

# SPARC V8 구조 CPU칩의 VHDL모델의 분석과 RTL 합성을 위한 코드 변환

도경선\*, 김남우\*, 허창우\*

\* : 목원대학교 대학원 전자 및 컴퓨터공학과

E-mail : halifax0@hanmail.net,

gotree94@mokwon.ac.kr,

chang@mokwon.ac.kr

## 요 약

기존의 범용시스템과 대별되는 임베디드 시스템의 수요가 급증하면서 하드웨어부분의 중심 축인 임베디드 프로세서에 대한 관심이 하루가 다르게 커지고 있다. 또한 사용자들이 작고 간편하면서도 기존의 범용시스템과 같은 기능들을 가지는 높은 수준의 성능을 요구하게 됨으로서 한 칩 안에 여러 가지 기능을 함께 구현하거나 시스템을 집적하는 시스템 칩의 상품화가 이루어지고 있는 추세이다. 날로 경쟁이 치열해져 가는 비메모리 설계 분야에서 누가 더욱 우수한 반도체 관련 IP를 확보하느냐가 승패의 관건이 될 것은 당연한 일이 되었다. 본 논문에서는 기존에 성능이 검증된 SPARC 아키텍처 V8을 근간으로 한 VHDL모델을 분석하고, 시뮬레이션을 통하여 그 기능을 검증하였으며, Synopsys FC2(FPGA Compiler 2)를 이용하여 로직 합성하였으며, 그 결과를 Xilinx VIRTEX 3000 FPGA를 이용하여 구현하였다.

## I. 서론

RISC(Reduced Instruction Set Computer) 마이크로프로세서는 지난 20여 년 간 마이크로 프로그램기법으로 설계되어 왔던 기존의 CISC(Complex Instruction Set Computer) 마이크로프로세서의 개념을 벗어나 고성능 컴퓨터의 마이크로프로세서로 각광을 받고 있으며 최근에는 컴퓨터뿐만 아니라 영상, 통신 시스템, 산업자동화 분야 등에서도 고성능 마이크로프로세서를 요구하고 있어서 RISC 마이크로 프로세서의 응용분야는 급격히 확대되고 있다. 뿐만 아니라 CISC보다 단순한 구조로 인하여 면적을 줄일 수 있어 다른 칩의 기능을 내장하고자 할 때 유리하며, 지금현재 시장이 급속히 성장하고 있는 이동통신에서부터 가전에 이르는 임베디드 분야에서 두각을 나타내고 있다.[1][2][3][4] 본 논문에서는 기존에 성능이 검증된 SPARC 아키텍처 V8을 근간으로 한 VHDL모델을 분석하고, 시뮬레이션을 통하여 그 기능을 검증하였으며, Synopsys FC2(FPGA Compiler 2)를 이용하여 로직 합성하였다. 그 결과를 Xilinx VIRTEX 3000 FPGA를 이용하여 구현하였다.

## II. SPARC V8의 특징

다음에서 사용된 모델은 SPARC V8 구조를 기반으로 한 32비트의 프로세서를 구현하였다. 분기 명령과 기억장소, 인터럽트 제어장치, 2개의 24비트 타이머, 2개의 UART, power-down 기능, 워치독, 16비트 I/O 포트와 유연한 기억 제어장치 그 위에 모델들은 on-chip AMBA AHB/APB 버스를 사용하여 쉽게 추가할 수 있도록 되어있다. 필요로 하는 기능의 추가를 위해 IP코어 source code를 추가 시켜 system-on-chip을 설계 하는데 사용될 수 있도록 되어 있다. VHDL 모델은 GNU Public License (GPL)와 Lesser GNU Public License (LGPL). 2개의 license의 적용을 받는다.

[1][2][3]

### 1. 기능적 특징[1][2][3][6][7]

블럭다이어그램은 그림 1과 같다.

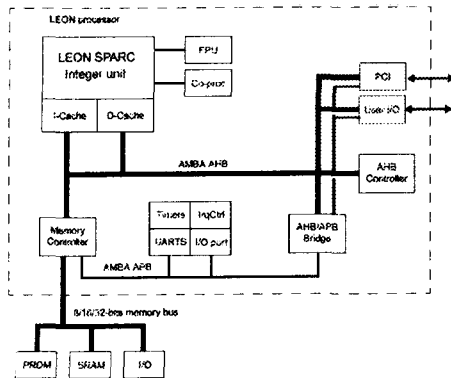


그림 1 CPU 블럭다이어그램

• **Integer unit**

IU(integer unit)는 SPARC V8 표준을 사용되고 있으며 ERC32와 완벽하게 호환이 가능하다. 윈도우 레지스터의 개수는 SPARC 표준인 2개에서 32개 내에서 사용가능하며 기본 값은 8개이다.

• **Floating-point unit and co-processor**

FPU(Floating-Point Unit)는 내장되어있지 않고, Meiko FPU코어와 직접 인터페이스 하거나 다른 FPU의 일반적인 인터페이스가 가능하다.

• **Cache sub-system**

각각 1에서 64 킬로 바이트로 구성가능한 명령과 데이터 캐시가 분리되어 라인 당 8-32 byte 크기로 제공된다. Sub-blocking은 32-bit word에 대하여 하나의 유효 bit로 구현된다.

• **Memory interface**

메모리 인터페이스는 직접 인터페이스 PROM, SRAM 그리고 메모리 맵 I/O 장치들을 제공한다. 메모리 영역은 8,16,또는 32bit의 데이터 길이로 프로그램 할 수 있다.

• **Timer**

두개의 24비트 타이머가 칩에 내장되어 있다. 타이머들은 주기적인 또는 one-shot 모드로 동작한다. 타이머 모두 공용 10비트 프리스케일러에 의해 클럭이 인가된다.

• **Watch Dog**

각각 1에서 64 킬로 바이트로 구성가능한 명령과 데이터 캐시가 분리되어 라인 당 8-32 byte 크기로 제공된다. Sub-blocking은 32-bit word에 대하여 하나의 유효 bit로 구현된다.

• **UART**

두 개의 8-bit UART는 칩 내부에 제공된다. 보드레이트는 개별적으로 프로그램 가능하며, 데이터는 한 개의 stop bit를 가지는 8bit frames 단위로 보내어진다.

• **Interrupt controller**

인터럽트 제어장치는 내, 외부의 소스로부터 총 15개의 인터럽트들을 관리한다.

• **Parallel I/O port**

16비트 병렬 입출력 포트를 지원한다. 각각의 비트는 입력 또는 출력으로 프로그램 할 수 있다. 일부 비트의 경우 예를 들어 UART 입출력과 외부 인터럽트 입력과 같이 두 가지다 사용되는 경우도 있다.

• **AMBA on-chip bus**

추가로 쉽고 AMBA를 사용하는 다른 프로세서들 안의 AMBA AHB/APB 인터페이스를 만들 수 있다. 프로세서는 AMBA AHB와 APB 온 칩 버스의 모든 기능을 구현할 수 있다. 유연한 배열 구조는 새로운 IP 코어들을 추가하기 쉽게 해준다. 또한, 주변 장치들은 AMBA AHB/APB 인터페이스의

• **Boot loader**

온칩 부트 로더는 선택적으로 외부에 boot 프로그램 없이 boot 프로세서와 다운로드 어플리케이션을 통해 부트 할 수 있도록 추가적인 기능을 지원해 준다. 이 특징은 대부분의 FPGA를 사용하는데 있어서 매우 유용하다.

2. 모델 체계

VHDL 모델 체계는 아래 테이블과 같다.

표 1 VHDL 모델 체계

Entity/Package	File name	Function
LEON	leono.svh	LEON top level entity
LEON_PCT	leoni_pci.vhd	LEON PCI top level entity
LEONMCORE	corent.vhd	Main core
LEONMCORECLKGEN	clkgen.vhd	Clock generator
LEONMCORERSTGEN	rstgen.vhd	Reset generator
LEONMCOREAHBAB	ahbab.vhd	AMBA AHB controller
LEONMCOREAPBMS	apbms.vhd	AMBA APB controller
LEONMCOREMCTR	mctr.vhd	Memory controller
LEONMCOREPROC	proc.vhd	Processor core
LEONMCOREPRIMCACHE	cache.vhd	Cache module
LEONMCOREPROCACHECACHEMEM	cachemem.vhd	Cache ram
LEONMCOREPROCACHEDCACHE	dcache.vhd	Data cache controller
LEONMCOREPROCACHEICACHE	icache.vhd	Instruction cache controller
LEONMCOREPROCACHEACACHE	acache.vhd	AHB cache interface module
LEONMCOREPROCPU	pu.vhd	Processor manager unit
LEONMCOREPROCFPU	fpicu.vhd	parallel FPU interface
LEONMCOREPROCFFILE	regfile.vhd	Processor register file
LEONMCOREPROCCTRL	regctrl.vhd	Interrupt controller
LEONMCOREIOPORT	ioport.vhd	Parallel I/O port
LEONMCORETIMERS	timers.vhd	Timers and watchdog
LEONMCOREUART	uart.vhd	UARTs
LEONMCORECONF	conf.vhd	LEON configuration register
LEONMCOREAHBSTAT	ahbstat.vhd	AHB status register

다음은 모델에 사용된 패키지들이다.

표 2 모델 패키지

Package	File name	Function
TARGET	target.vhd	Pre-defined configurations for various targets
DEVICE	device.vhd	Current configuration
CONFIG	config.vhd	Generation of various constants for processor and caches
SPARC_V8	sparc.vhd	SPARC_V8 specific definitions
IFACL	iface.vhd	Type declarations for module interface signals
MACRO	macro.vhd	Various utility functions
AMBA	amba.vhd	Type definitions for the AMBA buses
AMBA_OMP	ambacomp.vhd	AMBA component declarations
PADLIB	padlib.vhd	Pad generators
RAMLIB	ramlib.vhd	Register file and cache data generators
FPU_LIB	fpu.vhd	FPU interface package
DEBUG	debug.vhd	Debug package with SPARC disassembler
TECH_GENERIC	tech_generic.vhd	Generates regfile and pad models
TECH_VIRTEX	tech_virtex.vhd	Xilinx virtex specific regfile and ram generators
TECH_ATL35	tech_atl35.vhd	Atmel AT35 specific regfile, ram and pad generators
TECH_SYMPHONY	tech_sympathy.vhd	Synphony specific regfile and ram generators
TECH_LEONARDO	tech_leonardo.vhd	Leonardo specific regfile and ram generators

Synthesis는 750MHz(서) Pentium-III PC(256 MB RAM 환경)에서 약 35분 정도 걸린다. 아래 테이블은 synthesis에서 얻은 결과 일부를 보인 것이다. ( post-layout 타이밍 ):

표 4 post-layout 타이밍 결과표

Icache (Kbyte)	Dcache (Kbyte)	Regfile Implement.	Device	Freq (MHz)	Area
2	2	EAB	EPF10K200E-1	20	3,800 LC
8	4	blockRam	XCV300E-8	45	5,000 LUT
8	8	RAM16X1	XCV300E-8	48	6,300 LUT

FC2/FE는 ram cell 을 자동적으로 지원하지 않는다, ram은 목표 library로부터 직접 설명 할 수 있다. 현재, Xilinx VIRTEX 기술은 TECH\_VIRTEX package에서만 지원된다.

### III 시뮬레이션 및 Synthesis.

#### 1 시뮬레이션.

테스트벤치는 두 부분으로 나뉘어 작성되었는데, 일반적인 설정을 위해 구성과 다양한 메모리 크기에 의한 시스템의 모델을 생성하는 것과 모델의 동작을 검사하기 위한 부분으로 나뉘어 있고, 시뮬레이션으로부터의 결과는 다음과 같다

```
# # *** Starting LEON syntehtm test ***
# # Memory interface test
# # Register file
# # Interrupt controller
# # Timers, watchdog and power-down
# # Parallel I/O port
# # UARTs
# # Test completed OK,halting with failure
# # ** Failure : TEST COMPLETED OK,
ending with FAILURE
```

#### 2. Synthesis

Synthesis 과정을 단순화하기 위해, TARGET 패키지에 미리 정의해서 사용하는 구조로 설정된 사항에 맞추어 DEVICE 패키지에서 사용된다. 다음의 테이블은 여러 조합 구성을 보인 것이다:

표 3 TARGET 설정 조합 표

Configuration	cache	regfile	ram	pad	device	system
sympathy_2k2k	inferred	inferred	none	inferred	any	sympathy
sympathy_2k2k_ambacomp	inferred	inferred	inferred	inferred	any	sympathy
sympathy_2k2k_virtex300e	inferred	inferred	instance	inferred	virtex	leonardo
leonardo_2k2k	inferred	inferred	none	inferred	any	leonardo
leonardo_2k2k_ambacomp	inferred	inferred	inferred	inferred	any	leonardo
gen_virtex_2k2k	instance	instance	none	inferred	virtex	any
gen_virtex_2k2k_fpu	instance	instance	inferred	inferred	virtex	any
gen_virtex_2k2k_synphony	instance	instance	instance	inferred	any	any
gen_atl35	instance	instance	none	instance	ATL_35	any

### IV. 결론 및 향후 계획

이상에서 살펴본 것과 같이 SPARC V8 구조를 기반으로 하여 여러 가지 기술과 target에 적용될 수 있는 32비트 마이크로 프로세서를 구현하였다.

Xilinx사의 VIRTEX3000-PQ240-4 타입의 FPGA로 구현하였으며, 그림 2와 같은 FPGA의 place/routing 결과를 얻었다.

본 연구는 구현된 결과를 이용하여 내장형 마이크로 프로세서를 만드는데 기초 연구가 되었다는 점에서 의의를 가진다. 무엇보다 여러 가지 기술과 target에 적용이 가능하므로 사용하고자 하는 목적과 개발 환경에 바로 적용할 수 있는 장점이 있다. 또한 칩 면적이 작아서 다른 칩을 구현할 때나 한 칩에 여러 가지 기능을 내장하는 칩의 구현이 가능하고, 전력 소모가 작은 RISC형 마이크로 프로세서의 구조적 장점을 바탕으로 하기 때문에 통신 및 시스템 주변의 기능들을 함께 집적시킴으로서 시스템 온 칩(System on Chip) 형태의 마이크로 프로세서를 구현 할 수 있다.

구현된 LEON CPU는 Linux기반의 개발 틀을 지원하여 최근 각광을 받고 있는 실시간 OS(Real-time Operating System)를 적용한 임베디드 시스템을 구성할 수 있다.

향후 현재 널리 쓰이고 있는 마이크로 컨트롤러나 DSP칩을 대체하여 셋-탑-박스(set-top-box)나 소형 휴대용 통신단말기, PDA등의 통신 및 멀티미디어 등의 응용 분야에 넓게 이용 가능할 것으로 전망된다.

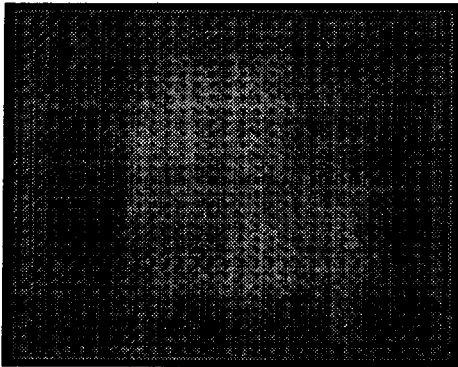


그림 2 Xilinx FPGA Place and routing 결과  
(VIRTEX 3000-pq240)

## V. 참고문헌

- [1] Gaisler Research, <http://www.gaisler.com>
- [2] LEON centre, <http://www.leoncenter.com>
- [3] LEON SPARC user group ,  
[http://groups.yahoo.com/group/leon\\_sparc](http://groups.yahoo.com/group/leon_sparc)
- [4] 32비트 SPARK RISC controller의 특징 및 응용, 전자공학회지 제20권 제 11호 p1180-1189, 1993.
- [5] 이희, 차세대 임베디드 마이크로프로세서 디자인, Electronic Systems, p66-76, 2001. 3.
- [6] 김진천, SPARC 811 CPU칩 Verilog모델의 분석과 시뮬레이션과 로직 합성, 포항공대, 1995.
- [7] Sun Micro system, The SPARC Architecture Manual Version 8, 1991.