

# 안드로이드 기반 임베디드 플랫폼 설계

윤찬호\* · 김광준\*\* · 장창수\*\*

Design of Embedded Platform based on Android

Chan-Ho Yoon\* · Gwang-Jun Kim\*\* · Chang-Soo Jang\*\*

## 요 약

본 논문은 안드로이드를 지원하는 ARM A8-cortex 프로세서를 기반으로 임베디드 플랫폼을 구현하였다. S5PV210의 삼성 CPU를 사용함으로써 32Bit RISC 마이크로컨트롤러(ARMv7) 구조에 적합하고, 주변기기에 호환가능하며, 응용에 확장 가능하도록 설계하였다. 또한 개발한 임베디드 플랫폼은 여러 가지 기능과 높은 효율성을 제공할 뿐만 아니라, 비교적 낮은 단가와, 낮은 전력사용, 높은 성능을 제공할 수 있다.

## ABSTRACT

This paper presents an implementation of embedded platform based ARM A8-cortex processor for android supporting. The development board for S5PV210 is a platform that is suitable for code development of SAMSUNG's S5PV210 32bit RICS micro controller(ARMv7) architecture for hand-held device and general applications. Embedded platform development board offers various function and high efficiencies. In addition to the high performance, the embedded platform offers low current consumption, ensuring low costs and power.

## 키워드

Embedded Platform, Android, ARM Cortex-A8 , Low Power

임베디드 플랫폼, 안드로이드, 암 코텍스-A8, 저전력

## 1. 서 론

임베디드 플랫폼의 특징이 낮은 속도의 마이크로프로세서, 적은 메모리 용량, 그리고 다양하지 못한 입출력 장치 등이 있기에 모바일 솔루션은 열악한 플랫폼에서 좋은 효율과 퍼포먼스를 가지는 것을 그 특징으로 하고 있다. 또한 임베디드 시스템에서는 제품의 신뢰성이 매우 중요하므로 운영체제인 OS와 시스템의 부팅 시간도 설계의 주요한 요소가 된다. 반면에

제품의 가격을 낮출 수 있고 원하는 기능에 최적화되어 있기 때문에 안정성을 충분히 확보할 수 있는 장점이 있어 임베디드 시스템의 이용 분야가 다양하게 늘어나고 있다[1],[2].

임베디드 시스템이란 일반적으로 특별한 업무를 수행하기 위한 하드웨어와 소프트웨어를 포함하는 특정한 응용 시스템이라고 할 수 있다. 즉, 마이크로프로세서 혹은 마이크로컨트롤러를 내장하여 단순히 회로만으로 구성된 것이 아니라 특정한 기능을 수행하도록

\* 전남대학교 전기·전자통신·컴퓨터공학부(yoonchanho@hotmail.com)

\*\* 교신저자(corresponding author) : 전남대학교 전기·전자통신·컴퓨터공학부(kgj@jnu.ac.kr)

\*\*\* 전남대학교 전기·전자통신·컴퓨터공학부(csjang@jnu.ac.kr)

접수일자 : 2013. 08. 20

심사(수정)일자 : 2013. 09. 23

게재확정일자 : 2013. 10. 21

록 프로그램이 입력되어 있는 내장시스템을 가진다. 예를 들면, 가전제품이 정보가전으로 변화되면서 제품에 프로세서, 운영체제, 응용 어플리케이션 소프트웨어 탑재되어 출하되고 있는 TV, 오디오, 냉장고, 세탁기 등의 가전제품과 의료기기, 네비게이션, 멀티미디어 플레이어, 홈 네트워크 등 무수히 많이 존재한다 [3].

본 논문에서는 안드로이드 기반의 ARM Cortex-A8를 사용하여 높은 성능을 보장함과 동시에 다양한 SoC (System on Chip) 추가하여 통합 운영이 가능하도록 임베디드 플랫폼을 설계하였다. 설계된 임베디드 플랫폼은 삼성의 S5PV210의 ARM Cortex-A8 CPU 코어를 기반으로 하여 PDA, 2.5G/3G 고성능 휴대전화, PMP(Portable Media Player)등에 최적화 되도록 하였으며, 주변기기와의 호환 및 확장성을 용이하도록 설계하였다.

## II. 안드로이드 플랫폼

안드로이드는 Google OHA(Open Handset Alliance)에서 발표한 모바일 오픈 솔루션이다. 그림 1은 안드로이드 플랫폼의 구조이다[4],[5],[6]. 그림 1에서 볼 수 있듯이 OS는 리눅스 커널을 사용하였고 Library는 많은 오픈 솔루션을 이용하였다. Android Runtime은 구글이 자체 개발한 Dalvik Virtual Machine과 Core Library로 구성되어 있고 이것 또한 소스 레벨로 공개된 상태이다. Application Framework은 다양한 기능의 프레임워크가 이미 완성되었고, Application은 전화 송·수신, SMS 송·수신, 인터넷, 이메일, 구글 맵스 등의 킬러 어플리케이션이 안드로이드 솔루션에 이미 포함되어 있다. 킬러 어플리케이션을 보면 알 수 있듯이 안드로이드는 핸드폰을 위한 솔루션이다. 하지만 안드로이드 솔루션이 반드시 핸드폰을 위한 솔루션만이 아니고 다른 플랫폼에도 충분히 올릴 수 있다[4],[5],[7],[8].

안드로이드 플랫폼은 크게 네 부분으로 나뉜다. 커널부분, 하드웨어 계층, 라이브러리, 응용프로그램 프레임워크. 라이브러리들은 C/C++로 구현되어 있다. 기존의 여러 프로젝트들을 통합하였기 때문에 C로 구현된 것들이 대부분이고 C++도 있다. 응용프로그램

프레임워크는 자바로 구현되어 있다. 프레임워크와 라이브러리 사이에는 자바와 C사이에는 자바와 C 사이의 서로 다른 호출 규약 등을 맵핑하는 JNI라는 마블러가 존재한다. 하드웨어 드라이버를 C로 만들면 JNI를 구현해야 한다. 하드웨어 부분은 Google에서 표준으로 정해두었기 때문에 따로 만들 필요가 없다. 추상레이어도 스펙이 다 되어 있어 응용프로그램 개발할 때는 API로 보이게 된다.

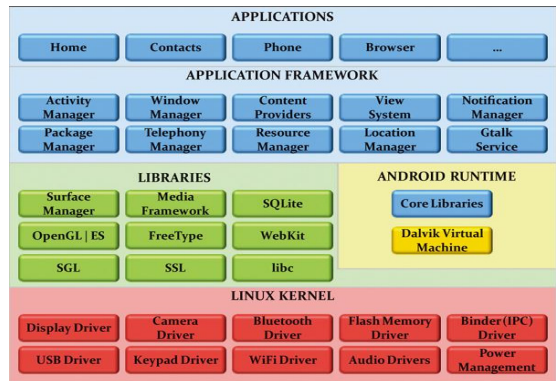


그림 1. 안드로이드 플랫폼 구조도 분석  
Fig. 1 Analysis of android platform structure

임베디드 리눅스와 안드로이드의 차이점 중 한 가지는 장치드라이버들(그래픽, 오디오, 카메라 등)이 커널영역이 아닌 유저모드에서 동작한다. 유저모드로 동작 시키면 자신에게 할당되어 있는 번지에만 접근할 수 있기 때문에 조금은 안전하지만 하드웨어 제어시에는 MMU등을 참조하는 등의 추가적인 작업이 필요하기 때문에 조금 불편 할 수 있다. 기존의 리눅스와는 다르게 GLIBC를 지원하지 않고 EABI(Embedded Arm Binary Interface)와 Binder를 사용한다. 그리고 커널 기능 향상을 위해 Alarm, Ashmem(공유메모리), Low memory-Killer, debugger, logger 컴퍼넌트가 추가되었다. 유저모드는 허가된 자원에만 접근이 가능하고 프로그램의 안정성을 높일 수 있지만 하드웨어 효율성은 떨어질 수 있다. 반대로 커널모드는 하드웨어 효율성은 높지만 안정성은 떨어질 수 있다. 안드로이드는 기본 드라이버들을 제외한 나머지 드라이버들은 유저스페이스에서 동작한다. 리눅스 커널을 빌드 할 경우에는 ARM서에서 표준으로 정해둔 ABI (Arm Binary Interface)규격의 컴파일러를 사용하며

안드로이드는 EABI(Embedded Application binary Interface) 기능을 지원한다. 기존의 ABI는 빌드옵션에서 VFP(Vector Floating Point)를 선택하거나 해제할 수 있었는데 이는 환경이 달라지면 다시 빌드해야 하는 단점이 있었다. EABI는 혼용이 되기 때문에 빌드의 부담이 줄어든다[9],[10].

### III. 임베디드 시스템 프로세서

#### 3.1 임베디드 프로세서

임베디드 프로세서는 단순히 장치 내에 내장되는 마이크로프로세서를 지칭하는 말이다. 주로 제어와 계산을 수행하는 임베디드 프로세서는 마이크로컨트롤러보다 성능은 높지만 범용 프로세서 성능에는 미치지 못한다. 그러나 그 적용 범위는 범용 프로세서를 훨씬 능가한다. 임베디드 프로세서에는 68K, x86, PowerPC, SPARC과 임베디드 정용 IP(Intellectual Property)인 ARM, MIPS, SuperH 등이 있다. ARM은 저 전력 소모를 요구하는 모바일 시장에서 독보적이며 오늘날 임베디드 프로세서 시장에서 막강한 세를 자랑하는 프로세서가 되었다. 인텔의 ARM코어 패밀리에는 StrongARM 아키텍처에서 파생된 Xscale 마이크로아키텍처가 있고 이 마이크로아키텍처를 기반으로 한 프로세서인 PXA26X, PXA250, PXA800F도 있다. 이와 함께 역시 Xscale 마이크로아키텍처에 기반을 둔 네트워크 스토리지 애플리케이션용 네트워크 프로세서 제품인 IXP2800, IXP2400, IXP425를 공급하고 있다[9],[10],[11].

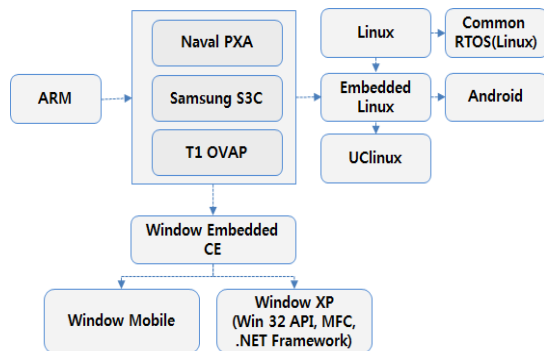


그림 2. 임베디드 플랫폼  
Fig. 2 Embedded platform

임베디드 플랫폼은 그림 2에서와 같이 윈도우 임베디드 CE와 임베디드 리눅스 등으로 구분할 수 있다. 임베디드 리눅스는 도입 비용 등이 적게 들어가는 장점이 있지만 필요시 관련 드라이버를 추가로 개발해야 하는 문제가 발생할 수 있고 시스템 문제 발생시 개발자가 자체적으로 해결해야 하는 어려움 등이 있다. 반면 윈도우 임베디드 CE는 구입비용이 많이 들어가는 문제점은 있지만 다른 플랫폼 보다 개발하기가 용이하고 관련 드라이버를 공급사에서 제공한다[12],[13],[14].

#### 3.2 ARM Core의 특성

ARM Core는 Advanced RISC Machine의 ARM의 약자로 임베디드 시스템(Embedded System)에 사용되는 범용적으로 쓰이는 모바일(Mobile)용 RISC (reduced instruction set computer) 코어(Core)로 저 전력사용 저가의 고효율 코어이다[13][14].

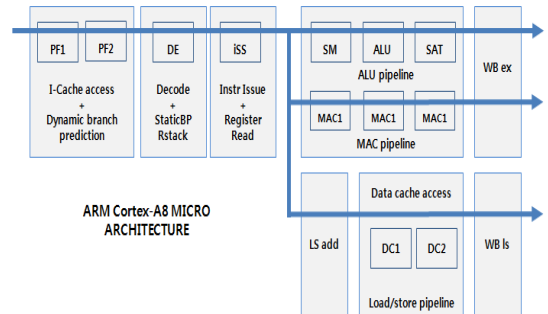


그림 3. ARM Cortex-A8 파이프라인 구조  
Fig. 3 Pipeline structure of ARM Cortex-A8

ARM Cortex-A8 마이크로 아키텍처(Micro architecture)는 효율적으로 고성능을 제공하도록 설계되어 코어 클럭(Core Clock)의 향상은 물론 전체적인 성능의 향상을 가져왔다. 그리고 여러 기기에 대한 컨트롤러를 내장하여 시스템의 확장성을 향상시켰다. ARM Cortex-A8 마이크로 아키텍처는 종전의 ARM10의 6 상태 파이프라인(State Pipeline)에 비해 13 상태 파이프라인으로 바뀌었다. 그림 3에서 나타낸 ARM Cortex-A8 마이크로 아키텍처의 13 상태 파이프라인은 종래의 ARM 코어에 비해 40%의 처리량 향상을 실현시켰다. 또한 13 상태 파이프라인에서는 13개의 처리 단계가 동시에 실행된다. 그러나 단계수가 많은

파이프라인 구조는 시스템에 지나친 지연 혹은 대기 시간을 초래하기 때문에 효율성이 떨어질 수 있다. 일반적으로, 단계수가 많은 파이프라인에서는 일부의 명령이 앞의 명령의 결과에 의존하기 때문에 지연될 수 있다. ARM Cortex-A8의 파이프라인은 전송을 완전히 활용함으로써, 이런 지연을 막을 수 있다. 또한, 단계수가 많은 파이프라인을 가진 프로세서에서는 파이프라인에서 처리되는 명령의 원활한 흐름에 인터럽트(Interrupt)가 일어날 경우, 예를 들면 분기 명령이 발생했을 때에도 성능이 떨어질 수 있다. ARM Cortex-A8의 파이프라인은 명령의 흐름을 예상하는 분기예측을 사용하여 이 지연을 막는다. 전송과 분기 예측은 파이프라인의 스탈(Stall)을 줄여 효율성을 유지한다. 스탈이란 다음 명령어가 아직 파이프라인을 통과중이기 때문에 프로세서가 기다리지 않으면 안 되는 상황을 가리킨다. 이상과 같은 파이프라인의 최적화에 의해, 유효 대기시간은 ARM9 마이크로 아키텍처의 5단계 파이프라인과 같으면서 훨씬 높은 처리량을 실현한다. 아키텍처에 있어 ARM Cortex-A8으로 바뀌면서 ARM은 그 동안 써온 32/16bit RISC 아키텍처인 ARMv4T의 명령어 구조에서 성능이 향상된 Powerful ARMv7 명령어 세트 아키텍처는 특히 차세대 사용자 기기, 무선통신, 네트워크, 자동차 제품의 요구를 염두에 두고 개발되었다[15],[16].

특히 ARMv7 아키텍처는 4가지 성능 개선에 초점을 두고 있다. 첫째 멀티미디어 처리 확장으로 2배 빠른 MPEG4 인코더/디코더, 2배 빠른 오디오 DSP (Digital Signal Processor)가 가능하다. 둘째로 캐시 구조(Cache Architecture)의 개선으로 물리적주소를 갖는 캐시, 문맥 교환에 걸리는 과부하가 감소하였다. 셋째로 예외 처리와 인터럽트 처리를 개선하여 실시간 태스크(Task)에서 성능 향상을 이루었다. 넷째로 정렬되지 않은 데이터, 바이트(Byte) 순서가 혼재된 데이터를 지원하고 데이터 공급, 어플리케이션 포팅 (Application Porting)이 간단해서 메모리를 절약할 수 있게 되었다. 또한 예외 처리와 인터럽트 처리의 시간이 단축되었다. 아래의 SRS와 RFE 명령어들을 추가함으로써 모드가 변할 때 상태 저장과 복구하는 과정을 더욱 효과적으로 할 수 있도록 하였다. 그리고 CPSID와 CPSIE 명령어를 새로 추가하여 사용자 모드(User Mode)를 제외한 모드 변화를 할 수 있도록

하였다. ARM Cortex-A8은 SIMD(Single Instruction Multiple Data) 미디어프로세싱(Media Processing)을 향상시켜 3D 그래픽 가속 효과를 월등히 상승 시켰고, 각각 16KB의 명령어, Data L1(Level1) 캐시 메모리를 보유하고 있으며, 하드웨어적으로 설계되었으나 레지스터 설정을 통해 사용 유무를 결정할 수 있도록 설계된 MMU(Memory Management Unit)가 있다. 이에 추가적으로 각각의 L1 캐시로 64bit의 인터페이스를 제공함으로써 더욱 빠른 접근이 가능케 했다. 또한 128Kbytes 의L2(Level 2) 캐시를 제공하며 64-bit 명령어, 64bit의 Level two 상호연결을 제공하며 32bit 주변기기와 64bit의 DMA(Direct Memory Access)를 AMBA(Advanced Microcontroller Bus Architecture) 인터페이스를 통하여 제공한다. 아래 그림 4는 Level two 상호연결 인터페이스를 보여준다.

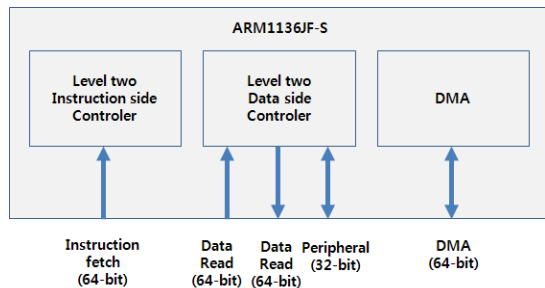


그림 4. Level two 상호연결 인터페이스  
Fig. 4 Level two interconnect interface

ARM Cortex-A8은 AMBA 2.0의 AHB 버스 구조를 가지고 있는데, AHB 버스는 위의 그림 4에서 보는 것과 같이 데이터의 Read/Write는 물론 DMA 컨트롤, 주변기기 컨트롤을 담당한다. 그림 5는 AMBA 인터페이스의 AHB-Lite 블록도를 보여준다. ARM Cortex-A8은 미디어 프로세싱과 네트워크 어플리케이션을 위한 고성능 64bits 메모리 시스템 데이터 액세스를 제공하며, 코어의 클럭 속도는 0.13u 프로세스에서 최대 660 드라이스톤 (Dhrystone 2.1)MIPS를 가진다.

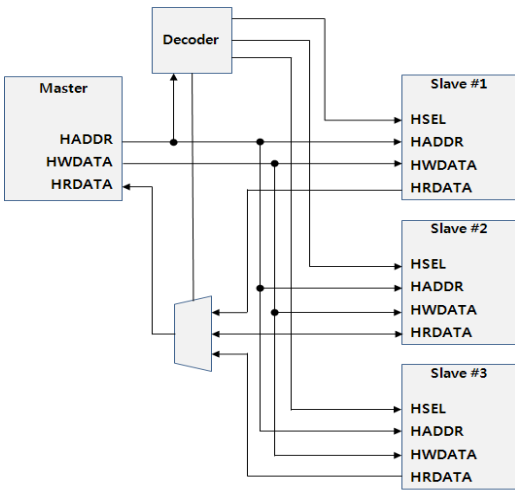


그림 5. AHB-Lite 블록도  
Fig. 5 Block diagram of AHB-Lite

ARM Cortex-A8 기반의 Core는 높은 성능을 목표로 하고 있지만, 동시에 다양한 SoC (System on Chip) 설계에 간단히 통합할 수 있어야 한다. 이 때문에 ARM Cortex-A8 제품은 세미 커스텀(Semi Custom) 구현 및 논리합성 설계 플로를 지원하고 있다. 또한, ARM Cortex-A8 마이크로 아키텍처는 OEM이 확신을 갖고 자유롭게 ARM 코어 기반 제품, 공급 회사가 설계 실현을 선택할 수 있도록 하기 위해 상위호환성을 확보하고 있다.

S5PV210은 삼성이 ARM Cortex-A8 CPU 코어를 기반으로 하여 PDA, 2.5G/3G 고성능 휴대전화, PMP(Portable Media Player)등에 최적화 되어 있다. 400/533MHz로 동작하며 64/32bit의 AXI, AHB로 구성된 내부 버스 구조를 가지며 DRAM 포트와 Flash/ROM/DRAM을 연결 할 수 있는 분리된 2개의 외부 메모리 포트를 가진다. 또한 확장성을 용이하게 해주는 많은 하드웨어 주변기기들을 가지고 있다. 4096 X 4096의 해상도를 지원하고 줌, 회전, 색 공간 변화(Color Space Conversion), LCD 컨트롤러 직접 연결 등의 여러 기능을 가진 카메라 인터페이스와 MPEG4, H.264/AVC 등을 지원하는 LCD컨트롤러, 비디오 포트 프로세서, TV 인코더, 오디오 인터페이스 등을 갖추고 있다. 또한 외부 확장으로 4채널의 UART, I2C, I2S, 2Ch SPI, HIPI HSI, IrDa 등의 여러 인터페이스와 MMC/SD Host, USB1.1 Host, USB

2.0 OTG(On-The-Go) 등을 지원한다[10],[11].

#### IV. 임베디드 플랫폼 설계

CPU는 Samsung 출시 제품 S5PV210를 사용하였다. S5PV210은 32Bit RISC 프로세서로서 비교적 낮은 단가와, 낮은 전력사용, 높은 성능을 제공하는 마이크로프로세서이다. S5PV210 CPU 특징은 ARM Cortex-A8 Core를 포함하고 있으며 ARMv7-A 아키텍처가 적용되어 있다. 비디오 가속을 위한 Video Processing, 제공되는 영상 처리를 위한 Display Control과 Scaling 등의 여러 종류의 하드웨어 가속기를 내장하고 있으며, MFC (Multi Format Codec)을 포함하고 있어 영상 포맷 지원을 위해 별도의 코덱 구현 없이 사용이 가능하다. MFC를 통한 하드웨어 가속기를 이용하여 TV Out, HDMI의 영상 지원이 가능하며, MPEG4, H.263, H.264의 인코딩/디코딩 코덱을 지원하고, 디코딩전용으로 MPEG1/2, VC-1을 지원하고 있다.

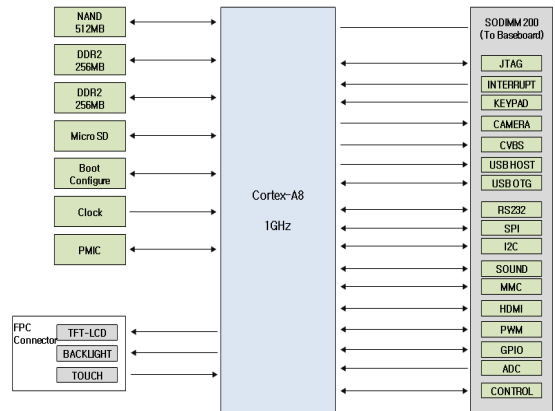


그림 6. CPU (S5PV210) logic diagram  
Fig. 6 CPU (S5PV210) logic diagram

CPU와 시스템에서 사용되는 전원은 블록별로 요구되는 사양에 따라 여러 가지로 분류되고 전원의 공급타이밍이나 불필요 시 전력소모를 줄이기 위해 Disable시켜야 하는 전원제어가 필요하다. 그림 7은 MPU Module 전원 설계 Block Diagram을 나타낸 것으로 Samsung사의 S5Px 시리즈에 최적화되어 있으

며, 3개의 Step-Down Converter, 9개의 LDO Regulator를 내장하고 있다.

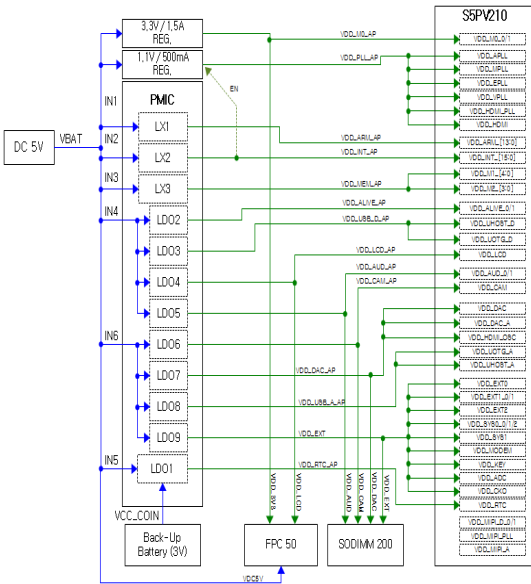


그림 7. MPU module 전원 설계 블록도

Fig. 7 Block diagram of MPU module power design

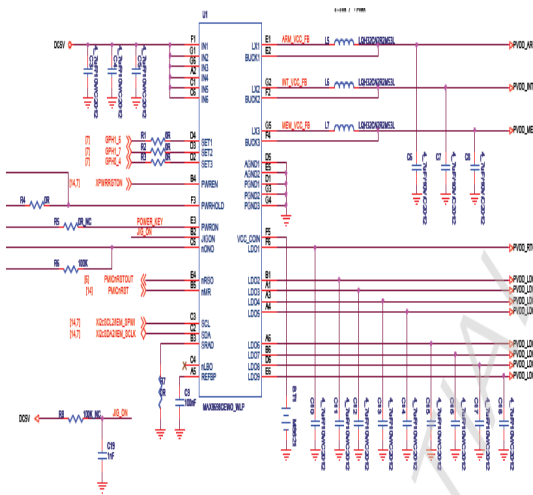


그림 8. PMIC 회로

Fig. 8 Circuit of PMIC

그림 8은 여러 전원을 관리하기 위해 PMIC(Power Management IC)를 사용하는데 본 논문에서는 Maxim사에서 개발한 Samsung 제품의 전용 PMIC를 사용 하

였다. 대부분의 전원은 CPU내부에서 사용되고 있고 VCC\_EXT전원은 외부 통신 인터페이스의 Reference 전원으로 사용되었다.

그림 9에서 나타난 5PV210의 Clock 회로는 4가지의 외부 클럭소스의 선택적 사용이 가능하며, 첫 번째로 XRTCXTI 클럭 소스는 CPU내부의 RTC(Real Time Clock)의 소스 클럭을 공급하고, 두 번째로 XXTI 클럭 소스는 시스템 클럭으로서 USB기능을 사용하지 않을 경우에 사용 가능하다. 세 번째 XHDMI 클럭 소스는 HDMI Video 출력 로직에 공급되는 클럭으로서 27MHz로 고정되었으며, 네 번째로 XUSBXTI 클럭 소스는 XXTI와 마찬가지로 시스템 클럭으로 사용할 수 있으며 USB를 사용할 경우 USB로직에 공급되는 클럭으로도 사용할 수 있다.

개발한 MPU 모듈의 경우 USB 기능을 사용하고 있으므로 USBTI에 24MHz의 클럭으로 사용 하고 있다. 외부 소스는 12M~50MHz까지 사용할 수 있으나 내부에서 사용하는 iROM의 1차 부트로더가 24MHz를 기준으로 설계되었기 때문에 삼성에서는 24MHz를 사용한다.

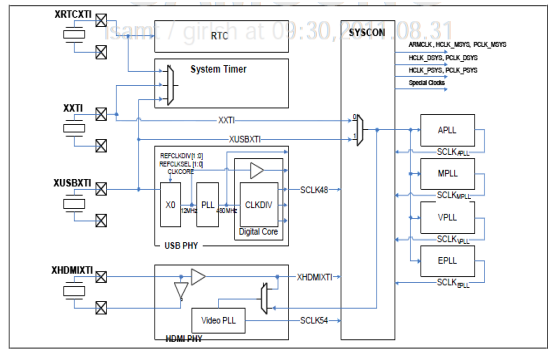


그림 9. MPU 모듈 (S5PV210)의 클럭 블록도

Fig. 9 Clock block Diagram of MPU module (S5PV210)

NAND Flash 는 8비트 버스구조로 되어 있는 Samsung의 제품을 사용한 것으로 그림 10에 나타냈다. NAND Flash는 일반 시스템버스와 동작하는 방식이 달라 별도의 컨트롤러가 필요하지만 S5PV210은 기본적으로 내부에 컨트롤러를 포함하고 있어 복잡한 설계 없이 쉽게 구현이 가능하다.

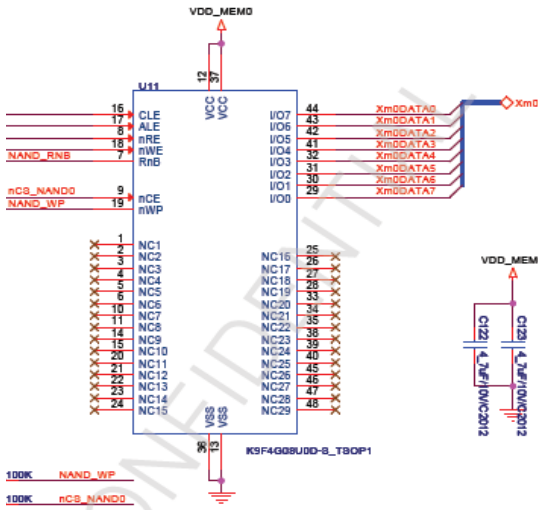


그림 10. NAND Flash 회로 설계  
Fig. 10 NAND flash circuit design

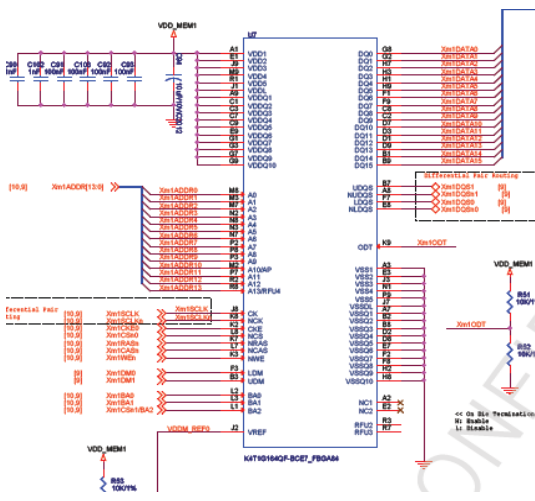


그림 11. DDR2 Memory 회로  
Fig. 11 DDR2 memory circuit

그림 11은 DDR2 Memory 회로를 나타낸 것으로 MPU모듈의 DDR2 메모리는 512MB를 기본적으로 사용할 수 있도록 설계되었으며, 추후 1GB까지도 업그레이드가 가능하도록 설계한다. DDR2 Clock소스는 200MHz로 동작하도록 되어 있으며 데이터 전송은 32 비트 버스에 400Mbps 동작 속도를 지원하고 있다. 높은 클럭을 사용하는 만큼 신호의 왜곡이 쉽게 발생할 수 있으며 Cross Talk의 영향에 취약할 수 있다

며, 이는 EMI의 발생을 일으키는 원인이 된다. 이러한 문제가 발생하는 근본적인 원인은 임피던스의 Miss Matching에서 비롯된 것으로 DDR2에서 부터는 이러한 문제를 극복하기 위해 On Die Termination기능을 제공함으로써 해결할 수 있다.

#### IV. 결론

ARM Cortex-A8 마이크로 아키텍처(Micro architecture)는 효율적으로 고성능을 제공하도록 설계되어 코어 클럭(Core Clock)의 향상은 물론 전체적인 성능의 향상을 가져왔다. 그리고 여러 기기에 대한 컨트롤러를 내장하여 시스템의 확장성을 향상시켰다.

본 논문에서는 임베디드 플랫폼 구현을 위해 S5PV210의 32Bit RISC 프로세서를 사용함으로써 비교적 낮은 단가와, 낮은 전력사용, 높은 성능을 제공하도록 하였다. S5PV210 CPU를 임베디드 플랫폼에 적용함으로써 ARM Cortex-A8 Core의 특징을 포함하도록 하였으며 동시에 ARMv7-A 아키텍처를 적용하여 구현하였다. 또한 안드로이드 기반의 ARM Cortex-A8 기반의 Core는 높은 성능을 달성할 수 있도록 설계함과 동시에 다양한 SoC 설계와 호환이 가능하도록 통합할 수 있다.

#### 감사의 글

본 연구는 교육과학기술부와 한국연구재단의 지역혁신 인력양성사업으로 수행된 연구 결과임

#### 참고 문헌

- [1] J. Leskela, J. Nikula, and M. Salmela, "OpenCL embedded profile prototype in mobile device," IEEE Workshop on Signal Processing Systems, 2009. SiPS 2009. pp. 279-284, Oct, 2009.
- [2] Vinh Quang La, " A study on Java Virtual Machine for Real-time embedded systems", CSSE, 2008.
- [3] HANBACK "Embedded Linux System"
- [4] <http://www.android.com>.
- [5] <http://www.Kandroid.org>.
- [6] Yeon-Jae Oh, Oh-Hoon Cho and ·Eung-Kon

Kim, "Design of 3D Ship Display System using Android", The Journal of The Korea Institute of Electronic Communication Sciences, Vol. 7, No. 5, pp. 1011-1016, 2012.

- [7] Phil-Joo Moon, "On the Availability of Anti-Forensic Tools for Android Smartphones", The Journal of The Korea Institute of Electronic Communication Sciences, Vol. 8, No. 6, pp. 855-861, 2013.
- [8] Chris Herring, "Microprocessors, Microcontrollers, and Systems in the New Millennium", IEEE Computer Magazine, Nov-Dec, 2000.
- [9] Steve Furber, ARM susem-on-chip architecture, 2nd edition. Addison-Wesley, Aug, 2000.
- [10] ARM, Architecture Reference Manual, 2003.
- [11] C. G. Kim, D. H. Lee, and J. Kim, "Optimizing Image Processing in Multi-core CPUs with Intel Parallel Programming Technologies," Multimedia Tools and Application, DOI:10.1007/s11042-011-0906-y, Nov, 2011.
- [12] Kuk-se Kim, Gil-choon Kim and Joon Lee, "Embedded Linux System for Self-Control System of Car", The Journal of The Korea Institute of Electronic Communication Sciences, Vol. 2, No. 1, pp. 62-66, 2007.
- [13] Hyun Huh and Jae-hak Lee, "A Study on Development of H8 MCU IDB(Integrated development board) for Embedded Education", The Journal of The Korea Institute of Electronic Communication Sciences, Vol. 4, No. 1, pp. 51-57, 2009.
- [14] ARM, Cortex-A8 Technical Reference Manual r3p2, ARM, 2009.
- [15] ARM, ARM1176JZF-S Technical Reference Manual r0p7, ARM 2008.
- [16] ARM, RealView Platform Baseboard for ARM-1176JZF-S User Guide, ARM, 2009.

저자 소개

**윤찬호(Chan-Ho Yoon)**



1997년 2월 호남대학교 컴퓨터공학과 졸업(공학사)  
 2000년 2월 조선대학교 대학원 컴퓨터공학과 졸업(공학석사)

2010년 전남대학교 대학원 컴퓨터공학과(박사과정)  
 ※ 관심분야 : 임베디드 시스템, ATM망, 인터넷 통신, 컴퓨터 네트워크, 실시간 통신 프로그래밍, 영상 처리 및 통신, 프로그래밍 언어(Visual C++, Java, 안드로이드)

**김광준(Gwang-Jun Kim)**



1993년 2월 조선대학교 컴퓨터공학과 졸업(공학사)  
 1995년 2월 조선대학교 대학원 컴퓨터공학과 졸업(공학석사)

2000년 2월 조선대학교 대학원 컴퓨터공학과 졸업(공학박사)  
 2000년~2001년 Dept. of Electrical & Computer Eng. Univ. of California Irvine Postdoc.  
 2003년~현재 전남대학교 컴퓨터공학과 부교수  
 ※ 관심분야 : 가상화, ATM망, 인터넷 통신, 컴퓨터 네트워크, 실시간 통신 프로그래밍, 영상 처리 및 통신, 프로그래밍 언어(Visual C++, Java, 안드로이드 등), 임베디드 시스템, 의료정보통신 등

**장창수(Chang-Soo Jang)**



1980년 2월 조선대학교 전자공학과(공학사)  
 1982년 8월 건국대학교 대학원전자공학과(공학석사)

1997년 2월 서강대학교 컴퓨터공학과(공학박사)  
 1984년~현재 전남대학교 컴퓨터공학과 교수  
 ※ 관심분야 : 가상화, 컴퓨터구조, 병렬처리 구조, 상호연결망, 마이크로 프로세서, 의료정보 통신 등