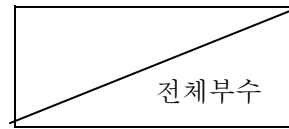


2009 년 12 월

보 고 서 번 호



# 초저전력 비동기 프로세서 개발 사업

Ultra Low Power Asynchronous Processor Development



# 제 출 문

본 연구보고서는 주요사업인 "초저전력 비동기 프로세서 개발에 관한 연구"의 결과로서, 본 과제에 참여한 아래의 연구팀이 작성한 것입니다.

2009 년 12 월

주관연구기관명 : 한국전자통신연구원

주관연구책임자 : 김 성 남

연 구 원 : 김 성 운

김 영 우

오 명 훈

이 재 성

신 치 훈

# 요 약 문

## I. 제 목

초저전력 비동기 프로세서 개발 사업  
(Asynchronous Low power Processor Development)

## II. 연구목적 및 중요성

### 1) 연구 목적

본 연구는 유비쿼터스 서비스(Ubiquitous service) 환경하에서 폭발적인 성장이 예상되는 휴대형, 이동형 정보 기기에서 핵심적인 역할을 수행하는 에너지 효율적인 초저전력 프로세서를 비동기 설계(Asynchronous design) 방식을 적용하여 개발하고 이를 통하여 관련 IPR 을 확보하는 것을 목적으로 한다.

#### ● 최종 목표

- 미래 휴대/이동 컴퓨팅을 위한 초저전력 비동기 프로세서 기술 확보
- 32 비트 초저전력 비동기 프로세서 마이크로아키텍처
- 150MHz 급 초저전력 비동기 프로세서 RTL

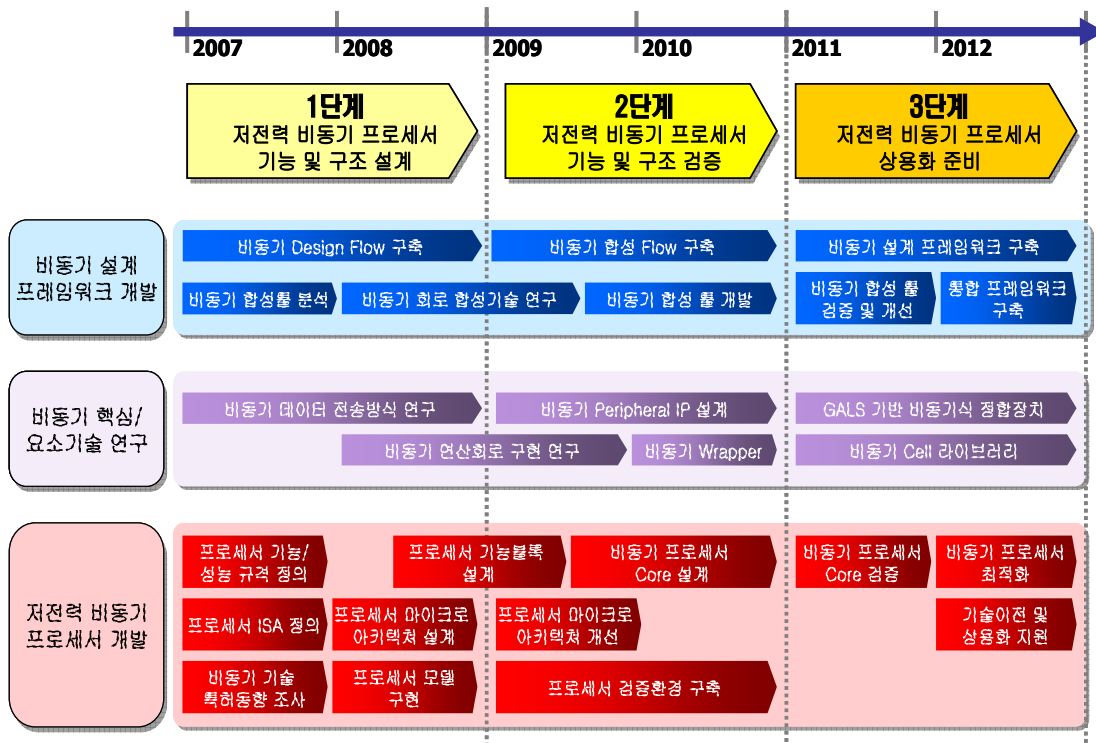
### 2) 연구 중요성

프로세서는 대부분의 정보 기기에서 필수적으로 사용되고 있으며 산업의 쌀로 불리울 만큼 그 중요성이 크지만, 현재 주요 선진국의 몇몇 기업에서 기술을 독점하고 있을 정도로 개발 난이도가 매우 높고 진입장벽이 높은 기술로서, 부가가치가 매우 높고 주변 산업에 큰 영향력을 행사하는 첨단 기술이다. 국내의 경우 현재 몇몇 중소기업 제외하고는 프로세서 개발에 적극적으로 나서고 있지

않으며, 대부분의 필요 기술을 대외 도입에 의존하고 있어서 국내 기술 및 제품의 대외 경쟁력이 취약한 상황이기 때문에 독자적인 기술개발 및 확보가 매우 시급하다. 따라서 본 사업에서는 저전력 프로세서에 대한 연구개발을 통하여 저전력 프로세서 기술을 독자적으로 확보하고 향후 이를 기반으로 다양한 응용분야에 적용하여 적극 활용될 수 있도록 본 연구개발을 추진하고자 한다.

### III. 연구내용 및 범위

본 연구개발은 전문연구사업의 일환으로 2007 년 1 월부터 총 6 년의 개발 일정으로 추진되고 있으며, 2007 년부터 2008 년까지 제 1 단계 연구개발을 수행하였음. 본 연구개발 사업의 각 단계별 로드맵은 다음에 나타낸 바와 같다.



<단계별 연구개발 로드맵>

당 부서에서는 상기 연구개발 로드맵과 같이 3 단계 연구개발 추진전략을 수립하고 1 단계 연구개발을 수행하였다. 본 연구개발 사업의 주요 단계별 주요 연구개발 내용은 다음에 나타낸 바와 같다.

● **1단계 (2007.1.1 - 2008.12.31): 기능 및 구조 설계 단계**

- 초저전력 비동기 설계기술 관련 특허동향 조사
- 초저전력 비동기 프로세서의 기능/구조/인터페이스 정의
- 초저전력 비동기 프로세서의 명령어셋(ISA) 정의
- 프로세서의 마이크로아키텍처 정의 및 설계
- ALU 등과 같은 비동기 프로세서 주요 기능 블록 설계
- 주요 기술 보유 유관 업체등과의 업무 협력을 통한 프로세서 개발환경 및 설계 환경 구축
- 프로세서 기능 및 성능 검증을 위한 행위모델 개발 및 검증환경 구축
- 주요 기관과의 위탁연구 수행을 통한 주요 요소기술 및 프로세서 성능향상 방안 연구

● **2단계 (2009.1.1 - 2010.12.31): 기능 및 구조 검증 단계**

- 전용 시뮬레이터 등을 통하여 비동기 프로세서의 기능/구조 검증
- 32비트 초저전력 비동기 프로세서 코어(RTL) 통합설계 및 검증
- 비동기 프로세서 주변장치 및 라이브러리 등의 개발
- 비동기 설계기술 및 프로세서 기술을 보유하고 있는 기관과의 교류를 통한 공동연구 개발
- 기 확보한 일부 요소기술의 업체 기술이전을 통한 상용화 지원
- 비동기 설계 및 합성기술, 인터페이스 기술 등의 요소기술 개발
- 비동기 설계기술의 적용을 통한 신규 응용분야 개발

● **3단계 (2011.1.1. - 2012.12.31): 상용화 지원 단계**

- GALS 기반의 비동기식 정합장치등과 같은 요소기술의 개발
- 저전력 비동기 프로세서 코어와 주변장치 통합 및 최적화 개발
- 저전력 비동기 프로세서 설계를 위한 비동기 설계 프레임워크 구축

- 국내 Fab. 벤더등과 공동개발을 통하여 주요 Cell 등 설계를 통한 검증
- 개발한 32 비트 초저전력 비동기 프로세서의 검증을 통한 IP 화
- 주요 수요 기업에 관련 기술의 기술이전 및 상용화 지원

#### IV. 연구결과

1 단계 연구개발 기간(2007-2008 년) 동안 본 연구를 통하여 얻은 주요 연구개발 결과에 대하여 1 차년도와 2 차년도로 구분하여 요약하면 다음과 같다.

##### 1) 1 차년도 (2007 년)

- 주요 작성 문서
  - 초저전력 비동기 프로세서 요구사항 정의서
  - 비동기 회로 설계 요소기술 분석서
  - 초저전력 비동기 프로세서 기능 및 성능 규격 정의서
  - 초저전력 비동기 프로세서 명령어셋 정의서
- 3P 현황
  - 특허 : 국내 2 건, 국제 1 건 출원
  - 논문 : 국제학술논문 2 건, 국내저널 1 건, 국내학술대회 1 건
  - 프로그램 : 4 건 등록
  - TM: 16 건, TDP: 3 건
- 기타 수행 업무
  - 비동기 기술 자료조사 및 특허 분석
  - 비동기 설계환경 구축 및 개발키트 분석
  - 위탁과제 : 비동기 회로 합성 및 시뮬레이션용 공개 소프트웨어 분석, 광주과기원

## 2) 2 차년도 (2008 년)

- 주요 작성 문서
  - 비동기 프로세서 명령어 microcode 상세분석서
  - 비동기 프로세서 마이크로아키텍처 설계서
  - 비동기 프로세서 블록설계서
  - 비동기 프로세서 기능 시험계획서
  - 비동기 프로세서 기능 시험절차 및 결과서
  - USN 을 위한 저전력 비동기 IP 응용기술 연구 보고서
- 3P 현황
  - 특허 : 국내 1 건 등록, 국내 출원 4 건, 국제 6 건 출원
  - 논문 : 국제학술논문 1 편 게재(SCIE), 국제학술회의 3 편 발표,  
국내저널 1 건 게재, 국내학술대회 1 건 발표
  - 프로그램 : 8 건 등록
  - TM: 16 건, TDP: 7 건
- 기타 수행 업무
  - 국제공동연구 : A Research on the Low Power Asynchronous IP technology for Ubiquitous Sensor Networks, Handshake Solutions
  - 용역과제 : 비동기 소프트웨어를 위한 개발환경 구축, AD 칩스
  - 위탁과제 : 비동기식 저전력 cache 설계에 관한 연구, 충북대학교

## V. 기대성과 및 건의

본 1 단계 연구개발의 기술혁신 차원의 성과에 대한 평가는 다음과 같다.

- 초저전력 비동기 프로세서 연구개발 사업은 전문연구사업으로서 2007 년 부터 총 6 년의 개발 일정으로 추진되고 있으며, 현재 1 단계 2 차년도 연구개발 일정을 완료하였음. 본 연구개발사업은 향후 유비쿼터스 서비스



환경하에서 큰 성장이 예상되는 다양한 정보기기나 저전력 응용분야 등에 적용할 수 있는 32 비트급의 초저전력 비동기 프로세서의 개발 및 관련 IPR(Intellectual Property Right)의 확보를 목표로 개발을 추진중임

- 제 1 단계 연구개발을 수행함에 있어 한정된 예산과 인력을 활용하여 최대의 성과를 도출하기 위하여 관련 전문기관 및 기업과의 협력을 추진하여 극복하고자 하였음. 이러한 노력의 일환으로서 국내의 프로세서 개발 전문 기업과 대학, 그리고 해외의 비동기 설계 전문기업과 협력을 추진하였으며, 이를 통하여 본 연구개발에 있어 취약한 부분을 보완하였음
- 1 단계 연구개발을 통하여 비동기 설계환경(CAD) 구축, 비동기 프로세서를 위한 소프트웨어 개발환경 구축, 비동기 프로세서 검증환경 구축, 비동기 프로세서 규격정의, 명령어셋 정의, 마이크로아키텍처 설계, 블록 설계 등 32 비트 비동기 프로세서 구현을 위한 기반 연구를 수행하였고, 이를 통하여 관련 논문, 특허 등과 같은 지적재산권을 다수 확보하는 등 가시적인 성과를 얻음
- 본 연구개발 단계에서는 32 비트 비동기 프로세서 코어 블록을 조기에 설계 완료하고 이에 대한 검증을 수행 함으로서 프로세서의 기능적인 동작을 확인하였음. 당 부서에서는 1 단계 연구개발을 통하여 구축한 비동기 설계환경 및 프로세서 개발환경을 초저전력 비동기 프로세서 설계에 적용 함으로서 선진국들에 비하여 매우 취약한 국내 비동기 설계 기반을 어느 정도 구축하는 성과를 거두었음

제 1 단계 연구개발 결과의 주요 활용 계획은 다음에 나타낸 바와 같다.

- 프로세서는 첨단 기술의 결정체로서 유관산업에 대한 파급효과가 매우 크며, 산업의 쌀으로 불리울 정도로 각종 정보기기 등에 필수적인 고부가가치의 기술임. 국내에서 이러한 기술에 대한 연구개발 투자가 상당히 미진

한 상황이며 관련 기술에 대한 저변이 취약하지만, 본 연구개발이 성공적으로 추진된다면 다양한 산업 분야에 활용이 가능함

- 당 부서에서는 본 연구개발 1 단계 과정을 통하여 확보한 32 비트급 초저전력 비동기 프로세서 코어 RTL 을 다음 단계의 연구개발 과정을 통하여 전력 및 성능 측면에서 최적화하고 프로세서 동작의 안정성을 확보하여 연구개발 성과의 경쟁력을 확보 할 계획임
- 향후 연구개발 단계에서는 저전력 프로세서 기술을 필요로 하는 수요 기업과 공동연구를 통하여 연구개발 시에 이들 잠재 고객의 요구사항을 적극 반영 함으로서 기술개발 결과의 완성도를 높이고 연구개발 완료 후에 상용화 추진 시 제품화 성공 가능성을 높일 수 있도록 할 계획임. 또한, 기술의 수요 기업과 함께 최적화된 저전력 프로세서 코어를 활용할 수 있는 신규 응용분야를 적극적으로 발굴하여 이에 적용할 수 있는 방안을 모색할 계획임
- 연구개발 결과 확보한 최적화된 초저전력 비동기 프로세서 코어를 IP 화 하여 수요 기업과 Fabless 반도체 설계 업체 등에 기술이전을 적극 추진하여 이를 다양한 저전력 응용 분야에 적용 함으로서 확산시킬 수 있도록 할 계획임
- 본 연구개발을 통하여 그간 거의 대부분을 외국의 프로세서 기술을 도입하여 사용하던 것을 일정부분 대체가 가능할 것으로 전망되며, 이로 인한 기술의 수입대체 효과가 기대됨. 본 연구개발 결과 및 확보한 IPR 을 활용하여 국내의 저전력 프로세서 기술기반을 확고히 하고 대외 경쟁력을 확보 함으로서 얻을 수 있는 기술적, 경제적 파급효과가 매우 클 것으로 기대됨

# ABSTRACT

## I. TITLE

Ultra Low Power Asynchronous Processor Development

## II. THE OBJECTIVES

The objective of this project is to develop a low powered 32 bit processor based on the asynchronous design technology. In this asynchronous design approach, all signalings between logic blocks are made by handshaking protocols instead of using a global clock signal. This processor can be applicable to various applications such as mobile handheld devices and ubiquitous sensor network devices.

## III. THE CONTENTS AND SCOPE OF THE STUDY

## IV. RESULTS

## V. EXPECTED RESULT & PROPOSITION

# CONTENTS

CHAPTER 1 Overview

CHAPTER 2 Status of Worldwide Technology Development

CHAPTER 3 Details of R&D of the 1<sup>st</sup> Stage

CHAPTER 4 R&D Results Summary

CHAPTER 5 Future Usage Plan for the Results

CHAPTER 6 Oversea Technology Information

CHAPTER 7 Reference

## 목 차

제 1 장 연구개발과제의 개요.....	13
제 2 장 국내외 기술개발 현황 .....	16
제 3 장 연구개발수행 내용 및 결과 .....	21
제 4 장 목표달성도 및 관련분야에의 기여도 .....	55
제 5 장 연구개발결과의 활용계획 .....	59
제 6 장 연구개발과정에서 수집한 해외과학기술정보 .....	61
제 7 장 참고문헌 .....	63

주의사항

# 제 1 장 연구개발과제의 개요

## 제 1 절 연구개발의 목적

유비쿼터스 서비스 패러다임으로 급격히 전환되고 있는 현 상황에서 각종 정보화 기기들의 수요는 크게 증가하고 있으며, 이들은 다음과 같은 방향으로 진화되고 있다.

- 지능형 / 맞춤형
- 정보기기의 융합화
- Personal / Mobility
- Energy Efficiency

본 연구에서는 미래 유비쿼터스 환경하에서 폭발적으로 수요가 증가할 것으로 전망되는 휴대/이동형 정보 기기에 필수적인 에너지 효율적인 저전력 프로세서 기술의 개발을 목표로 한다. 이를 위하여 본 연구개발에서는 다음과 같은 특징을 갖는 저전력 프로세서의 규격을 설정하였다.

- 최종 결과물 : 32 비트급 비동기 프로세서 (RTL)
- 저전력 설계 기술 : 비동기식 설계방식 적용 (25uW/MHz 급 전력소모)
- 고효율 마이크로아키텍처 : 32 비트의 범용성과 확장성을 고려한 ISA 내장 (150MHz 급 성능)

이러한 연구개발을 통하여 그린 IT 기반 저전력 프로세서 기술에 대한 수요에 부응하고 국내 반도체 산업의 경쟁력 확보에 기여함으로써 향후 큰 성장이 기대되는 미래 휴대/이동 기기에서 독자적인 기술 기반을 확보하고자 한다.

## 제 2 절 연구개발의 필요성

본 사업은 향후 큰 수요가 예상되고 있는 저전력 프로세서 기술의 확보를 위하여 중장기적인 관점에서 추진되고 있는 전문연구사업으로서, 본 연구개발의 필요성에 대하여 요약하면 다음과 같다.

- 초저전력 프로세서 기술 및 비동기 설계기반 저전력 설계 기술의 확보를 통한 Green IT 기반 기술 확보 및 유비쿼터스 패러다임에서 큰 수요가 예상되는 u-서비스 기반 모바일 기기를 위한 에너지 효율적인 프로세서 기술의 자체 확보
- 초저전력 프로세서 기술의 높은 부가가치와 기술적, 경제적 파급효과로 인한 유관 산업 분야의 동반 성장에 기여하고, 연구개발 성과의 국내 기업 기술이전을 통하여 제품의 차별화 및 대외 경쟁력 향상
- 시장에서 수요가 매우 큰 저전력 프로세서 기술 분야에 대한 전략적인 연구개발을 통하여 날로 치열해지는 글로벌 경쟁환경에서 경쟁력 있는 기술로 시장과 트렌드를 주도할 필요성 증대
- 큰 성장세를 보이고 있는 휴대형/이동형 정보기기에 필요한 핵심 기술의 개발을 통한 경쟁력 확보를 통하여 국내 비메모리 분야의 산업 육성에 일조하고, 이를 통한 관련기술에 대한 수입대체 및 수출증대 효과 기대
- 비동기 설계 기반 프로세서 IP 기술을 활용한 응용기술 개발 및 이를 통한 신규 수요처 발굴 및 신시장 개척

## 제 3 절 연구개발의 내용 및 범위

본 연구개발 사업에서 추구하고 있는 저전력 프로세서 기술 개발에 대한 각 단

계별 연구개발의 내용 및 범위는 다음과 같다.

● **1 단계 (2007.1.1 - 2008.12.31) : 설계 단계**

- 초저전력 비동기 설계기술 관련 특허동향 조사
- 초저전력 비동기 프로세서의 기능/구조/인터페이스 정의
- 프로세서의 마이크로아키텍처 정의
- ALU 등과 같은 주요 기능 블록 설계
- 주요 대학 및 유관 업체등과의 업무 협력 및 위탁연구 수행을 통한 연구 개발 효율성 제고

● **2 단계 (2009.1.1 - 2010.12.31) : 검증 단계**

- 전용 시뮬레이터 등을 통하여 비동기 프로세서의 기능/구조 검증
- 저전력 비동기 프로세서 코어(RTL) 및 핵심 주변장치 모듈 및 라이브러리 등의 개발
- 비동기 설계 관련 주요 기술을 확보하고 있는 기관 내 부서 및 학제와의 교류를 통한 요소 기술확보
- 확보한 일부 요소기술의 업체 기술이전을 통한 상용화 지원

● **3 단계 (2011.1.1. - 2012.12.31) : 상용화 지원 단계**

- GALS 기반의 비동기식 정합장치의 개발
- 저전력 비동기 프로세서 코어의 최적화 개발
- 저전력 비동기 프로세서 설계를 위한 비동기 설계 프레임워크 구축
- 국내 Fab. 벤더등과 공동개발을 통하여 주요 Cell 등 설계를 통한 검증
- 주요 수요 기업에 관련 기술의 기술이전 및 상용화 지원



## 제 2 장 국내외 기술개발 현황

### 제 1 절 연구동향

#### 1. 국내 연구 동향

과거 90년대 후반부터 지금까지 학계를 중심으로 꾸준히 비동기 설계에 관련된 연구가 수행되고 있다. 그러나, 상대적으로 국외와 같이 활발한 수준은 아니며, 관련 업체 또한 거의 전무한 실정이다.

#### 2. 국외 연구 동향

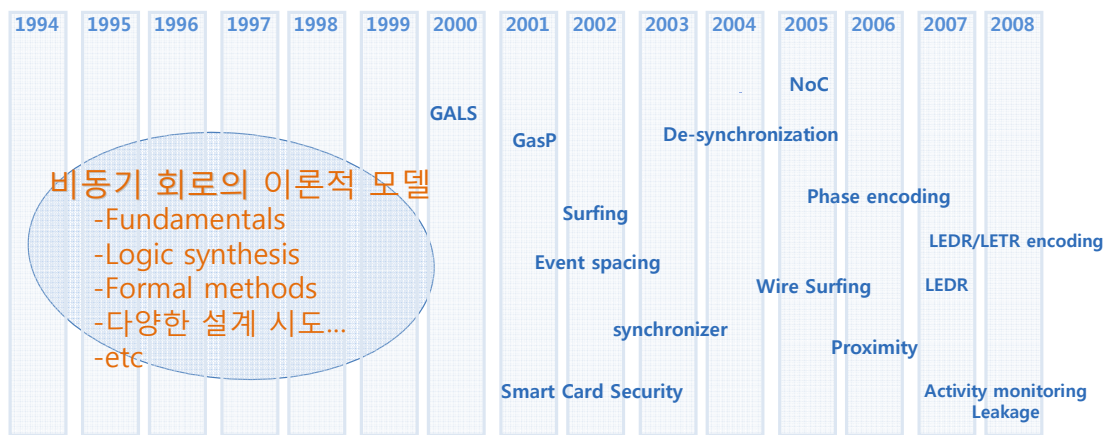


그림 1. 비동기식 설계 기술의 국외 연구동향

미국, 영국의 학계를 중심으로 90년대에 로직 합성, formal 방식, delay 모델 등 비동기 회로의 이론적 모델에 대한 연구가 수행되었고, 다양한 비동기식 설계 방식이 시도 되었다. 2000년대에 접어들면서, security, global wiring, wave pipeline, 저전력, multi-clocked domain에서의 동기화, 비동기식 핸드셰이크 프로토콜의 overhead 감소와 같은 연구 분야가 활발하게 진행되고 있다. 그림 1은 관련 기술의 국외 연구 동향을 나타내고 있다.

## 제 2 절 특허동향

본 절에서는 비동기식 설계 및 구현과 관련한 국내 및 국제 출원/등록 특허의 현황과 동향[2]에 대하여 정리하도록 한다.

### 1. 국내 특허 동향

국내에서 비동기식 회로 및 시스템의 연구는 1990 년대 초반부터 활발해 지기 시작하여 그 연구 결과가 1995 년 이후 특허의 출원과 등록으로 이어진 것으로 판단된다. 2000 년대 후반에는 다소 주춤한 경향이 있으나, 비동기식 연결망에 대한 연구가 국내외 적으로 활발히 이루어지고 있는 점을 감안할 때 앞으로도 관련 특허가 계속 출원 될 것으로 판단된다(그림 2 (a)).

국내 출원 특허의 관련 기술 대분야(하드웨어/소프트웨어)의 비중을 볼 때, 하드웨어 분야에 대한 출원이 약 83%로 높은 비중을 차지하고 있으며, 하드웨어 관련 특허를 세분화 하여 보았을 때, 비동기식 하드웨어 설계의 핵심이 되는 비동기식 파이프라인과 FIFO 에 대한 특허가 64%로서 대다수를 차지 하고 있음을 알 수 있다(그림 2 (b)). 대표 출원인을 분석하여 보았을 때 ETRI, 광주 기술원, 현대전자가 상위 1, 2 위 의 다수 출원인으로 분석 되었으며, 외국 유수 업체(필립스, ARM, 세이코 엡슨, 썬 등)의 특허가 50%이상을 차지하고 있음을 알 수 있다.

### 2. 국외 특허 동향

그림 2 (c)~(e)는 국가별, 연도별 관련 특허의 출원 및 동향을 나타낸 것이다(총 337 건). 국제적으로는 1990 년대 초반부터 특허의 출원 및 등록 건수가 꾸준히 상승 하였으며, 국가별로 출원 건수를 분석하여 보았을 때, 미국이 202 건으로 1 위이며 다음으로 일본, 한국, 유럽의 순으로 나타났다. 특허의 양적인 측면에서 볼 경우 일본 혹은 유럽과 대소동이 한 건수를 보이고 있으나, 질적인 측면을 보았을 때 확연

한 차이점이 있음을 알 수 있다. 그림 2 (f) 는 비동기식 설계 기술과 관련하여 특허를 출원한 국가들의 기술 수준을 분석한 기술수준 분석표 이다. 기술 수준의 분석에 사용한 지표로는 다음과 같은 지표를 사용하였다.

- 특허 피 인용수(CPP, Cites Per Patent)

특정연도/기간에 등록된 특허들이 이후 특허들에 의해 평균적으로 인용된 회수.

$$CPP_i = \frac{\sum_{i=1}^{n_i} C_i}{n_i}, n_i \text{는 } t \text{ 년도에 등록된 특허건수, } C_i \text{ 는 } i \text{ 특허의 피인용수}$$

- 특허 영향력 지수(PII, Patent Impact Index)

전체 CPP 에 대하여 정규화된 국가별 CPP 값. PII 값이 1 일 경우 평균 기술 수준을 1 보다 클 경우 질적 수준 우위를 나타냄

- 기술력 지수(TS, Technology Strength)

특정국가(혹은 기관)의 기술적 역량을 나타내는 지표

$$TS^* = PII \times x, x \text{ 는 분석구간 특허건수. * 원 지표는 } CII \times \text{ 건수}$$

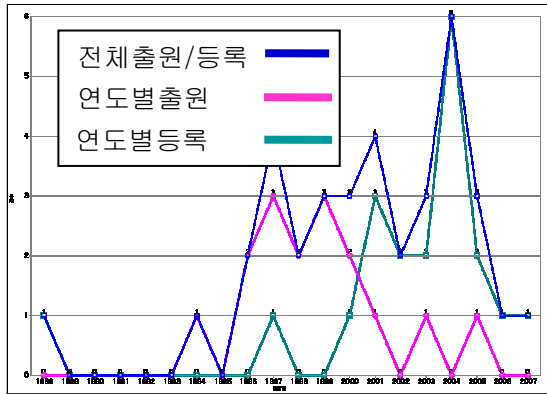
- 특허 패밀리 규모(PFS, Patent Family Size)

해당 특허의 지역적 보호범위, 기술적 중요성, 가치 정보를 반영하며, A, B, C, D 의 네 개 영역으로 구분된다. A 영역은 특허의 질적 수준과 시장의 규모가 큰 영역, B 는 질적 수준은 낮으나 시장 규모가 큰 영역, C 는 질적 수준은 높으나 시장이 작은 영역, D 는 질적 수준 및 시장 규모가 작은 영역을 나타낸다.

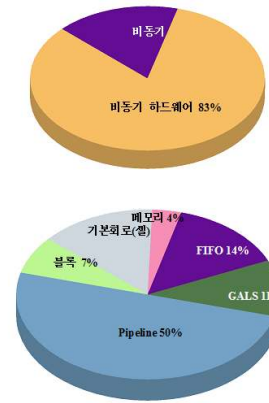
$$PFS(1) = \frac{\sum_{i=1}^N F_i}{N}, M \text{ 분석국가의 수, } N \text{ 해당 국가 특허수, } F_i \text{ 각 특허 패밀리수}$$

$$PFS(2) = \frac{\sum_{i=1}^N F_i}{(\sum_{j=1}^M \sum_{i=1}^N F_{ij}) / M}$$

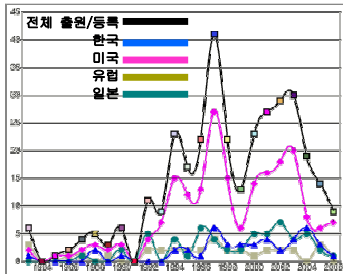
위와 같은 특허 기술수준 분석 지표를 통한 분석 결과, 기술력지수 측면에서는 미국이, 패밀리 규모에서는 미국, 유럽(영국, 네덜란드, 독일 등)이 높은 지수를 나타내어 시장 확보 및 기술 측면에서 우위를 점하고 있는 반면, 한국의 비동기 설계 기술 관련 기술 수준은 관련국 중 최하위 수준을 나타내고 있다.



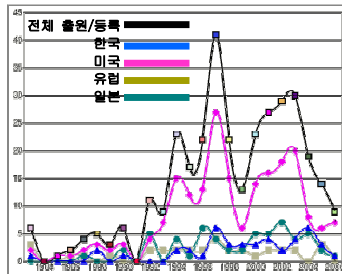
(a) 국내 출원 연도별 동향



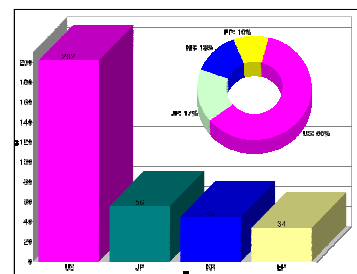
(b) 국내 출원 분야별 동향



(c) 국제 출원현황

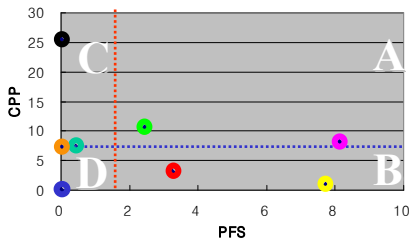


(d) 국제 등록현황

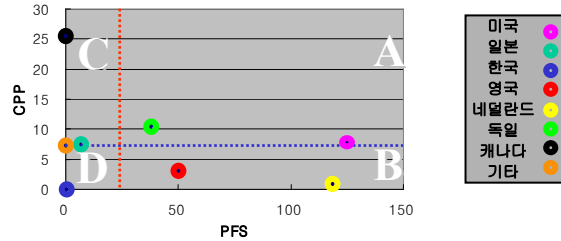


(e) 국가별 출원 건수

국가별 시장력과 피인용비(1)



국가별 시장력과 피인용비(2)



국가	등록 특허수	점유율	CPP (특허 당 피인용 수)	PII (특허영향지수)	TS (기술력지수)	PFS(1) (시장확보지수)	PFS(2) (시장확보지수)
미국	145	71.78%	8.03	1.13	163.51	8.13	124.73
일본	15	7.43%	7.27	1.02	15.31	0.42	6.46
한국	14	6.93%	0.07	0.01	0.14	0.00	0.00
영국	10	4.95%	3.10	0.44	4.35	3.25	49.89
네덜란드	8	3.96%	0.88	0.12	0.98	7.72	118.49
독일	5	2.48%	10.60	1.49	7.45	2.44	37.42
캐나다	2	0.99%	25.50	3.58	7.16	0.00	0.00
기타	3	1.49%	7.33	1.03	3.09	0.00	0.00

(f) 국가별 기술수준 분석

그림 2. 국내외 특허동향

## 제 3 절 산업동향

### 1. 국내 산업 동향

지금까지 비동기식 프로세서에 관련된 국내의 산업은 전무한 실정이다. 다만, 몇몇 IP 벤더를 중심으로 외국 업체의 비동기식 IP 와 툴 [14], 고성능 비동기식 FPGA [15]을 공급하고 있다.

### 2. 국외 산업 동향

비동기식 EDA 툴 분야에서는 2004 년 Handshake Solutions 사 (네델란드)에서 상위 수준 비동기식 설계 flow 인 TiDE 툴 [17] 출시했고, 2007 년 Silistix 사 (영국)에서 비동기식 NoC 를 적용한 CHAINworks [18], 그리고, 2008 년 Elastix 사 (미국)에서 저 전력 설계 solution 인 ELASTIX 툴 [19] 출시하였다.

비동기식 회로를 응용한 제품으로는 2005 년 Handshake Solutions 사 (네델란드)에서 최초의 상용 32 비트 비동기 프로세서 (ARM996HS)를 출시했고, 비슷한 시기에 출시된 비동기식 8 비트 프로세서(HT80C51)는 전자여권에 장착되어 세계적으로 80%를 점유하고 있다[17]. 이외에도 2005 년 Fulcrum 사 (미국)에서 비동기식 설계를 이용한 10G 용 Ethernet 스위치 출시했고 [20], 2008 년 Achronix 사 (미국)에서 최초의 비동기식 상용 FPGA(GHz 급) 출시[21]하여 시장을 늘려가고 있다. 또한, Intel, IBM, Sun Microsystems 등 메이저 급 회사에서 state-of-the-art 제품에 비동기 설계 기술을 적용하고 있다.

## 제 3 장 연구개발수행 내용 및 결과

### 제 1 절 연구개발 목표

본 저전력 비동기 프로세서 개발 사업의 1 단계 연구개발 목표는 저전력 비동기 프로세서 기능 및 구조설계이다. 이를 위하여 다음과 같은 세부 목표를 설정하여 추진하였다.

- 비동기 설계 기술관련 동향 분석(연구동향/특허분석)
- 비동기 프로세서의 기능/구조/인터페이스 정의
- 비동기 프로세서의 마이크로아키텍처 정의
- 비동기 프로세서 기능블록 설계

본 전문연구 사업에서는 1 단계 1 차 년도에 비동기설계 기술과 관련한 국내외 연구 동향에 대한 분석과 함께, 비동기 프로세서의 기능 및 구조를 정의하였다. 1 단계 2 차년도에는 1 차년도의 연구 결과에 기반하여 비동기 프로세서의 마이크로아키텍처를 정의하고, 비동기 프로세서 코어의 기능 블록 설계와 기능 검증을 위한 검증 환경을 개발 하였다. 본 과제의 효과적인 수행을 위하여 1 단계에서 국내의 기업/대학과의 공동 연구 및 용역을 추진하였으며, 비동기 설계 기술을 보유하고 있는 국외 업체와의 공동연구 추진을 통하여 연구결과물의 완성도를 높이고자 노력하였다.

### 제 2 절 기능 및 성능 규격 정의

본 절에서는 연구 개발 목표인 저전력 비동기 프로세서의 기능 및 성능에 대한 규격을 정의하고[3], 개발한 저전력 비동기 프로세서의 기능 검증을 위한 개발 환경에 대하여 설명한다.

## 1. 기능 규격 정의

저전력 비동기 프로세서는 최종적으로 다음과 같은 기능을 지원하며, 상세 기능 규격은 표 1 과 같이 정의 하였다. 이러한 기능 규격 중 내장 cache 및 메모리 관리 기, 내장 TCM, 버스인터페이스의 개발은 1 단계 연구 개발의 범위에 속하지 않는다.

- 비동기식 프로토콜
  - Bounded delay 모델 기반 4-phase bundled data 방식
- 프로세서 코어
  - 핸드셰이킹 방식의 비동기식 프로토콜
  - 32 비트 프로세서 코어
  - 내장 cache 및 메모리 관리기
  - 내장 TCM(Tightly Coupled Memory)
- 버스 인터페이스
  - Point-to-point 방식의 동기/비동기식 혼합 방식의 내부 버스
  - 외부와의 AMBA(Advance Microcontroller Bus Architecture) Bus 호환 기능

표 1. 저전력 비동기 프로세서의 기능 규격

기능 항목		기능 규격
비동기식 프로토콜	Signaling 방식	4-phase bundled data 방식
	Delay 모델	Bounded delay 모델
프로세서 코어	프로세서 기본 구조	Harvard 구조
	어드레스 및 데이터 인터페이스	32 비트
	명령어 Cache	최대 16Kbyte
	데이터 Cache	최대 16Kbyte
	메모리 제어기	최대 8 개의 메모리 Bank
내장 TCM	Tightly coupled memory 지원	
버스 인터페이스	내부 버스	Point-to-point 방식의 heterogeneous 타이밍 지원
	외부 버스	AMBA 2.0 및 3.0 호환

## 2. 성능 규격 정의

저전력 비동기 프로세서의 최종 성능 목표는 다음과 같다.

- 프로세서 코어
  - 동작 속도 150MHz 이상
  - 소모 전력 25uW/MHz 이하
- 외부 버스 인터페이스
  - 동작 속도 100MHz 이상

저전력 비동기 프로세서의 주요 성능 규격을 다음의 표 2에 요약하였다. 상기 성능 규격은 1 단계 연구 개발의 범위에 속하지 않으며, 성능 규격에 따른 저전력 비동기 프로세서의 개발은 다음 단계에서 구현 예정이다.

표 2. 저전력 비동기 프로세서의 성능 규격

성능 항목		성능 규격
프로세서 코어	프로세서 개수	1 개
	동작 속도	150 MHz 이상 (평균 동작속도)
	프로세서 ISA 구조	EISC AE32000
	프로세서 파이프라인	5 단 비동기 파이프라인
	메모리 사이즈	최대 32GBytes 외장 메모리
	부가 기능	Exception 처리
외부 버스 인터페이스	버스 구조 및 성능	AMBA 2.0/3.0. 호환 버스 100MHz 이상

## 3. 개발 환경 기능 정의

저전력 비동기 프로세서의 기능 검증을 위하여 개발 및 검증 환경은 다음과 같은 기능을 가져야 한다.

- 프로그램 개발환경
  - GCC 기반의 C 언어 컴파일러 환경
- 저전력 비동기 프로세서 검증 환경
  - 명령어 checker 및 프로파일러 환경



### 제 3 절 저전력 비동기 프로세서 구조

본 절에서는 저전력 비동기 프로세서의 구조(architecture)에 대하여 설명한다. 1 단계 연구개발 목표인 저전력 비동기 프로세서 코어의 구조 및 설계에 앞서서 본 연구과제에서 이루고자 하는 저전력 비동기 프로세서의 구조에 대하여 설명한 후, 프로세서 코어의 구조[6]에 대하여 상세 설명한다.

#### 1. 저전력 비동기 프로세서 구조

본 연구개발을 통하여 구현하고자 하는 저전력 비동기 프로세서 (ALTHEA, Asynchronous Low power processor based on THE EISC Architecture)는 크게 프로세서 코어 유닛(ALTHEA core unit), 버스 인터페이스 유닛(BIU), 캐쉬/메모리 관리 유닛(Memory Management Unit: MMU)와 기타 기능 블록으로 구성되는 코프로세서 유닛을 기본으로 하며, 선택사항으로 임베디드 응용분야를 위한 고속 SRAM 으로 이루어진 근접 메모리(Tightly Coupled Memory: TCM)을 포함한다. ALTHEA 는 프로세서 파이프라인의 성능을 향상시키기 위하여 하바드 아키텍처를 사용하고 있다. ALTHEA 의 최상위 수준의 구조는 다음의 그림 3 과 같다.

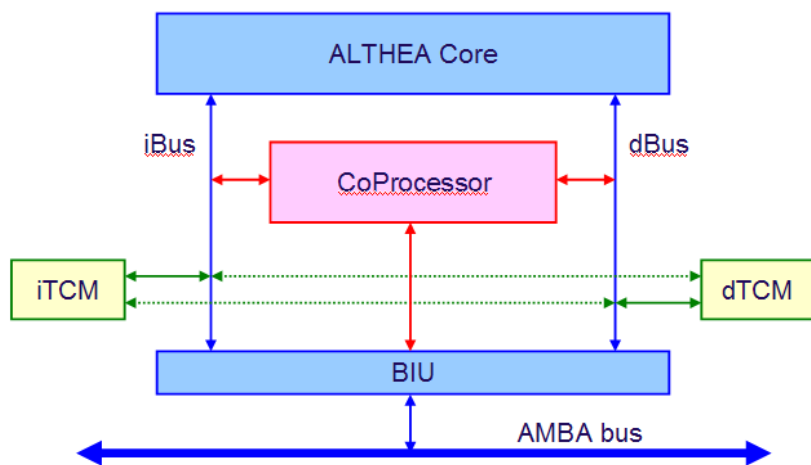


그림 3. ALTHEA의 최상위 수준 구조도

## 2. 저전력 비동기 프로세서 명령어 구조

ALTHEA 프로세서의 명령어 셋 구조(ISA, Instruction Set Architecture)로서는 EISC(Extendable Instruction Set Computer) 구조를 가지는 프로세서인 AE32000C 에서 사용하는 명령어 셋 구조를 채용하였다. 명령어는 6 개의 그룹으로 구분되며, 총 100 개의 명령어를 사용한다. 명령어의 효과적인 구현을 위하여 본 연구과제에서는 비동기 프로세서에서 사용할 명령어 100 개를 정의하고, 개별 명령어에 대하여 비동기식 동작을 마이크로코드 형태로 분석 하였다[4][5][9][10]. 다음의 표 3 은 ALTHEA 프로세서 코어에서 지원하는 명령어 셋을 정리한 것이다.

표 3. ALTHEA 프로세서 코어 명령어 셋

Class	type	Instruction
Load ER with immediate (1)		LERI
Memory Access (26)	Load	LD(2)/LDB(2)/LDBU(2)/LDS(2)/LDSU(2)/LDAU(2)/LEA(4)
	Store	ST(2)/STB(2)/STS(2)/STAU(2)
	multi	POP/PUSH
Data Move (9)	move	MTMH/MTML/MFMH/MFML/LDI/MTCR0/MTCR1/MFCR0/MFCR1
Branch (19)	Cond	JNV/JV/JP/JM/JNZ/JZ/JNC/JC/JGT/JLT/JGE/JLE/JHI/JLS
	Uncond	JMP/JAL/JR/JALR/JPLR
Arith & Logical (32)	ALU	ADD/ADC/SUB/SBC/AND/OR/XOR/CMP/TST/ADDQ/CMPQ/NEG/SET/CLR
	Shift	ASR(2)/LSR(2)/ASL(2)/SSL(2)
	Unary	EXTB/EXTS/CVB/CVS/CNT0/CNT1/NOT
	Multiply	MUL/MULU/MAC
Misc (13)	Cop	CPCMDn/GETCn/MVTCn/MVFCn/LDCn/STCn/EXECn
	Control	NOP/SYNC/HALT
	interrupt	SWI/BRKPT/STEP
<ul style="list-style-type: none"> <li>● Total 100 instructions (in binary)</li> <li>● Supports only AE32000C version</li> <li>● Blue instructions are not yet modeled</li> <li>● Instructions are described in order</li> </ul>		

### 3. 저전력 비동기 프로세서 코어 구조

ALTHEA 저전력 비동기 프로세서 코어는 5 단 비동기 파이프라인 구조를 가진다. 각 단계의 기능은 다음 표 4 와 같이 정의된다.

표 4. 파이프라인 단계 구분 및 기능

Stage		기능
<b>IF (Instruction Fetch)</b>		Instruction prefetch LERI instruction folding Asynchronous instruction queue
<b>DE (Decode)</b>		Instruction decoding Interrupt handling Hazard detection/processing Branch prediction Register file (RF) access Co-processor access
<b>EX (Execution)</b>		Multiple execution units (ALU, SHIFTER, Multiply, BYPASS) In-order issue multiple instruction issue In-order completion Register forwarding
<b>ME (Memory)</b>	<b>MEM</b>	Data align Register forwarding
	<b>WB</b>	Register file access

프로세서 코어는 기본적으로 5 단 비동기 파이프라인 이나, WB(Write Back) 단계의 주된 동작이 레지스터에 대한 쓰기 동작이므로, 표 4 에서 보듯이 MEM 단계와 WB(Write Back) 단계가 통합된 구조로 되어 있다.

프로세서 코어의 비동기 파이프라인 외에 프로세서 코어를 위한 블록으로서 레지스터 파일, 시스템 코프로세서(CP, Co-Processor) 블록이 ALTHEA 프로세서 코어의 내부에 포함된다. 레지스터 파일은 16 개의 32 비트 범용 레지스터와 8 개의 32 비트 특수 레지스터를 포함한다. 시스템 코프로세서는 ALTHEA 프로세서 코어와 추후 개발 예정인 캐쉬 제어기 및 메모리 제어기의 동작을 제어하기 위한 32 개의 32 비트 레지스터를 내장한다. 그림 4 는 ALTHEA 저전력 비동기 프로세서 코어를 구성하는 주요 블록들 간의 관계를 나타낸 구조도 이다.

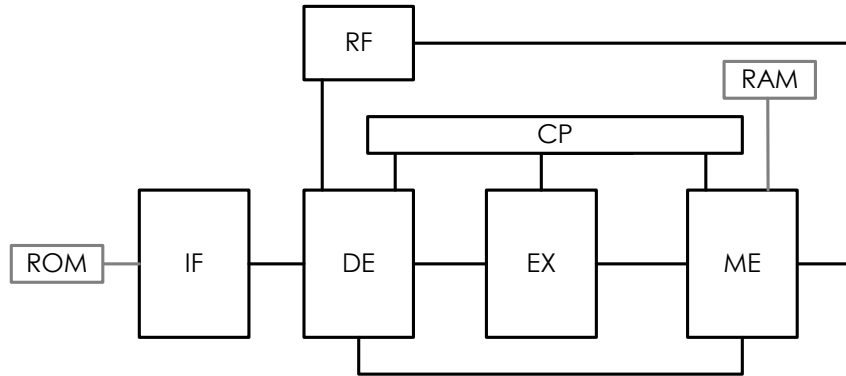


그림 4. ALTHEA 프로세서 코어의 파이프라인 구조도

## 제 4 절 저전력 비동기 프로세서 구조 설계 및 구현

본 절에서는 ALTHEA 저전력 비동기 프로세서의 설계와 그 구현 결과에 대하여 설명한다.

### 1. 프로세서 코어 구조 설계 및 구현

#### 가. 인터페이스 설계

ALTHEA 프로세서 코어는 비동기 식으로 설계됨으로 인하여 블록 및 외부 인터페이스에 전통적인 wire 인터페이스와 비동기식의 채널 인터페이스, 그리고 외부의 ROM 과 RAM 을 연결하기 위한 ROM/RAM 인터페이스로 구성된다. 다음의 그림 5 에 ALTHEA 프로세서의 블록 I/O 및 외부 인터페이스를 나타내었다.

- Wire 인터페이스
  - 일반적으로 회로를 기술하는데 사용하는 신호형식. 핸드쉐이킹을 사용하지 않으며, HDL 언어에서 사용하는 signal 혹은 wire 형식이다.
- Channel 형식

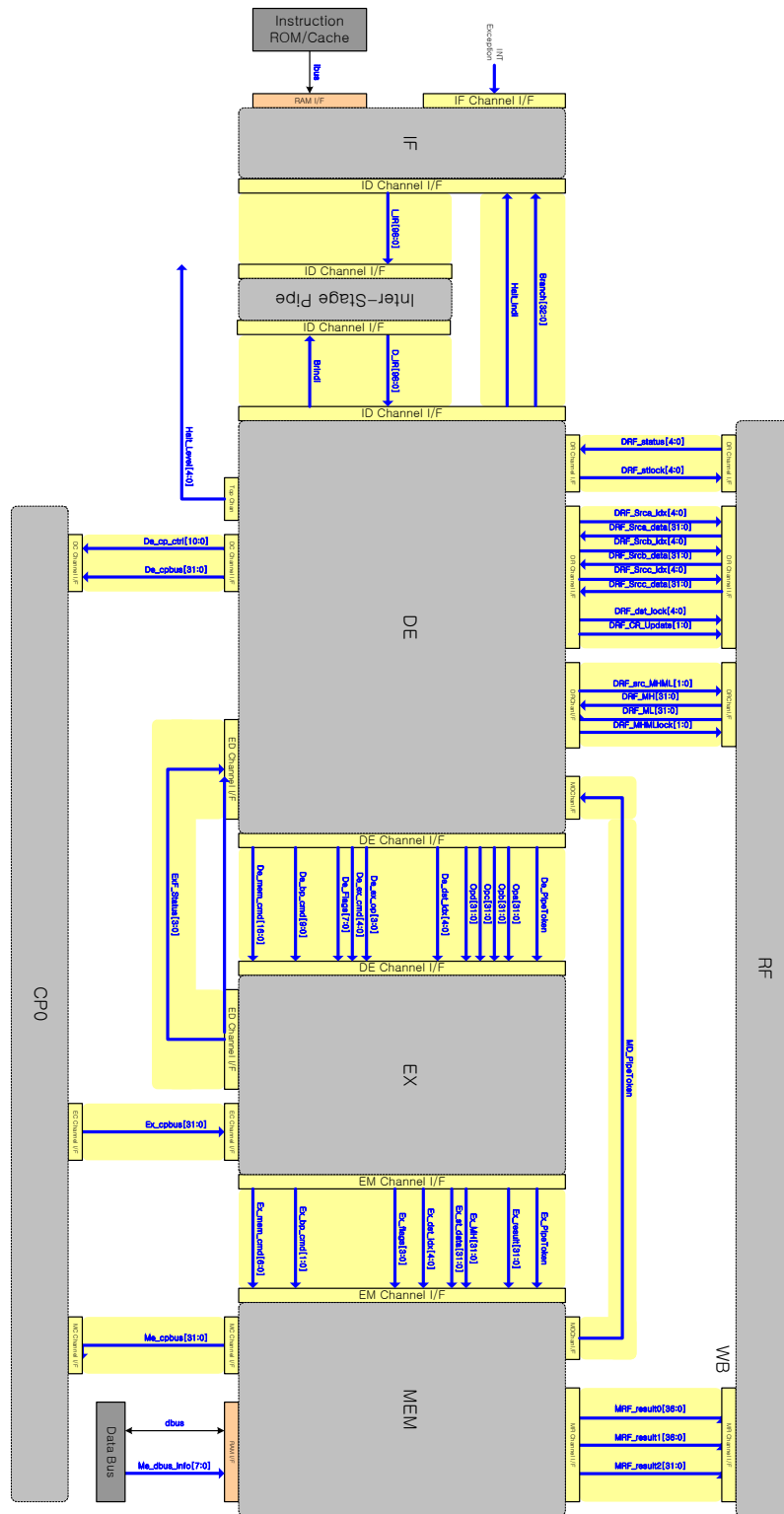


그림 5. ALTHEA 블록 I/O 및 외부 인터페이스

- 채널 형식은 비동기식 인터페이스를 구현하기 위한 인터페이스 형식으로, 실제 데이터의 값을 표시하는 wire 신호와 핸드셰이킹을 구현하기 위한 요청신호(\*\_R), 확인 신호(\*\_A)의 3가지 종류의 wire 신호의 묶음으로 구성되는 인터페이스 형식이다.

- ROM/RAM 형식

- ROM/RAM 인터페이스 형식은 메모리(ROM/RAM 형식의 메모리)와의 비동기식 인터페이스를 구현하기 위한 인터페이스 형식이다.

#### 나. 명령어 인출기

Instruction Fetch(IF) 블록은 명령어들을 메모리에서 읽어와 다음 파이프라인 단계인 Decoder(DE) 블록에서 명령어에 해당하는 동작을 처리할 수 있도록 1 차적으로 가공하는 역할을 한다. 명령어 메모리(ROM)를 통해 가져온 명령어는 종류에 따라 분류 및 가공 되고, 즉치값의 유무에 따라 몇 개의 명령어들이 묶어져 다음 단계로 전달되게 된다.

IF 단계는 프로세서 내부의 다른 단계들과 독립적(Decoupled)으로 동작한다. AE32000 은 이종으로 구성된 명령어들을 지연 없이 효율적으로 처리하기 위하여, 명령어 패치와 재가공을 담당하는 전위(Front-end) 파이프라인과 명령어의 본격적인 처리를 위한 후위(Back-end) 파이프라인으로 나뉘는데, IF 단계는 이 중 전위 파이프라인에 해당한다. IF 블록은 그림 6 과 같이 크게 다음과 같은 5 개 기능 블록으로 이루어져 있다. 하는 IF\_LERI 모듈, 임시 명령어 패킷을 저장하기 위한 패치 큐 (Fetch\_Q), 저장된 임시 명령어를 불러와 선 해독 하여 최종 패킷을 생성하는 선 해독기(PreD), 최종 패킷이 다음 파이프라인인 Decoder 블록으로부터 인출되기 전까지 보관 할 명령어 큐(Inst\_Q), 마지막으로 Decoder 파이프라인으로부터 분기 명령 등을 컨트롤 해주는 De\_recv 블록이 있다.

(1) Instruction Fetch & LERI Folding (IF\_LERI): IF\_LERI 모듈은 명령어 처리 순서에

따라 명령어 인출과 LERI 폴딩을 담당한다. 외부 메모리(ROM)으로부터 읽어오는 데이터는 32 비트 폭을 가지며 두 개의 명령어를 포함하고 있다

따라서 그림 6 의 IF\_LERI 블록의 내부에서 보는 바와 같이, 명령어 쌍의 첫 번째 명령어를 처리하는 블록과 두 번째 명령어를 처리하는 블록이 따로 필요 (First Path, Secd Path)하고 이들은 의존성을 가지기 때문에 순차적으로 연결하는 것이 편리하다.

ROM 으로부터 읽어 오는 명령어는 크게 일반 명령어와 LERI 명령어로 구분할 수 있다. 일반 명령어는 간단히 명령어를 그대로 패치 큐에 저장하면 된다. 다만, 문맥 상 선행해서 들어온 LERI 명령어를 포함하게 되므로, 만약 먼저 들어온 LERI 명령어가 존재한다면, LERI 명령어를 저장해둔 ER 레지스터 값을 함께 큐에 저장하여야 한다.

LERI 명령어는 해석되어 ER 에 저장한다. LERI 명령어에는 헤더 2 비트를 제외한 14 비트의 데이터가 있고 두 개의 LERI 명령어가 연속으로 들어올 수 있다. 첫 번째 LERI 명령어가 들어오면 ER 에 그냥 저장되지만 연이어 두 번째 LERI 명령어가 들어올 경우 ER 을 14 비트 밀어낸 후 두 번째 명령어 14 비트를 저장한다.

그림 7 (a), (b)는 LERI 디코더의 구조 및 동작을 나타낸 그림이다. 그림 7 (a)는 첫 번째 LERI 명령어가 들어왔을 경우를 나타낸 그림인데, 첫 번째 LERI 의 즉치값은 ER 의 하위 14 비트에 저장되고 나머지 18 비트는 LERI 의 즉치값에 대한 부호 확장된 값이 채워지게 된다.

첫 번째 LERI 가 저장되고 부호 확장이 완료되면 EF 를 설정하여 두 번째 명령어가 올 때 부호 확장 된 값이 아닌 첫 번째에 들어왔던 14 비트가 복사될 수 있도록 구성되어 있다. 그림 7 (b)는 두 번째 명령어가 들어올 경우 일어나는 동작을 설명한 그림이다. EF 가 설정 되어 있으므로 처음 들어왔던 하위 14 비트는 다음 14 비트로 복사되고 최상위 4 비트는 값을 유지한다.

(2) 선 해독기(Pre-Decoding): 선 해독기는 ER flag (EF)을 참고하여, 명령어의 IMM 을 읽어와 ER 을 완성하는 동작을 수행한다. ER flag 가 설정되어 있으면 일반 명령어의 전에 LERI 명령어가 읽혀졌음을 의미하므로 명령어 내의 IMM 의 크기에 따라 ER 을 시프트 한 후 ER 의 최하위 비트에 IMM 을 넣어주어 ER 을 완성시킨다. 반대로 ER flag 가 설정되어 있지 않으면 IMM 을 최 하위비트에 넣어주고 이외의 비트는 0 으로 채워준다.

그림 8 은 선 해독기의 구성 및 동작을 보여주는 그림이다. 먼저 명령어의 IMM 을 추출하기 위해 내부의 Decoder 가 명령어를 해석한다. 추출된 IMM 의 크기에 따라 나머지 비트가 Shifter 나 ZEXT (Zero Extension)에 의해 채워지게 되고, IMM 은 그림의 맨 오른쪽 ER 의 최하위 비트에 쓰이게 된다. 명령어와 ER 은 명령어의 종류와 EF 비트에 따라, 선 해독기에 의해 여러 종류로 해독되어 ER 레지스터 값에 변경 저장되게 된다.

(3) 패치 큐, 인스트럭션 큐: IF 단계의 내부에 존재하는 큐는 명령어, ER, PC, Control 등을 다음스테이지로 전달하는 과정에서 버퍼로서 기능한다. PC, IR 과 같은 다른 내부자료구조와는 다르게 큐는 여러 가지 동작을 수행하게끔 요구 받으므로 자체적인 관리 메커니즘을 가지고 있어야 한다.

본 프로세서에서는 순환큐를 채용한다. 순환큐는 큐에 대한 입출력이 일어날 때 shifting 이 일어나지 않고 포인터의 조작으로 큐의 입출력을 조절한다. Flush 와 같은 reset 동작도 큐를 비우는 실제적인 동작 없이 포인터를 초기화함으로써 해결할 수 있다.

(4) 명령어 패킷 전달 및 분기, 정지 명령 처리 모듈: IF 모듈은 DE 모듈에 완성된 명령어 패킷을 전달해 주고, DE 모듈은 IF 모듈에 명령어를 해석하여 결과에 따라 분기 혹은 정지 명령을 내릴 수 있다. 먼저, DE 모듈에서 명령어 패킷을 요구할 경



우, IF 모듈은 명령어 큐에 저장된 명령어 패킷을 인출하여 DE 모듈에 전달한다. 그리고 DE 모듈로부터 분기 명령을 받을 경우 그림 6 의 De\_recv 블록의 Halt\_cmd 를 통해 큐를 포함한 각 모듈의 동작을 정지 시키고, 큐 등의 자료 구조를 초기화 한 후, New\_PC 를 통해 분기 주소를 업데이트 하고, 다시 모듈들의 동작을 재개시킨다. 정지 명령의 경우 모듈들의 동작을 재개시키지 않는 것만 제외하고 분기 명령을 처리할 때와 동작이 같다.

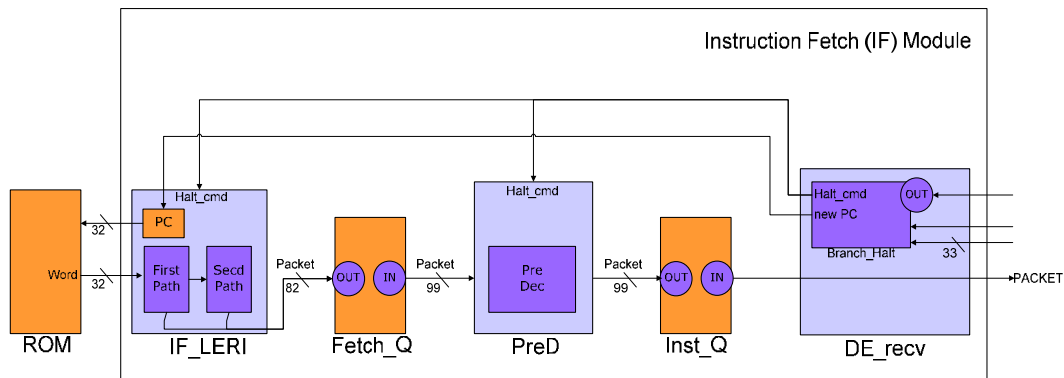


그림 6. Instruction Fetch 블록의 구성도

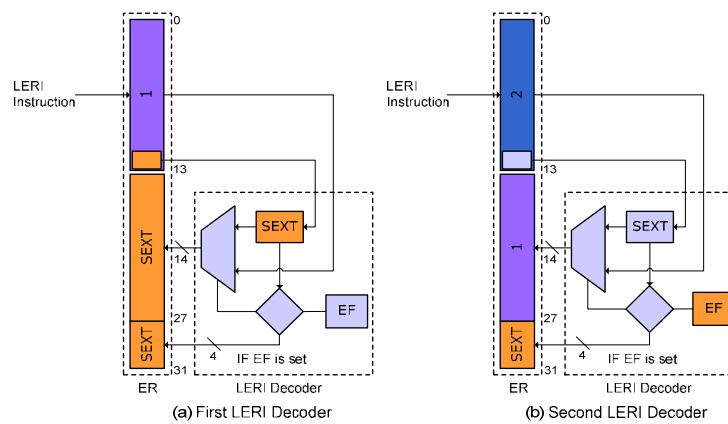


그림 7. LERI Decoder

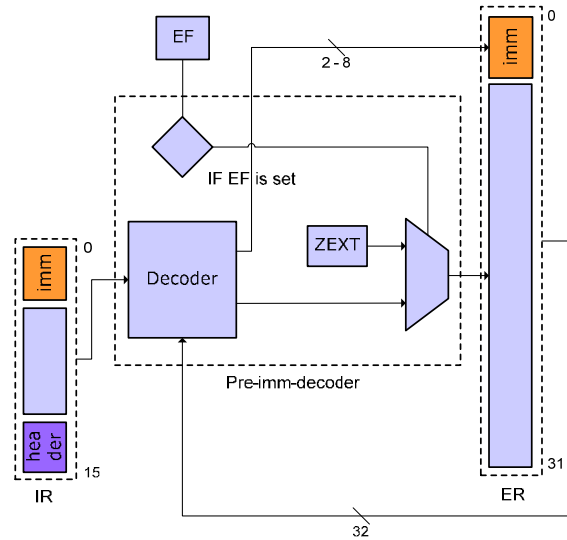


그림 8. 선해독기

다. 명령어 해독기

Decoder(DE) 블록은 IF 블록을 통하여 인출된 명령어를 전달 받아, 이를 실제 수행을 위하여 명령어에 대한 해독과 필요한 리소스(operand)를 인출하여 EX 블록으로 전달하는 역할을 수행한다. Decoder 는 총 99 개의 명령어를 지원하며, ALTHEA 프로세서의 decoder 블록은 다음의 표 5 와 같은 기능과 규격을 만족하도록 설계하였다.

표 5. ALTHEA decoder 블록의 주요 기능 및 규격

기능	규격
명령어셋 지원	DSP 명령어를 제외한 AE32000c 프로세서 명령어 99 종
기능	명령어셋의 해독
동작 방식	Clockless 비동기식 Handshaking
인터페이스 블록	IF, EX, RF, ME, CP 블록

(1) 명령어 해독기 구조

그림 9 는 DE 블록의 여러 기능 블록을 개념적으로 정리한 블록도이다. DE 블록은 내부에 개념적으로 다음과 같은 기능 모듈로 설계 하였다.

- 명령어 해독 모듈 : 인출된 명령어의 동작 해석
- 분기 처리 모듈 : 분기 명령에 대한 분기 어드레스 및 조건 확인
- 오퍼랜드 처리 모듈 : 명령어의 수행에 필요한 오퍼랜드들에 대한 인출
- 전방전달 검출 모듈 : 전방전달을 통해 입력되는 데이터들에 대한 검출
- HALT 처리 모듈 : HALT 명령어에 따른 동작 정지 및 레벨 처리
- Co-processor I/F 모듈 : 코프로세서로의 명령 및 데이터 전달

(2) 명령어 해독기 동작

DE 블록은 전술한 바와 같이 IF 단에서 인출된 명령어의 해독 및 실행에 필요한 정보의 전달을 주 기능으로 한다. DE 블록에서 해독하는 명령어는 크게 두 가지 종류로 구분 된다.

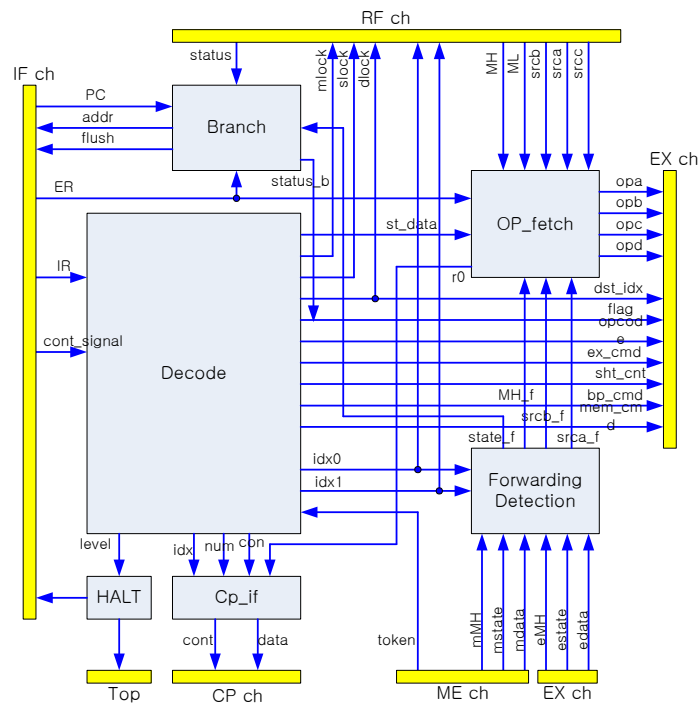


그림 9. Decoder 블록 내부 구성도

- 단일 사이클 명령어 해독
- 멀티 사이클 명령어 해독

단일 사이클 명령어는 해당되는 명령어의 해독이 하나의 사이클 안에 완결되는 명령어에 대한 해독으로서 대부분의 명령어가 이에 해당된다. 멀티 사이클 명령어는 PUSH/POP 명령어와 exception 의 경우에 해당된다. 멀티 사이클 명령어의 해독은 하나의 사이클 안에서 명령어에 대한 해독이 완결되지 않으므로 해당 명령어에 대한 상태 변수를 두어 사이클이 진행됨에 따라 동작을 변화하면서 해독 작업을 수행한다. DE 블록에서의 명령어 해독과정은 다음의 그림 10 과 같은 절차에 따라 수행한다.

(가) 리셋 및 채널 초기화

리셋 이후 내부 변수 대한 초기화와 전방전달(forwarding), 채널 초기화에 따른 deadlock 방지를 위한 채널 쓰기 및 읽기 과정

(나) 토큰 입력

파이프라인 동기화를 위한 토큰 확인. 토큰을 보낸 이력이 있을 경우는 명령어 인출동작을 수행하지 않는다(그림 10 의 1.3 과정).

(다) 명령어 인출 혹은 멀티 사이클 처리

단일 사이클 명령어 처리의 경우는 IF 단에서 새로운 명령어에 대한 인출을 수행하며, 멀티 사이클 명령어 처리의 경우는 멀티 사이클 처리 상태변수를 확인하여 해당 동작을 처리

(라) 명령어 디코딩 준비

명령어 디코딩을 위한 준비과정으로, 단일 사이클 처리의 경우는 토큰 상태를 초기화하고, 멀티 사이클 처리의 경우는 멀티 사이클 처리를 위한 상태 변수 및 정보에 대한 갱신을 수행

(마) 명령어 해독

단일 사이클의 경우는 입력 받은 명령어에 대한 해독을 수행, 멀티 사이클의 경우는 이전 명령어에 따른 현재 동작을 수행(PUSH/POP)

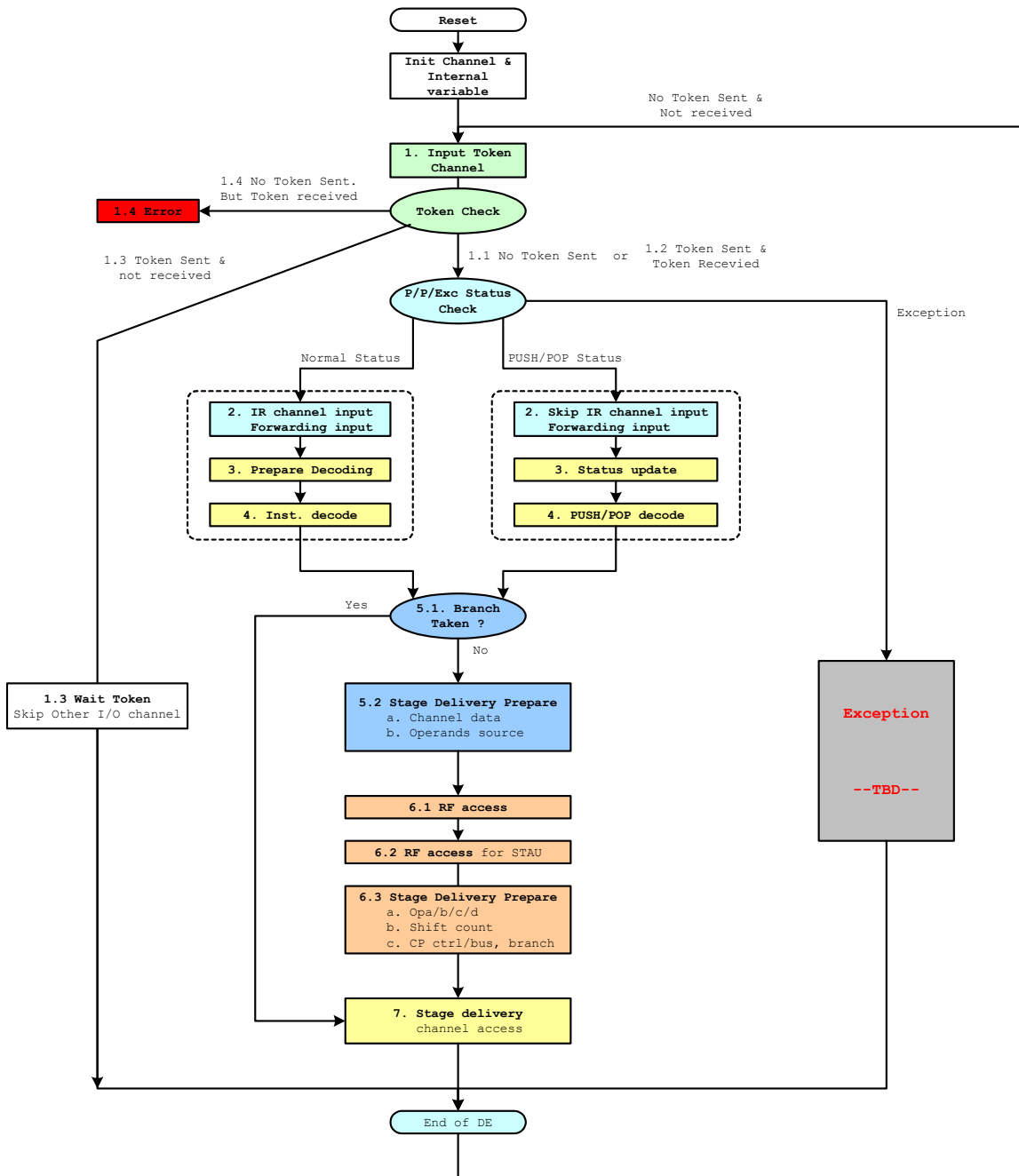


그림 10. DE블록에서의 명령어 해독과정

(바) 분기 명령 처리

입력 받은 명령어가 분기 명령일 경우 분기를 위한 처리를 수행. 전단계(그림 10의 5.1 과정)에서 RF 블록에 대한 접근이 불필요한 조건 분기 혹은 절대분기를 처리

하고, 두번째 단계는 일반 명령어 혹은 RF 블록에 대한 접근이 필요한 분기의 처리

(사)RF 접근

각 오퍼랜드에 대한 데이터 생성을 위하여 RF 블록에 대한 레지스터 읽기 작업을 수행하여 데이터를 인출

(아) 생성된 모든 오퍼랜드, 명령 등을 EX 블록으로의 전달

그림 11 은 멀티 사이클 처리를 필요로 하는 PUSH 와 POP 명령어 처리에 관한 상태 머신을 나타낸 그림이다.

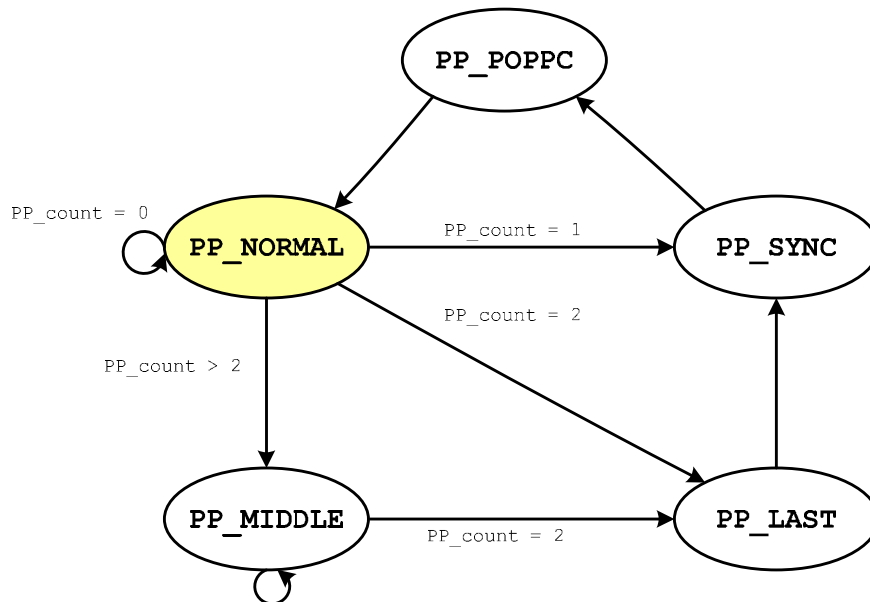


그림 11. 멀티 사이클 명령어 처리 상태 머신 - PUSH/POP

라. 명령어 실행기

명령어 실행기(Execution 블록, 이하 EX 블록)는 5 단 파이프라인 중 EX 단을 실행시키는 블록으로 주로 DE 블록으로부터 소스 오퍼랜드를 입력 받아 연산을 수행하고 결과값을 ME 블록으로 전달하는 역할을 수행한다. 다만, DE 블록 이외에 CP 블록에서 데이터를 입력 받을 수 있고 연산 수행 결과를 ME 블록 이외에도 다시 DE 블록으로 전달할 수도 있다.

(1) EX 블록의 구조

앞에서 정의된 인터페이스 정의에 의해 De\_ex\_op 채널은 EX 블록에서 수행되는 명령어들을 크게 ALU, memory, shift, unary, multiply, co-processor, 단순 move 의 7 가지 그룹으로 나눈 정보를 포함한다. 이들 명령어 그룹들은 수행되는 연산 종류가 다를 뿐만 아니라, 명령어 그룹에 따라 연산 후 발생하는 3 가지 결과 신호 Ex\_result, Ex\_flags, Ex\_mh 채널의 출력 저장 형태도 달라진다.

ALTHEA EX 블록은 이러한 7 가지 그룹의 명령어 처리 방식을 가능한 준수하여 모듈별로 설계되었다. 그림 12 는 EX 블록의 구조도를 나타낸다.

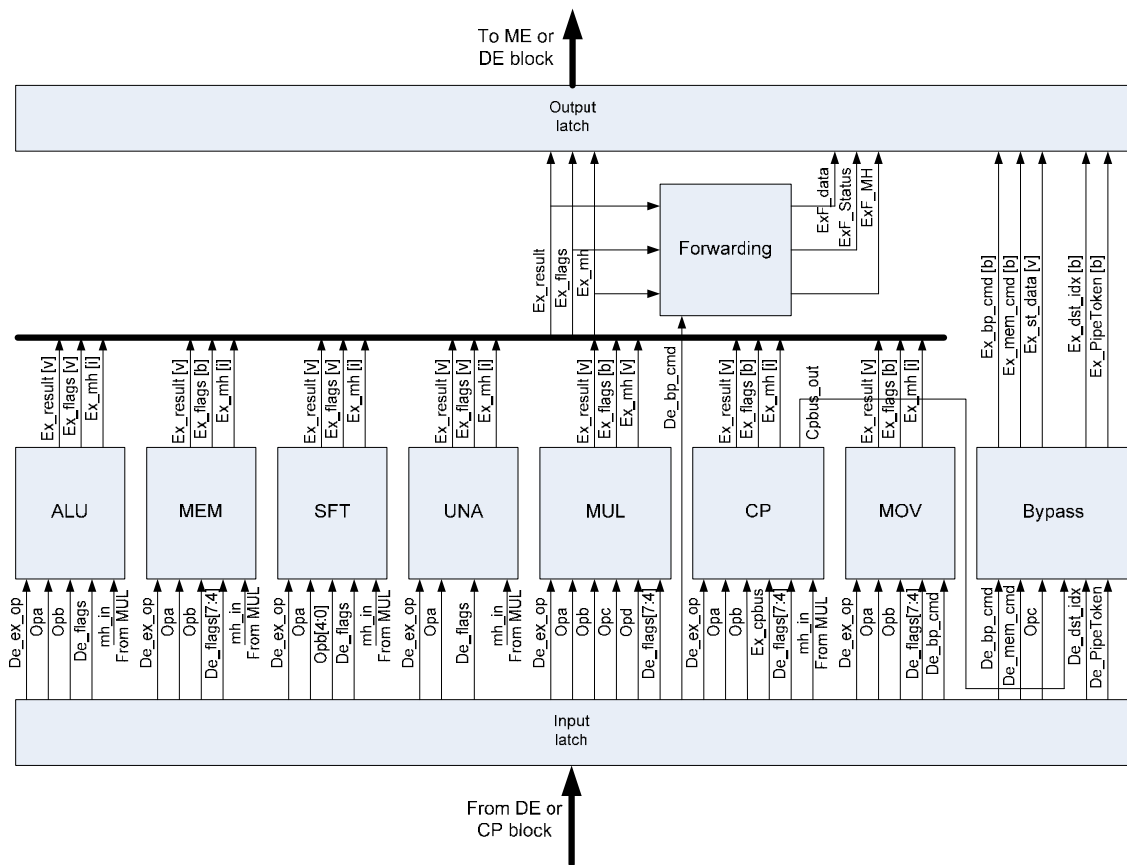


그림 12. EX 블록 구조도

앞에서 말한 7 가지 모듈 이외에 Bypass 모듈과 Forwarding 모듈이 존재하는데, Bypass 모듈은 파이프라인 구조 때문에 생기는 ME 블록으로의 단순 데이터 전달을

위한 것이고, Forwarding 모듈은 연산 결과를 DE 블록으로 선전달하기 위한 것이다. 각 모듈들의 출력 신호에 표시된 [v], [i], [b]는 현재 출력 채널에 담긴 데이터가 유효한 경우([v])로 변경하거나 실제 데이터를 변화시키지는 않은 경우([i])로 변경, 혹은 입력 채널 중의 하나를 그대로 전달만 하는 경우([b])를 의미한다. DE 블록 혹은 CP 블록에서 입력되는 데이터는 입력 래치에 저장되고 DE\_ex\_op 신호에 따라 연산을 위한 모듈과 입력들이 선택된다. 연산결과는 유효성 정보를 덧붙여서 단순 전달되어야 되는 데이터와 함께 ME 혹은 DE 블록으로 전달된다.

## (2) Execution 블록의 동작

다음에서 EX 블록의 각 서브 모듈에 대해 설명한다.

### (가) ALU 모듈

De\_ex\_op[0] 신호가 세팅되었을 때 구동되는 모듈이다. 두 소스 오퍼랜드 Opa, Opb 로 연산을 수행하며 연산의 종류는 De\_ex\_cmd 정보에 기반한다. 실제로 각 연산을 위해 가산기, 감산기 및 각종 논리 연산기가 구현되었다. 연산 시에 사용될 flag 값과 연산 결과 반영되어야 할 flag 비트의 정보를 가진 De\_flags 신호가 입력된다.

### (나) MEM 모듈

De\_ex\_op[1] 신호에 의해 구동되며, 메모리 참조 명령어의 메모리 주소값을 계산한다. 내부적으로는 두 소스 오퍼랜드인 Opa, Opb 를 이용한 32 비트 가산 연산이 이루어진다. Flag 비트 반영은 없으며, Ex\_mh 도 비활성화된다.

### (다) SFT 모듈

De\_ex\_op[2] 신호에 의해 구동되며, 쉬프트 명령어의 연산을 수행한다. 소스 오퍼랜드 Opa 의 값을 Opb[4:0] 만큼 쉬프트시킨다. 연산 결과에 따라 overflow 를 제외한 C, Z, S 비트를 반영한다.

### (라) UNA 모듈



단일 소스 오퍼랜드를 사용한 연산을 수행하며, De\_ex\_op[3]에 의해 구동된다. 연산 결과 및 flag 비트를 반영한다.

#### (마) MUL 모듈

곱셈 연산과 MAC 연산을 수행하며, De\_ex\_op[4]에 의해 선택된다. Flag 비트 반영은 없으며, 그전 flag 값을 bypass 시킨다. 32 비트 곱셈 연산을 위해 Opa, Opb 를 사용하거나, 32 비트 MAC 연산을 위해 Opa, Opb, Opc, Opd 오퍼랜드를 사용한다. 64 비트 결과값은 상위 32 비트는 Ex\_mh 에 하위 32 비트는 Ex\_result 에 유효 비트를 첨가하여 출력된다.

#### (바) CP 모듈

Coprocessor 관련 명령어 중에서 EX 블록에서 처리하는 명령어는 GETCn, MVFn, LDCn, STCn 이며, De\_ex\_op[5]에 의해 active 된다. CP 모듈은 이러한 명령어를 위한 데이터 패스를 제공하는데, 구체적으로 LDCn, STCn 을 위한 어드레스 계산, MVFn 을 위한 Ex\_cpbus 데이터의 Ex\_result 로의 전달, 그리고 GETCn 을 위한 Ex\_cpbus 데이터 값에 따른 Zero flag 변경 등을 수행한다. 이 모듈에서는 GETCn 명령어 처리시에 Zero flag 를 반영하는 것을 제외하고 flag 값은 bypass 시킨다.

#### (사) MOV 모듈

LEA 명령어를 제외한 move 계열 명령어와 PUSH, POP 의 EX 단 처리를 위한 모듈로 De\_ex\_op[5:0]가 '0' 으로 세팅될 때 구동된다. Opa, Opb 데이터가 De\_bp\_cmd 값에 의해 선택되어 ex\_result 로 출력 되며, flag 는 bypass, ex\_mh 는 invalid 상태로 출력된다.

#### (아) Bypass 모듈

ME 블록에서 필요한 제어 신호인 Ex\_bp\_cmd, Ex\_mem\_cmd 와 store 명령어의 데이터(Ex\_st\_data), 목적지 오퍼랜드의 레지스터 주소(Ex\_dst\_idx), 그리고, 파이프라인 토큰(Ex\_PipeToken)에 DE 블록에서 생성된 데이터를 전달한다. 일반 store 명령어는 Opc, STCn 명령어는 Ex\_cpbus 데이터를 사용하는데 CP 계열 명령어가 입력될때만

Ex\_cpbus 채널을 active 시키기 위해, Ex\_cpbus 를 저장하는 CP 모듈의 Cpbus\_out 신호를 사용하여 STCn 을 처리한다.

(자) Forwarding 모듈

EX 블록의 결과인 Ex\_result, Ex\_flag, Ex\_mh 값을 De\_bp\_cmd 에 의해 DE 모듈로 forwarding 할것인지 아니면, invalid 임을 알릴 것인지를 판별하여 출력한다. 현재 ALTHEA 버전은 forwarding 기능을 지원하지 않으므로 이 모듈은 구현되어있지 않다.

라. 메모리 접근

메모리 접근 블록(Memory 블록, 이하 ME 블록)은 그림 13 과 같이 메모리 접근 및 데이터 align 기능을 수행하는 Mem\_if 모듈, PUSH/POP 명령어 수행을 위해 SP 값을 변경시키는 SP\_inc 모듈, WB 블록으로 출력 값을 선택하는 Port\_sel 모듈, DE 블록으로의 forwarding 데이터를 생성하는 Forwarding 모듈, 그리고, 코프로세서 인터페이스를 위한 Cp\_if 모듈로 구성된다.

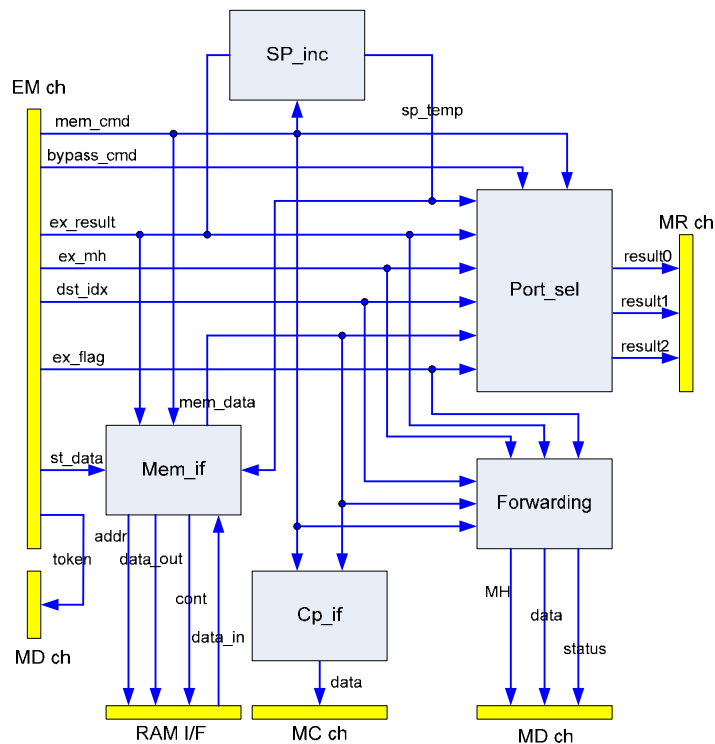


그림 13. ME 블록 구조도

### (1) 메모리 접근 블록의 구조

Mem\_if 모듈은 메모리 명령어인 경우 활성화되며 EX 블록의 결과값인 ex\_result 를 주소 값으로 사용한다. mem\_cmd 에 의해 메모리 쓰기/읽기 동작이 구분되며 쓰기 동작의 경우는 저장할 데이터(st\_data)를 입력 받는다. 메모리와의 인터페이스는 주소(addr), 출력 데이터(data\_out), 제어신호(cont), 입력 데이터(data\_in)로 구성된다. 메모리 읽기의 동작의 경우 mem\_cmd 에 정의된 size 정보로 정렬 및 부호확장 연산을 수행하여 결과값(mem\_data)을 출력한다. PUSH/POP 명령어인 경우에는 ex\_result 대신 SP 를 변경시킨 sp\_temp 를 사용하여 주소 값을 생성시킨다.

SP\_inc 모듈에서는 초기 주소 값 즉, SP 값은 ex\_result 로부터 입력되고, 처음 메모리 접근 동작을 수행한 후, 초기 SP 값을 4 씩 증가(POP 명령어), 혹은 4 씩 감소(PUSH 명령어) 연산을 수행한다. 이때, 입력되는 dst\_idx (POP 명령어), st\_data (PUSH 명령어)와 동기화되어 다중 메모리 접근 동작을 수행한다. 이후, mem\_cmd 로부터 최종 메모리 접근 동작임을 알리는 신호를 감지하여 최종 SP 값을 출력(sp\_temp)한다. ALTHEA 코어의 레지스터 파일에는 총 3 개의 쓰기 포트가 존재하며 Port\_sel 모듈에서 선택된다. 이들 포트에 EX 블록의 결과값인 ex\_result, 메모리 읽기 결과값인 mem\_data, EX 블록에서 전달되는 MH 값(ex\_mh), status 값(ex\_flag)을 선택하는 기능을 수행한다. Cp\_if 모듈은 코프로세서 명령어 중 메모리의 값을 읽은 후 코프로세서 내부 레지스터에 쓰기 동작을 수행하는 명령어를 위해 mem\_data 를 코프로세서에 전달한다. Forwarding 모듈에서는 EX 단의 결과 값인 ex\_result, 혹은 메모리 읽기 명령어인 경우는 mem\_data, 그리고, ex\_mh, ex\_flag 를 DE 블록으로 전송한다. 이때 인덱스로 구별되는 ex\_result 나 mem\_data 는 dst\_idx 와 동기화되어 같이 전송된다. 현재 Forwarding 기능은 구현되어 있지 않다.

### (2) Memory 블록의 동작

ME 블록에서는 Ex\_mem\_cmd, Ex\_bp\_cmd 값에 의해서 총 9 가지의 case 로 나뉘

어서 내부 동작을 수행하는데 다음의 4 가지 수행으로 요약할 수 있다.

(가) 일반 메모리 접근 명령

코어 프로세서의 메모리 읽기 수행의 경우 메모리 참조 후 입력 데이터를 크기에 맞게 정렬한 후 포트 MRF\_result0 에 출력한다. 다른 포트는 invalid 비트를 더하여 출력되며, pipe 는 bypass, cpbus 는 역시 invalid 비트가 추가되어 출력된다. 쓰기 수행의 경우에는 RF 블록쪽의 출력은 모두 invalid 로 출력 되며, 나머지 출력은 읽기의 경우와 동일하다.

(나) Bypass 명령

메모리를 접근하지 않는 명령어의 처리를 위한 것으로 RF 블록으로의 쓰기 동작의 경우는 valid 로 출력되고, 그렇지 않는 경우는 invalid 로 출력된다

(다) 코프로세서 메모리 접근 명령

코프로세서의 메모리 접근 명령을 처리한다. 일반 메모리 접근 명령과 유사하나 읽기의 경우 Me\_cpbus 채널로의 valid 출력이 발생한다. 쓰기의 경우에는 32 비트 데이터만 발생되므로 align 을 제외한 모든 동작은 일반 메모리 접근 명령의 쓰기 동작과 동일하다.

(라) Stack 메모리 접근 명령

구현상 PUSH/POP 명령어는 달리 단일 명령어 접근뿐만 아니라, 다중 접근이 가능하며, 항상 SP 값을 참조 혹은 변경해야 하므로 일반 메모리 접근 명령과 다르다. 편의상 PUSH/POP 처리를 위해서 ALTHEA 코어에서는 단일 stack 메모리 접근(Stack One), 다중 stack 메모리 접근 중에서 첫번째(Stack First), 중간(Stack Middle), 마지막(Stack Last) 으로 구분하여 구현하였다.

바. 레지스터 파일

레지스터 파일 블록(Register File, 이하 RF 블록)은 프로세서 코어의 실행에 필요한 임시 변수를 저장하는 레지스터들의 집합으로 그림 14 와 같이 설계 되었다. RF 블

록은 내부에 다음과 같은 기능 모듈로 구성된다.

- Register File 레지스터 모듈 : GPR 과 SPR
- Lock 레지스터 모듈 : 상호의존성 해결을 위한 Lock 레지스터
- 레지스터 읽기 모듈 : 레지스터에 대한 읽기 처리
- 레지스터 쓰기 모듈 : 레지스터에 대한 쓰기 처리
- Lock 처리 모듈 : 목적 레지스터 Lock 처리

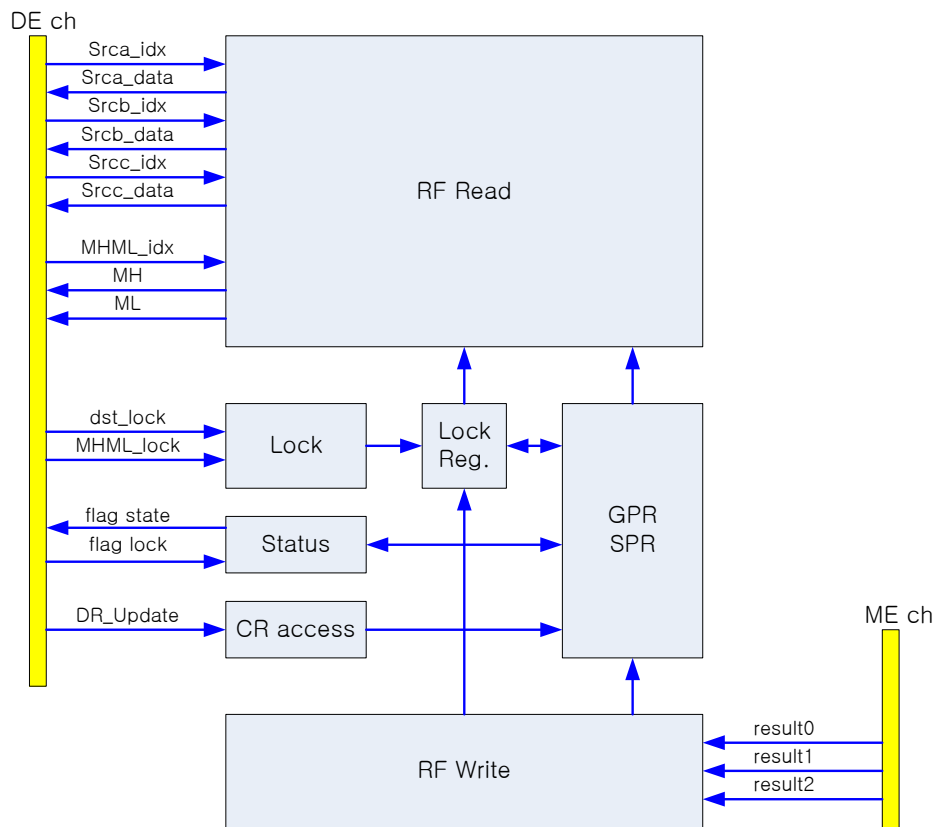


그림 14. RF 블록 내부 구성도

레지스터 파일의 설계에 있어서 중요한 점은 비동기 식으로 프로세서 코어가 설계됨에 따라, 레지스터의 읽기와 쓰기의 시점이 일치 하지 않음으로 인하여 발생 가능한 데이터의 일관성 문제를 해결하는 것이다. 데이터에 대한 일관성이 유지 되

지 않을 경우 코어의 동작을 정지시키거나(deadlock), 인출 및 쓰기 과정의 중재오류, 잘못된 lock 으로 인한 동작 정지와 같은 문제를 야기할 수 있다. 이와 같은 문제점을 해결하기 위하여 RF 블록의 액세스는 다음과 같은 절차에 따라 이루어 지도록 설계 하였다.

#### (1) 리셋 및 채널 초기화

이 단계에서는 리셋 이후 내부 변수에 대한 초기화 과정 채널 초기화에 따른 deadlock 방지를 위한 채널 쓰기 및 읽기 과정을 포함한다.

#### (2) DE 블록으로부터의 읽기 요청

DE 블록으로부터 RF 블록에 대한 레지스터 인덱스요청을 처리. 이 과정에서는 요청된 레지스터에 대한 lock 상태 확인을 수행

#### (3) 레지스터 파일 읽기

이전 과정에서 레지스터에 대한 lock 상태확인 결과, lock 이 되어 있지 않은 레지스터에 대하여 데이터를 DE 블록으로 전달

#### (4) 목적 레지스터 lock 설정

요청된 모든 레지스터에 대한 읽기 과정이 종료된 후, DE 블록으로부터의 목적 레지스터에 대한 lock 요청을 확인하고, lock 요청이 있을 경우 lock 처리

#### (5) ME 블록으로부터의 쓰기 요청

ME 블록으로부터의 쓰기 요청을 처리. 이 과정은 lock 의 확인 및 해지 과정을 포함, 이때 lock 이 걸리지 않은 레지스터에 대한 쓰기 요청을 받을 경우 RF 블록은 정지함에 유의

#### 사. 보조프로세서

보조프로세서 블록(Co-Processor 블록,이하 CP 블록)은 ALTHEA 처리하는 명령어 가운데 코프로세서 관련 명령어를 처리한다. 주로 CP 블록내의 레지스터 파일의 읽

기, 쓰기 기능을 수행한다.

### (1) CP 블록의 구조

현재 구현된 CP 블록의 구조는 그림 15 와 같이 내부에 16 개의 32 비트 레지스터를 갖는 레지스터 파일과 제어 신호를 해석하는 디코더로 구성되어 있다. DE 블록과의 인터페이스로부터 입력 받은 제어 신호와 데이터를 해석하여 레지스터 파일에 저장할 수 있고, 레지스터 파일의 값을 EX 블록과의 인터페이스를 통해 출력, 혹은 ME 블록과의 인터페이스로 입력 받아 레지스터 파일에 저장한다.

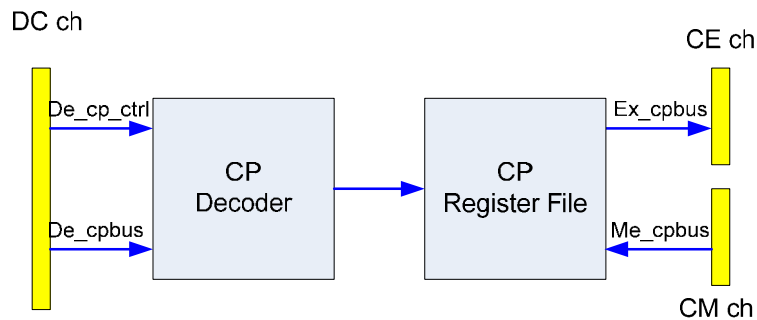


그림 15. CP 블록 구조도

### (2) CP 블록의 동작

현재 Co-Processor 관련 명령어 중 구현된 것은 EXEC 를 제외한 CMD, GETC, MVTC, MVF, LDC, STC 등이다. 이중 CP 블록 입장에서 GETC, MVFC, STC 는 CP 레지스터 파일의 값을 읽어서 EX 블록에 출력하는 동작을 수행한다. MVTC 명령어는 DE 블록과의 인터페이스를 통해서 레지스터 파일에 데이터를 저장할 수 있고, LDC 는 ME 블록과의 인터페이스를 통해 데이터를 저장한다.

## 2. 검증 환경 설계 및 구현

설계된 저전력 비동기 프로세서의 기능을 검증하기 위하여는 기능의 검증을 위한 다양한 환경을 구현하여야 한다. 가장 기본적으로 요구되는 것은 상위수준의 언어

를 이용한 검증 프로그램의 작성을 위한 프로그램 개발 환경이며, 다음으로 하드웨어 행위 수준 혹은 RTL 수준의 설계에서의 신호 타이밍, 동작/기능, 오류 등의 검증에 필요한 검증 환경이다. 검증 환경을 통하여 검증하고자 하는 비동기 프로세서의 기능은 다음과 같다.

- ALTHEA 기능 블록의 기능 및 동작
- ALTHEA 개별 명령어의 동작
- ALTHEA 명령어 집합으로 이루어진 테스트 프로그램의 동작
- ALTHEA 명령어 사용 빈도

이와 같은 기능의 검증을 위하여 본 연구과제에서는 ALTHEA 프로세서를 위한 C 컴파일러의 개발 및 검증을 위한 검증 환경을 설계/구현 하였다.

#### 가. 검증환경 구조

그림 16 은 검증환경의 전체 구조를 나타낸 그림이다. 검증 환경은 행위수준 모델과 RTL 수준 코드의 검증을 위한 환경으로써, C 혹은 어셈블리 언어를 이용하여 검증 코드를 작성하고, 이를 C 컴파일러를 이용하여 바이너리 코드를 생성한다[11][12]. 생성된 코드는 ROM 데이터로 변환되어 행위수준 혹은 RTL 수준의 프로세서 코어의 입력 스티뮬러스로 적용된다. 검증 환경은 프로세서 코어 코드와 검증을 위한 코드로 나뉘어지며, 검증을 위한 코드로는 성능 프로파일러와 모니터, 명령어 동작 체커로 구성하였다.

프로파일러는 개별 명령어의 빈도, 성능을 측정하며, 명령어 체커는 파이프라인의 DE, EX, ME 단계에서의 각 명령어의 실행 동작을 모니터링하여 정상 동작 여부를 확인한다. 또한 프로파일러에서 외부 메모리 액세스, 레지스터 파일 액세스에 대한 텍스트 출력을 기록하여 모델 간의 동작 일치성을 확인할 수 있도록 하였다. 이를 다음의 그림 17 에 나타내었다.



## 나. 검증환경 설계 및 구현

### (1) 성능 프로파일러의 설계 및 구현

성능 프로파일러는 향후의 동작에 대한 성능을 확인하기 위한 것으로, 프로그램 상에서 개별 명령어의 사용 빈도와 실제 명령어가 수행된 최소 시간, 최대 시간 그리고 전체 평균시간을 계산한다. 이를 위하여 성능 프로파일러는 DE 단으로 입력되는 명령어들에 대한 디코딩과, 각각의 명령어들의 입력 시간, 실제 수행시간, 명령어의 출현 빈도를 계산하여 저장하고 검증의 종료에 따라 결과를 출력하도록 설계하였다.

### (2) 명령어 동작 체커의 설계 및 구현

명령어 동작 체커는 DE, EX, ME 블록 별로 작성하였다. 각 동작 체커는 각 블록의 입력 신호에 따른 예상 출력 신호의 값을 판별하도록 설계하였다. 예를 들어 EX 단계에서 ADD 연산을 수행하도록 입력이 주어질 경우 오퍼랜드의 값을 이용하여 체커에서 연산 결과를 동시에 수행하고, 블록의 최종 출력값과 비교한다. 또한 연산 결과에 따른 다음 단계의 제어신호 출력을 비교하여 각 블록에서의 동작을 확인하도록 설계하였다.

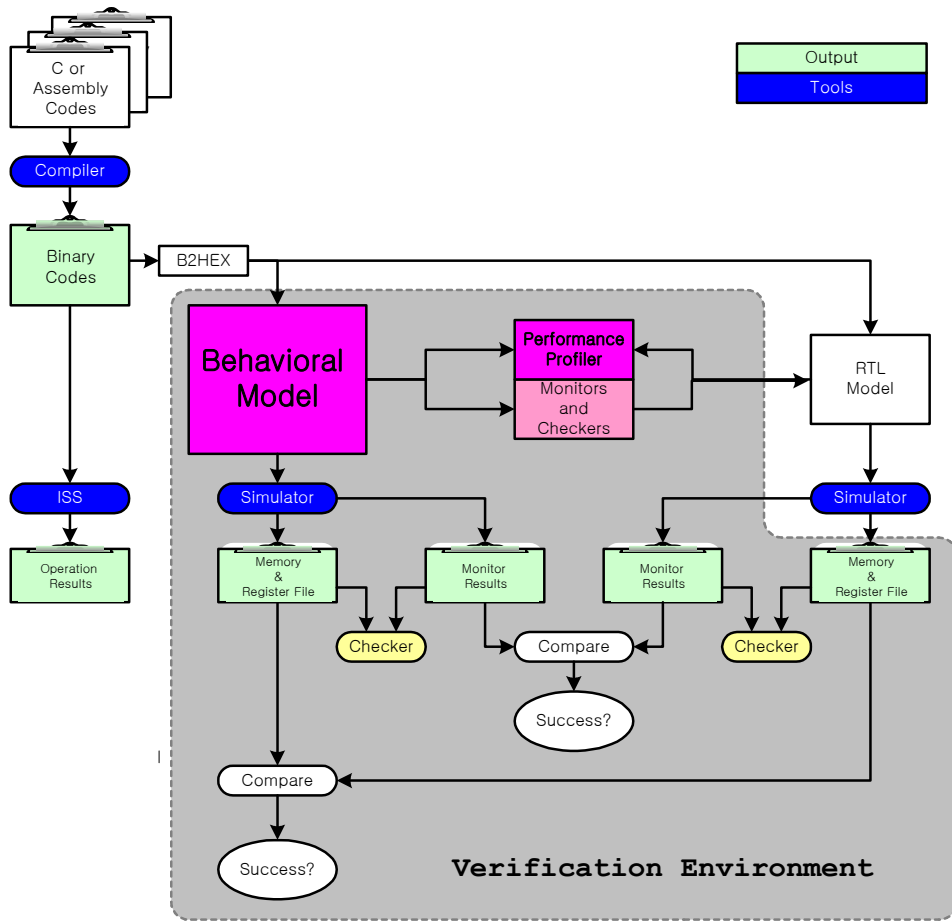


그림 16. 검증환경 구조도

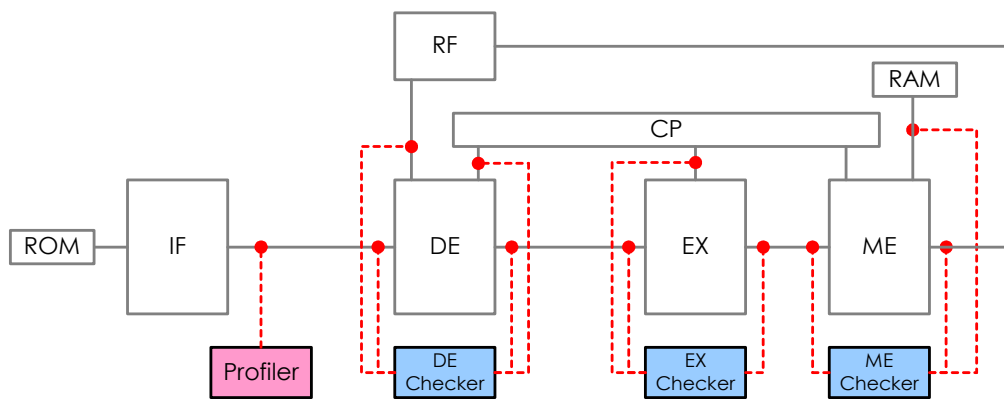


그림 17. 성능 프로파일러 및 기능 체커

## 제 5 절 개발 결과

본 절에서는 개발한 비동기 프로세서 코어의 기능에 대한 검증 결과를 설명한다.

### 1. 행위수준 기능 검증

행위수준의 기능 검증은 ALTHEA 비동기 프로세서 코어가 지원하는 100 개의 명령어 들에 대한 개별 명령어 동작을 검증하는 것으로써, 기능 검증을 위하여 시험 계획서 및 절차서를 작성하고 이에 따라 검증을 수행하였다[7][8].

행위수준의 기능 검증은 개별 설계 블록에 대한 컴파일 및 통합 오류를 검사한 후, 기능 검증을 수행하였다.

기능 검증의 1 단계에서는 개별 명령어에 대한 ROM 스티뮬러스를 작성하고 이를 검증환경을 통하여 검증하였다. 1 단계 검증은 개별 명령어의 파이프라인상에서의 기능 및 동작 여부를 확인하는 검증으로써, 1 단계 검증 과정을 통하여 69 개의 파이프라인 기능상의 오류를 확인한 후, 수정 조치 완료 하였다.

기능 검증의 2 단계에서는 C 언어로 작성된 명령어의 종류에 따른 명령어 검증용 테스트 벡터를 사용하여 명령어의 그룹별 테스트를 진행하였다. 테스트 벡터는 총 20 개의 세트로 구성되며, 각각의 세트는 개별 명령어 그룹의 동작, 각종 해저드의 발생 및 확인을 위한 코드로 구성되어 있다. 검증 결과 20 개의 테스트 벡터 세트를 통하여 11 개의 파이프라인 기능상의 오류를 확인한 후, 수정 완료 하였다.

각 검증 단계에서 프로파일러는 그림 18 과 같이 실행된 명령어의 개수 및 각 명령어에 대한 실행 통계를 출력하여 명령어의 정상 실행 여부를 확인하였다.

```

ALTHEA_MSG : Performance Profile Statistics -----
ALTHEA_MSG :   Inst : Occurrence :   Max :   Avg :   Min
ALTHEA_MSG : -----:-----:-----:-----:-----
ALTHEA_MSG :   LD   :           16 :    178 :    178 :    178
ALTHEA_MSG :   ST   :           17 :    181 :    181 :    181
ALTHEA_MSG :   LDB  :            8 :    178 :    178 :    178
ALTHEA_MSG :   LDS  :            4 :    178 :    178 :    178
ALTHEA_MSG :   STB  :            4 :    181 :    181 :    181
ALTHEA_MSG :   STS  :            2 :    181 :    181 :    181
ALTHEA_MSG :   LDBU :            8 :    178 :    178 :    178
ALTHEA_MSG :   LDSU :            4 :    178 :    178 :    178
ALTHEA_MSG :   LD2  :           16 :    178 :    178 :    178
ALTHEA_MSG :   ST2  :           82 :    181 :    181 :    181
...
ALTHEA_MSG : -----:-----:-----:-----:-----
ALTHEA_MSG : Invalid :            0 :            0 :            0 :    100000
ALTHEA_MSG : -----:-----:-----:-----:-----
ALTHEA_MSG : Performance Statistics Summary -----
ALTHEA_MSG : Executed Instructions           =    1078
ALTHEA_MSG : Average Execution time per Inst =    188
ALTHEA_MSG : !! WARNING: Summary is calculated without PUSH, POP and NOP !!
ALTHEA_MSG : -----:-----:-----:-----:-----

```

그림 18. 성능 로그 예

행위수준의 검증 결과 100 개의 ALTHEA 명령어 중 1 단계에서 구현 완료된 95 개의 명령어들에 대하여 기능 및 동작이 정상적으로 동작함을 확인하였다.

## 2. RTL 수준 기능 검증

행위수준에서 기능이 검증된 코드, 즉 HASTE 원천 코드는 TiDE, Synopsys Design Compiler 등의 합성 툴을 거쳐 RTL 수준의 코드(여기에서는 Gate 수준의 netlist)로 변환된다. 이 과정을 통해 실제 공정 라이브러리로 매핑된 netlist 는 pre-layout 단계의 성능 및 전력 측정에 사용될 수 있다. 성능 및 전력 측정 전에 생성된 RTL 코드의 보다 실제적인 환경에서의 functionality 를 검증하기 위해 행위 수준의 개별 명령어가 아닌 몇 개의 application 을 수행시켰다. Application 으로는 embedded processor 용 benchmark suite 로 academic 용으로 널리 사용되는 MiBench [13]와 Dhrystone[14]를 사용하였다.

그림 19 는 RTL 수준에서의 기능 검증 흐름을 나타낸다. C 언어로 작성된 benchmark program 의 main 함수 종료 부분에 Halt 수준 0 를 발생시키는 코드를 삽입하여 컴파일 후 32 비트 ROM 형식의 실행 코드를 생성시킨다. 한편, HASTE 수준의 ALTHEA 코어의 코드를 합성하여 gate 수준의 netlist 를 생성시키고, 이를 검증할 환경 파일을 작성한다. 이 환경 파일에 실행 코드를 include 하여 verilog 시뮬레이션을 수행한다. 해당 benchmark 프로그램의 정상 실행 여부는 시뮬레이션 종료 시점에서 halt 수준을 검사하여 0 인지를 검사하여 판명하였다. 표 6 에는 해당 benchmark 프로그램을 수행시켰을 때의 명령어 수와 실행 시간, 그리고 정상 실행 여부를 요약하였다.

차후 성능 개선 시에 개별 명령어 종류에 따라 구조가 변경될 수 있으므로, 실제 환경에서 ALTHEA 코어가 지원하는 모든 명령어의 사용 빈도 측정 데이터는 성능 개선에 중요한 자료가 될 수 있다. 수행 application 에 따라서 명령어 발생 빈도가 달라지겠지만, 현재 구현된 96 개의 명령어에 대하여 실행시킨 benchmark 프로그램의 총 24,310,516 개 명령어의 발생 빈도를 측정하였다. 그 결과 빈도 순서상 상위 20 개의 명령어가 약 93 %를 차지하며 그 명령어 및 빈도는 그림 19 에 나타나있다.

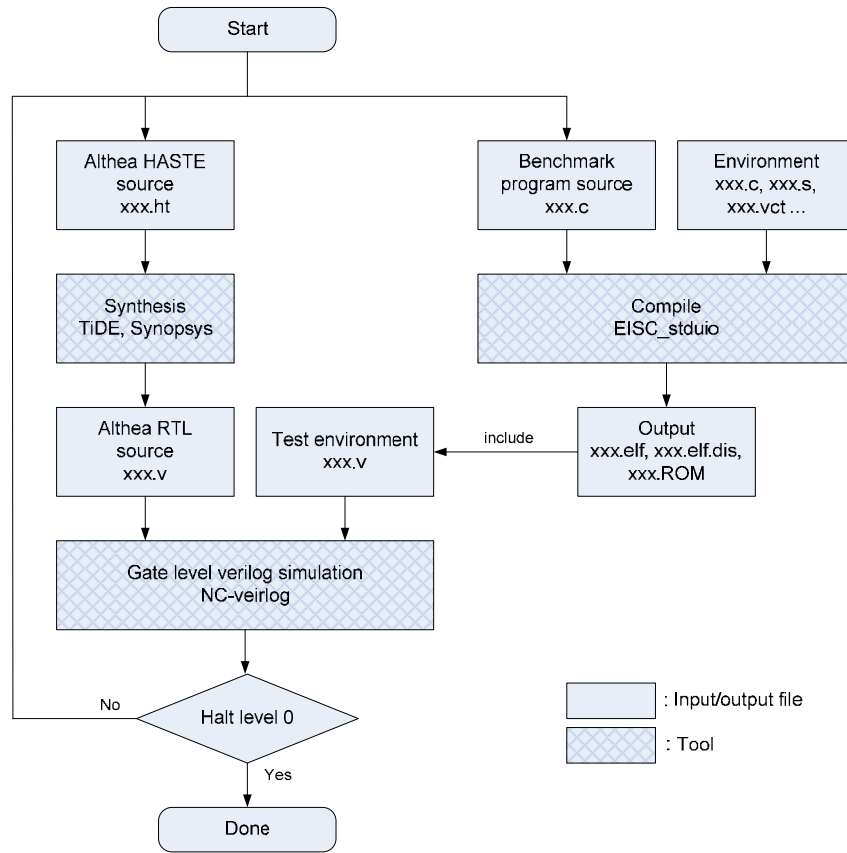


그림 19. RTL 수준 기능 검증 흐름

표 6. Benchmark 프로그램 수행 결과

Benchmark program	수행명령어 수	시뮬레이션 시간 (시:분:초)	정상 동작 여부
Stringsearch	100,421	30:04	yes
Qsort	105,825	29:56	yes
Bitcount	485,343	2:17:40	yes
Dijkstra	6,708,414	9:34:53	yes
Sha	16,049,554	37:32:19	yes
Dhrystone (x1000)	860,959	1:30:05	yes

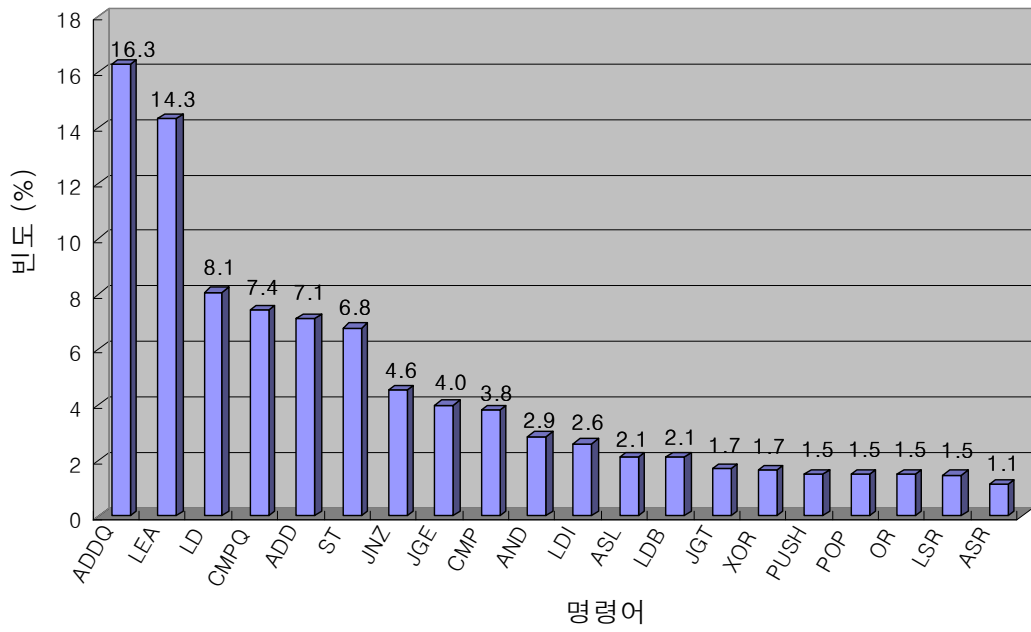


그림 19. 명령어 발생 빈도

## 제 4 장 목표달성도 및 관련분야에의 기여도

### 제 1 절 연구목표 및 달성도

본 연구개발 사업의 1 단계 과정을 통하여 초저전력 비동기 프로세서의 기능 및 성능 규격, 명령어 셋, 그리고 마이크로아키텍처를 정의하였으며, 이를 기반으로 하여 비동기 프로세서 블록 레벨의 설계를 진행하였다. 본 사업의 1 단계 연구개발에서 설정한 목표는 다음과 같다.

- 1 단계 (2007.1.1 - 2008.12.31): 기능 및 구조 설계 단계
  - 초저전력 비동기 설계기술 관련 특허동향 조사
  - 초저전력 비동기 프로세서의 기능/구조/인터페이스 정의
  - 프로세서의 마이크로아키텍처 정의
  - ALU 등과 같은 주요 기능 블록 설계
  - 주요 대학 및 유관 업체등과의 업무 협력 및 위탁연구 수행을 통한 연구 개발 효율성 제고

설정된 1 단계 연구개발 목표를 달성하기 위하여 수립한 연도별 연구목표 및 달성도를 주요 연차별로 정리하면 각각 다음의 표 7 과 8 에 나타난 바와 같다.

표 7. 2007 년도 연구목표 및 달성도

주요 연구결과(계획)	주요 연구결과(실적)
o 비동기 설계기술 동향에 대한 특허 분석	비동기 기술 국내외 특허 분석서 작성



o 비동기 설계기술 요소기술 분석	-[TDP] 비동기 회로 설계 요소기술 분석서 작성
o 비동기 프로세서 기능/성능 정의	-[TDP] 초저전력 비동기 프로세서 요구사항 정의서 -[TDP] 초저전력 비동기 프로세서 기능 및 성능 규격 정의서 -[TDP] 초저전력 비동기 프로세서 명령어셋 정의서
o 2007 년 연구성과 목표 - 국내특허 출원 (목표 2 건) - 논문 (비 SCI 4 건) - 문서	-국내특허: 2 건 출원, 국제특허 1 건 출원 -논문: 4 건 출원(비 SCI) -TDP 3 건, TM 16 건

표 8. 2008 년도 연구목표 및 달성도

주요 연구결과(계획)	주요 연구결과(실적)
o 비동기 프로세서 마이크로아키텍처 설계	-[TDP] 비동기 프로세서 명령어 microcode 상세분석서 작성 -[TDP] 비동기 프로세서 마이크로아키텍처 설계서 작성 -[TDP] 비동기 프로세서 블록설계서 작성 -[TDP] 비동기 프로세서 기능 시험계획서 작성 -[TDP] 비동기 프로세서 기능 시험절차 및 결과서 작성
o 비동기 프로세서 코어 설계 (RTL)	- 비동기 프로세서 코어 기능블록별 RTL
o 비동기 프로세서 코어 검증환경	기본설계 및 기능 검증 완료

<p>구축</p> <p>o 2008 년 연구성과 목표</p> <ul style="list-style-type: none"> <li>- 국내특허 출원 (목표 3 건)</li> <li>- 국제특허 출원 (목표 2 건)</li> <li>- 논문 (비 SCI 4 건)</li> <li>- 문서</li> </ul>	<ul style="list-style-type: none"> <li>- 비동기 프로세서 행위모델 RTL 설계 및 명령어수준 동작 및 통합검증 완료</li> <li>- 비동기 프로세서 RTL 코드의 검증을 위한 검증환경 구축 완료</li> <li>- 비동기 프로세서 RTL 기능 검증을 위한 FPGA 기반 검증환경 구축 완료</li> <li>- 국내특허 : 4 건 출원, 1 건 등록</li> <li>- 국제특허 : 6 출원</li> <li>- 국내외 논문 6 건 발표 및 게재 완료</li> <li>- TDP 7 건, TM 16 건 작성 및 등록</li> </ul>
--	---

## 제 2 절 기술적 성과 및 사회 경제적 성과

본 연구개발을 통하여 얻을 수 있는 기술적 측면의 성과는 다음과 같다.

- 유비쿼터스 서비스 환경으로 급격하게 전이됨에 따라 수요가 급증하고 있는 저전력 프로세서 기술에 대한 요구를 본 연구개발을 통하여 충족 가능
- 저전력 프로세서 기술에 대한 연구를 통하여 에너지 효율적인 그린 IT 기술에 대한 산업체의 수요에 적극 대응하고 비동기 설계 기반 저전력 프로세서 설계 기술에 대한 IPR 을 확보하여 국내 반도체 기반 산업의 국제경쟁력 확보하고, 이를 통하여 국내 관련 연구분야의 활성화를 꾀함
- 프로세서 기술은 최첨단 기술의 집합체로서 고난이도, 고부가가치의 기술로서 현재 세계적으로 선진 몇몇 기업에서 관련 기술을 독점하고 있어 이로 인한 기술적 산업적 종속현상이 심화되고 있음. 이러한 상황에서 본 연구개발을 통하여 저전력 프로세서 기술을 확보 함으로서 기술의 독자 자립 기반

을 확보할 수 있음

본 연구개발을 통하여 얻을 수 있는 산업 경제적인 측면에서의 성과는 다음과 같다.

- 초저전력 프로세서 관련 핵심 기술을 확보를 통한 비메모리 산업 분야를 육성하고, 초저전력 프로세서 및 저전력 설계 기술의 확보를 통한 관련기술의 수입대체 효과 및 수출 증대에 기여
- 확보한 저전력 프로세서 기술의 수요업체 기술이전을 통한 제품의 경쟁력 확보 및 기업의 대외 경쟁력 향상이 가능
- 현재 급격히 성장하고 있는 휴대형 정보기기 등에 폭넓게 적용하고, 신규 시장 및 관련 서비스를 창출하여 미래성장동력으로 육성 가능
- 정보단말기기에 의하여 주도 되는 유비쿼터스 사회에 있어서, 각종 유비쿼터스 정보 단말기기는 2008 년에 230 억 개에서 2012 년 1 조개까지 증가 전망
- SoC 시장은 한해 200 억불씩 증가하고 있으며, 그 중 마이크로 컴포넌트(단품)은 2004 년 기준 23%를 차지하고 있음. 전세계 마이크로 컴포넌트 시장은 2010 년 까지 꾸준한 규모의 성장세를 보이며 2010 년 약 660 억불 규모의 시장을 형성할 것으로 예측(Gartner Dataquest, '05. 9)
- 이와 같이 급격한 성장세가 기대되고 있는 이들 분야에서 자체 확보한 비동기 설계기술을 적용한 제품을 출시하는 경우 수입대체 효과 및 수출증대 효과도 높은 성장세가 예상됨

## 제 5 장 연구개발결과의 활용계획

본 연구개발 1 단계 과정을 통하여 32 비트급 초저전력 비동기 프로세서 코어의 기능 및 구조를 설계하고 명령어셋을 정의하였으며 이를 기반으로 하여 프로세서 마이크로아키텍처를 설계하였다. 또한, 비동기 설계 환경(CAD)과 검증환경, 그리고 비동기 프로세서를 위한 소프트웨어 개발환경을 구축하였으며, 이를 32 비트급 초저전력 비동기 프로세서 코어의 설계에 적용하여 블록 레벨에서 프로세서 코어를 RTL 로 구현하여 기능을 검증하였다. 이와 같은 2 년 동안의 연구개발을 통하여 32 비트 비동기 설계 기반 프로세서 코어 RTL 과 비동기 설계 및 저전력 설계에 대한 주요 요소기술에 대한 IPR 을 확보하였다.

1 단계 연구개발을 통하여 확보한 연구개발 결과들은 향후 제 2 단계, 3 단계 연구개발을 통하여 전력 및 성능 측면에서 최적화되고 동작의 안정성을 확보할 수 있도록 계속 개선을 추진할 계획이다.

우선 2 단계 연구개발 기간에는 32 비트급 저전력 프로세서 기술을 필요로 하는 수요 기업을 적극 발굴하여 이들 기업과 적극적인 협동연구를 통하여 연구 개발 시에 업체의 요구사항을 적극 반영할 수 있도록 함으로서 기술개발 결과의 완성도를 높일 수 있도록 노력할 계획이다. 이를 통하여 개발한 최적화된 32 비트 초저전력 비동기 프로세서 IP 를 수요 기업이나 Fabless 반도체 설계 업체 등에 적극 이전할 수 있도록 할 계획이다. 이를 위하여 관련 기업과 함께 최적화된 32 비트 저전력 프로세서 코어를 활용할 수 있는 SoC 설계 응용분야를 적극적으로 발굴하여 이를 적용할 수 있는 방안을 모색 함으로서 본 연구개발을 통하여 확보한 검증되고 최적화된 초저전력 비동기 프로세서 코어 IP 를 다양한 저전력 응용 분야에 확산시킬 수 있도록 할 계획이다.

3 단계 연구개발 기간에는 앞서 확보한 1 단계, 2 단계 연구개발 과정의 결과물을 활용하여 이를 관련 기업에 적극 기술이전 하여 상용화할 수 있도록 지원할 계획이다. 이를 위하여 그간 연구개발 과정을 통하여 확보한 32 비트 비동기 프로세서 코어의 안정적인 동작을 보장할 수 있도록 다양한 방법을 적용하여

검증하고, 저전력 비동기 프로세서가 적용될 수 있는 분야를 적극 발굴하여 이를 적용할 수 있도록 함으로서, 이 기술에 대한 안정적인 수요 기반을 확보할 계획이다.

본 연구개발을 통하여 확보한 초저전력 32 비트급 프로세서와 비동기 저전력 설계 기술은 향후 다양한 저전력 기반 제품의 개발시에 활용할 수 있고, 그간 외국 주요 기업들에 의해 주도되어 온 프로세서 기술 분야에서 경쟁력을 확보하는데 기여할 수 있을 것으로 기대된다. 또한 그간 대부분을 외국의 프로세서 기술을 도입하여 사용하던 것을 일부 대체할 수 있을 것으로 기대되며, 이를 통하여 국내의 관련 산업의 발전과 활성화에 크게 기여할 수 있을 것으로 전망된다.

## 제 6 장 연구개발과정에서 수집한 해외과학기술정보

본 사업은 소규모의 예산과 인력을 이용하여 중장기적으로 원천 요소기술에 대한 연구개발을 수행하고자 하는 목적에서 시작된 전문연구사업으로서, 연구개발 수행 시에 예산 및 인력 면에서 여러 가지 제약이 따른다. 또한, 이러한 사업상의 제약 요소뿐만 아니라 우리나라와 같이 비동기 설계 분야나 저전력 프로세서 개발 관련 저변이 폭넓지 않고 지원이 활성화되지 않은 경우에는 이러한 연구개발의 수행하는데 있어 어려움이 더욱 크다. 당 부서에서 2007 년부터 2008 년까지 본 사업의 1 단계 연구개발을 수행함에 있어서 전문연구사업이 내포하고 있는 이러한 한계를 극복하기 위하여 다각적인 노력을 기울였다.

당 부서에서는 이러한 전문연구사업이라는 기본 틀 안에서 최대의 연구개발 효과를 달성하기 위하여 국내 및 국외의 기업 및 전문연구 기관과의 교류를 확대하고자 하였다. 연구개발 제 1 단계 기간 동안에 당 부서에서는 국내의 프로세서 관련 기업과 비동기 설계 관련 대학들과 함께 프로세서 개발환경 구축 및 비동기 설계 관련 요소기술들을 함께 개발을 추진하였다. 또한, 영국의 New Castle 대학과 6 개월 정도의 교류를 통하여 비동기 설계 및 합성기술에 대하여 연구하였으며, 네덜란드에 위치한 비동기 설계 관련 전문기업인 Handshake solutions 사와는 비동기 설계 기반 IP 개발을 위한 공동연구 수행 및 비동기 설계에 필요한 설계환경(CAD)을 구축하였다.

당 부서에서는 1 단계 연구개발 기간 동안 비동기 프로세서 설계를 주제로 영국 소재 뉴캐슬 대학의 Asynchronous Circuit Lab.에서 TI 사의 MSP430 을 TiDE 를 이용해 모델링하고 검증 환경을 구축하였다. 뉴캐슬 대학의 Asynchronous circuit lab.은 비동기 VLSI 시스템 설계, Petri Nets 과 concurrency model 연구, Hardware Description Languages 연구, VLSI 를 위한 CAD Systems 연구, Fault Tolerance, Reliability 연구 등을 수행하고 있는 영국 소재의 유수의 대학으로서 비동기 설계 관련 분야에서 상당한 실적과 기술력을 보유하고 있으며, 연구 책임자는 Alex Yakovlev 교수이다. 뉴캐슬

대학과의 교류를 통하여 현재 영국의 비동기 설계 및 합성 기술관련 연구개발 동향을 파악할 수 있었다.

네덜란드에 위치한 Handshake Solutions 사는 (주)Phillips 에서 spin-off 한 비동기 설계 관련 전문기업으로서 최초로 비동기 설계 툴(CAD)인 TiDE 를 상용화하였고, 비동기 설계 기반 32 비트 프로세서를 개발하는 등 비동기 설계 분야에서 높은 기술력을 가진 전문 기업이다. 당 부서에서는 1 단계 연구개발 기간 동안에 이 기업과 수 차례의 미팅을 통하여 관계를 구축하였고, 1 단계 2 차년도 연구개발 기간 동안에는 비동기 IP 기반 저전력 응용 IP 개발을 위한 공동연구를 수행하였다. 이를 통하여 현재 비동기 설계 기술 및 관련 제품의 전 세계적인 시장동향 및 기술의 발전 방향에 대하여 파악할 수 있었고, 특히 비동기 IP 기반 응용 기술의 응용분야 및 관련된 Handshake Solutions 사의 연구개발 수행 현황에 대하여 파악할 수 있었다. 현재 Handshake Solutions 사에서는 이 회사가 보유하고 있는 비동기 설계 기술 및 관련 IP 를 활용하여 Smart Card, pager 등에 성공적으로 적용하여 상용화하고 있고, 향후 Automotive In-Vehicle Networking(IVN), Embedded microcontroller, Tire Pressure Monitoring System(TPMS), Bluetooth radio 등에 적용하기 위한 연구개발을 수행 중에 있다.

이러한 대내외 관련 기업 및 연구기관과의 협력연구를 통하여 당 부서에서는 비동기 저전력 프로세서 설계에 필요한 주변기술을 확보하고 이를 비동기 설계 기반 저전력 프로세서 설계에 적용 함으로서 1 단계 연구개발을 성공적으로 수행할 수 있었고, 이들과의 적극적인 관계 구축을 통하여 상호 부족한 점을 보완하면서 장점을 공유 함으로서 향후 시너지를 창출 할 수 있는 기반을 마련하였다. 이러한 연구개발 응용분야에 대한 정보는 향후 본 연구개발의 성과를 상용화하고 적용하는데 있어 아주 유용하게 활용할 수 있을 것으로 기대된다.

## 제 7 장 참고문헌

- [1] 오명훈 외, “비동기 회로 설계 요소기술 분석서,” 한국전자통신연구원 기술문서, TM20070357, 2007
- [2] 김영우, 김성운, “비동기 설계의 기술 및 특허 동향,” 주간기술동향, 제 1301 호, 2007
- [3] 오명훈 외, “초저전력 비동기 프로세서 기능 및 성능규격 정의서 V1.0,” 한국전자통신연구원 기술문서, TM200800557, 2008
- [4] 김영우 외, “명령어셋(ISA) 정의서,” 한국전자통신연구원 기술문서, TM200707713, 2007
- [5] 오명훈 외, “ALTHEA 명령어 마이크로코드 상세분석서,” 한국전자통신연구원 기술문서, TM200805998, 2008
- [6] 김성남 외, “초저전력 비동기 프로세서 기능 및 마이크로아키텍처 설계서 V1.0,” 한국전자통신연구원 기술문서, TM200800606, 2008
- [7] 김영우 외, “초저전력 비동기 프로세서 기능 시험계획서,” 한국전자통신연구원 기술문서, TM200806830, 2008
- [8] 김영우 외, “초저전력 비동기 프로세서 기능 시험절차 및 결과서,” 한국전자통신연구원 기술문서, TM200806836, 2008
- [9] Instruction Set Reference Manual for AE32000: An 32bit EISC microprocessor, ADChips Inc., 2008
- [10] AE32000C(Lucifer): Hardware Reference Manual, ADChips Inc., 2007
- [11] ETRI Platform Specification: Lucida Base, ADChips Inc., 2008
- [12] EISC Software Developer Guide: Extendable Instruction Set Computer, ADChips Inc., 2008



- [13] [mibench] <http://www.eecs.umich.edu/mibench/>
- [14] [dhry] R. Weicker and S. Nixdorf. Dhystone, CACM, Vol. 27, No. 10, Oct. 1984
- [15] [seloco] <http://www.seloco.com/>
- [16] [unitron] <http://unitrontech.co.kr/>
- [17] Handshake Solutions Home Page, <http://www.handshakesolutions.com/>
- [18] Silistix Home Page, <http://www.silistix.com/>
- [19] Elastix Home Page, <http://www.elastix-corp.com/>
- [20] Fulcrum Home Page, <http://www.fulcrummicro.com/>
- [21] Achronix Home Page, <http://www.achronix.com/>

## 주 의

1. 이 보고서는 전문연구사업의 연구보고서입니다.
2. 이 보고서 내용을 발표할 때에는 반드시 전문연구사업의 연구결과임을 밝혀야 합니다.
3. 국가과학기술 기밀유지에 필요한 내용은 대외적으로 발표 또는 공개하여서는 아니됩니다.