

논문 2012-50-6-11

# Core-A 프로세서 기반의 멀티미디어 SoC 플랫폼 설계

(The Design of Multi-media SoC Platform Based on Core-A Processor)

서 학 용\*, 허 경 철\*, 정 승 표\*, 박 주 성\*\*

(Xuelong Xu<sup>Ⓞ</sup>, Jingzhe Xu, Seungpyo Jung, and Jusung Park)

## 요 약

최근 주목받는 스마트 폰, 스마트 TV 등 스마트 전자기기들은 전통기기의 기능과 컴퓨터를 결합하는 공통점을 갖고 있다. 단순히 프로세서가 내장되어 연산만 수행하는 것이 아니라 OS(Operating System)가 포함되고 사용자가 개인 용도에 따라 새로운 기능을 추가할 수 있고 유무선 통신으로 인터넷 또는 PC와 연결하여 통신할 수 있는 개장된 멀티미디어 SoC 플랫폼이 필요하다. 본 논문에서는 Core-A 프로세서와 AMBA 버스 기반으로 영상, 음성 또는 각종 통신 형태를 지원하는 다기능 SoC 플랫폼을 설계하여 FPGA로 구현과 검증을 하였다. SoC 플랫폼의 전체 성능을 검증하기 위해 JPEG 디코딩 알고리즘과 ADPCM 인코딩 디코딩 알고리즘을 실행하고 실행 결과를 모니터 또는 스피커로 출력하여 검증했다.

## Abstract

Recently smart devices which combine traditional electronic devices and personal computers, such as smart phones and smart TV, have caught people's eyes from all over the world. A multi-media SoC platform which embeds not only a calculating processor but also an operating system could provide an user-customized environment of several types of communication methods to PC or Internet. In this paper, we describe a multi-functioning SoC platform with video, audio and other communicating protocols based on Core-A processor and AMBA buses. To verify the designed multi-media SoC platform, JPEG decoding and ADPCM encoding/decoding algorithms are applied on it and the final decoding results are confirmed by video monitors and audio speakers.

**Keywords** : Multi-media, SoC, Core-A, AMBA bus, FPGA

## I. 서 론

기존의 스마트 기기들은 대부분 ARM 프로세서 기반의 플랫폼으로 구성되어 있다. 대기업의 입장에서 이러한 플랫폼에 다양한 유료 IP를 연동하여 쉽게 멀티미디어 스마트기기를 만들 수 있는 장점은 있으나, 비싼 ARM 프로세서의 사용료와 프로세서에 제한된 플랫폼

구성에 따른 낮은 경쟁력은 중소기업의 활발한 시장참여를 막는 장벽이 되고 있다.

본 논문에서는 국내 특허청 사업의 지원으로 개발된 한국형 프로세서인 Core-A 프로세서<sup>[1]</sup>와 ARM사에서 공개한 AMBA(Advanced Micro-controller Bus Architecture) 버스<sup>[2]</sup>로 구성된 고성능 멀티미디어 SoC 플랫폼을 구현함으로써 multi-media 시스템을 경제적 또는 효율적으로 설계하는 방법을 제안한다. 이에 따라 다양한 전자기기의 기능을 시스템에 단순히 나열하여 탑재하는 것이 아니라 개발 단계에서는 집적도가 높은 SoC(System on Chip) 설계를 FPGA로 구현하는 것이 안정적이다.

본 논문의 구성은 다음과 같다. II장에서는 멀티미디어 플랫폼의 구조와 기능을 소개하고 III장에서는 기본

\* 학생회원, \*\* 평생회원 부산대학교 전자전기공학과  
(Department of Electronics and Electrical  
Engineering, Pusan National University)

\* 본 연구는 지식경제부가 지원하는 산업융합원천기술개발사업을 통해 개발된 결과임을 밝힙니다.  
(10039173, 융복합 혁신 반도체 기술 개발).

Ⓞ Corresponding Author(E-mail: xuxuelong@pusan.ac.kr)  
접수일자 2012년11월17일, 수정완료일 2013년5월17일

플랫폼의 설계를 설명한다. IV장에서는 기본 플랫폼에 다양한 기능을 추가하여 멀티미디어 SoC 플랫폼의 설계를, V장에서는 플랫폼의 FPGA 구현과 검증을 소개한다. 마지막으로 VI장에서는 결론을 맺는다.

## II. 멀티미디어 SoC 플랫폼의 구조

본 멀티미디어 SoC 플랫폼은 그림 1에 표시된 바와 같이 각 외부 기능 블록과 FPGA 내의 IP 형태로 설계된 해당 인터페이스로 구성돼 있다. 인터페이스 구조가 간단한 SRAM 메모리<sup>[3]</sup>와 대용량 DRAM 그리고 비휘발성인 SD-Card/NOR-Flash 메모리는 플랫폼에 필수적인 저장 공간을 제공한다. 입력 장치는 PS/2 (Keyboard and Mouse), GPIO 로 구성돼 있으며 출력은 오디오 AC'97 CODEC<sup>[4]</sup>과 비디오 DAC를 사용한다. 통신 장치는 제어가 용이한 UART, 전송 속도가 빠른 USB Slave, 그리고 network 연결을 지원하는 Ethernet으로 구성돼 있다.

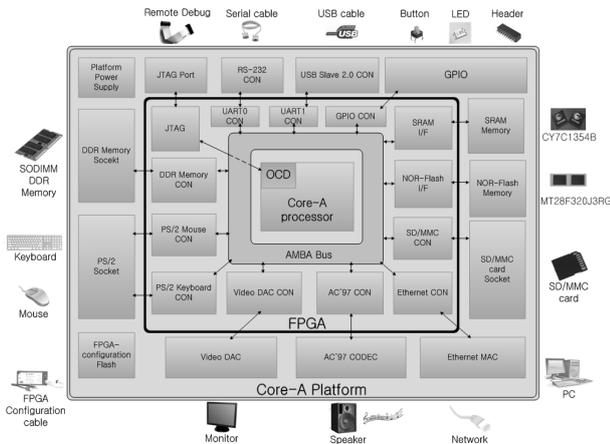


그림 1. 멀티미디어 SoC 플랫폼의 구조  
Fig. 1. Architecture of multi-media SoC platform.

### 1. Core-A 프로세서

멀티미디어 SoC 플랫폼에 내장된 한국형 프로세서는 3단 Pipeline 구조를 가진 Core-A 프로세서다. Core-A 프로세서는 명령어와 데이터 참조 경로가 분리된 RISC 프로세서이며, 32bit 명령어(16bit 단축 명령어)와 32bit 데이터를 사용한다. 명령어 내에 조건비교 기능이 있어 자체적으로 명령어의 수행 여부를 결정할 수 있다. Core-A 프로세서는 32bit 프로세서로서 최대 4GB 명령어 메모리 공간과 4GB 데이터 메모리 공간을 지원할 수 있고 메모리 접근에는 Big-endian과 Little-endian을

모두 지원한다. 프로세서에 내장된 레지스터(Register)는 범용 레지스터(GPR, General Purpose Register) 16개에 특수 레지스터(SPR, Special Purpose Register) 2개 그리고 연동프로세서(Co-Processor)용으로 16개로 구성된 4개의 묶음이 내장돼 있다.

프로세서의 Tool Chain에 포함된 프로그램은 C 언어나 Assemble 코드를 해당 프로세서에서 실행할 수 있도록 변환해준다. 변환된 코드를 플랫폼으로 다운로드하여 전체 시스템 동작을 확인하기 위해 Core-A 프로세서에는 OCD(On-Chip Debugger) 로직이 내장되어 있으며 FPGA 레벨에서 플랫폼을 검증할 수 있다.

### 2. AMBA 버스

AMBA 버스는 ARM사에서 공개한 SoC 버스 표준이다. AMBA 버스는 SoC 시스템의 기능 모듈들을 연결해주는 32bit On-chip 버스 구조로써 SoC 설계의 필수적인 요소이며 버스의 효율성은 시스템의 전체 성능에 큰 영향을 준다. 본 논문에서는 AMBA 2.0 표준에 AMBA AHB/APB 버스를 연동하여 사용한다.

## III. 기본 플랫폼의 설계

Core-A 프로세서에는 내부 RAM 메모리 구조가 포함되지 않아 Core-A 프로세서를 AMBA 버스에 연결하여 외부 SRAM 메모리 공간에 접근한다. 멀티미디어 SoC 플랫폼 설계의 첫 단계로 OCD 로직을 내장한 Core-A 프로세서, AMBA AHB/APB 버스 또는 SRAM 메모리만을 연결하여 기본 플랫폼을 구현한다. 그림 2에서는 기본 SoC 플랫폼의 FPGA 구현을 보여주며 이를 바탕으로 다양한 기능을 추가하여 멀티미디어 SoC 플랫폼을 구현한다.

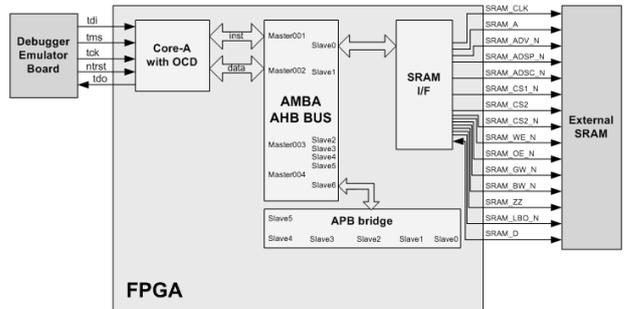


그림 2. 기본 SoC 플랫폼의 FPGA 구현  
Fig. 2. FPGA implementation of basic SoC platform.

#### IV. 멀티미디어 SoC 플랫폼의 설계

새로운 플랫폼을 설계할 때 모든 기능 블록의 인터페이스 IP와 해당 제어 C 프로그램을 모두 새로 설계하는 방식은 비효율적이다. 그래서 사용자의 요구에 따라 검증받은 하드웨어를 선정하고 해당 부품에 맞게 설계된 공개 IP들을 시스템 버스에 맞게 수정하여 짧은 개발 주기 내에 플랫폼을 구현하는 설계 방식은 현재 플랫폼 설계에서 많이 사용되고 있다<sup>[5]</sup>.

Opencores 도메인에서 공개한 IP core는 대부분 wishbone 버스, AMBA 버스 등 널리 사용되고 있는 시스템 버스에 맞게 설계돼 있으므로 하드웨어 설계에 상대적으로 쉽게 접근할 수 있는 경로를 사용자에게 제공해준다. 그래서 본 플랫폼 설계과정에서는 기본적으로 Opencores 도메인에서 공개한 IP와 제어 C 프로그램을 채택하여 수정하거나 재설계를 한다. 디버깅 기능을 내장한 Core-A 프로세서로부터 메모리 인터페이스 또는 각종 멀티미디어 기능 블록의 인터페이스는 Verilog HDL 형태의 IP로 제공돼 있으며 실제 로직의 타이밍, 면적 또는 전원 소모 측면에서 예측 불가능한 단점이 있지만, 사용자의 요구에 따라 수정 용이한 장점도 있다.

그림 3에서는 멀티미디어 SoC 플랫폼의 FPGA 구현을 보여준다. AMBA AHB 버스의 master 포트에는 Core-A 프로세서의 instruction/data 버스, Video DMA, Audio DMA 그리고 Ethernet DMA 블록이 연결돼 있고 slave 포트에는 SRAM/Flash 메모리 인터페이스, DDR 메모리 인터페이스, Video DAC controller,

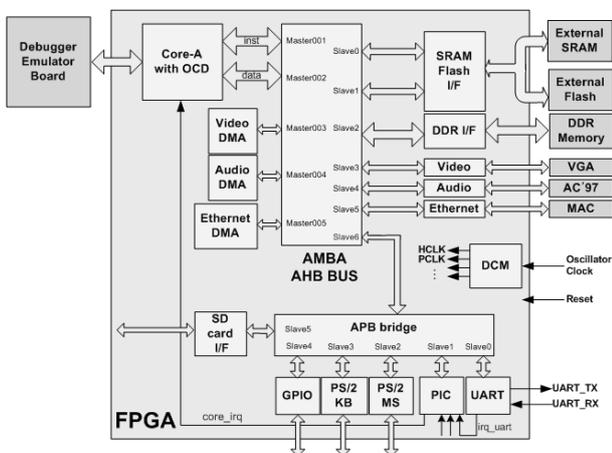


그림 3 멀티미디어 SoC 플랫폼의 FPGA 구현  
Fig. 3 FPGA implementation of multi-media SoC platform.

AC'97 controller, Ethernet MAC controller 그리고 AHB to APB bridge가 연결돼 있다. AMBA AHB 버스와 달리 AMBA APB 버스에는 저속으로 동작하는 UART, PIC, PS/2 Mouse, PS/2 Keyboard, GPIO 또는 SD card I/F가 연결돼 있다. 이상 기능 블록의 조합으로 전체 시스템은 사용자에게 아주 다양한 기능을 하나의 플랫폼으로 제공할 수 있다.

전체 시스템의 클럭(Clock)을 생성해주는 DCM 블록은 Xilinx FPGA에서 제공한다. HCLK(80MHz), PCLK(40MHz), 또는 DDR 블록에서 sync 신호를 발생하기 위해 사용하는 클럭(200MHz)은 모두 외부 Oscillator 클럭(50MHz)을 변환하여 생성된 것이다. DCM 블록에는 DLL(Delay locked loop), DFS(Digital Frequency synthesize), DPS(digital phase shifter) 또는 DSS(Digital spread spectrum) 기능이 내장돼 있으므로 시스템은 클럭의 skew/delay로 미친 영향을 최소화시킬 수 있다.

#### V. 검증

플랫폼의 구조가 복잡해지면서 기능 블록을 설계하고 모든 기능이 정상적으로 동작하는 것을 확인하기 위해 설계를 검증해야 한다. 실제 하드웨어로 구현하기 전에 function simulation을 진행하고 필요에 따라 post-time simulation도 할 수 있다. 그러나 simulation 과정은 많은 시간과 집중력이 필요해서 모든 상황을 검증하지 못할 수 있으며 실제 하드웨어 플랫폼을 구현한 뒤 효율적인 검증할 필요가 있다. 따라서 하드웨어 플랫폼에서 간단한 동작을 확인하고 플랫폼의 완성도와 신뢰성을 높이기 위해 복잡한 알고리즘을 플랫폼에 실행하여 통합 검증을 해야 한다.

##### 1. Function level verification

멀티미디어 SoC 플랫폼을 설계한 뒤 전체 SoC 플랫폼의 기본 구성 부분인 Core-A 프로세서와 AMBA 버스 인터페이스를 검증한다. AMBA 버스 master 포트에 연결된 Core-A instruction 버스와 data 버스의 신호 전송 과정을 ModelSim 시뮬레이터로 검증한다. 그림 4와 같이 AMBA AHB 버스의 Master001과 Master002의 동작을 검증한다. Master002에서 HBUSREQ002 신호로 버스 사용권한 요청을 AHB arbiter로 보낸다. AMBA AHB 버스의 AHB arbiter는 버스 사용 상태를 확인한 뒤 사용권한을 Master002에 넘겨준다. Signal

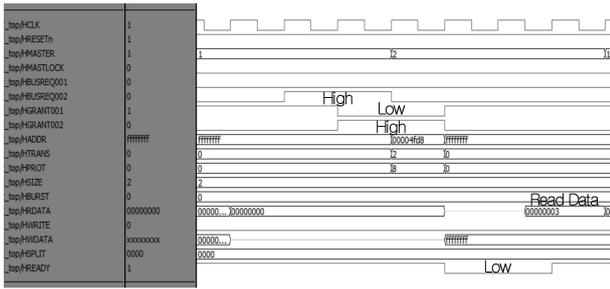


그림 4 AMBA 버스 master의 행위 레벨 검증  
Fig. 4 Funtion level simulation of AMBA bus master.

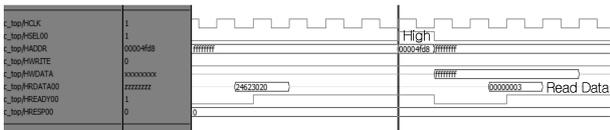


그림 5 AMBA 버스 slave의 행위 레벨 검증  
Fig. 5 Funtion level simulation of AMBA bus slave.

레벨에서 설명하면 HGRANT001은 'LOW'로, HGRANT002는 'HIGH'로, HMASTER번호는 '2'로 변환한다. MASTER002는 버스 사용권한을 받고 HADDR[31:0]를 통해 버스 slave로 주소(0x00004FD8)를 보낸다. HADDR[31:0]과 동시에 HTRANS, HPORT, HSIZE, HWRITE 등 제어 신호도 같이 전송된다. AMBA AHB slave 측에는 SRAM 메모리 인터페이스가 Slave0에 연결돼 있으며 SRAM 메모리의 주소(0x00004FD8)를 접근한다.

AMBA 버스의 slave 포트에 연결된 SRAM 메모리 인터페이스 부분을 검증한다. AMBA AHB master의 메모리 접근 동작에 응답하는 AMBA AHB slave 측의 동작은 그림 5와 같다. HSEL00 신호는 'HIGH'로 인가되고 SRAM 메모리 인터페이스는 선택된다. 그리고 HADDR[31:0] 어드레스 버스를 통해 주소(0x00004FD8)가 전달된다. AMBA AHB slave는 이에 응답하여 SRAM 메모리의 주소(0x00004FD8)에 저장된 데이터를 읽어온다. HRDATA00[31:0]에 나타난 '0x00000003'이 바로 SRAM 메모리 주소(0x00004FD8)에 저장됐던 데이터다.

그림 4와 그림 5를 통합하여 Core-A 프로세서가 주도한 SRAM 메모리의 '0x00004FD8' 주소로부터 데이터 '0x00000003'을 읽어오는 과정을 검증한다. 유사한 과정을 통해 AMBA 버스의 master/slave의 write/burst read/burst write 등 동작도 확인할 수 있다.

기본 SoC 플랫폼의 동작을 검증한 뒤 멀티미디어 SoC 플랫폼에 연결된 다양한 기능 블록의 인터페이스 부분을 검증했다. 구체적인 검증 과정은 그림 4와 그림

5에 메모리 접근 또는 레지스터(register)접근 방식과 유사하여 본문에서는 생략한다.

### 2. FPGA level verification

행위 레벨의 검증을 통해 SoC 설계의 기본 동작을 확인한 뒤 PCB 설계 및 제작 단계를 거쳐 SoC 설계를 FPGA 내부의 실제 하드웨어로 구현한다. 구현한 시스템의 실제 동작을 검증하기 위해 Core-A 디버거<sup>[6]</sup> 환경을 구축한다. Core-A 디버거 환경을 구축하는 과정과 사용법은 참고문헌[6]에 자세히 설명되어 있다. 그림 6과 같이 Core-A GDB, Core-ISS 또는 Eclipse(GUI)<sup>[6]</sup>로 구성된 호스트 디버거 환경에서는 로컬 디버깅을 실행할 수 있으며 RDA, Emulator Board, JTAG을 통해 Core-A 프로세서에 내장된 OCD 디버거 로직과 연동하여 실제 보드 디버깅을 할 수 있다. 또는 ModelSim 시뮬레이터와 Core-A GDB를 연동하여 검증할 수 있다.

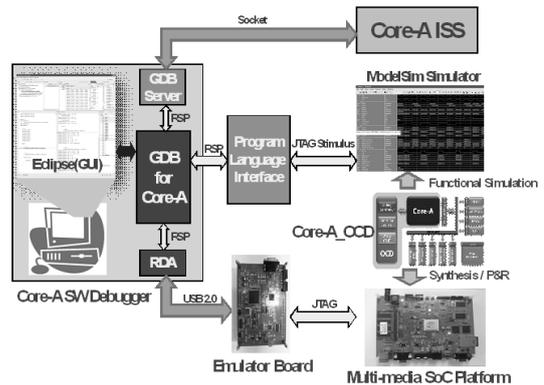


그림 6 멀티미디어 SoC 플랫폼의 FPGA 레벨 검증  
Fig. 6 FPGA level verification of the multi-media SoC platform.

### 3. Algorithm level verification

#### 가. JPEG 디코딩 알고리즘을 이용한 검증

멀티미디어 SoC 플랫폼의 Video 기능을 통합 검증하기 위해 표준 JPEG 디코딩 알고리즘을 Core-A 프로세서와 본 플랫폼에 탑재된 메모리 size에 적합하게 수정한다. JPEG 디코딩 알고리즘을 실행할 때 Huffman Decoding Table과 Image Frame buffer 공간이 필요하므로 알고리즘 초기화 시 heap 공간을 미리 할당해줘야 한다. 그러나 SRAM 메모리 크기보다 큰 공간이 필요하여 대용량 DDR memory<sup>[7]</sup>는 필수적이다. 그림 7에서는 JPEG 디코딩 알고리즘을 이용한 검증 과정을 보여

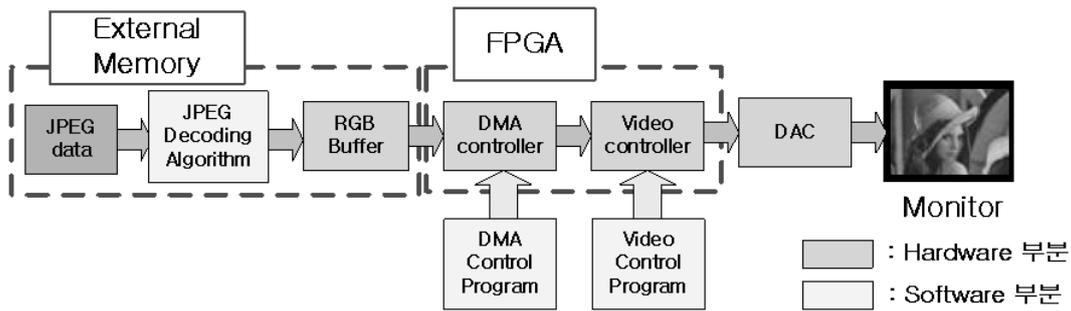


그림 7 JPEG 디코딩 알고리즘을 이용한 검증  
Fig. 7 Verification of the multi-media SoC platform using JPEG decoding algorithm.

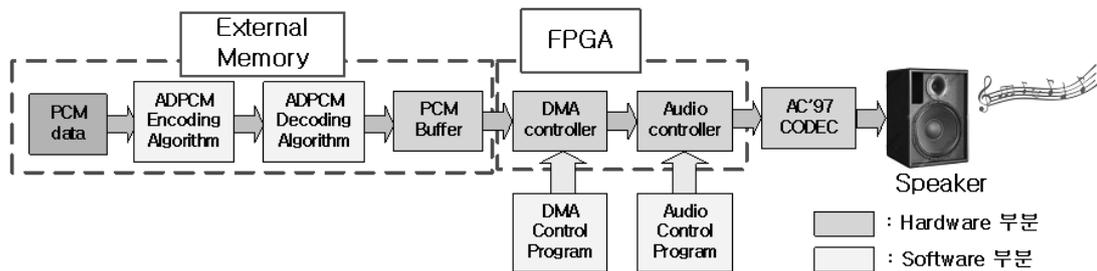


그림 8 ADPCM인코딩/디코딩 알고리즘을 이용한 검증  
Fig. 8 Verification of the multi-media SoC platform using ADPCM encoding/decoding algorithm.

준다. 결과로 320 x 240 크기의 JPEG 이미지를 압축 풀어 모니터로 출력한다.

#### 나. JPEG 디코딩 알고리즘을 이용한 검증

멀티미디어 SoC 플랫폼의 Audio 기능을 통합 검증하기 위해 ADPCM 인코딩 디코딩 알고리즘을 Core-A 프로세서에 맞게 포팅 한다. 그림 8에서는 sample rate 가 8kHz / 16kHz인 PCM 음성 데이터를 ADPCM 인코딩 알고리즘으로 압축하고 압축된 ADPCM 데이터를 다시 디코딩하여 Audio controller를 통해 스피커로 재생하는 검증 과정을 보여준다.

## VI. 결 론

본 논문에서는 Core-A 프로세서와 AMBA 버스 기반으로 멀티미디어 기능을 지원하는 SoC 플랫폼의 설계 및 구현을 완성하였다. Opencores 도메인에서 공개한 IP core들을 필요에 따라 수정하여 SRAM 메모리, NOR Flash 메모리, DDR 메모리, SD card 메모리, UART 통신, PS/2 Keyboard, PS/2 Mouse, Video, Audio, Ethernet, GPIO, USB Slave 기능을 하나의 SoC 플랫폼에 구현했다. 표 1에서는 제안한 플랫폼을 기존

표 1. 제안한 플랫폼의 추가 기능

Table 1. Additional functions of the proposed platform.

Functions	Basic Func.	DDR	Debug	USB slave
Available	●	●	●	●

플랫폼과 비교했을 때 추가적인 기능을 표시한다.

플랫폼의 부품 선정부터 PCB Schematic 설계, PCB Placement, PCB Soldering 과정을 수행하여 플랫폼 PCB 보드를 제작하고 FPGA방식으로 SoC 시스템 설계를 완성하였다. 각 기능 블록의 인터페이스 설계를 행위 레벨에서 검증하고 전체 SoC 설계의 timing을 분석하여 설계를 개선했다. 최종적으로 제안한 SoC 플랫폼은 기존 플랫폼(50MHz)보다 빠른 80MHz의 시스템 클럭 주파수에서 정상 동작하며 90nm copper 공정인 Virtex-4 FF668LX40<sup>[8]</sup> FPGA에 약 15,377개(83%)의 Slices와 23,678개(64%)의 LUTs를 사용한다.

기본 SoC 플랫폼을 검증하기 위해 간단한 메모리 읽기/쓰기 동작을 참고문헌<sup>[6]</sup>에 설명된 디버그 시스템으로 검증하고 플랫폼 전체를 통합 검증하기 위해 JPEG 디코딩 알고리즘과 ADPCM 인코딩 디코딩 알고리즘을 도입했다.

본 논문에서는 기존 플랫폼보다 다양한 기능을 지원

하고 보다 동작속도가 빠른 멀티미디어 SoC 플랫폼을 제안한다. 또한, 기존 멀티미디어 SoC 플랫폼에 존재한 문제점을 디버깅 과정을 통해 보완하여 FPGA 구현을 개선했다. 추후 본 설계는 ASIC으로 Core-A 프로세서를 내장한 AMBA 버스 기반의 Multi-media SoC Controller로 구현하는 데에 기반 플랫폼이 될 것으로 보인다.

REFERENCES

[1] 이종열, 이강, 김시호, 최병운, 이광엽, “Core-A 프로세서 기반 System-on-Chip 설계”, 홍릉과학출판사, 56-71쪽, 2011년  
 [2] “AMBA Specification (Rev 2.0)”, ARM Ltd, 1999.

[3] “CY7C1354B 9-Mb (256K x36 / 512K x18) Pipelined SRAM Manual”, CSC, 38-05114 Rev.\*C, pp. 3-7, 2004.  
 [4] “AC '97 Rev 2.1 Multi-Channel Audio Codec with Stereo Headphone Amplifier, Sample Rate Conversion and National 3D Sound”, National Semiconductor, 2004.  
 [5] 기안도, “Core-A 프로세서를 활용한 플랫폼 설계”, 홍릉과학출판사, 81-108쪽, 2010년  
 [6] 허경철, 박형배, 정승표, 박주성, “Core-A를 위한 효율적인 On-Chip Debugger 설계 및 검증”, 전자공학회논문지, 제47권 SD편, 제4호, 50-61쪽, 2010년 4월  
 [7] “Memory Interface Guide(MIG)- DDR1 Memory interface IP”, Xilinx, 2010.  
 [8] “Virtex-4 FPGA User Guide”, Xilinx, 2008.

— 저 자 소 개 —



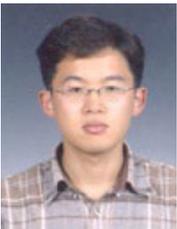
서 학 용(학생회원)  
 2010년 중국 하얼빈공정대학  
 자동제어학과 학사 졸업.  
 2013년 부산대학교  
 전자공학과 석사 졸업.

<주관심분야 : SoC 설계, 멀티미디어 플랫폼 설계, 비디오 알고리즘 구현 및 최적화>



허 경 철(학생회원)  
 2005년 중국 연변과학기술대학  
 전자통신학과 학사 졸업.  
 2008년 부산대학교  
 전자공학과 석사 졸업.  
 2011년 현재 부산대학교  
 전자전기공학과 박사과정.

<주관심분야 : 프로세서 설계, 디버거 설계, SoC 설계, 멀티 프로세서 플랫폼 설계, 오디오 알고리즘 구현>



정 승 표(학생회원)  
 2007년 부산대학교  
 전자공학과 학사 졸업.  
 2009년 부산대학교  
 전자공학과 석사 졸업.  
 2011년 부산대학교  
 전자공학과 박사 과정.

<주관심분야 : 신호처리, SoC 플랫폼 설계>



박 주 성(평생회원)-교신저자  
 1976년 부산대학교  
 전자공학과 학사 졸업.  
 1978년 한국과학기술원(KAIST)  
 전자공학과 석사 졸업.  
 1989년 University of Florida  
 전자공학과 박사 졸업.

1998년~2007년 부산대학교 IDEC 센터장  
 1991년~현재 부산대학교 전자공학과 교수  
 2008년~2009년 부산대학교 공과대학 학장  
 <주관심분야 : DSP 설계, ASIC 설계, 반도체 소자 모델링, 음성/사운드 신호처리 및 구현, SoC 설계>