H. Chang, "A DIGITAL SIGNAL PROCESSOR FOR LOW POWER", 1999 IEEE Asia-Pacific Conference on ASIC, Aug.1999.
- [2] Yamashita, T.; Yoshida, N.; Sakamoto, M.; Matsumoto, T.; Kusunoki, M.; Takahashi, H.; Wakahara, A.; Ito, T.; Shimizu, T.; Kurita, K.; Higeta, K.; Mori, K.; Tamba, N.; Kato, N.; Miyamoto, K.; Yamagata, R.; Tanaka, H.; Hiyama, T , "A 450MHz 64b RISC Processor using Multiple Threshold Voltage CMOS", ISSCC., 2000 IEEE International ,Page(s): 414 - 475, Feb.2000.
- [3] Naehyuck Chang, Kwanho Kim, and Jinsung Cho, "Bus Encoding for Low-Power High-Performance Memory Systems", DAC'00, Page(s): 800 - 805, June.2000.
- [4] S. Bhattacharyya *et. al.*, Ptolemy 0.7 User's Manual, EECS, UC Berkeley, 1996.
- [5] Qualcomm Inc., CDMA System Engineering Training Handbook, 1993.
- [6] Computer Economics homepage (URL: <http://www.computereconomics.com>)
- [7] Intel, Intel StrongARM, SA-1110 Microprocessor Advanced Developer's Manual,1996.
- [8] ARM Ltd., ARM Software Development Toolkit User Guide, v2.50, 1998.
- [9] IEEE Std 1149.1-1990 (including IEEE Std, 1149.1a-1993), "IEEE Standard Test Access Port and Boundary-Scan Architecture."
- [10] Test Technology Standards Committee, "IEEE Standard Test Access Port and Boundary-Scan Architecture," IEEE Computer Society Press, 1990.
- [11] F. Abu-Nofal, et al, "The SuperSPARC Microprocessor" , Proceedings of 1992 International Solid State Circuit Conference, Feb. 1992.
- [12] H. Hao and K. Bhabuthmal, "Clock Controller Design in SuperSPARC II Microprocessor" , Proceedings of 1994 International Conference on Computer Design, Oct 1995.
- [13] W. Needham, N. Gollakota, "DFT Strategy for Intel Microprocessors," International Test Conference, pp. 396-399, 1996
- [14] Design Power, "Synopsys Power Product Reference v1997.01", Synopsys Inc., Mountain View, CA.
- [15] COSSAP, Synopsys COSSAP Reference Manual, Synopsys Inc., Mountain View, CA.
- [16] TMS320C54x User's Guide, Texas Instruments Inc., 1995.
- [17] SPW(Signal Processing WorkSystem) User's Guide, Alta Group of Cadence Design System, Inc., 1996
- [18] XILINX, Inc., Alliance Series Quick Start Guide 1.4, 1997.
- [19] ALTERA, Corp., Data Book, 1998
- [20] 삼성종합기술원 homepage(URL <http://www.sait.samsung.co.kr/newsait/global/g.html>)
- [21] Mentor Graphics homepage(URL <http://www.mentorg.com>)

- [22] Synopsys homepage(URL <http://www.synopsys.com>)
- [23] Motorola homepage(URL <http://www.motorola.com>)
- [24] DSP Group homepage(URL <http://www.dspg.com>)



별첨 1. QMF 전체 블록도

Project

■ Object

■ Design Digital Signal Processor for Signal Analysis

■ Application : All-Pole Modeling by Linear Prediction with Application to Signal Analysis

■ Schedule

DATE TASK

- 1st week DSP가 수행할 알고리즘 선정 및 Proposal 작성
- 2nd week C language를 이용하여 Linear Prediction 알고리즘 구현
- 3rd week Instruction Set Design
- 4th week Assembler Design 및 Assembly Implementation
- 5th week Define Pipeline Stage
- 6th week Data-Path Design(VHDL/ALTERA 이용)
- 7th week Control-Part Design(FPGA, VHDL/ALTERA 이용)
- 8th week I/O-Part Design(FPGA, VHDL/ALTERA 이용)
- 9th week Fault-Simulation Test
- 10th week Power Evaluation
- 11th week FPGA Board Implementation 및 DEMO Presentation
- 12th week Overall Test & Evaluation(Performance, Power, Size)
- 13th week Final Report 작성

■ DEMO Application 요약 : original signal을 입력하여 predictor parameter(coefficients)를 계산한후 이 parameter를 이용하여 다시 signal을 synthesis하여본다. original signal과 synthesise된 signal을 비교하여본다.

Project

■ Date: March 15, 1999

■ Author: 김승현

■ Title: Implementation of a low-pass FIR filter based on a DSP processor

■ Objectives:

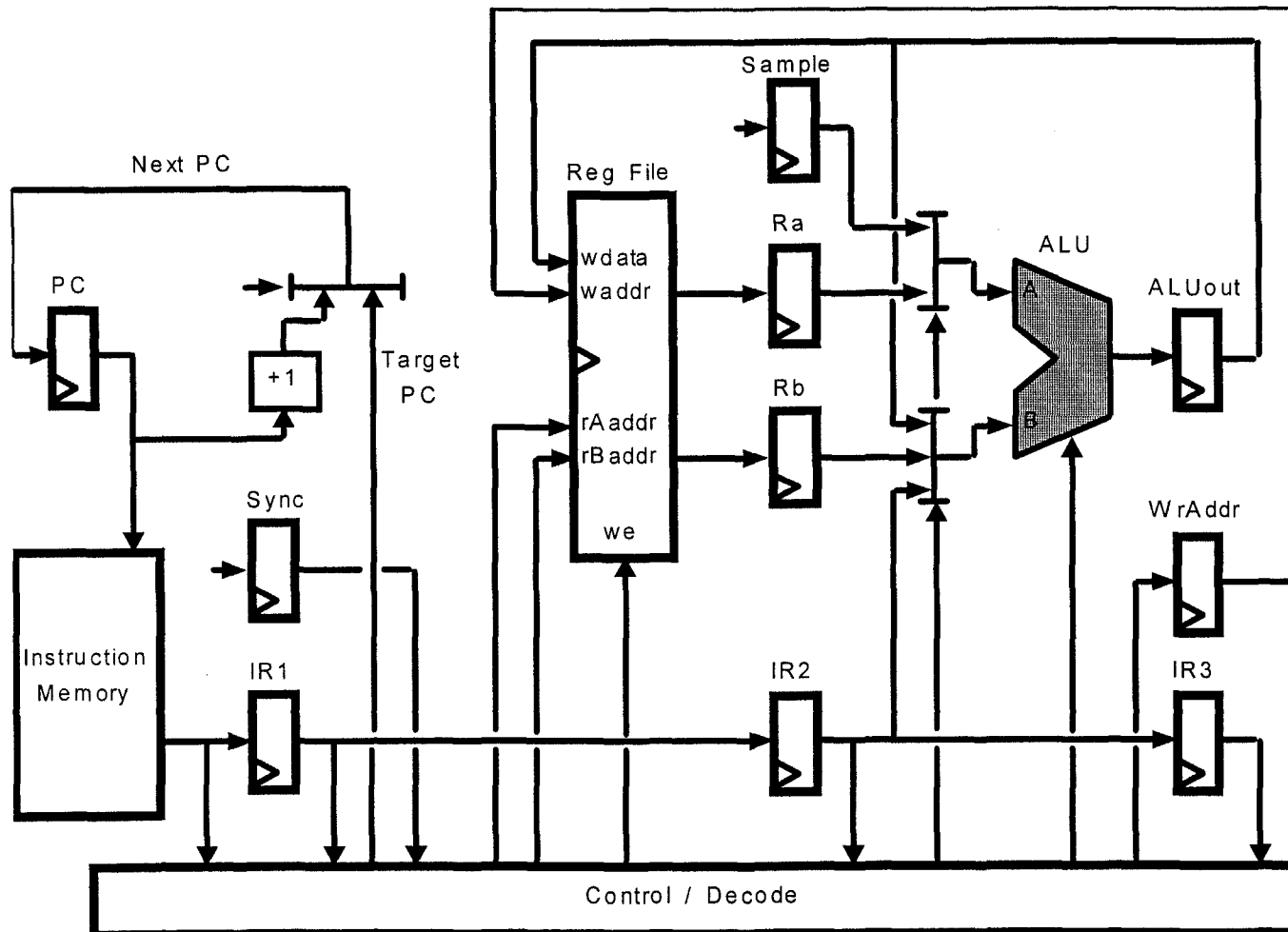
- Design and implement a DSP processor (ISA and Assembler)
- Design and implement a low-pass FIR filter
- Power estimation and fault simulation

■ Specification:

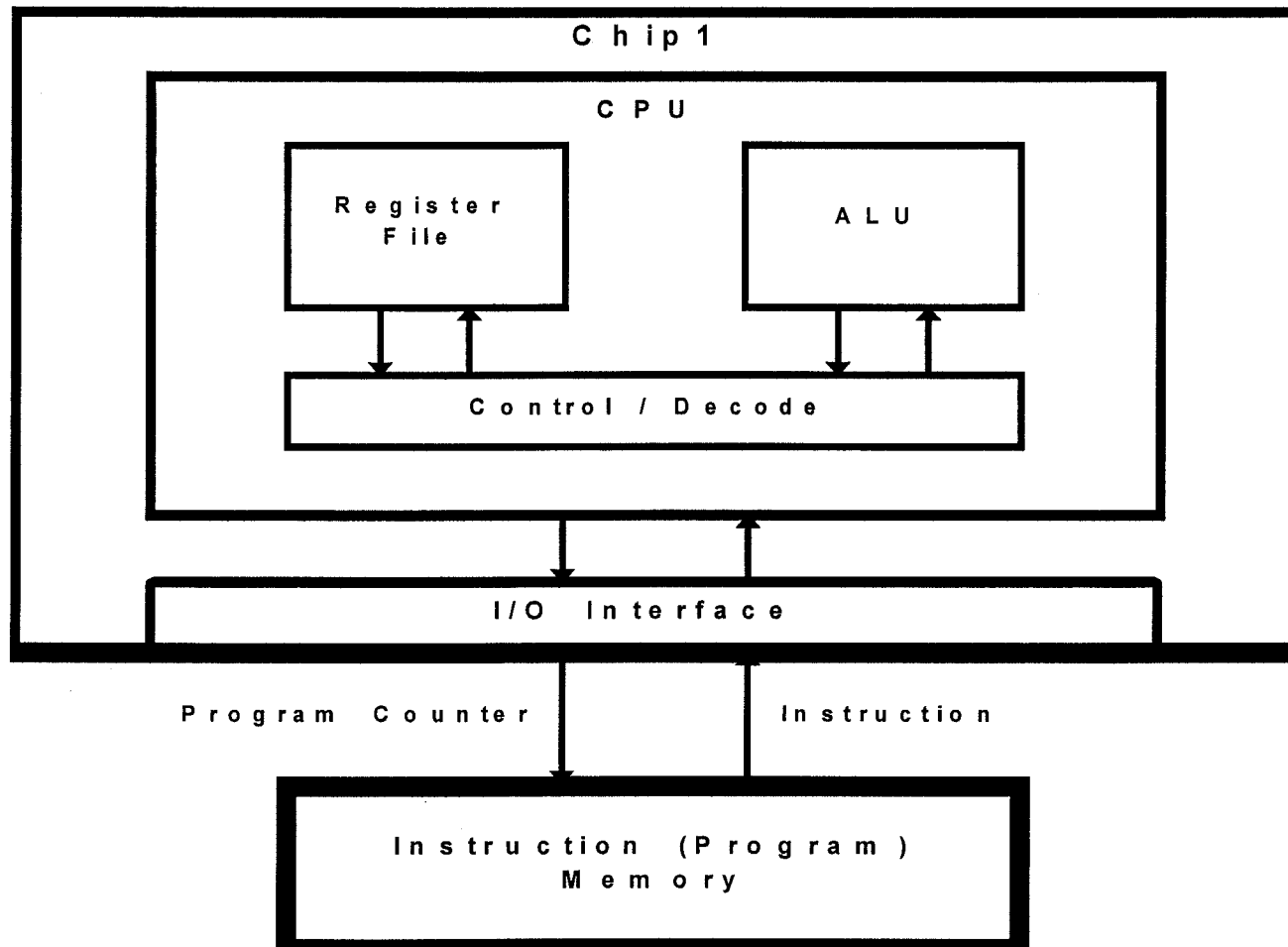
- Sampling frequency: 16 KHz
- Passband frequency: 5 KHz
- Stopband frequency: 6 KHz
- Gate size: less than 50,000 gates
- Power: less than 100 mW
- Data bus width: 16 bits
- Fault coverage: 80 %
- Passband ripple: 3 dB
- Stopband attenuation: 60 dB

■ Demo Plan:

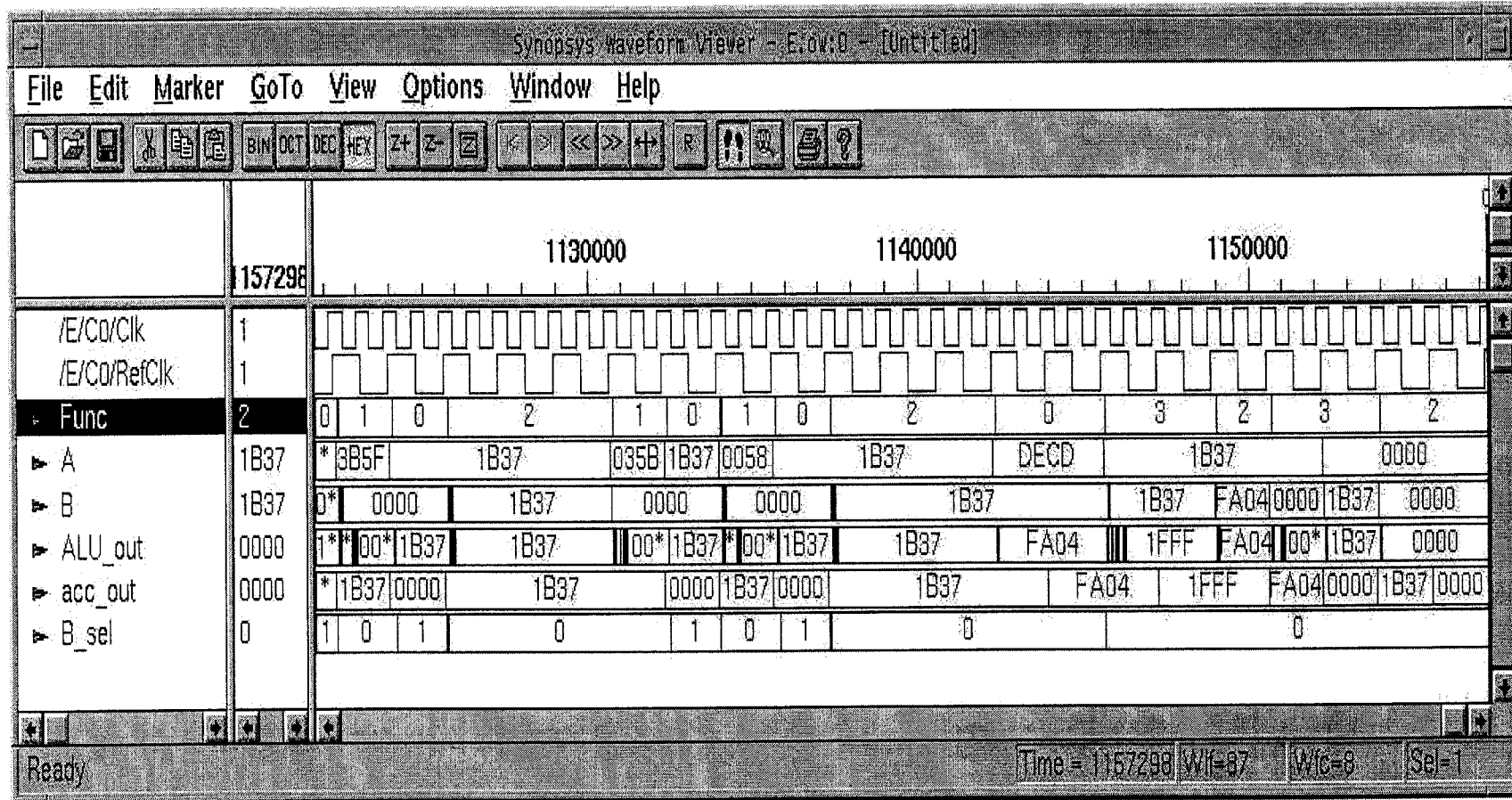
The input signal (sound) below a specified passband frequency is passed (heard), and the input signal above the passband frequency is rejected (not heard).



별첨 4. FIR LP 필터 설계 칩 구조



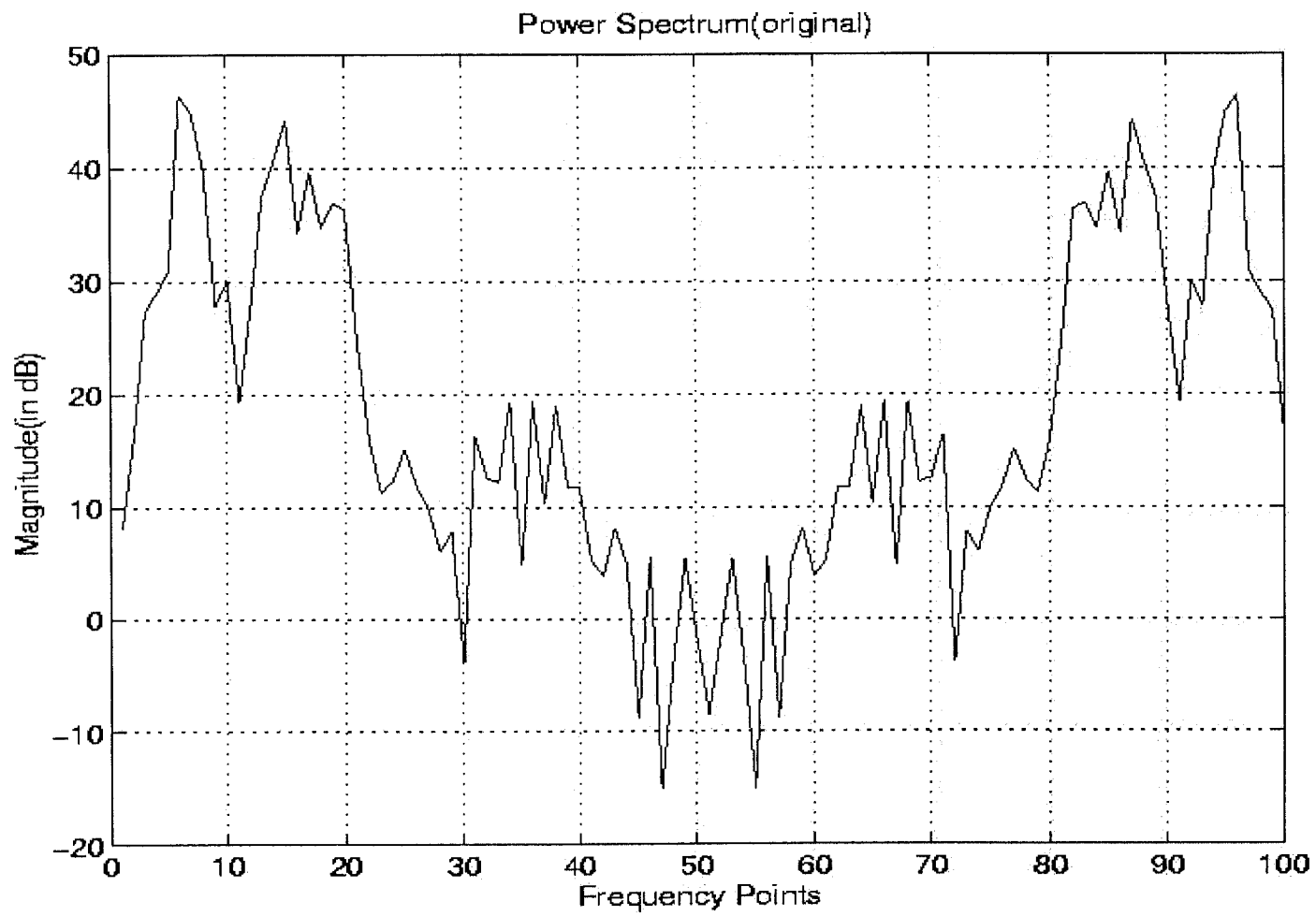
별첨 5. FIR LP 필터 전체 블록도



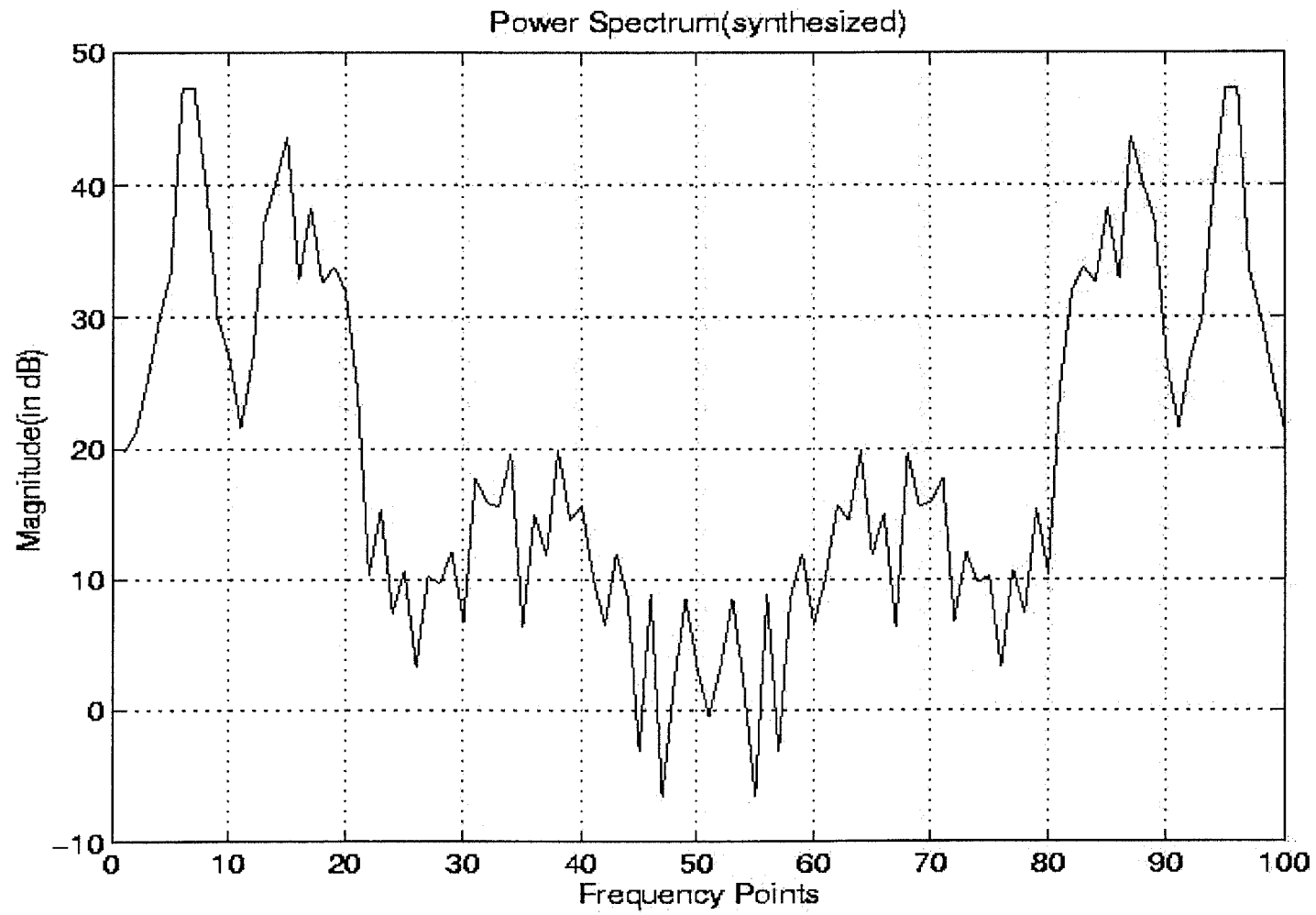
별첨 6. 설계된 ALU 관련 신호도

LPC Coefficients:	Original Coefficients:
1.0000	1.0000
-1.9551	-2.1346
2.0439	2.5473
-1.3301	-1.9539
1.2354	1.6693
-1.5762	-1.8591
1.3301	1.5996
-0.4316	-0.5565
0.1045	-0.1212
-0.2070	0.1773
0.3906	0.1189
-0.0977	-0.0243
0.0107	0.0167

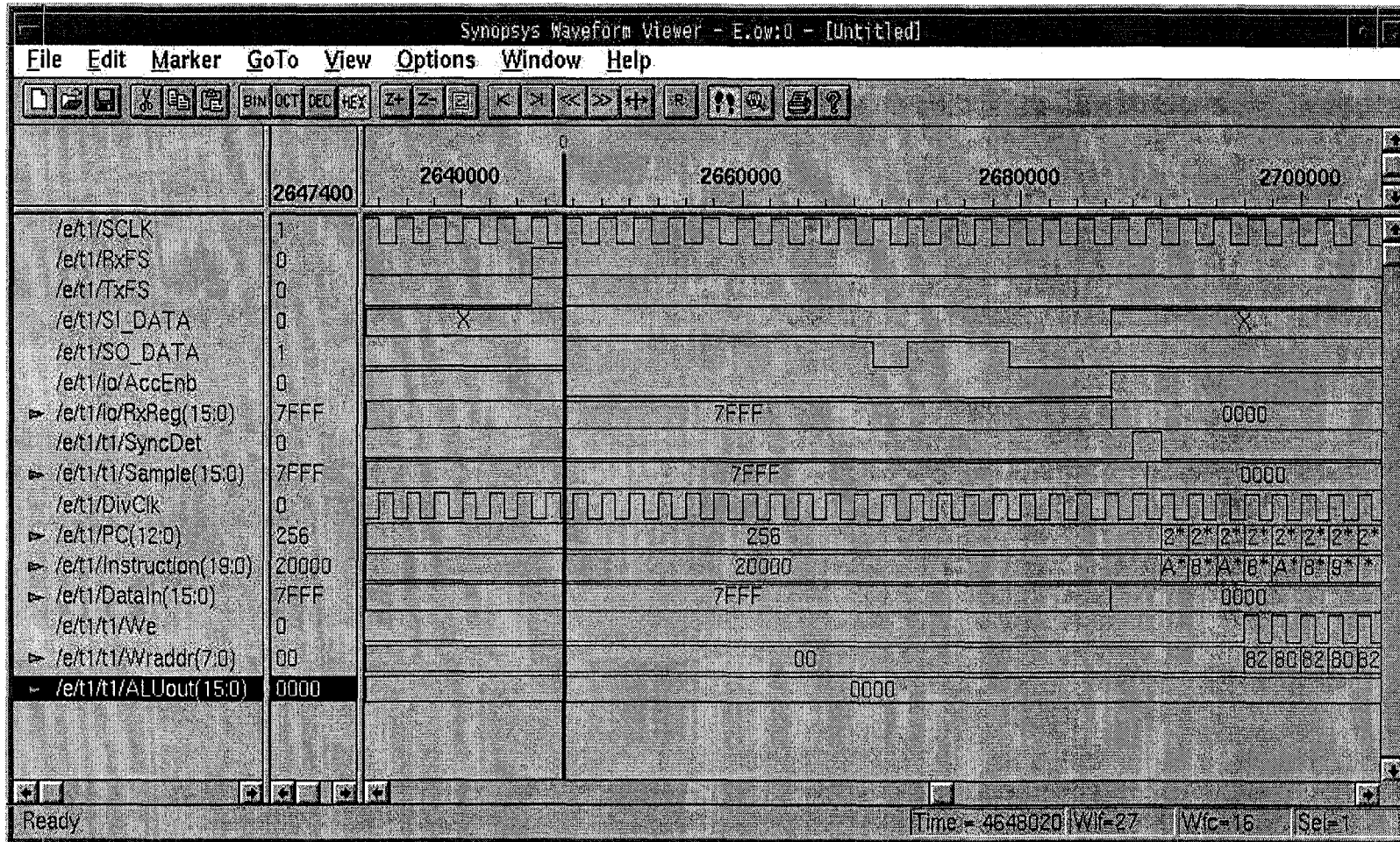
별첨 9. 추출된 Coefficient 값



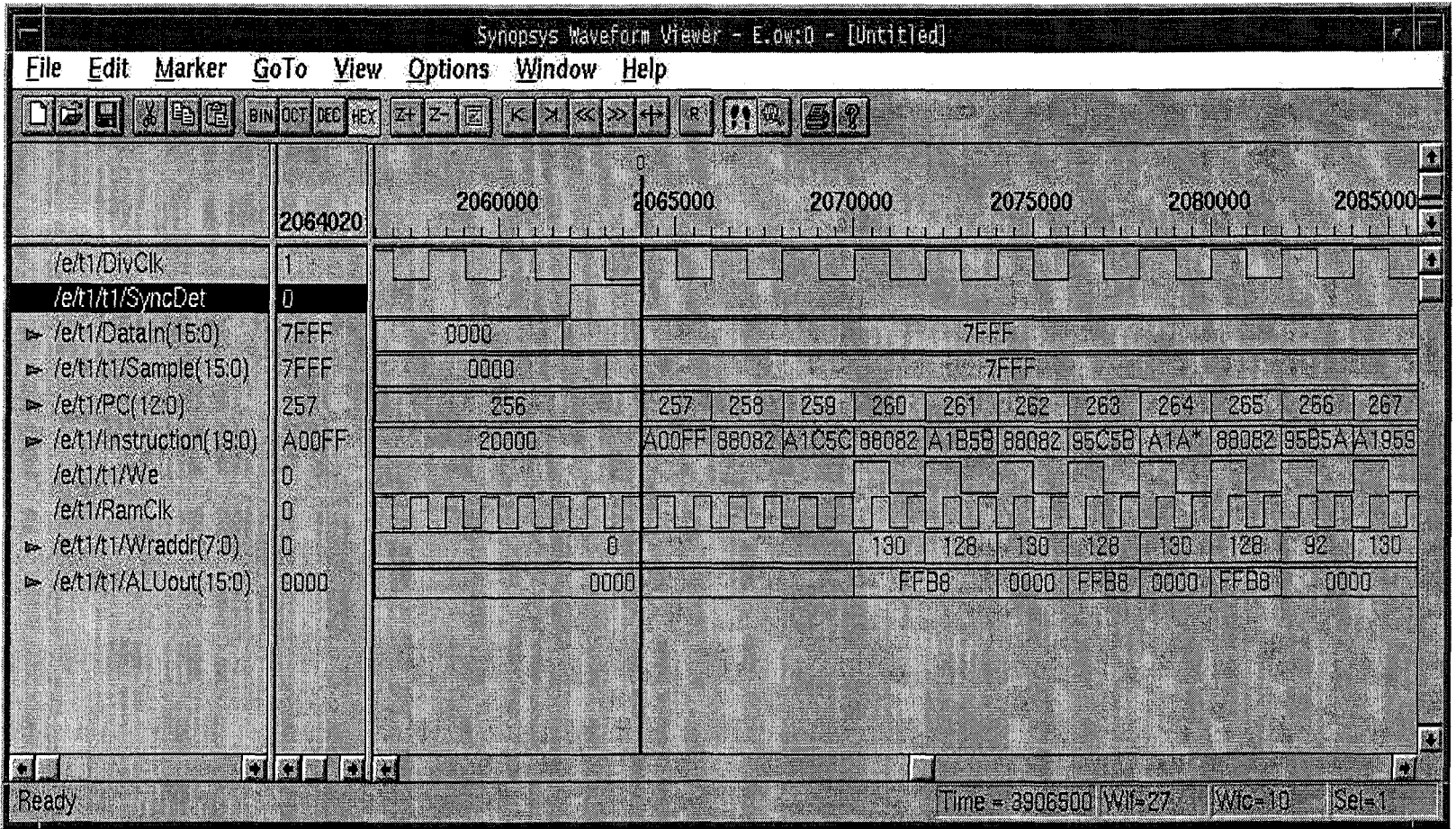
별첨 10. 입력신호 스펙트럼



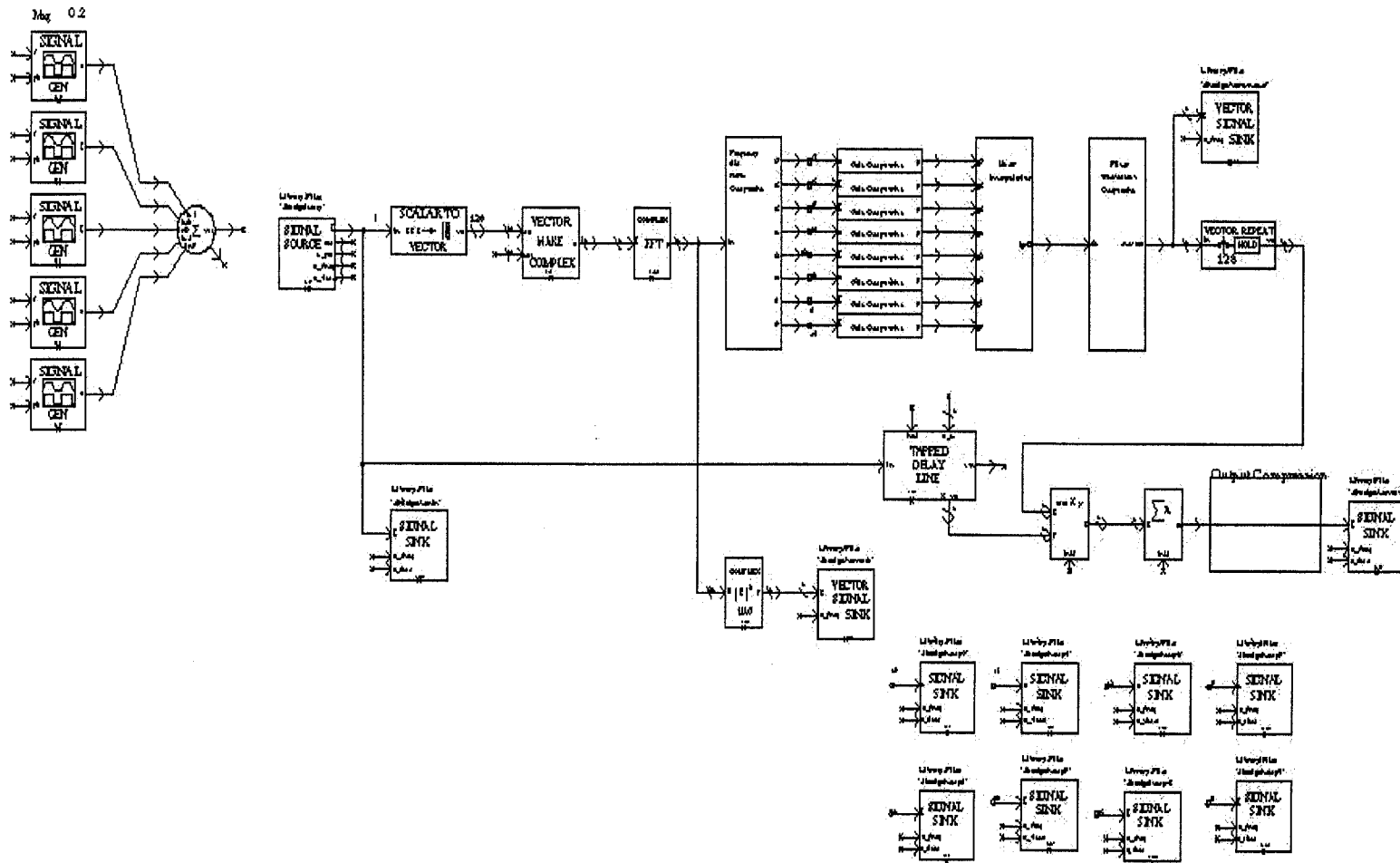
별첨 11 . 출력신호 스펙트럼



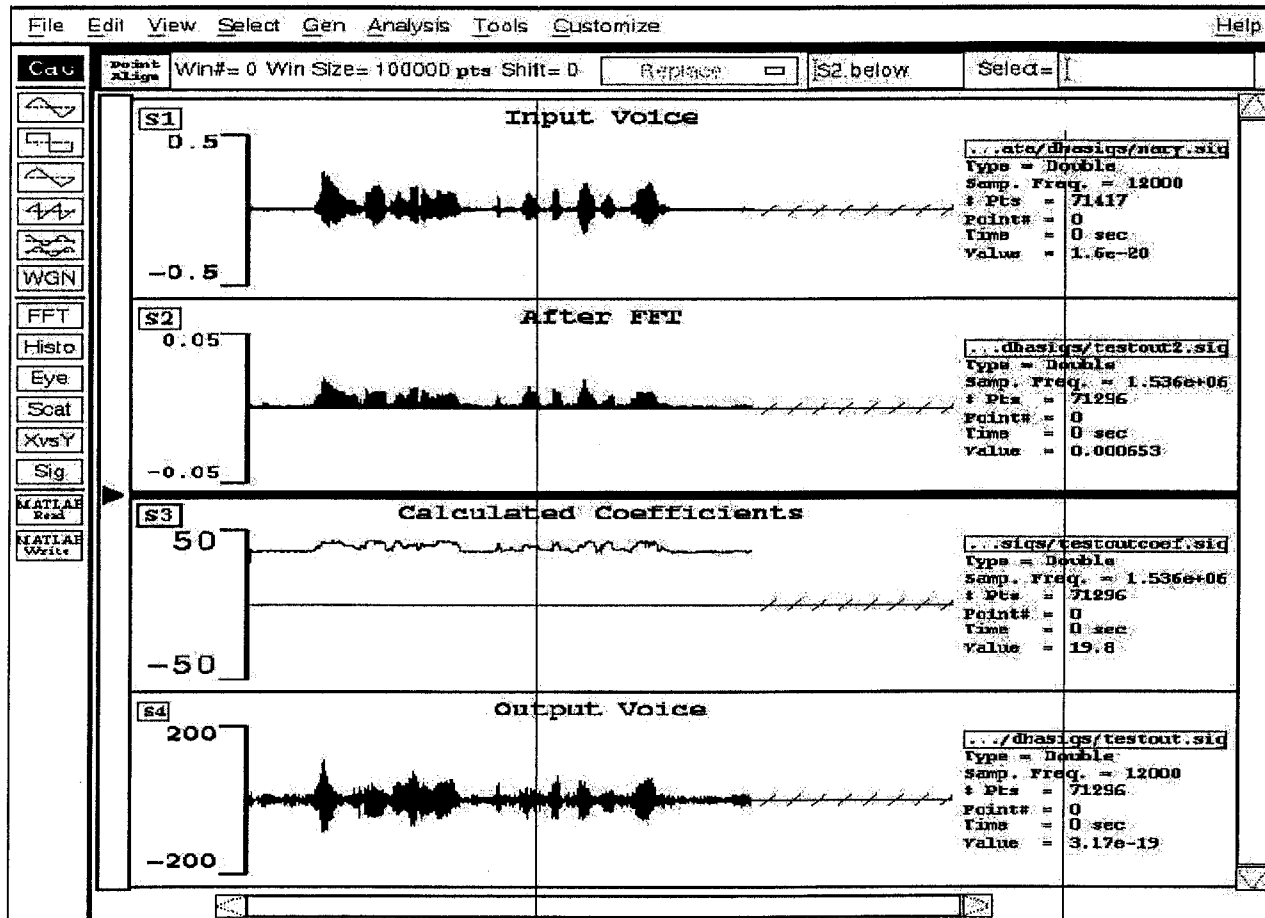
별첨 12. FIR LP 필터의 입/출력 관련 타이밍도



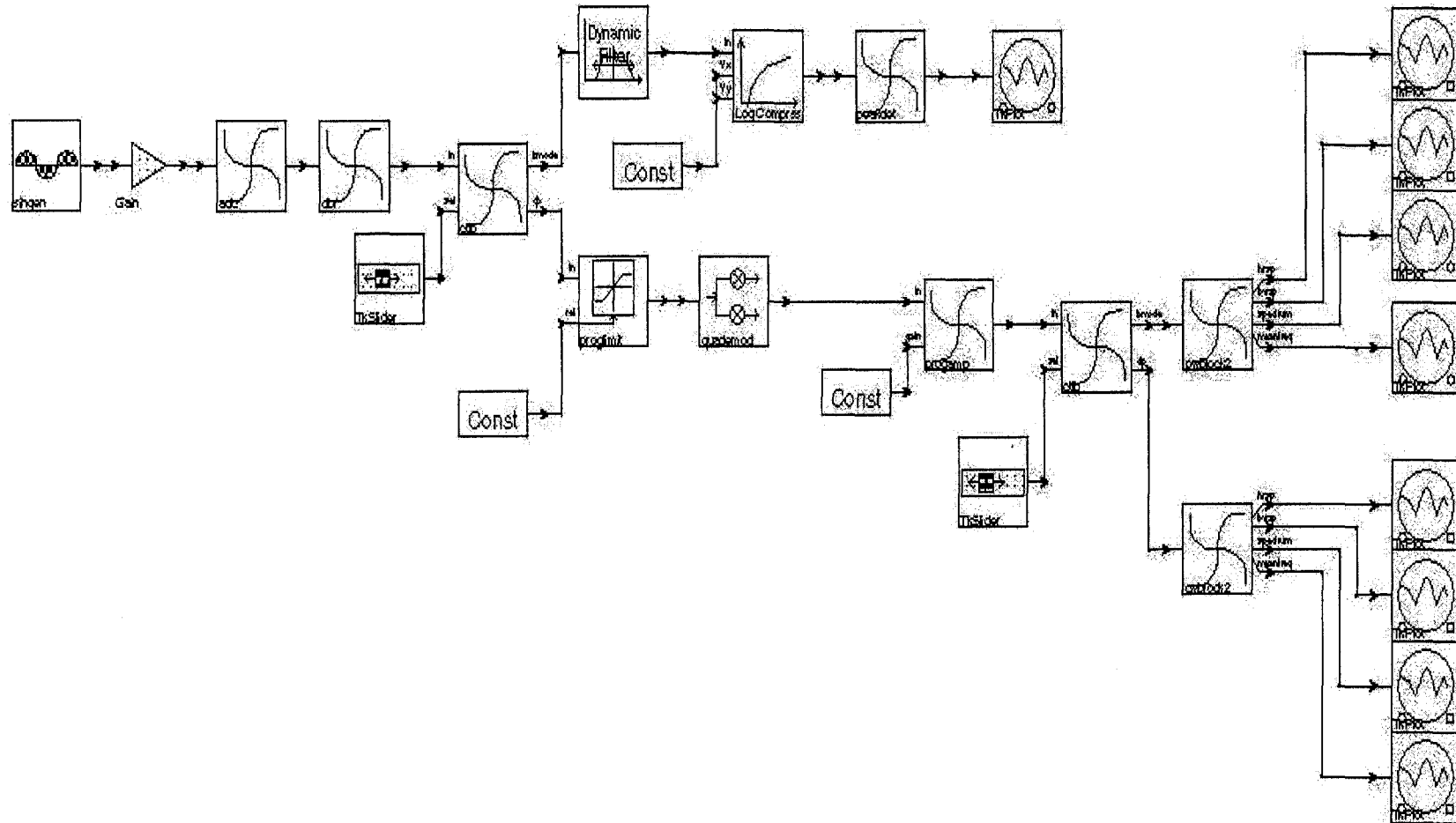
별첨13. FIR LP 필터의 파이프라인 관련 타이밍도



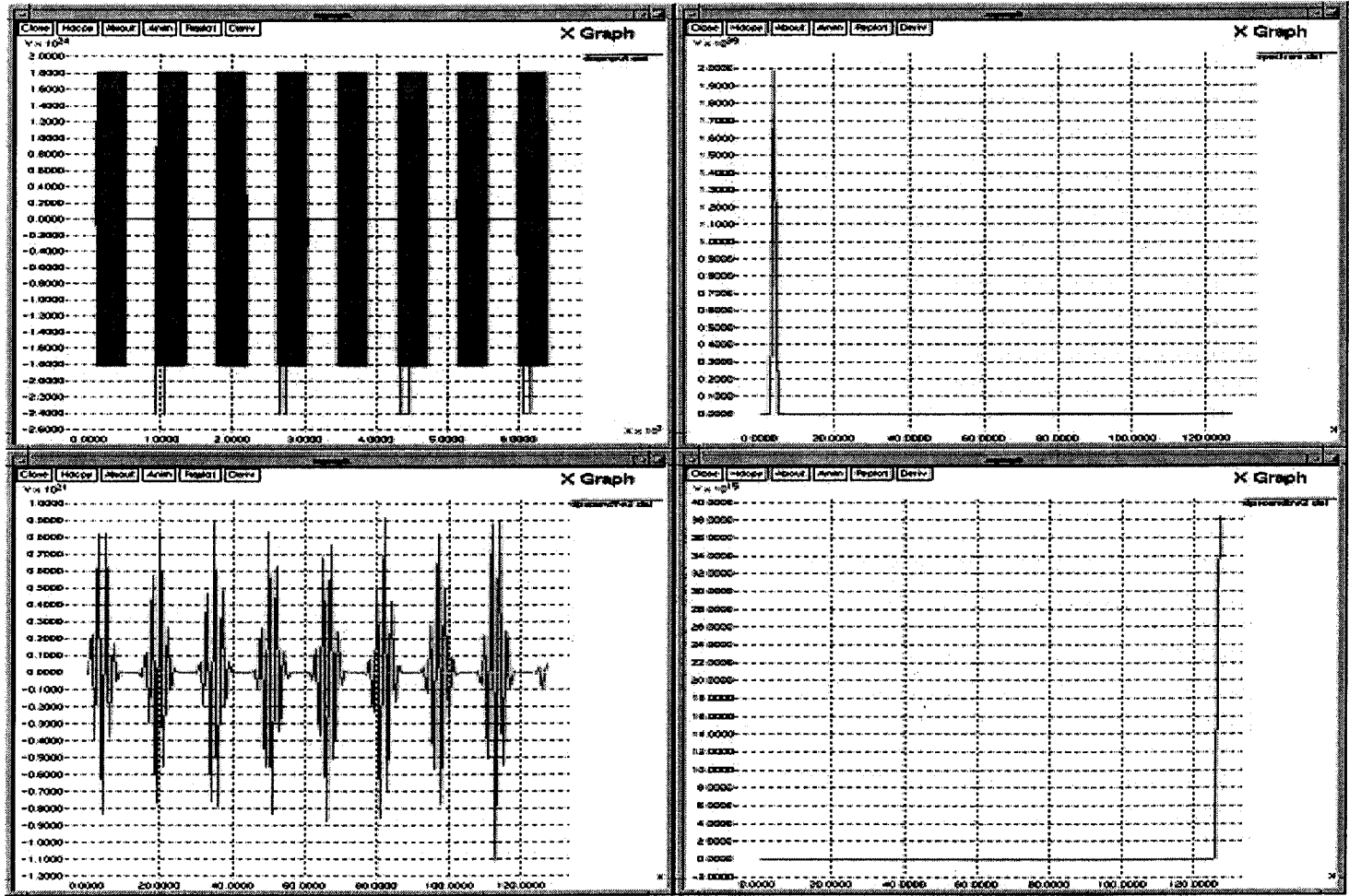
별첨 14. SPW Voice signal processor 블록도



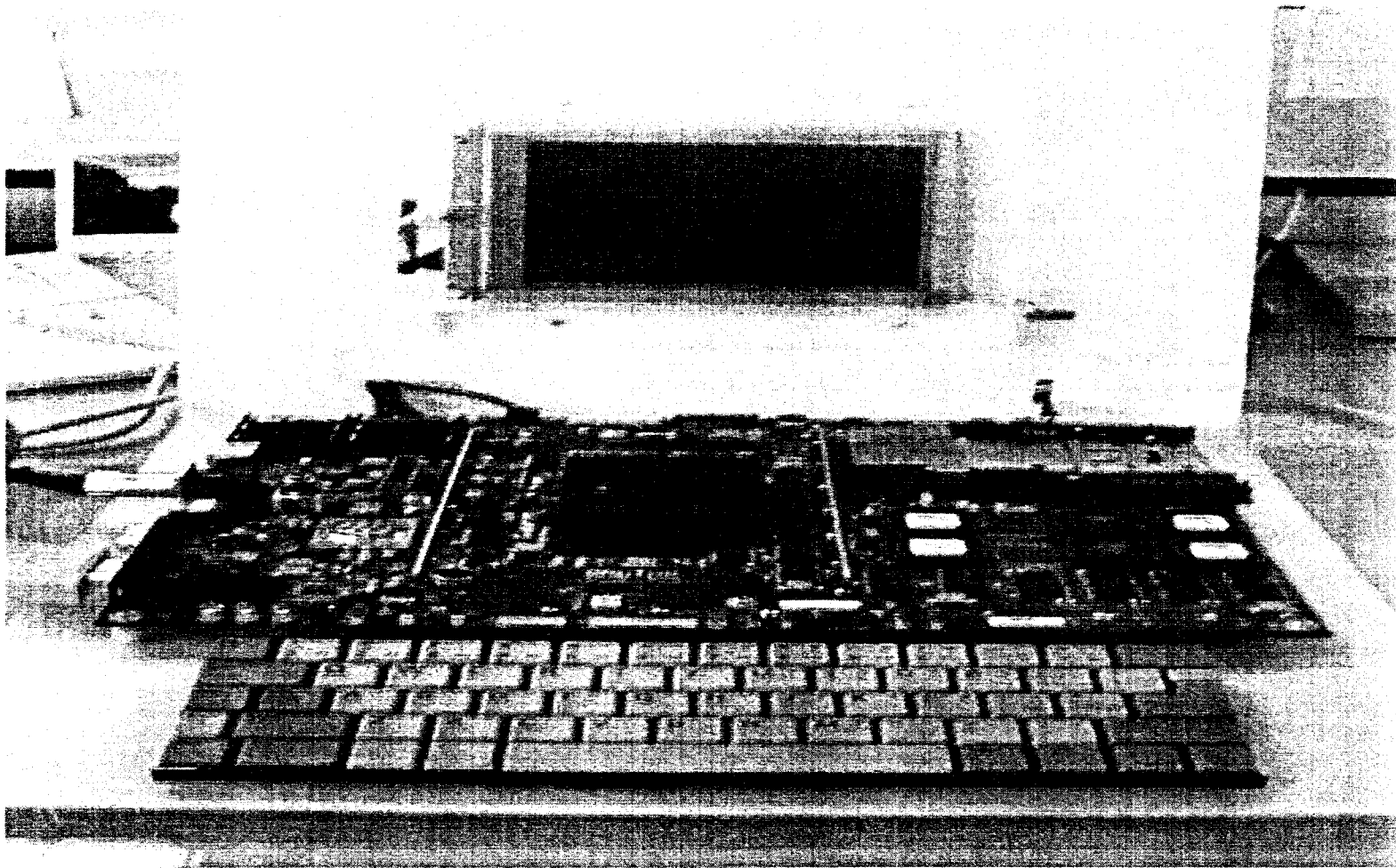
별첨 15. SPW 로 설계된 Voice signal processor 신호도



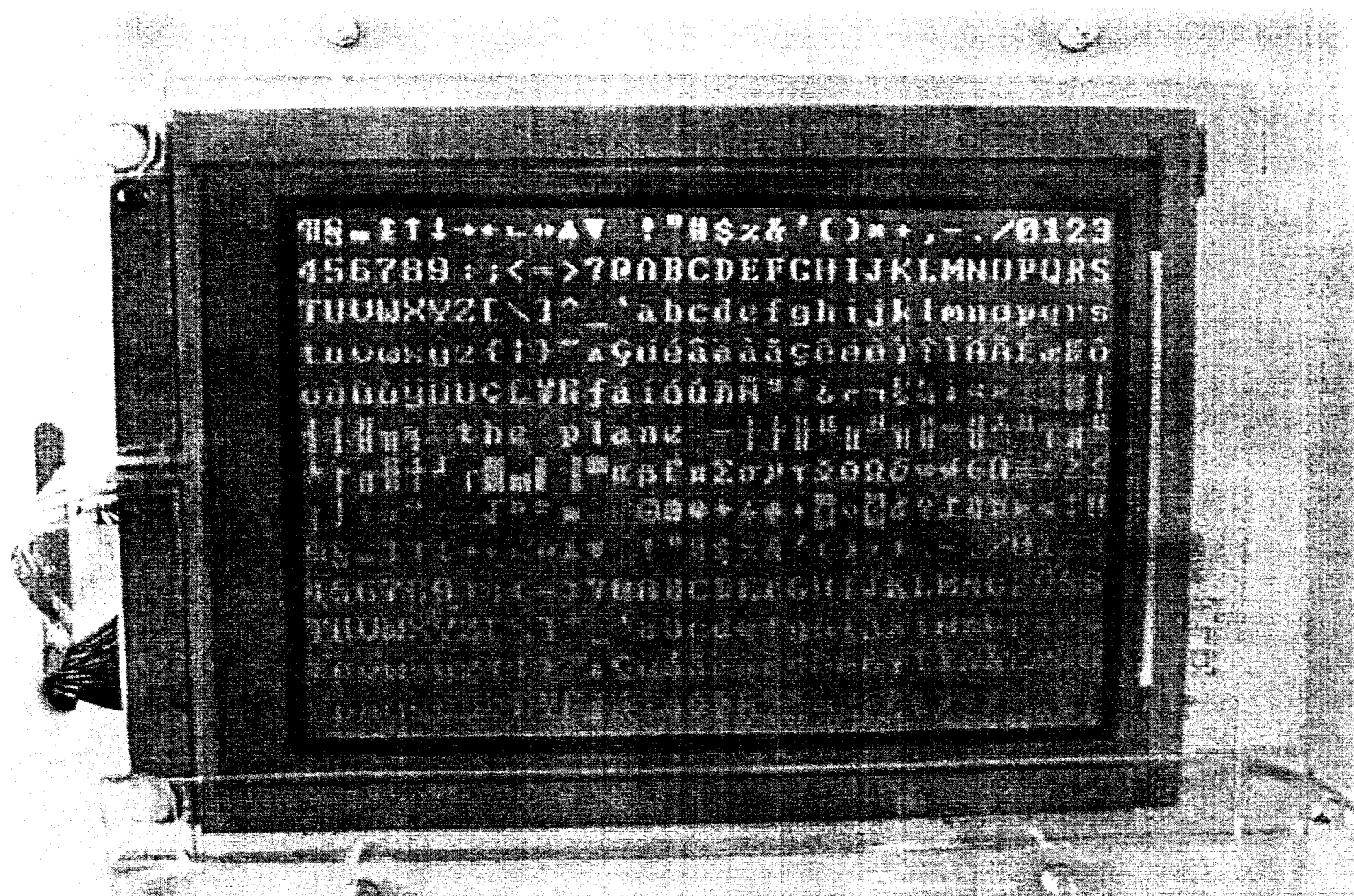
별첨 16.Ptolemy로 구현된 sonic 블록도



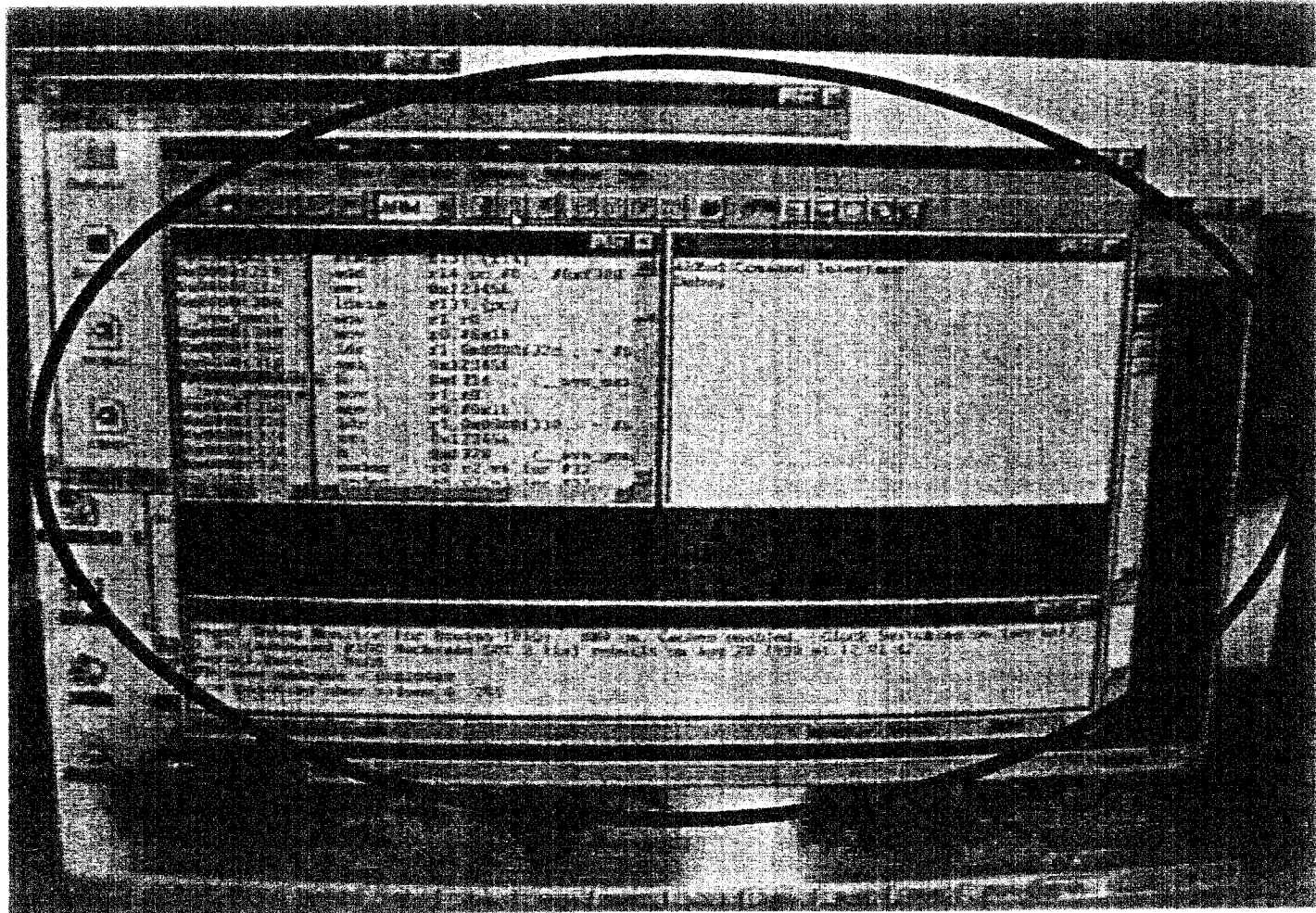
별첨 17. Ptolemy로 구현된 sonic 동작신호



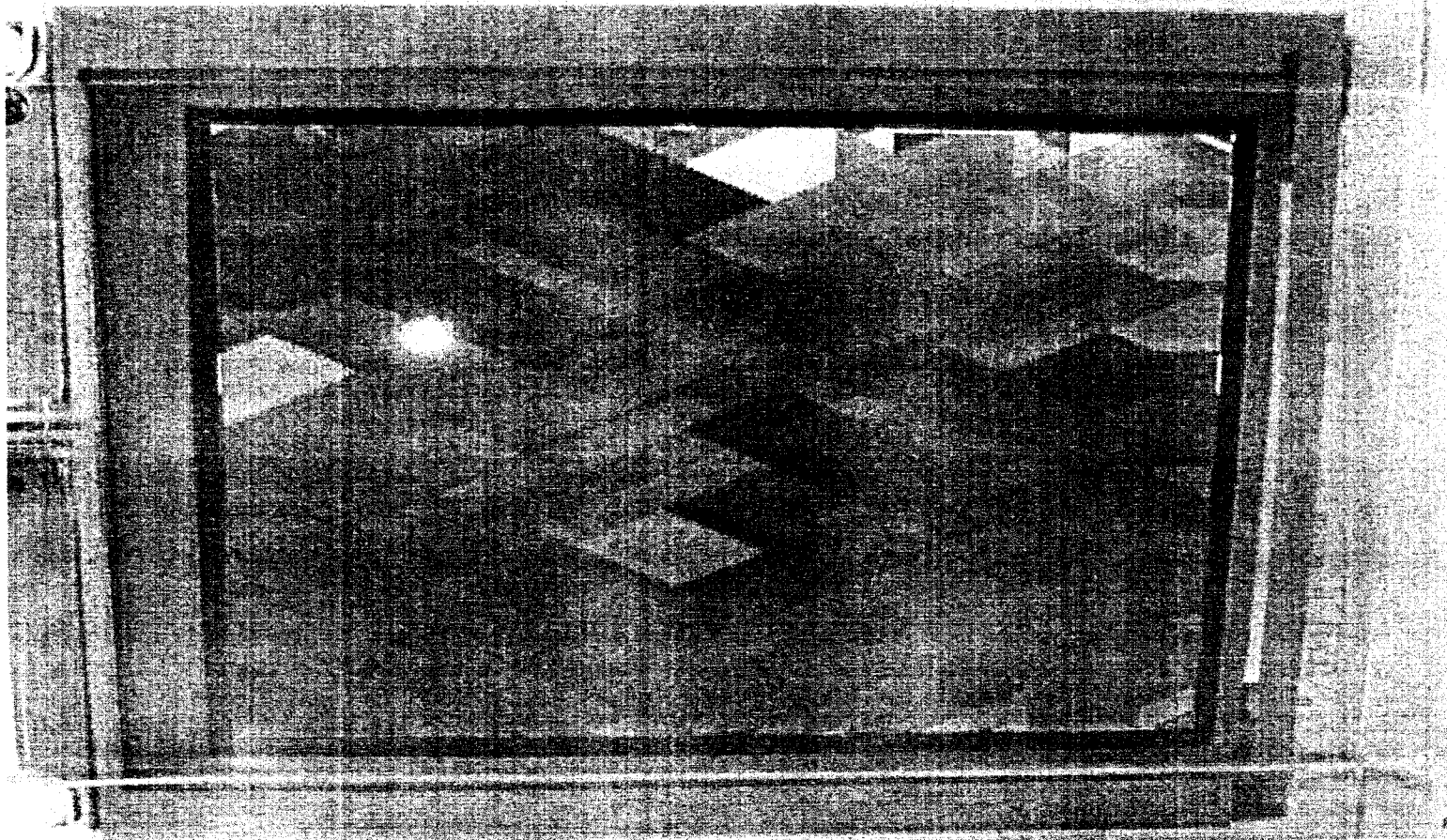
별첨 18. StrongARM 개발도구



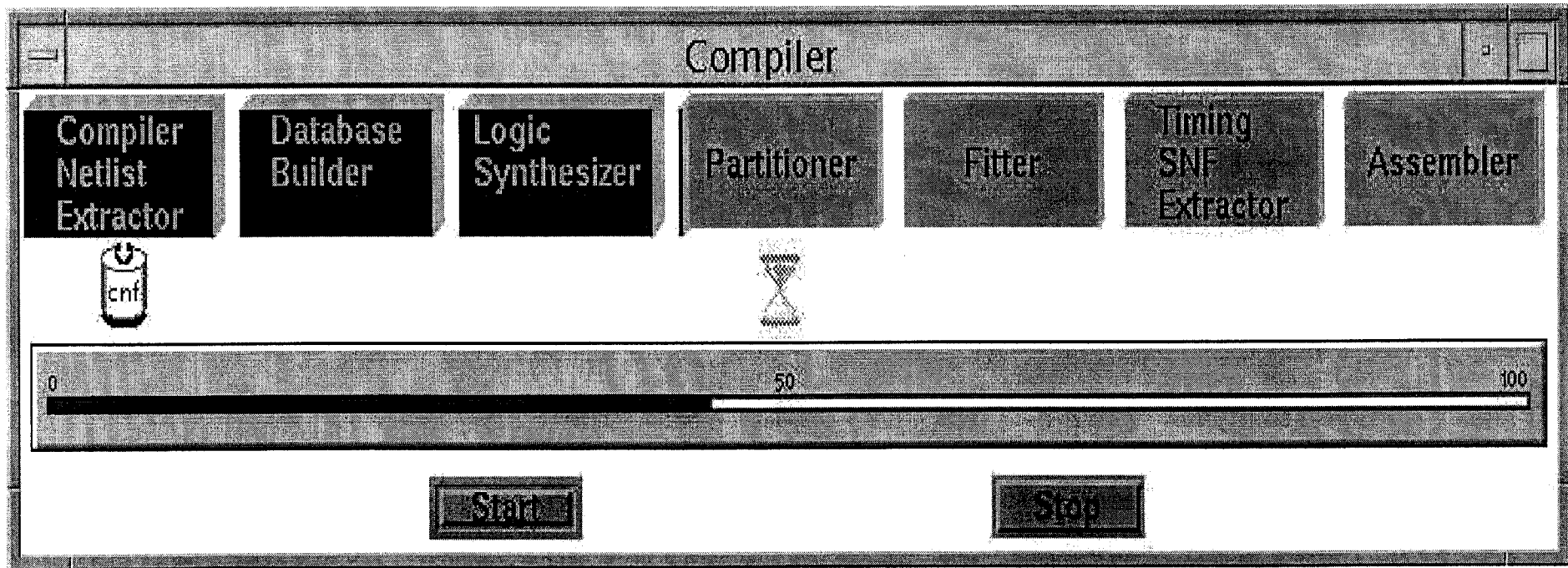
별첨 19. ROM 동작 확인 (보드에 장착되어 있는 프로그램 R/W)



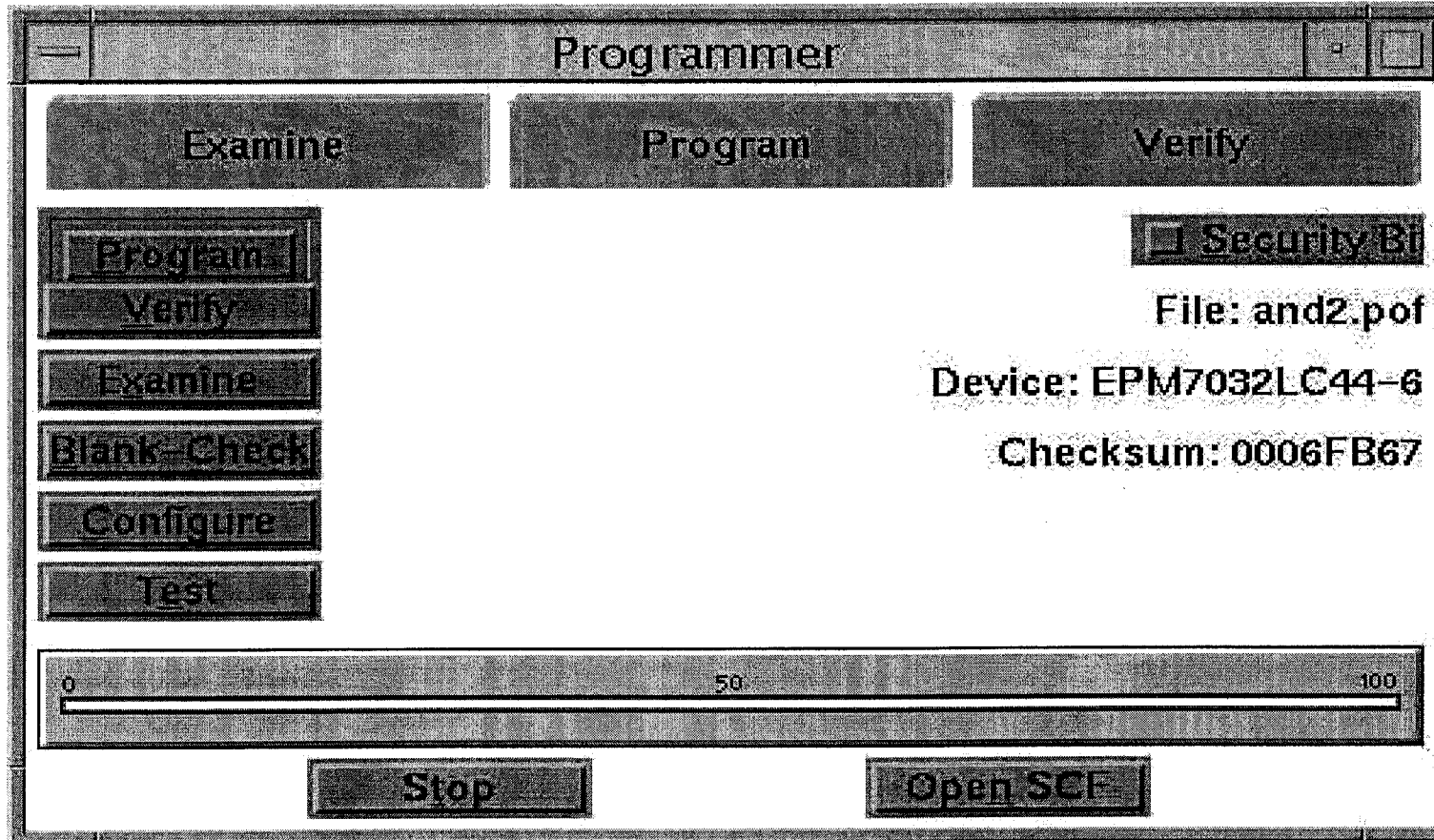
별첨 20. RS-232C 송수신 체크 (PC 와의 송수신)



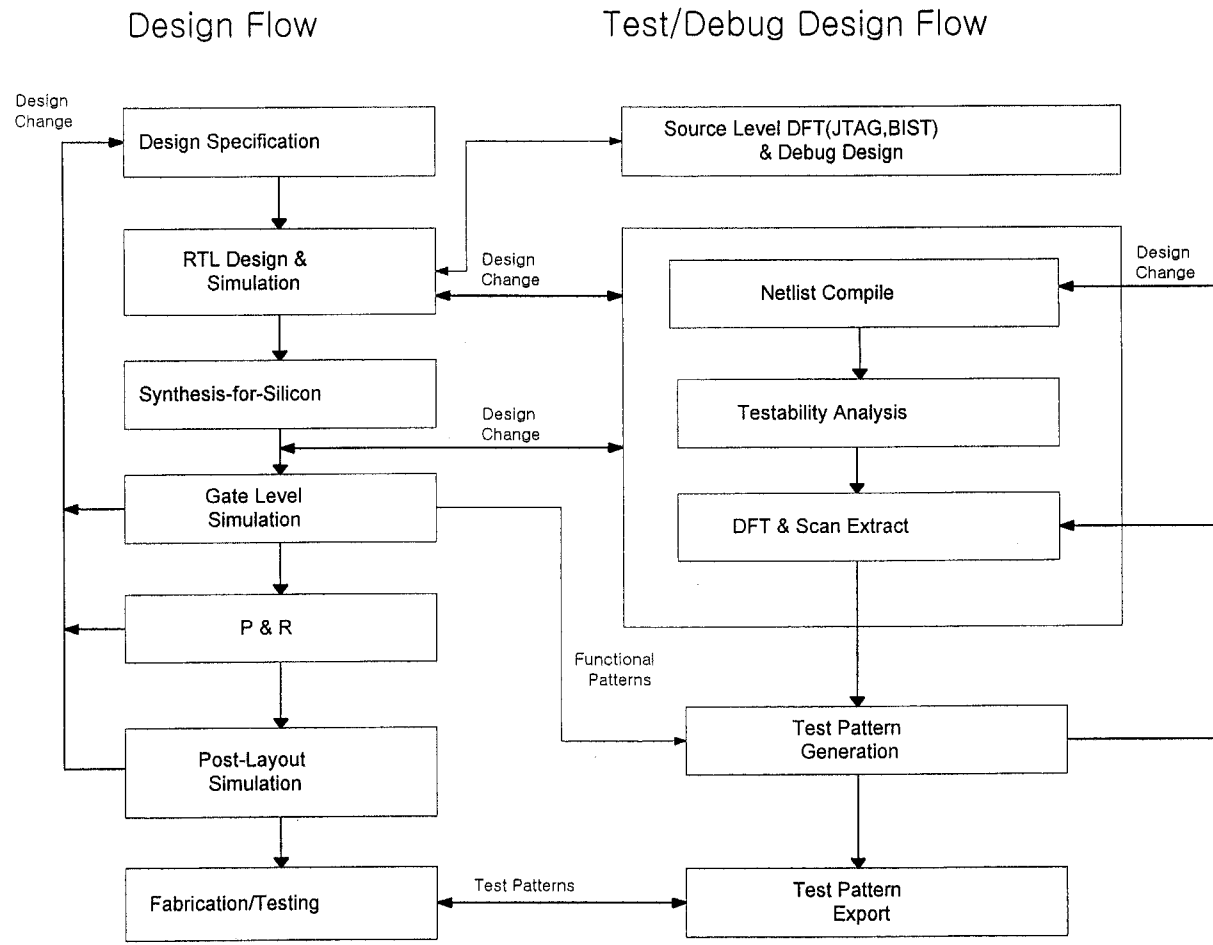
별첨 21. LCD 그래픽 표시 테스트



별첨 22. FPGA placement & routing



별첨 23. FPGA device programming



별첨 24. 설계와 테스트의 협동연구 추진 체계도

80개 sample 당 계산량	PreProcessing block		
	PreEmphasis	Energy	FFT
곱셈	160	240	4096
덧셈	160	239	6144
나눗셈			1
제곱근			1
10*log10() loge()			
합계	곱셈		4496
	덧셈		6543
	나눗셈		1
	제곱근		1
	10*log10())		0
	자연로그		0

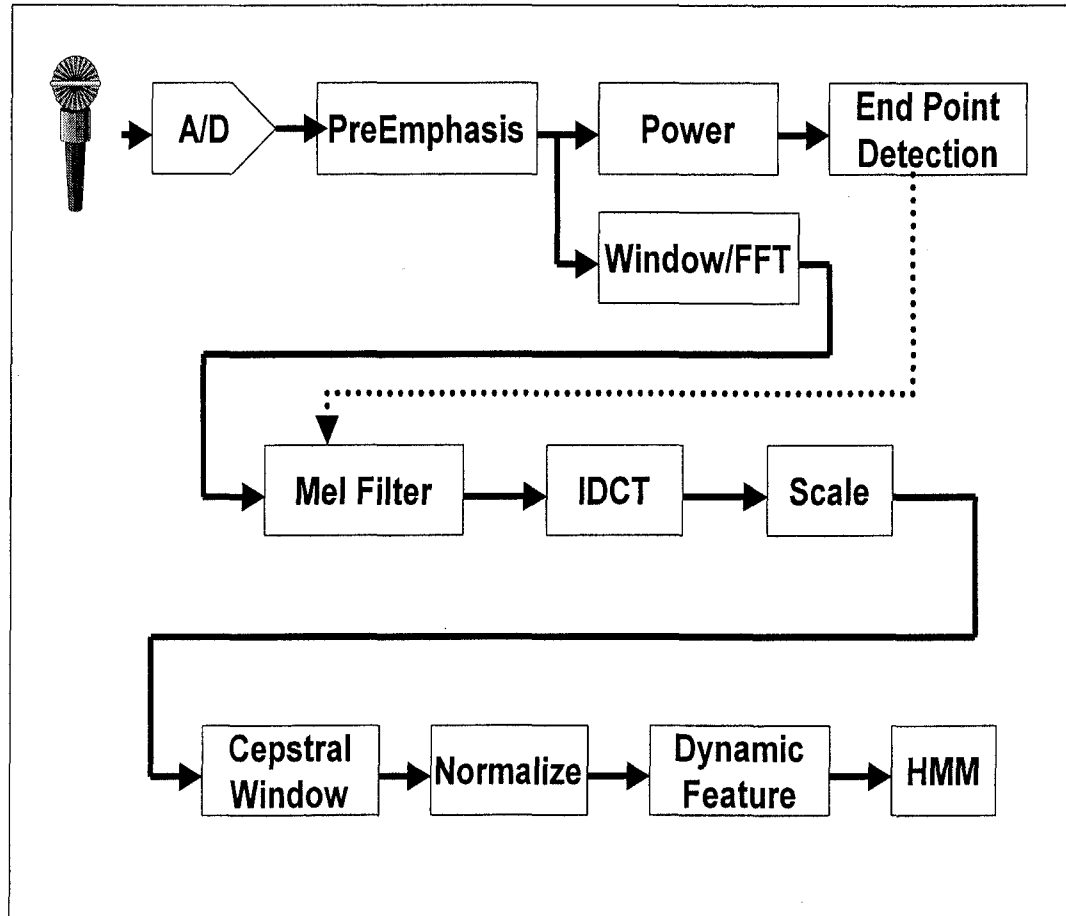
메모리 사용 량	Pre-Processing									
구분	Static Memory						Dynamic Memory			
내용 (Byte)	m_fpEne rgy	m_pInEn ergy	m_dpInS pec	m_fpWin dow	m_fpDat aA,B	m_dpOut Spec	Pre-Emp hasis	Energy	FFT	
		120	1016	131064	960	2048	164088	fpInBuf		
								10240		
										fdpSpec
										30720
										farray
										960
총합	300k						10240	10240	41920	

별첨 25. Preprocessing 부분 계산량/메모리 소요량

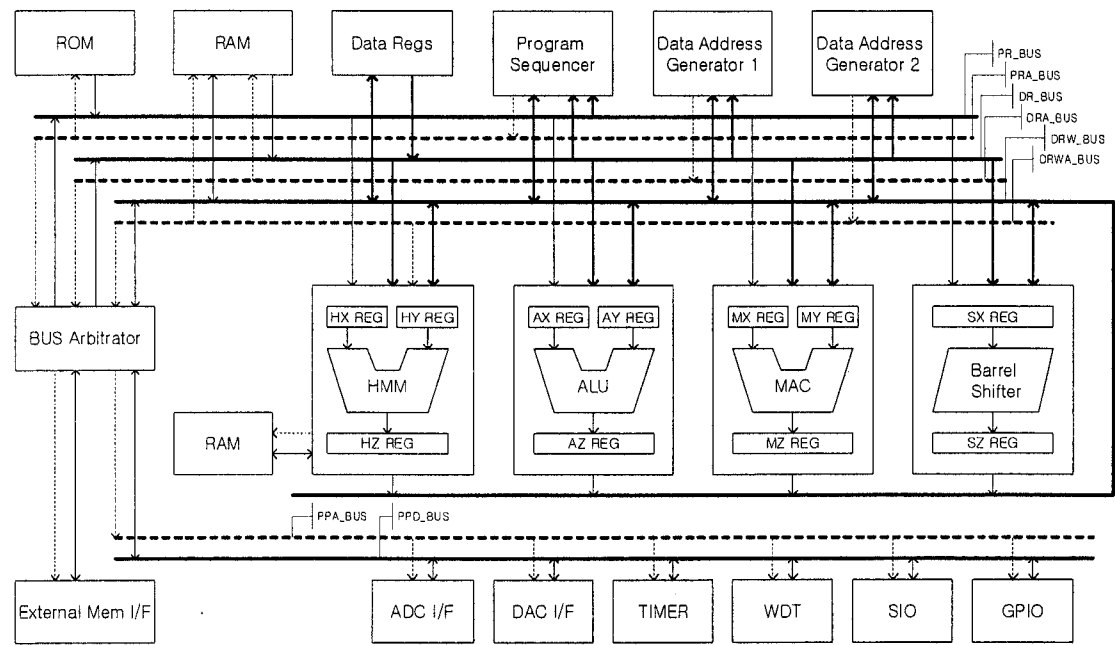
80개 sample 당 연산 량		Melfiltering & Cepstrum				
	Melfilter	IDCT	Scale	C e p Window	Normalize	D y n a m i c Cepstrum
곱셈	234	288	9	8	1	27
덧셈	202	279			1	
나눗셈						9
제곱근						
10*log10()	32					
loge()				1		
합계	곱셈					567
	덧셈					482
	나눗셈					9
	제곱근					0
	10*log10()					32
	loge()					1

메 모 리 사용량		Melfiltering & Cepstrum				
구분		Dynamic Memory				
내용 (Byte)	Melfilter	IDCT	Scale	C e p Window	Normalize	D y n a m i c Cepstrum
	fft_coefss					
	162816					
	mfsc->spectral_data					
	12800					
	mfcc->spectral_data					
	3600					
						cep
						7632
						o_data
					7344	
총합	162816	175616	16400	3600	3600	18576

별첨 26. 음성 특성치 추출을 위한 계산량/메모리 소요량



별첨 27. 화자독립/어휘독립 50단어 음성인식 알고리즘의 전체 Block Diagram



별첨 28. 음성인식용 ASSP의 기본 아키텍처

실현 명령 세트1(데이터전송)

Syntax		Expression
Mnemonic	Operand	
MOV	src, dst	dst = src
MOVI	#num, dst	dst = #num
LOAD	[src], dst	dst = [src] ; [src]: = Source memory address
STORE	src, [dst]	[dst] = src ; [dst]: = Destination memory address
POP	<dst1, dst2, ...>	dst1 = Data_mem(SP++), dst2 = Data_mem(SP++), ...
PUSH	<src1, src2, ...>	Data_mem(--SP) = src1, Data_mem(--SP) = src2, ...
LOADD	\$Smem, dst	dst = \$Smem
STORED	src, \$Dmem	\$Dmem = src
MDTD	src, dst, {#size}	[dst] = [src] / \$Dmem = \$Smem
	\$Smem, \$Dmem, {#size}	
MDTP	src, dst, {#size}	Having no size factor, controller uses repeat count register.
	\$Smem, \$Dmem, {#size}	
MPTD	src, dst, {#size}	
	\$Smem, \$Dmem, {#size}	

실현 명령 세트2(산술연산)

Syntax		Expression
Mnemonic	Operand	
ADD	src1, src2, dst	dst = src1 + src2
	\$Xmem, \$Ymem, dst	dst = \$Xmem + \$Ymem
ADDC	src, dst	dst = dst + src + C
	\$Smem, dst	dst = dst + \$Smem + C
SUB	src1, src2, dst	dst = src1 - src2
	\$Xmem, \$Ymem, dst	dst = \$Xmem - \$Ymem
SUBB	src, dst	dst = dst - src - c
	\$Smem, dst	dst = dst - \$Smem - c
INC	src, {dst}	dst = src + 1 or src = src + 1
DEC	src, {dst}	dst = src - 1 or src = src - 1
CMP	src1, src2	src1 - src2 ; only changes flag register
CMPI	src, #num	src - #num ; only changes flag register
MUL	src1, src2, dst	dst = src1 * src2
	\$Xmem, \$Ymem, dst	dst = \$Xmem * \$Ymem
SQUR	src, dst	dst = src ²
	\$Smem, dst	dst = \$Smem ²
MAC	src1, src2	MAC_reg = src1 * src2 + MAC_reg
	\$Xmem, \$Ymem	MAC_reg = \$Xmem * \$Ymem + MAC_reg
		* MAC_reg selection => MAC_reg0, MAC_reg1
SQURA	src	MAC_reg = src ² + MAC_reg
	\$Smem	MAC_reg = \$Smem ² + MAC_reg
		* MAC_reg selection => MAC_reg0, MAC_reg1
ADDAC	src1, src2	AZ_reg = src1 + src2 + AZ_reg
	\$Xmem, \$Ymem	AZ_reg = \$Xmem + \$Ymem + AZ_reg

실현 명령 세트3(논리연산, 시프트)

Syntax			Expression
Mnemonic	Operand		
AND	src1, src2, dst	dst = src1 & src2	
	\$Smem, src, {dst}	dst = \$Smem & src or src = \$Smem & src	
OR	src1, src2, dst	dst = src1 src2	
	\$Smem, src, {dst}	dst = \$Smem src or src = \$Smem src	
XOR	src1, src2, dst	dst = src1 ^ src2	
	\$Smem, src, {dst}	dst = \$Smem ^ src or src = \$Smem ^ src	
NOT	src, {dst}	dst = src or src = src	
	\$Smem, dst	dst = \$Smem	

Syntax			Expression
Mnemonic	Operand		
SHR	src, dst, #num	dst = src >> (#num)	
SHL	src, dst, #num	dst = src << (#num)	
SAR	src, dst, #num	dst = src >> (#num), dst[MSB] = src[MSB]	
SAL	src, dst, #num	dst = src << (#num), dst[MSB] = src[MSB]	

실현 명령 세트4(제어)

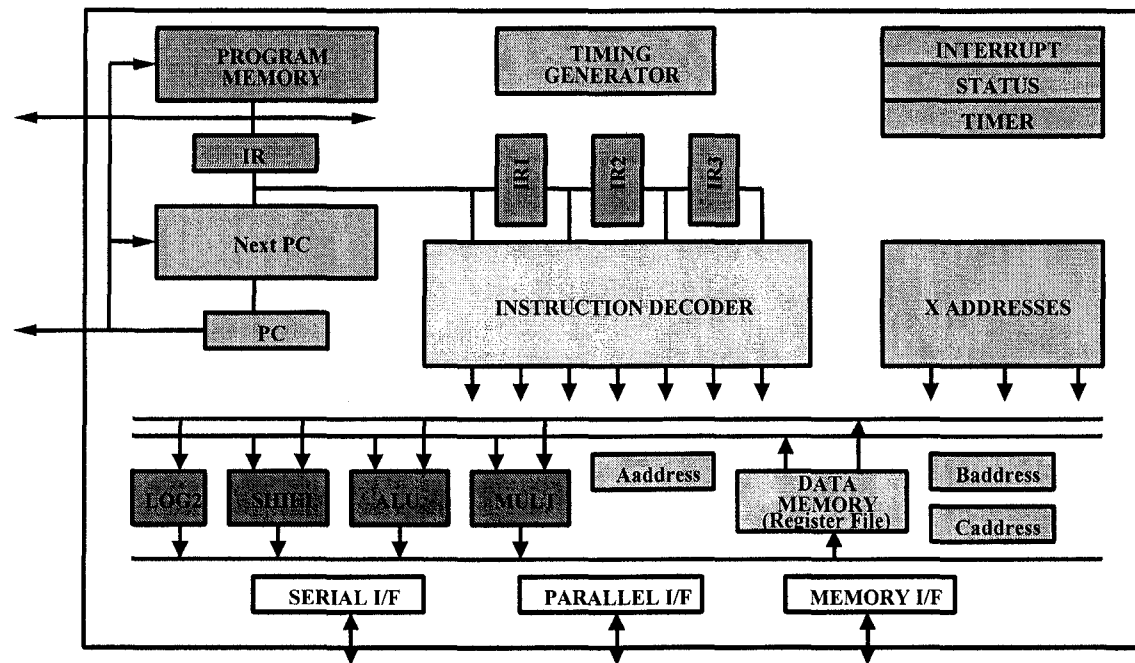
Syntax		Expression
Mnemonic	Operand	
NOP		No operation
RESET		Software reset
WAIT	#num	PC = PC + 1, #num indicate power down mode
INTR	#num	Data_Mem(--SP) = PC+1, PC = IVAR + #num * Software interrupt, IPR : Interrupt Vector Address Register
CALL{D}	\$Padr	Data_Mem(--SP) = PC+1(or PC + 3), PC = \$Padr * SP : Stack Point, Padr : Program address
RET{D}		PC = Data_Mem(SP++)
RETI{D}		PC = Data_Mem(SP++), Interrupt control bit = enable
JMP{D}	\$Padr	PC = \$Padr, * Unconditional jump
JMPC{D}	\$Padr, @cond	PC = \$Padr if @cond is true, else PC = PC + 1
RPT{D}	#num	Repeat next single instruction by n-times, RC = #num
RPTB{D}	\$Padr	Repeat next instruction block, End_address = \$Padr Start_address = PC + 1 or (PC + 3) BRC be set by data transfer instruction

실현 명령 세트5(HMM전용)

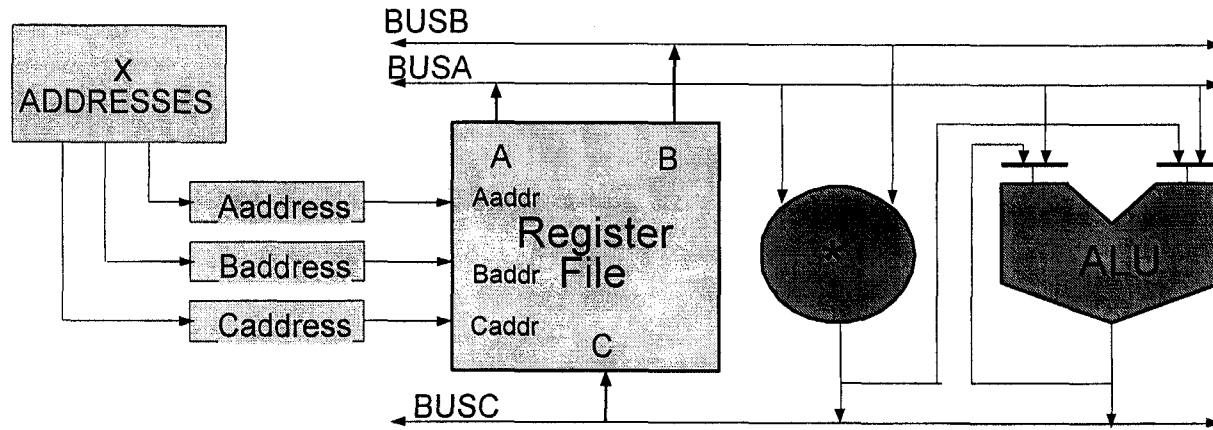
Syntax		Expression
Mnemonic	Operand	
HMM_INIT		Initialize HMM block
HMM_RD	hmm_src, dst	dst = hmm_src ; to read HMM register
HMM_WR	src, hmm_dst	hmm_dst = src ; to write HMM register
HMM_MDH	\$Smem, \$Hmem, #num	\$Hmem++ = Smem++, #num times
HMM_MHD	\$Hmem, \$Smem, #num	\$Smem++ = Hmem++, #num times

별첨 5. 음성인식용 ASSP의 기본 ISA

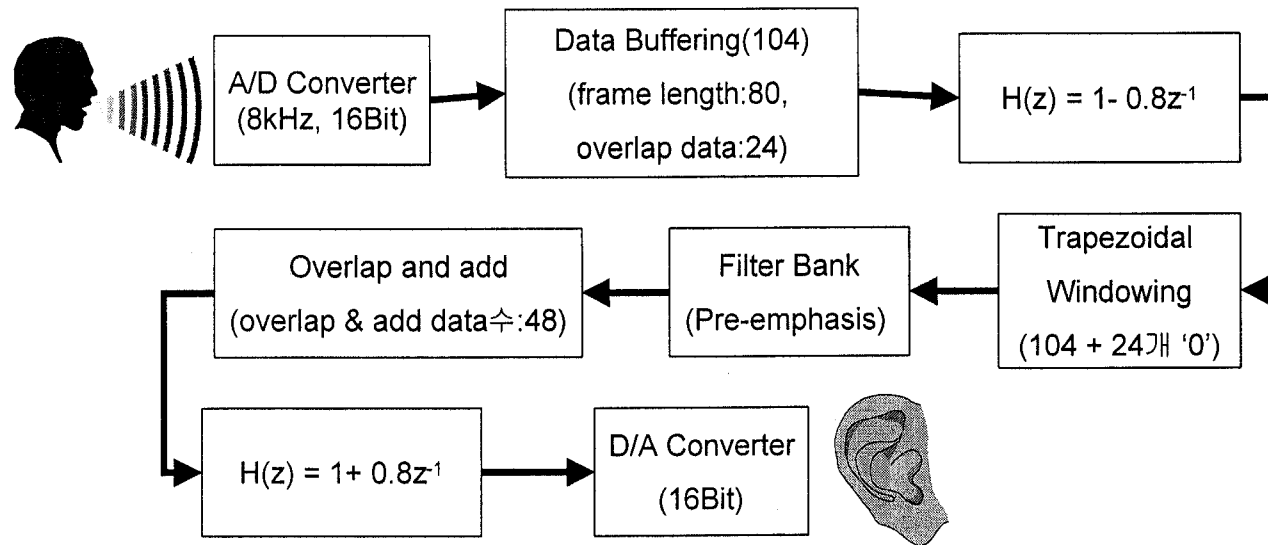
1. Block Diagram for Processor Architecture



2. H/W for Matrix operation



별첨 30. Preprocessing을 위한 ASSP의 아키텍처



별첨 31. Preprocessing(Pre-emphasis) 동작 Block Diagram

세계 장난감 시장 현황 및 전망

	1998	2000	2003
세계 시장 규모 (\$억)	678	770	950
세계 시장 성장률(%)	3.0	6.6	7.8
음성인식관련 시장규모(\$억)	-	6	95
음성인식 관련 시장 예측	2003년 전체 시장의 10% 예상		

Source : International Council of Toy Industries)

세계 TAD 시장 현황 및 전망

	1998	1999	2000	2001	2002
Units by Recoding Media (K)	27,931	29,983	31,674	33,139	34,335
Tape-based(K)	12,010	8,995	4,751	3,313	1,716
Tape-less(K)	15,921	20,988	26,923	29,825	32,618
Units by Feature Category(K)	27,931	29,983	31,674	33,139	34,335
Standalone(K)	13,966	14,092	14,253	14,250	14,078
Integrated with Corded Phone(K)	3,212	3,298	3,167	2,983	2,747
Integrated with Cordless Phone(K)	10,754	12,593	14,254	15,907	17,511
Factory ASP [*] by Recording Media : Digital(\$)	51.5	49.1	46.9	46.7	45.6
Factory Revenue by Recording Media : Digital (\$M)	819.5	1,031.5	1,262.4	1,392.6	1,487.0
Total Semiconductors (\$M)	408.4	441.9	488.0	538.4	582.0
Notes : Answering Machine 성장률 (5.5%) / Semiconductor 시장 성장률 (9.3%)					

*ASP : Average Selling Price

Source : DataQuest Jan.18, 1999

별첨 32. 장난감 및 TAD(Tapless Answering Device)에서 필요한 음성인식 칩 세계 시장 전망 예측

표.1 COMPARISON OF THE OVERALL POWER CONSUMPTION DURING OFF CHIP DRIVING

Pattern Names	Coffchip (pF)	Unencoded P(mW)	BI encoding		BITS encoding		HIHR encoding	
			P(mW)	Red.(%)	P(mW)	Red.(%)	P(mW)	Red.(%)
SpeechOutput	10	27.9	28.2	-1.1	18.1	35.1	17.6	36.9
	15	38.2	36.9	3.4	23.5	38.5	23.7	38.0
	20	48.5	45.6	6.0	28.9	40.4	29.8	38.6
	25	58.9	54.3	7.8	34.3	41.8	35.9	39.0
	30	69.1	63.0	8.8	39.7	42.5	42.0	39.2
ClassicMusic	10	29.7	28.1	5.4	23.2	21.9	22.5	24.2
	15	40.7	36.7	9.8	30.4	25.3	30.4	25.3
	20	51.7	45.3	12.4	37.7	27.1	38.4	25.7
	25	62.6	53.9	13.9	44.9	28.3	46.3	26.0
	30	73.6	62.5	15.1	52.2	29.1	54.3	26.2

표.2 COMPARISON OF AREA, DELAY, AND POWER OF ENCODERS AND DECODERS

	Encoder			Decoder		
	BI	BITS	HIHR	BI	BITS	HIHR
Area(μm^2)	19076	18626	4659	2662	11392	9968
Delay(ns)	3.29	3.87	0.38	0.15	0.38	0.38
Power(μW)	2309	2409	411	120	2102	1618

별첨 33. IRSIM이라는 버클리 Simulator를 사용하여 Power를 비교표(Time Delay는 HSPICE를 사용)

Technology

About SAIT

Research Area

Global Partnership

Job Opportunities

Global Purchasing

QUICK

Search input field with a 'GO' button

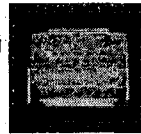
- Digital - Nano - Bio
- Opto - Energy

Low Power System Design Group

BBS | Tech Discussion | Utilities | Tech Report | Contract | Research Calendar | Manager

Introduction

Advances in semiconductor processing technology and computer-aided design (CAD) tools have made integration of many electrical components on a single chip possible. Combined with the market requirements for the smaller and lighter systems, even an entire system can be integrated on a chip (system on a chip). Heat generated by a high-performance and high-density chip not only shortens the operating hour but may also lowers performance. Low power design is imperative to reduce the power dissipation of high performance VLSI, especially in portable systems. SAIT's Mobile System Laboratory has formed the global partnership with leading research institutions, such as the University of Utah, Seoul National University and Sun Kyun Kwan University, to work on the practical solutions for low power portable system. Our research interests are system on a chip (SoC), low power design, and hardware-software co-design and co-simulation.



What's Cool

▶ Contratulation!

Hot utilities

별첨 34. 글로벌 사이버 연구센터의 형태