

Advanced Microcontroller Bus Architecture 기반의 MCU 설계 가이드라인⁺

(MCU Development Guideline based on Advanced Microcontroller
Bus Architecture)

노 찬 휘¹⁾, 오 연 상²⁾, 백 돈 규^{3)*}
(Chanhwi Roh, Yeonsang Oh, and Donkyu Baek)

요 약 마이크로 컨트롤러(MCU)는 Advanced Microcontroller Bus Architecture (AMBA)에 다양한 모듈을 연결시켜 프로그래밍을 통해 각 모듈들을 적절히 활용하도록 설계되어 있다. 범용 MCU는 많이 사용되는 모듈들을 미리 설계 및 제작하여 소비자가 적절히 사용하도록 제작되어 판매되고 있다. 하지만, 특정 산업군에서는 MCU를 직접 설계하여 분야에 적합한 모듈 종류 및 수량을 사용자가 희망하는대로 설계하여 활용할 필요가 있다. 하지만, 대부분의 소비자는 MCU를 직접 설계하기에는 상당한 진입장벽이 있다. 본 연구에서는 MCU 교육 및 연구단계에서 손쉽게 MCU를 설계할 수 있는 개발 가이드라인을 제공한다. 먼저, AMBA를 활용하기 위해 요구되는 필수 모듈을 소개하고, 해당 모듈이 AMBA 및 인터럽트 동작을 통해 제대로 동작하는지 검증하고, 최종적으로는 온칩으로 설계하여 사용될 수 있도록 방법론을 제공한다.

핵심주제어: Advance Microcontroller Bus Architecture, MCU, ASIC, Interrupt

Abstract Microcontroller (MCU) is designed to properly utilize each module through programming by connecting various modules to Advanced Microcontroller Bus Architecture (AMBA). General-purpose MCUs are designed for consumers to use them appropriately in their research or industry area. However, in a specific area such as networking and AI autonomous vehicles, it is necessary to design MCU suitable for the field directly. However, there is a significant barrier for most consumers to directly design an MCU. In this paper, we provide a development guideline that can easily design an MCU for education or research purpose. First, we introduce AMBA system with open IPs, and we verify that the module operates properly through AMBA and interrupt operation. Finally, the MCU system is designed as an on-chip.

Keywords: Advance Microcontroller Bus Architecture, MCU, ASIC, Interrupt

* Corresponding Author: donkyu@cbnu.ac.kr

+ 이 논문은 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. 2020R1A6A1A12047945)
Manuscript received November 02, 2022 / revised
December 21, 2022 / accepted December 22, 2022

1) 충북대학교 전자공학부, 제1저자
2) 충북대학교 전자공학부, 제2저자
3) 충북대학교 전자공학부, 교신저자

1. 서 론

에너지경제연구원(2021)의 세계 에너지시장 인사이트에 의하면 다양한 사물에서 데이터 수집/처리/전송이 가능한 Micro Controller Unit (MCU)의 수요가 폭발적으로 증가하고 있다. 예

를 들어, 전기차의 수요는 연간 약 30% 씩 증가하는 추세를 보이는데, 이러한 전기차 한 대당 약 200~300개의 MCU가 필요하며, 앞으로 자율주행을 위해 추가로 요구되는 MCU 수는 더욱 증가하고 있다.

시중에 판매하는 범용 MCU는 소비자가 많이 활용하는 모듈들(통신 모듈, 연산 모듈 등)을 미리 설계 및 제작하여 분야별로 판매하고 있다. 사용자는 판매되는 MCU를 프로그래밍하여 직접 활용하게 모듈들을 활용한다. 하지만, MCU의 사용범위가 확대되고, 각 활용되는 MCU 별로 필요로 하는 모듈의 수량 및 종류가 다양해지면 분야에 따라서 판매용 MCU 대신에 직접 사용 목적에 적합한 MCU를 설계할 필요성이 생긴다.

엣지컴퓨팅의 등장으로 인해 각 노드들이 기존엔 정보수집/처리/전송을 각각 담당했다면 앞으로는 모두 수행하게되면서 필요로 하는 통신 모듈과 연산 모듈이 많아지게 되었다. 특히, 데이터 처리량 및 전송량이 많아지는 경우에는 효과적으로 데이터를 처리하거나 우회하여 전달하는 알고리즘이 적용됨에 따라 그에 따른 MCU가 요구된다. 차량용 반도체의 경우, 기존 IoT 통신 규격과는 다르게 차량 정보는 CAN 통신을 주로 활용하고, 센싱/이미지 정보는 기존의 I2C, SPI 통신을 활용하고, 서버와는 UART 통신을 복합적으로 활용함에 따라 다양한 통신 모듈을 요구하게 되었다.

하지만, MCU를 직접 설계하기 위해서는 디지털 회로설계, 동작 검증, 온칩 설계에 이르기까지 상당히 많은 전문 과정이 요구된다. 기존 연구는 다양한 BUS 구조를 제안하거나 BUS에 연동되는 다양한 모듈을 제안하는 등 전문적인 내용을 다룬다. 하지만, 설계 초심자가 본인이 희망하는 시스템을 MCU 설계를 통해 구축하려면 어떤 모듈이 필수적이고, 어떻게 기능검증을 할 것인지 알기가 어렵다. 따라서, 본 논문에서는 이러한 문제를 해결하고자, 산업체 및 교육기관에서 MCU를 활용한 교육/실무/연구 목적으로 손쉽게 MCU를 설계 및 제조할 수 있는 개발 가이드라인을 제공한다. 먼저, MCU를 구동하기 위해 필수적인 모듈을 AMBA에 연결하고, 해당 모듈이 제대로 동작하는지 검증하고,

최종적으로는 온칩으로 설계하여 사용할 수 있도록 방법을 제시한다.

2장에서는 본 논문에서 제시하는 개발 프레임워크를 구성하는 MCU 라이브러리 및 구조를 소개한다. 3장에서는 해당 MCU를 설계 및 검증하는 프로세스를 소개하며, 4장에서는 검증이 완료된 회로를 온칩으로 설계하는 내용을 담고 있다. 전체 연구내용은 5장에서 정리되었다.

2. 시스템 구성

본 논문에서는 MCU를 구성하기 위해 일반에게 공개된 intellectual property (IP, 지적재산권) 중에 FPGA를 활용한 성능 검증 및 온칩 설계까지 가능한 GRLIB을 활용하였다(Gaisler). GRLIB 내 BUS 시스템인 Advanced Microcontroller Bus Architecture (AMBA) 위에 MCU 필수 기능을 구현하였다. 각 핵심 IP들이 올바르게 동작하는지 검증하기 위해 UART 통신으로 PC와 데이터를 주고받아 모니터로 올바르게 인터럽트가 발생하는지 검증하였다. 이때, GRLIB 전용 Debug 툴을 사용하여 AMBA를 직접 컨트롤함으로써 검증하였다.

2.1 AMBA 기반 MCU 구조

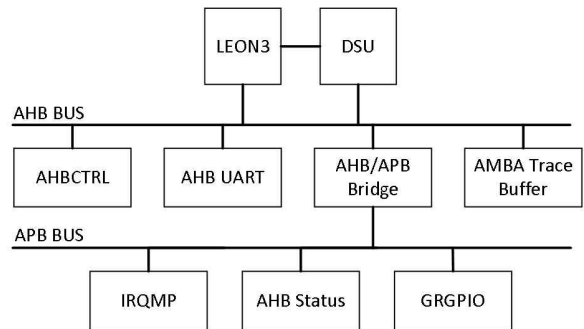


Fig. 1 Control System Diagram.

Fig. 1은 AMBA 기반의 필수 MCU 구조를 보여준다. Core (LEON3, DSU)와 Bus 모듈 (AHBUART, AHB/APB Bridge, AMBA Trace Buffer, IRQMP, AHB Status, GRGPIO)로 구성

되어있다. LEON3, DSU, AHB/APB Bridge, AMBA Trace Buffer는 AHB Bus에 연결되어 있으며, 나머지 모듈들은 APB Bus에 연결되어 있다. AHB BUS의 경우 고속 동작을 하기 위한 BUS로써 한 번에 많은 데이터를 처리하는데 장점이 있다. APB Bus의 경우 비교적 느린 속도의 주변장치를 제어하는데 쓰인다. 속도가 다른 모듈을 사용하게 될 때 속도가 맞지 않아 효율적인 동작을 할 수 없어 분리하여 통신한다.

2.2 Core 모듈

1) LEON3

LEON3는 SPARC V8 아키텍처를 준수하는 32bit 프로세서의 합성 가능한 VHDL 모델이다. 이 모델은 다양한 FPGA board에 합성할 수 있고 여러 모듈을 쉽게 연결할 수 있어서, 특히 SoC 설계에 적합하다. 7단계 파이프라인, IEEE-754 FPU 그리고 256 KB 캐쉬 등을 지원하며 AMBA 2.0 Bus에 접속될 수 있다. LEON3와 함께 제공되는 DSU를 AMBA Bus를 통하여 접근하면 LEON3의 동작을 제어하거나 동작 상태를 파악할 수 있으며, 이를 이용하여 LEON3를 기반으로 동작하는 임베디드 시스템의 하드웨어와 소프트웨어를 개발하거나 디버깅할 수 있는 환경을 갖출 수 있다.

2) DSU

DSU(Debug Support Unit)는 Debug 모드 동안 프로세서를 제어하는데 사용된다. DSU는 AHB Slave 역할을 하며 모든 AHB Master가 여러 종류의 UART, JTAG, PCI, USB 등 직렬통신을 통하여 액세스할 수 있다. 또한, 멀티프로세서 시스템을 지원하며 최대 16개의 프로세서를 처리할 수 있다. AHB Slave 인터페이스를 통해, 모든 AHB Master는 프로세서 레지스터와 명령어 추적 버퍼의 내용에 접근할 수 있다. DSU 제어 레지스터는 언제든지 접근할 수 있지만, 프로세서는 Debug 모드로 들어갔을 때만 접근할 수 있다. Debug 모드에서는 프로세서 파이프라인이 유지되고 DSU가 프로세서 상태에 액세스할 수

있다.

2.3 BUS 모듈

1) AHBCTRL

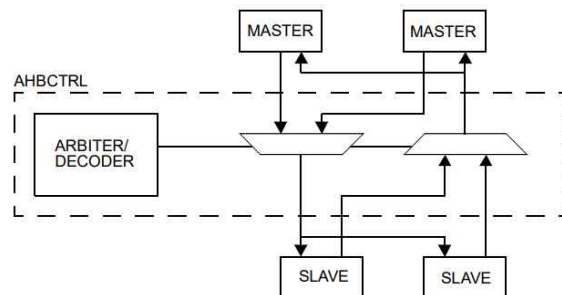


Fig. 2 AHBCTRL Block Diagram.

AHBCTRL은 Fig. 2와 같이 ARBITER, BUS 멀티플렉서, DECODER로 구성되어 있다. AHBCTRL은 Fixed-priority와 Round-priority 방식으로 ARBITER를 동작시킬 수 있다. Fixed-priority 방식일 때는 Master의 Bus index에 따라 우선순위가 결정된다. 요청이 없을 때는 우선순위가 가장 낮은 Bus index가 0인 Master가 동작하게 된다. Round-priority 방식일 때는 한 번의 통신 이후에 다음 우선순위를 갖는 Master의 동작을 수행하게 한다. 요청이 없을 때는, 가장 마지막에 동작한 Master가 동작하게 된다.

2) AHBUART

UART 통신을 하기 위한 Master 모듈로써 AHB BUS에 연결되어있다. Access parameter와 데이터를 전송하기 위해 사용하였다. UART 통신을 통해 AHB BUS에 레지스터 주소를 Read/Write 동작을 할 수 있다.

3) AHBTRACE

AHBTRACE는 Fig. 3과 같이 구성되어 있다. AHB BUS의 주소, 데이터 및 다양한 제어 신호가 저장되고 나중에 분석을 위해 사용된다. 제어 신호를 저장한 버퍼의 주소는 버퍼 인덱스 레지

스터에 유지되고, 다음 신호는 자동으로 그다음 주소에 저장되게 된다. AHBTRACE를 활성화하는 EN 신호가 Low가 되거나, breakpoint에 도달하면 동작이 멈추게 되며, breakpoint에 도달하게 되면 Interrupt를 발생시킨다.

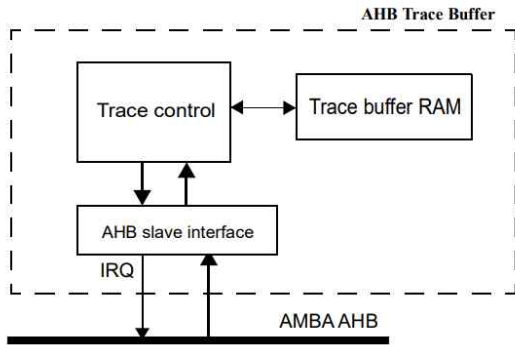


Fig. 3 AHBTRACE Block Diagram.

4) AHB/APB Bridge

AHB/APB Bridge는 AMBA 2.0 표준에 따른 APB Bus Master이며, 최대 16개의 Slave를 제어할 수 있다.

5) IRQMP

GRLIB의 AMBA 시스템은 Interrupt 라인이 나머지 AHB/APB Bus 신호와 함께 연결되어 Interrupt Bus를 형성하는 Interrupt 방식을 사용하고 있다. 멀티프로세서 Interrupt 컨트롤러 코어는 APB 슬레이브로 AMBA Bus에 부착되어 결합한 Interrupt 신호를 모니터링한다. Interrupt Bus에서 생성된 Interrupt는 모두 컨트롤러로 전달된다. 컨트롤러는 Interrupt의 우선순위를 매기고, 가장 높은 우선순위로 프로세서에 전달한다.

6) AHB Status

AHB Status는 AMBA AHB 액세스에 대한 정보를 저장하여 오류 응답을 트리거한다. 실패한 AMBA 버스 트랜잭션의 제어 및 주소 신호 값을 캡처하는 상태 레지스터와 다른 주변기기에서 보내는 수정 가능한 오류 신호를 저장하는 오류 주소 레지스터로 구성되어 있다.

3. 동작 검증

3.1 동작 검증 환경

설계한 MCU가 정상적으로 동작을 하는지 확인하기 위해 FPGA를 사용하여 검증하였다. FPGA 보드는 여러 통신 기능을 제공하여 이후에 확장하여 설계하기 용이한 Xilinx사의 Spartan-6 FPGA SP605 Evaluation Kit를 사용하였다(Xilinx Inc).



Fig. 4 Spartan-6 SP605 Evaluation Kit.

3.2 Interrupt 및 UART 통신을 활용한 동작 검증

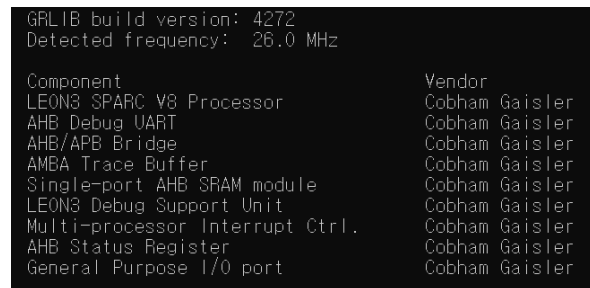


Fig. 5 GRMON Debug Mode.

설계된 MCU의 동작을 확인하기 위해 GRMON을 사용하였다. GRMON은 Debug 모니터로 LEON 프로세서를 기반으로 하는 Debug 툴이다. 해당 툴은 AHBUART와 상호동작하며 이를 통해 AMBA에 신호를 주어 레지스터에 저장된 데이터를 확인하여 동작을 검증한다.

Fig. 5와 같이 GRMON을 사용하면 AMBA에 어떤 모듈들이 구성되어 있는지 확인할 수 있다.

동작을 검증하기 위해 프로세서에 Interrupt 발생 시 동작하는 Pending Register를 GRGPIO와 연결되어있는 Switch를 통하여 확인하였다.

Interrupt를 발생시키는 조건을 컨트롤하는 Interrupt Mask Register, Interrupt Polarity Register, Interrupt Edge Register에 값을 Write 하여 Interrupt를 발생시켰다.

Interrupt Mask Register에 '1'을 Write 하여 Interrupt 신호를 발생시킬 수 있게 하였다. Interrupt Polarity Register는 값이 '1'이면 Active High일 때 Interrupt를 발생시키고 '0'이면 Active Low일 때, Interrupt를 발생시킨다. Interrupt Edge Register는 값이 '1'이면 Rising Edge에서 Interrupt를 발생시키고, '0'이면 Falling Edge에서 Interrupt를 발생시킨다.

스위치를 눌렀을 때 Interrupt를 발생시키기 위의 레지스터에 '1'을 Write 해주었다. Write 이후 레지스터를 확인하였고 Fig. 6을 통해 결과를 확인할 수 있다.

```
0xf800040c  I/O interrupt mask register  0x0000ff00
0xf8000410  I/O interrupt polarity register  0x0000ff00
0xf8000414  I/O interrupt edge register  0x0000ff00
```

Fig. 6 GRGPIO Interrupt Register.

스위치를 눌렀을 때, Data Register에 스위치 값이 저장되어 0에서 Fig. 7과 같이 값이 저장됨을 확인할 수 있다.

```
0xf8000400  I/O port data register  0x00000000
0xf8000400  I/O port data register  0x00000200
```

Fig. 7 GRGPIO Data Register.

스위치에서 발생한 Interrupt로 인해 Pending Register에 저장된 값이 0에서 Data Register의 값과 같이 변하게 된 것을 Fig. 8를 통해 확인할 수 있다.

```
0xf8000004  Interrupt pending register  0x00000000
0xf8000004  Interrupt pending register  0x00000200
```

Fig. 8 Interrupt Pending Register.

위의 과정들을 통해 각 레지스터가 AMBA Bus에 연결되어 데이터를 주고받는데 이상이 없

음을 확인할 수 있다.

4. On-Chip 설계

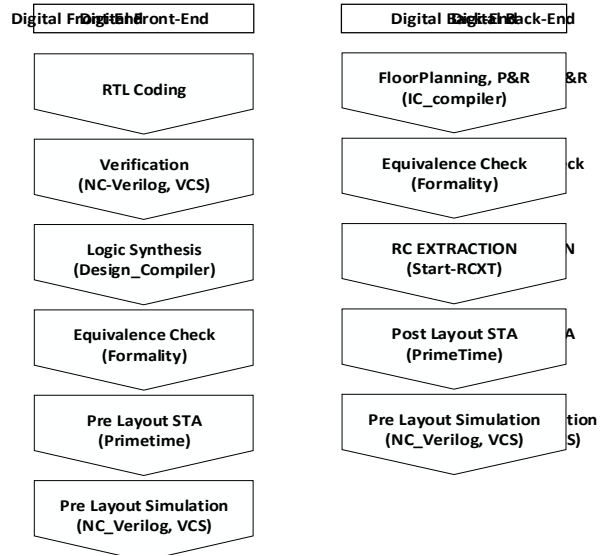


Fig. 9 Digital Design Flow.

VHDL로 설계된 모듈을 하나의 시스템으로 구현하였다. 구현한 디자인은 Cadence Inc. 사에서 판매하는 NC Verilog 또는 Synopsys Inc. 사의 VCS를 통해 검증하였다. 이후 Target Library를 이용하여 Fig. 9에 있는 과정대로 설계를 진행하였다. 그 이후 Synopsys Inc. 사의 Design Compiler를 통해 합성을 진행하였다.

GRLIB에서 제공하는 LEON3는 Xilinx 사의 Spartan6 모델로 설계되어 있으므로 ASIC 설계 시 Digital Clock Manager (DCM), Input Buffer (IBUF), Output Buffer (OBUF) 등 사용할 수 없는 하위 모듈들이 생긴다. IBUF와 OBUF는 Port와 로직들 사이에 합성된다. IBUF는 CLK가 일반 I/O에 연결된 경우 생성될 수 있는 오류를 방지하기 위해 자동으로 생성된다. 해당 모듈이 갖는 특성을 확인한 후에 Target Library에 있는 버퍼와 비교하여 다시 매칭하여 합성하였다. DCM은 Digital Clock Manager로써 CLK를 관리하는 Xilinx의 자체 모듈이다.

우리가 구현하고자 하는 시스템에서는 특정 주기를 설정하여 신호를 내보내는 CLKFX 기능만 사용한다. 해당 주기를 확인하고 분주하는 모듈을 만들어 다시 합성하였다.

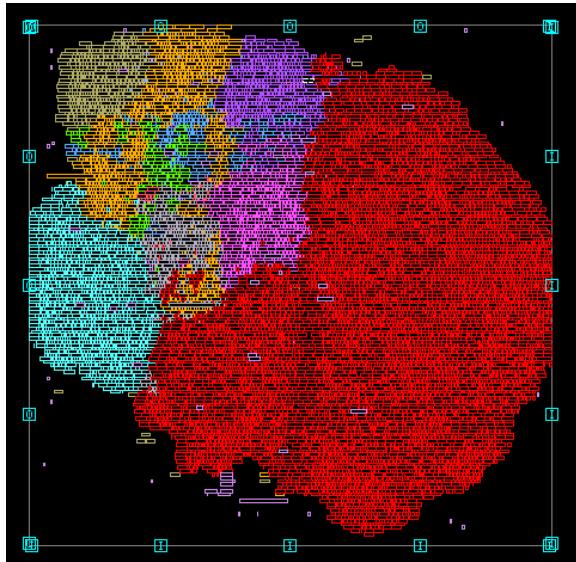


Fig. 10 Chip Layout.

Fig. 10의 칩 레이아웃 중에 Core 모듈은 빨간색 부분은 LEON3, 분홍색인 DSU 모듈이며 면적 절반 이상을 차지한다. 나머지 부분은 AMBA 및 그에 연결된 모듈로 구성되어있다. 하늘색은 AHBTRACE, 주황색은 AHBUART, 보라색은 IRQMP, 황토색은 GRGPIO, 파란색은 AHB/APB Bridge, 회색은 AHBCTRL, 연두색은 AHB Status이다.

Floorplan시 전원을 안정적으로 공급하기 위해 Metal4와 Metal5로 Power ring을 만들어 주었다. 기본적으로 IC Compiler은 CTS 합성 시, CLK를 기준에 신호를 연결하는 Metal과 동일하게 연결한다. 하지만 이렇게 연결하게 될 경우 주변 Metal에 영향을 받아 오동작할 수 있어 CLK Metal의 Width를 2배, Metal 과 Metal 간격을 2배로 설계하였다. 또한, 합성될 때 Metall, Metal2를 사용하지 않고 최소 Metal3부터 사용하여 안정성을 확보하였다. 이후 Routing을 진행해 Layout을 완성하였다.

4.1 온칩 설계 결과 비교

일반적으로 MCU를 설계하고 FPGA를 활용하여 구축하는 경우 별도의 비용없이 시스템을 구축할 수 있다. 하지만, IoT 소형 시스템 등을 구성해야하는 경우, 즉 실내나 차량용 반도체에서 MCU 시스템의 크기를 소형화해야하는 환경에서는 온칩으로 설계해야 하는 필요성이 있다. 이때, MCU를 온칩으로 설계하는 경우, 시스템이 얼마나 작아지는지, 소비전력은 얼마나 감소하는지에 대해 분석하였다.

4.1.1 면적 비교

Fig. 1에서 설명한 AMBA 시스템을 Spartan-6에 합성한 결과 FPGA에 구현된 Slice는 총 2081개이다. 순차회로를 FPGA로 구현할 때 사용하는 기준단위인 Slice는 LUT6 8개, Flip-Flop 16개, MUX 1개로 구성되어 있다 (Xilinx). LUT6는 M. Renovell(1988)에 의하면 SRAM으로 구현되며 SRAM의 면적과 표준 셀 (Standard cells) 면적은 Balobas, D.(2015)를 참고하였다. 180nm 공정으로 FPGA에 구현된 구성물의 면적을 유추한 결과는 Table 1에 정리하였다. 따라서, Table 1에 의하면 Slice 1개의 면적은 약 $2044\mu\text{m}^2$ 이다. 총 2081개의 Slice로 이루어져 있는 AMBA 시스템을 구성하기 위해 사용한 전체 면적은 $4,273,477\mu\text{m}^2$ 다.

Table 1 Area of Components.

	Area (μm^2)
LUT6	69.12
Flip-Flop	79.65
MUX	226.21

한편, 같은 AMBA 시스템을 DBhitech 180nm 공정을 이용하여 합성을 통해 주어진 회로를 다양한 종류의 표준 셀의 연결 형태로 구성하였다. Fig. 10의 전체 면적은 $850\mu\text{m} \times 850\mu\text{m}$ 이며, 합성된 모듈 구성 및 면적은 Table. 2와 같다. ASIC 합성을 통해 얻은 면적은 FPGA에 구성된 면적보다 약 10배 작게 설계된 것을 확인하였다.

Table 2 Area of The Proposed Design.

	Area (um ²)
LEON3	251,486
DSU	20,667
AHBTRACE	42,064
AHBUART	44,314
AHBCTRL	6,442
GRGPIO	14,747
AHB/APB Bridge	15,646
IRQMP	10,430
AHB Status	5,298
Total	411,127

4.1.2 소비전력 비교

Xilinx Power Analyzer를 사용하여 FPGA에 사용한 전력을 측정된 결과와 ASIC으로 합성한 이후에 측정된 전력은 Table 3과 같다.

Table 3 Power of The Proposed Design.

	Power (mW)	
	FPGA	ASIC
LEON3	47.1	15.40
DSU	0.12	1.78
AHBTRACE	1.53	2.90
AHBUART	0.76	2.09
AHBCTRL	0.87	1.01
GRGPIO	0.14	1.01
AHB/APB Bridge	0.12	0.45
IRQMP	0.44	1.18
AHB Status	0.03	0.46
Total	51.11	26.28
Ratio	×1.94	×1.00

각각을 비교하였을 경우 상대적으로 작은 Bus 모듈들은 ASIC으로 구현하였을 때 FPGA

로 구현하였을 경우보다 많은 전력을 소비하지만, 면적이 큰 모듈의 경우는 적은 전력을 소비하여 동작함을 확인할 수 있다.

5. 결 론

오픈 라이브러리인 GRLIB을 활용하여 AMBA 기반의 기본형태의 MCU를 설계할 수 있는 프레임워크를 제안하였으며, 설계한 MCU의 검증 및 온칩 설계, 그리고 온칩 설계 시 면적 및 소비전력 면에서 어느정도의 차이가 있는지를 분석하였다. 향후 본 논문에서 제안한 개발용 프레임워크를 이용하여 엣지컴퓨팅 및 차량용 반도체 등에 활용한 연구를 수행할 예정이다.

References

- Balobas, D., & Konofaos, N. (2015). Design and evaluation of 6T SRAM layout designs at modern nanoscale CMOS processes.
- Cadence Inc, NC-Verilog Simulator User Guide
- Gaisler, GRLIB IP Library User's Manual
- Korea Energy Economics Insstitute, "World Energy Market Insight", 21-14, 2021.7.12.
- M. Renovell, J. M. Portal, J. Figueras and Y. Zorian, "SRAM-based FPGA's: testing the LUT/RAM modules," Proceedings International Test Conference 1998 (IEEE Cat. No.98CH36270), 1998, pp. 1102-1111.
- Synopsys Inc, VCS User Guide, 1999
- Synopsys Inc, Design Compiler User Guide, B-2008.09, 2008
- Synopsys Inc, IC Compiler User Guide: Implementation, B-2008.09, 2009
- Xilinx Inc, www.xilinx.com



노 찬 휘 (Chanhwi Roh)

- 충북대학교 전자공학과 공학사
- (현재) 충북대학교 전자정보대학 반도체공학과 석사과정
- 관심분야: SoC, Energy harvesting system Design



오 연 상 (Yeonsang Oh)

- 충북대학교 전자공학과 공학사
- (현재) 충북대학교 전자정보대학 반도체공학과 석사과정
- 관심분야: SoC, Energy harvesting system Design



백 돈 규 (Donkyu Baek)

- 정회원
- 한양대학교 전자통신컴퓨터공학부 공학사
- 한국과학기술원 전기전자공학부 공학석사
- 한국과학기술원 전기전자공학부 공학박사
- (현재) 충북대학교 전자정보대학 전자공학부 조교수
- 관심분야: IoT system, SoC design, Cyber-physical system