특 집

무선 멀티미디어 응용을 위한 ARM 기반 SoC 플랫폼 설계

장 준 영, 김 원 종, 조 한 진, 김 종 대

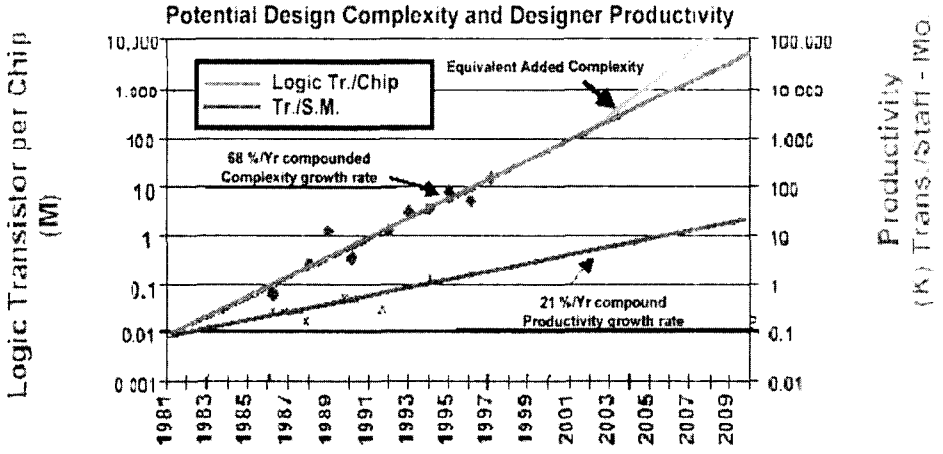
한국전자통신연구원 집적회로연구부

I. 서 론

실리콘 처리 기술의 고속화 요구와 유무선 환경하에서 동영상 통신이 가능한 비디오 폰, 영상회의 시스템, 3G-324M을 이용한 이동 통신용 단말기 등의 전자 제품 사용자의 급증은 시스템을 하나의 칩에 집적화하는 SoC(System-On-a-Chip) 설계 기술을 요구하고 있다. 〈그림 1〉과 같이 칩의 복잡도와 SoC 제품의 생산성 차이가 계속적으로 증가함에 따라 현재의 IC 설계 방법으로는 SoC 제품의 성능과 요구의 변화를 만족시킬 수 없다. 칩의 면적을 최소화하고 성능을 최대화하며 게이트 수준의 최적화를 통한 기존의 셀 기반 설계 방법으로는 설계의 생산성 문제를 해결할 수 없다. 이러한 문제를 해결 위한 새로운 설계 방법인 IP 재사용을 기반으로 한 플랫폼 기반 설계가 제시되었다[1, 2].

플랫폼 기반 설계는 SoC 제품을 빠르게 개발하기 위한 응용 기반 통합 플랫폼과 재사용이 가능한 IP(Intellectual Property) 또는 VC (Virtual Component)를 이용한 플랫폼 기반 설계(Platform-Based Design) 방법이다. 새로운 설계 방법은 90% 이상의 IP 재사용을 통해서 설계 시간을 단축하며, 시스템 수준에서의 최적화를 통해서 제품의 시장 경쟁력(Time-to-Market)의 문제를 해결하기 위한 방법이다[3, 4].

II장에서는 플랫폼 기반 설계에 대해서 설명하고, III장에서는 플랫폼 기반 설계 방법을 이용한 ARM/ASB 기반의 MPEG-4 코덱 설계, IV장에서는 AMBA AHB 버스를 이용한 고성능 MPEG-4 코덱 설계에 대해서 설명하고 결론을 맺는다.



〈그림 1〉 SoC 설계의 생산성 위기[6]

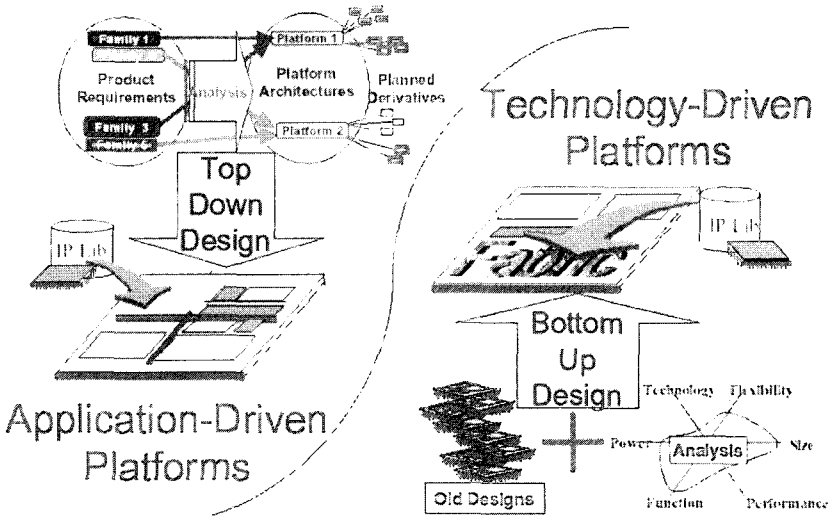
II. 플랫폼 기반 설계

1. 플랫폼 기반 설계 정의

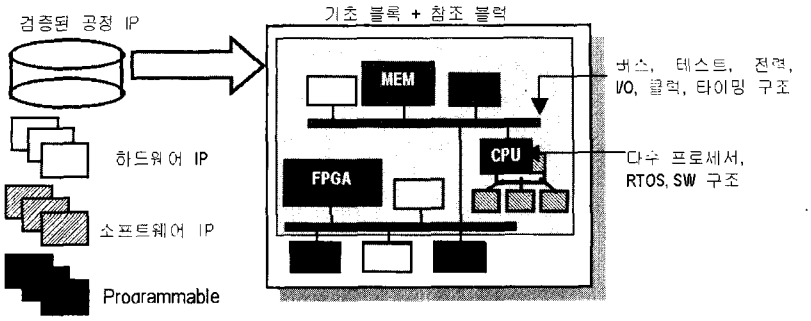
VSIA(*www.vsia.org*)의 플랫폼 기반 설계 그룹의 정의[7]에 따르면 SoC 설계에서 "플랫폼"이란 미리 정형화되고 통합된 하드웨어, 소프트웨어 IP 블록이나 모델 및 설계 도구를 말한다. 플랫폼의 라이브러리와 설계 방법에 따라서 구조적 탐색을 통해서 플랫폼의 구조를 결정하고 기존의 정의된 IP 블록이나 모듈을 적용 및 수정 보완하여 통합하고 검증한다. "플랫폼 기반 설계"는 복잡한 SoC 제품을 개발하기 위해 IP 재사용을 강조한 플랫폼 기반 통합 설계 접근 방법이다.

플랫폼 기반 설계에는 기술-기반 플랫폼(Technology-Driven Platform)와 응용-기반 플랫폼(Application-Driven Platform) 있다[8]. 기술-기반 플랫폼은 모듈 설계를 한 후에 성능 분석을 통해서 최적화 및 검증을 실행 한 후에 IP 라이브러리를 사용하여 칩을 설계하는 상향식 설계(Bottom-up Design) 방법이다. 응용-기반 플랫폼은 응용 분야의 요구 사항에 따라서 다양한 계열로 나누고, 이를 분석하여 적합한 플랫폼의 구조를 결정한 다음, 실제 IP 라이브러리를 적용하여 칩을 설계하는 시스템 수준 설계의 하향식 설계(Top-Down Design) 방법이다. 응용-기반 플랫폼은 상위 수준에서 응용 분야 따라서 다양한 성능 분석이 가능하므로 개발 실패 부담을 최소화할 수 있다. 미리 정의된 블록이나 모델들을 재사용하므로 개발 시간을 단축할 수 있는 장점이 있다. 플랫폼을 이용한 설계 방법은 높은 초기 플랫폼 설계 비용과 미리 정의된 플랫폼으로 인해서 설계의 유연성이 제한되는 어려움은 있으나 개발된 플랫폼을 이용하여 다양한 응용 분야를 개발하는데 시간과 비용의 감소를 통해서 보완될 수 있다. 〈그림 2〉는 이들의 개념을 나타낸 것이다.

응용-기반 플랫폼의 구조를 결정하는 가장 핵심적인 요소인 코어 모듈은 〈그림 3〉과 같이 프로세서와 버스 및 연결 구조, 주변 장치 IP, 응용 목적의 IP 등이 있다. 프로세서에서 동작하는 소프트웨어 모듈은 하드웨어를 동작시키기 위한 디바이스 드라이버, 펌웨어, RTOS, 응용 목적에 필요한 소프트웨어 및 라이브러리가 있다. 또한 플랫폼을 검증하기 위해서는 하드웨어와 소프트웨어 혼합 검증 환경이 필요하다.



〈그림 2〉 플랫폼 기반 설계



〈그림 3〉 플랫폼 구성 요소

2. 플랫폼 기반 설계 방법

플랫폼 기반 설계는 플랫폼을 설계하는 과정과 플랫폼을 사용하여 응용 분야 제품을 적용하는 과정으로 구분된다. 플랫폼을 설계하는 과정에서는 응용 분야에 따른 IP나 하드웨어 및 소프트웨어 모듈을 특성화하여 선택하며, 시스템 구조를 결정하고 성능과 전력 소모를 최적화하기 위한 성능 분석 과정을 통해서 플랫폼을 결정하고 선택한다. 플랫폼을 사용하는 과정에서는 플랫폼의 응용 분야에 필요한 IP 및 하드웨어 및 소프트웨어 모듈을 최적화하여 통합하고 검증을 실행한다.

〈그림 4〉는 SoC 설계를 위한 플랫폼 기반 설계 방법의 흐름도이다[2, 5]. 플랫폼을 구성하는 과정에서는 먼저 적용하고자 하는 응용 분야의 영역과 제품의 발전 방향을 조사하고 정의한다. 초기 플랫폼을 구성하기 위해서 초기 플랫폼 구조를 구성하는 프로세서와 DSP를 선정하고 데이터 전송 및 처리를 위한 온-칩 버스와 내부 및 외부 메모리를 선택한다. 응용 분야에 적합한 하드웨어 및 소프트웨어 IP를 결정하고 IP에 필요한 Wrapper를 설계한다. 성능 분석 및 플랫폼 선택 및 구성 과정에서 메모리 Bandwidth와 전력 소모 및 성능들을 분석 및 평가하여 응용 분야의 요구 사항을 만족하는 플랫폼을 선택하고 구성한다. 플랫폼 Refinement 단계에서는 응용 분야를 개발하기 위한 플랫폼을 조정하고, 최적화하는 과정으로 버스 크기, 클럭 속도 데이터 전송 프로토콜에 따라서 각 모듈들을 최적화한다. 하드웨어와 소프트웨어 모듈을 분할하고 소프트웨어의 싸이클을 측정하여 태스크 스케쥴링을 생성한다. 플랫폼 구현과 검증 단계에서는 분할된 소프트웨어 모듈은 플랫폼의 프로세서에서 실행되며 하드웨어 모듈은 플랫폼에 연결된다. 소프트웨어와 하드웨어 사이의 혼합 검증을 통해서 구현된 플랫폼을 검증한다.

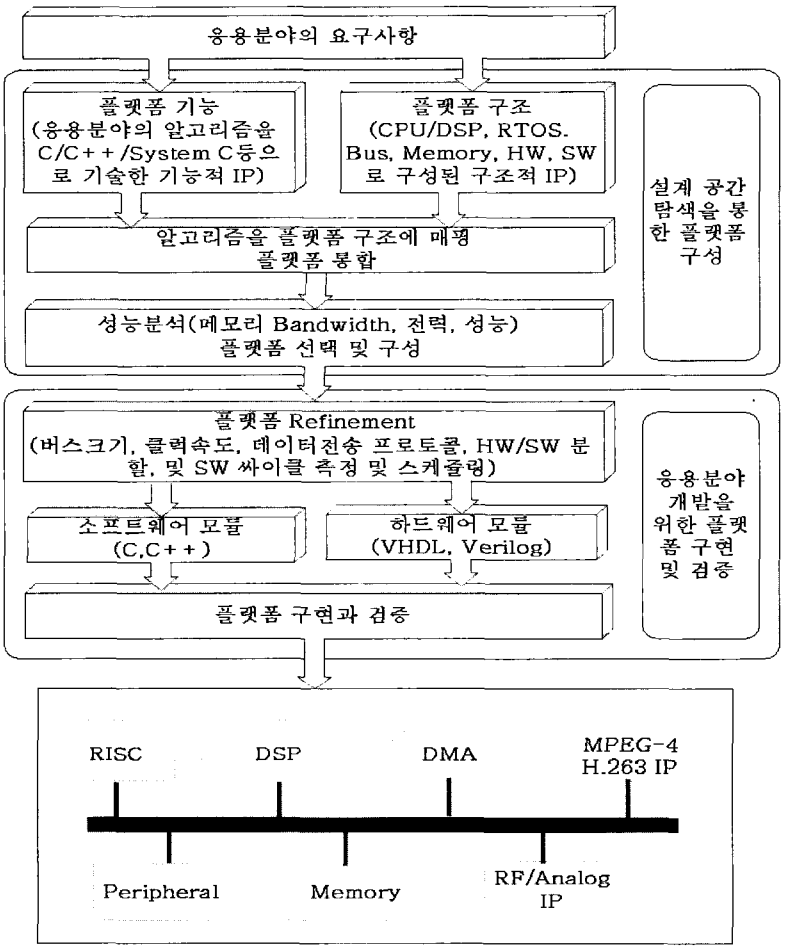
플랫폼의 종류는 프로세서 코어와, 버스 구조와 소프트웨어 구조를 통해서 응용 분야에 맞는 플랫폼을 설계한다. 현재 사용되는 SoC 플랫폼의 형태는 다음과 같이 4가지 형태로 분류된다.

1) 완전 응용 플랫폼
(Full Application Platform)

하드웨어와 소프트웨어 라이브러리 세트와 다양한 매핑 및 응용 예제가 지원되는 플랫폼으로 TI's OMAPTM(tm), Philips's Nexperia(tm), Infineon's Mgold, Tality's ARM-based SoC 등이 있다. 이 플랫폼은 응용 분야에 따라 플랫폼을 모델링을 위한 많은 노력이 필요하다.

2) 프로세서 중심 플랫폼
(Processor Centric Platform)

프로세서와 주변 장치들로 구성된 플랫폼으로 기본적인 소프트웨어 드라이버와 기본 응용 루틴이 지원되며 플랫폼 모델링을 위한 제한적인 노력이 필요하다. 대표적인 플랫폼으로는 Improv's PSATM Jazz, ARM's Micropack(tm), ADS (ARM Development System), ST's Starcore Platform 등이 있다.



〈그림 4〉 플랫폼 기반 설계 방법

3) 통신 중심 플랫폼

(Communication Centric Platform)

OCP(Open Core Protocol)을 사용하는 통신 프레임 워크와 주변 장치를 지원한다. 제한된 플랫폼 모델링 노력이 필요하며 응용 분야의 모듈을 가장 쉽게 연결할 수 있는 플랫폼이다. 대표적인 플랫폼으로는 Sonic's SiliconBackplane (tm) MicroNetwork, Palmchip's PalmPak (tm) 등이 있다.

4) 프로그램이 가능한 플랫폼

(Highly Programmable Platform)

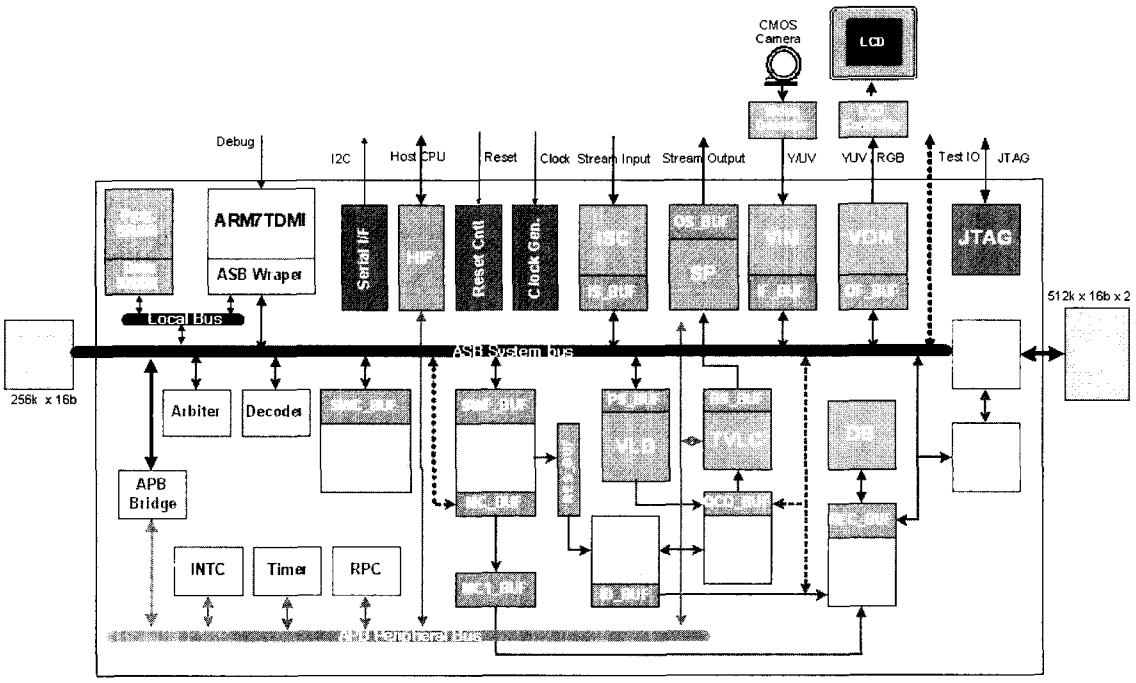
플랫폼 재구성에 중점을 둔 플랫폼으로 재구성이 가능한 프로세서와 프로그램이 가능한 로직이 제공된다. 대표적인 플랫폼으로 Triscend's A7, Altera's Excalibur, Xilinx's Platform FPGA 등이 있다.

Ⅲ. ASB 기반 MPEG-4 플랫폼

MPEG-4 영상 코덱 칩은 저전력 32 비트 내장형 RISC 프로세서인 ARM7TDMI 코어와 최적화된 ABMA 버스 인터페이스 하드웨어 및 펌웨어로 설계되었다[9]. 설계된 영상 코덱 칩은

27MHz에서 동작하며 QCIF는 초당 30 프레임 또는 CIF에서 초당 7.5 프레임을 처리할 수 있다. AMBA의 ASB 버스를 기반으로 한 ASB 플랫폼에 매핑된 MPEG-4 영상 코덱용 칩의 전체 구조도는 〈그림 5〉와 같다. MPEG-4 영상 코덱용 칩은 크게 3개의 블록으로 구성되는 데 영상 데이터의 픽셀 전후 처리 모듈인 VIM (Video Input Module), VOM(Video Output Module)과 스트림 인터페이스 모듈인 ISC(Input Stream Controller)와 SP(Stream Producer)로 구성된다. 영상 데이터의 압축 효율을 높이기 위한 움직임 추정 및 보상을 위한 모듈은 동작 추정(ME : Motion Estimation), 동작 보정(MEFMC : Motion compensation)과 영역 재구성을 위한 REC(Re-Construction) 및 양자화를 위한 이산여현변환(DCT : discrete cosine transform), 역이산여현변환(IDCT : inverse DCT)과 가변 길이 변환 모듈인 TVLC (Variable Length Coding), LVLD(Variable Length Decoding) 구성된다. 영상 코덱 칩의 제어와 스케줄링 기능 및 테스트를 실행하는 ARM 프로세서와 영상 코덱 처리를 위한 파라메터 입력을 위한 Host Interface와 외부 메모리와 하드웨어 내부 메모리 사이의 데이터 전송을 위한 DMA와 외부 메모리 인터페이스를 위한 EMI(External Memory Interface)와, 타이머, 인터럽트 제어기로 구성된다.

MPEG-4 영상 코덱 칩은 저전력 내장형 RISC 프로세서인 ARM7TDMI 코어와 변형된 AMBA의 ASB와 APB 버스를 사용한다. 메모리 계층 구조는 외부 메모리로 프로세서에서 처리될 프로그램 저장을 위한 EPROM(256K×16bit)과 영상 데이터를 저장하기 위한 SDRAM(512K×16bit×2bank)로 구성된다. 내부 메모리는 프로세서에 의해 실행될 프로그램과 데이터를 위한 내부 메모리인 SRAM과 병렬 처리를 위해 필요한 최소한의 하드웨어 모듈의 내부 로컬 메모리로 구성된다. 내부 로컬 메모리는 I/O 패드를 통한 외부 프레임 메모리의 접근을 줄이기 위해 사용된다. 16M 비트의 외부 프레임 메모리인



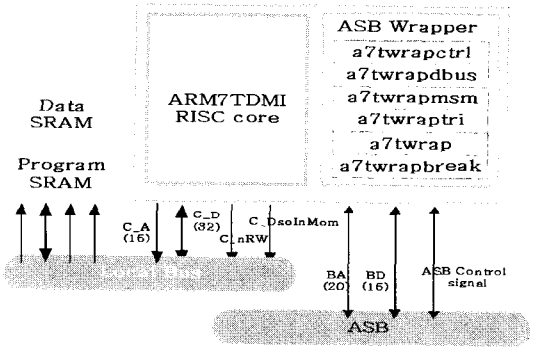
〈그림 5〉 MPEG-4 영상 코덱 구조도

SDRAM은 16비트 메모리 버스를 통해서 EMI에 연결된다. 내부 로컬 메모리는 EMI를 통해서 외부 SDRAM메모리에 데이터를 전송한다. EMI의 DMA 제어기를 사용하므로 일반적인 DMA보다 전력 소모가 적고, 고성능인 데이터 전송을 할 수 있다. 일반적인 DMA 제어기는 내부, 외부 메모리에 데이터를 전송하기 위해 내부에 임시 버퍼를 가지고 있으나, EMI의 DMA 제어기는 내부에 데이터 전송을 위한 로컬 버퍼를 가지고 있지 않고 내부, 외부 메모리의 시작 주소, 전송할 데이터의 양과 전송 방향에 따라서 데이터를 전송한다.

1. ARM7TDMI와 ASB Wrapper

MPEG-4 영상 코덱 칩의 프로세서는 저전력 내장형 RISC 프로세서인 ARM7TDMI 코어는 내부 메모리에 펌웨어로 저장된 32KB의 명령어와 데이터를 처리한다. 변형된 AMBA 인터페이스를 갖는 시스템 버스 인터페이스는 16비트 ASB 버스와 8비트 APB 버스로 구성된다. 프로세서는 시스템을 기동하기 위한 부트 코드를 실행하고, 비트 스트림 구문과 속도 제어(rate control)와 파이프라인 처리를 위한 내부 모듈에 있는 레지스터의 파라메터 초기화와 전체 파이프라인 스케쥴링과 같은 복잡한 과정을 처리한다. RISC 프로세서는 EMI를 포함한 모든 하드웨어 모듈을 제어한다. ARM 프로세서는 일반적으로 AMBA의 ASB Wrapper를 통해서 ASB 버스에 연결되어 있고 프로그램이나 데이터를 저장하는 내부 메모리도 ASB 버스에 연결된다. RISC 프로세서가 프로그램이나 데이터를 처리하기 위해서 시스템 버스인 ASB 버스를 사용한다.

영상 코덱 칩에서는 〈그림 6〉에서와 같이 내부 메모리를 접근하기 위해 ASB 버스를 사용하는 것을 분리하기 위해서 내부 메모리를 로컬 버스를 통해서 직접 ARM 프로세서와 연결한다. 내부 로컬 버스는 프로그램이나 데이터 접근을 위해서 32비트 주소와 32비트 데이터 버스를 사용하며 시스템 버스인 ASB와는 독립적으로 동작한다. ARM 프로세서와 내부 메모리 사이의 로컬 버스를 사용하여 프로그램의 명령어와 데이터를 처리하고, ASB 시스템 버스는 시스템 내의 각 하드웨어 모듈의 영상 데이터를 처리하는데 할당하므로 병렬 처리를 통해서 전체 시스템의 성능을 향상시킬 수 있다. ARM 프로세서는 ASB Wrapper를 통해서 ASB 버스와 인터페이스를 한다. ARM 프로세서는 ASB 버스 마스터로 동작하며 20비트의 주소 버스와 16비트 데이터 버스와 ASB 버스 제어를 위한 제어 신호로 연결된다. ARM 코어와 ASB 버스 인터페이스를 위해서 ASB Wrapper 모듈은 영상 코덱 칩 설계에 필요한 형태로 수정된 6개의 모듈로 구성된다.



〈그림 6〉 내부 메모리 분리와 ASB Wrapper

A7WrapMSM는 main state machine으로 ASB 버스의 사용권을 인가하는 모듈이다. ARM 프로세서는 ASB 버스의 반응 신호를 입력으로 받아 상태에 따라서 버스의 사용권을 인가하는 신호를 발생한다. A7Wraptri는 ARM 프로세서의 출력인 주소(BA), 데이터(BD), 제어 신호를 출력하기 위한 Tri-state Driver이다. A7WrapDbus는 ARM 프로세서로 입력과 출력되는 데이터 버스를 연결하는데 사용된다. ARM 프로세서는 32비트 단일 방향 데이터 입력(DIN), 데이터 출력(DOUT)을 사용하고, DIN과 DOUT은 내부 메모리를 위한 로컬 버스의 32비트 데이터 버스인 32비트의 C_D에 직접 연결된다. ASB의 데이터 버스인 BD는 두 개의 신호를 함께 연결하여 16비트 양방향 버스로 사용한다. 이는 영상 프레임을 처리하는 외부 메모리인 SDRAM (512K×16bit×2bank)은 16비트 단위로 데이

터를 처리하므로 ASB 버스의 데이터 버스는 16 비트 데이터 전송을 사용한다. A7WrapCtrl은 ARM 프로세서의 정상 모드와 테스트 모드 동작에 필요한 제어 신호를 결정하기 위해 사용된다. A7WrapBreak 모듈은 버스 마스터인 ARM 프로세서가 DMA에게 ASB 버스 사용권을 인계하기 위한 제어 신호로 사용된다. A7Wrap 모듈은 ASB Wrapper의 최상위 모듈로 ARM 프로세서 신호를 ASB 버스에 연결하기 위한 인터페이스 모듈이다.

2. ASB 버스 구조

영상 코덱 칩을 위한 ASB/APB 버스 구조는 AMBA의 시스템 버스인 16비트 ASB 버스와 내부 메모리를 접근하기 위한 로컬 버스와 주변 장치 버스인 8비트 APB 버스로 구성되고 <그림 6>과 같다. 영상 코덱 칩의 ASB 버스 구조는 준-이중 버스 구조로 구성되는데 ARM 프로세서와 내부 메모리를 직접 연결하는 로컬 버스와 16비트 시스템 버스인 ASB 버스와 8비트의 주변 장치 버스인 APB 버스로 구성된다. ARM 프로세서와 내부 메모리 사이의 로컬 버스를 사용하여 프로그램의 명령어와 데이터를 처리하므로 ASB 버스를 DMA와 하드웨어 모듈 사이의 영상 데이터를 처리하는데 할당하므로 병렬 처리를 통해서 전체 시스템의 성능을 향상시키는 구조이다. ARM 프로세서는 로컬 버스를 통해서 내부 메모리로부터 명령어를 읽어서 처리하고 생성된 제어 데이터를 ASB 버스를 통해서 하드웨어 모듈의 제어 레지스터에 전달한다.

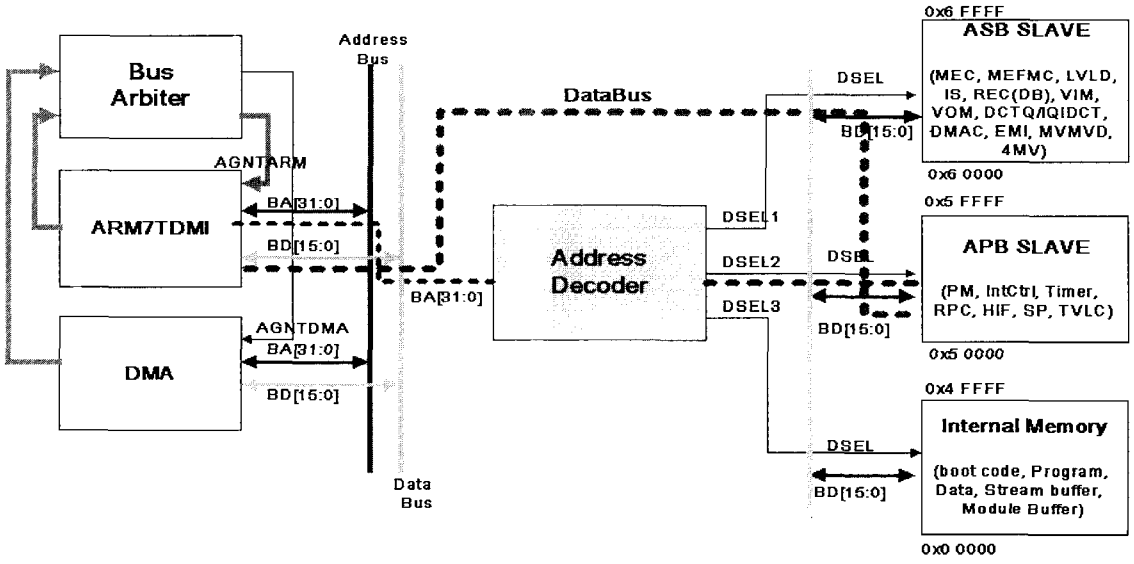
ASB 버스는 고성능, 파이프라인 처리, 버스트 모드 데이터 전송 다중 버스 마스터 기능을 지원하는 시스템 버스로서 ARM 프로세서와 DMA 버스 마스터, 내부 메모리와 하드웨어 모듈의 로컬 메모리인 SRAM으로 구성된 원-칩 메모리 블록, 프로그램과 데이터를 저장하고 있는 프로그램을 저장하는 외부 메모리인 EPROM과 영상 프레임을 저장하는 SDRAM을 위한 외부 메모리 인터페이스(EMI)로 구성된다. ASB 버스 인터페이스를 위해 버스 마스터를 인가하기 위한 state machine으로 구성된 버스 중재기(Arbiter)와 메모리-맵된 ASB 하드웨어 모듈의 주소 디코딩을 위한 Decoder와 APB 모듈과 인터페이스를 하기 위한 APB Bridge로 구성된다. HIF는 ARM 프로세서와 외부의 Host 프로세서 사이의 데이터 전달을 하는 모듈로서 I2C 직렬 버스 인터페이스와 인텔/모토롤라의 8비트 병렬 인터페이스를 지원한다. HIF를 통해서 MPEG-4 영상 코덱을 처리하기 위한 파라메타를 전송한다. 영상 코덱 칩의 파이프라인 동작과 스케쥴링 및 테스트를 수행하는 ARM 프로세서와 외부 메모리와 내부 메모리 사이의 영상 데이터를 처리하기 위한 DMA 및 EMI와 영상 데이터 입출력을 위한 VIM, VOM이 연결된다. 비디오 입력 모듈인 VIM은 외부 CMOS 이미지 센서로 입력된 영상 데이터를 MPEG-4의 표준 입력 형태인 4 : 2 : 0(Y : Cr : Cb) 형태로 변환하여 내부 메모리에 저장한다. 비디오 출력 모듈인 VOM은 RGB와 LCD 디스플레이 장치를 위한 다양한 Y, Cr, Cb 형태를 출력한다. 개략적인(Coarse Resolution) 움직임 추정을 위한 MEC와 완벽한 영상(Fine Resolution) 움직임 추정을 위한 MEF 모듈은 전력 소모를 줄이기 위해 3단계 계층적 움직임 추정을 수행한다. 움직임 보상을 위한 MC 모듈은 움직임 추정을 위한 데이터를 생성을 조정하고 또한 움직임 보상을 실행한다. 영상 데이터의 공간적 여분을 줄이기 위해 이산여현변환(DCT : discrete cosine transform), 역이산여현변환 IDCT(Inverse DCT)가 사용되고, 영상 데이터의 압축률을 높이기 위해 AC/DC 예측 알고리즘이 사용된다. 가변 길이 변환 모듈인 TVLC(Variable Length Coding), LVLD(Variable Length Decoding)는 DCT에서 계산된 Coefficient를 가변길이부호화를 하거나 가변길이부호화된 데이터에서 DCT에 사용될 Coefficient를 생성한다. DB(De-Blocking) 모듈은 압축된 비디오 영상의 품질을 높이기 위해서 사용되는 모듈로서, 블록 기반 압축 부호화 후 복원된 영상에 대해 화질 저하를 느끼게 하는 요소 중 하나인 블록킹 효과를 제거

하기 위한 후처리 모듈이다.

APB 버스는 AMBA의 주변 장치 버스로서 비교적 느린 데이터 접근을 필요로 하는 모듈들로 구성되며, 속도가 느린 반면 매우 적은 전력(power)를 소모한다. APB에 연결된 주변 장치 중 인터럽트 제어기는 ARM프로세서 코어에 인터럽트를 걸기 위하여 하드웨어 인터럽트 신호를 생성한다. 인터럽트를 원하는 모듈이 여러 개가 있는 경우 이들 중에서 우선 순위가 가장 높은 모듈을 선택하는 기능도 수행한다. 영상 코덱 칩에서 인터럽트 제어기는 7개 이상의 인터럽트 채널을 지원한다. ARM의 소프트웨어 인터럽트, 예외 상황 처리를 위한 인터럽트, ARM 프로세서를 재-기동시키기 위한 인터럽트, DMA 종료를 알리기 위한 DMA 인터럽트와 Decoder용 타이머 인터럽트, Encoder용 타이머 인터럽트, 프레임용 타이머 인터럽트를 사용한다. 타이머(Timer)는 영상 코덱 처리의 각 단계별로 필요한 인터럽트 신호를 발생하며 Timer0,1,2 3개의 타이머를 사용한다. Timer0는 프레임 데이터의 동기화를 위한 타이머로 사용되며 프레임율에 따른 전력 감소 지원 및 프로그램 제어 지원 기능을 수행한다. 영상 코덱의 인코딩 과정에서 인코딩 초기화를 위해서 Timer1을 사용하고 파이프라인 주기에 따라 타이머 인터럽트를 발생한다. 디코딩의 시작 시점은 Timer2를 ARM 프로세서가 초기화함으로 주기적으로 타이밍 인터럽트를 발생한다. RPC(Remap/Pause Controller)는 리셋 초기 단계에서 초기 메모리 맵을 설정하고 전력 소모를 줄이기 위한 인터럽트를 기다리는 pause 모드를 실행한다. 설계된 영상 코덱 칩은 H.263 ITU-T 권고 사항인 27Mhz로 QCIF 영상을 초당 30프레임(30 frames/second)을 엔코딩과 디코딩이 가능하며, CIF 영상을 코덱 모드로 초당 7.5 프레임(7.5 frames/second) 처리할 수 있다.

3. Arbitration과 Decoding 구조

AMBA의 버스 중재기(Arbiter)는 ASB 버스의 사용권을 요구하는 버스 마스터에게 우선순위에 따라 사용권을 인가한다. 버스 중재기는 표준 다중 마스터 버스로서 버스 중재기는 한 시점에서 하나의 버스 마스터에게만 버스 사용권을 허가한다. 영상 코덱 칩에서는 리셋 후 고정 마스터는 ARM 프로세서에게 주어진다. ARM 프로세서와 DMA 두 개의 버스 마스터를 사용하며



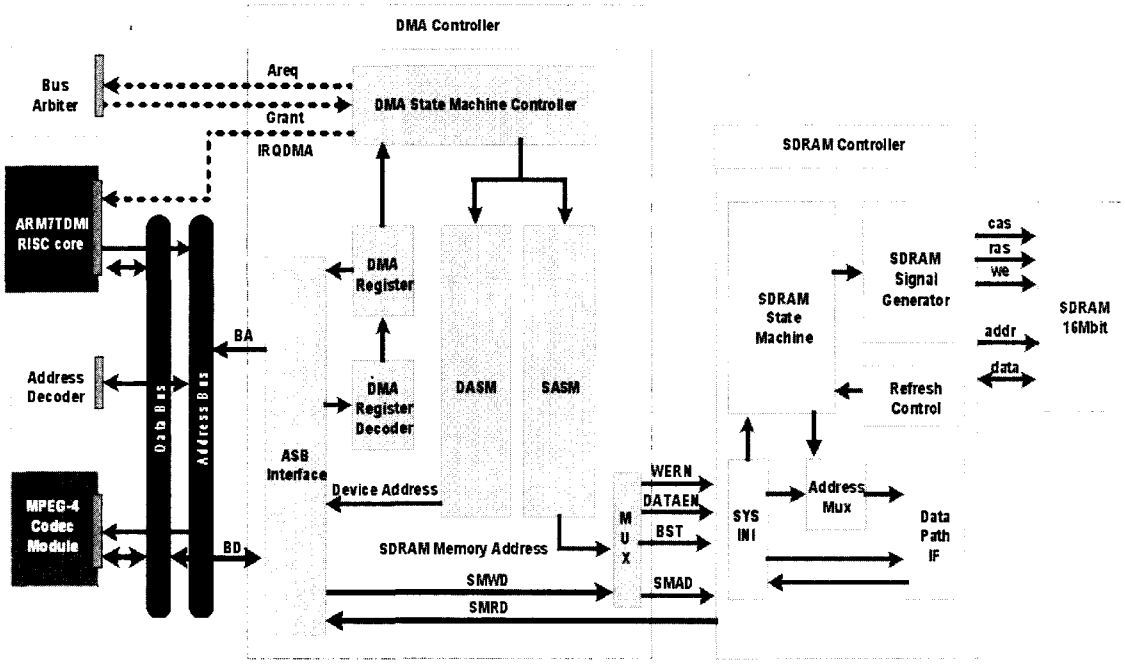
〈그림 7〉 버스 중재기와 Decoder 블록도

우선 순위는 DMA가 가장 높고 다음으로 ARM 프로세서가 우선순위를 갖는다. 각 버스 마스터는 버스 사용을 요청하고 버스 중재기는 우선순위가 가장 높은 버스 마스터를 결정하고 버스 사용권을 인가한다.

ASB의 Decoder는 주소로부터 지정된 ASB 모듈을 기동시키는 역할을 수행한다. 〈그림 7〉에서 영상 코텍 칩의 주소 Decoder에 의해서 할당된 메모리는 크게 ARM 프로세서에 의해 사용되는 내부 메모리의 프로그램이나 데이터를 위한 주소 공간과 MPEG-4 하드웨어 모듈의 로컬 메모리를 위한 스트림 및 버퍼 모듈 공간, APB 버스에 연결된 주변 장치를 위한 APB 모듈 공간과 ASB 버스에 연결된 ASB 모듈 공간으로 구성된다. 영상 코덱 칩에서는 시스템 리셋 후에 ARM 프로세서에 의해서 프로그램을 실행하고 DMA에 의해서 ASB 버스 요청이 들어오면 DMA의 제어 레지스터를 설정 한 후에 ASB 버스의 사용권을 DMA에 양도한다. ASB 버스 사용권을 인계받은 DMA는 ARM 프로세서와는 독립적으로 ASB 버스를 이용하여 슬레이브 하드웨어 모듈과 데이터 전송을 수행한다. 따라서 DMA 버스 마스터를 통해서 MPEG-4에서 처리되는 영상 데이터를 ASB 버스를 통해서 전송할 수 있다.

4. MPEG-4용 DMA 제어기

DMA 제어기는 시스템 버스(한 개의 채널)를 통하여 시스템 ARM 프로세서의 중재 없이 여러 내부 모듈과 외부 메모리(SDRAM)간에 데이터 전송을 수행하는 동작을 한다. 〈그림 8〉에서 DMA 동작은 슬레이브 모드와 마스터 모드로 구분되고, 슬레이브 모드에서 ARM 프로세서에 의해서 데이터 전송 소스와 목적지의 레지스터를 초기화되고, 소프트웨어적으로 DMA request가 시작된다. 그때부터 DMA 제어기는 마스터가 되어 시스템 주소와 데이터 버스를 제어할 수 있어서 stop 조건을 만날 때까지 데이터를 전송한



〈그림 8〉 DMA 제어기의 구조

다. DMA 제어기는 소스 또는 목적지 주소를 생성하며, 버퍼 없는 이중 주소 모드 동작을 수행한다. 데이터 전송이 끝나면 DMA 동작 완료를 알리고, 슬레이브 모드로 돌아간다. 영상 코덱 칩을 위해서 설계된 DMA는 동작 속도는 27MHz에서 동작하며 ASB 버스 형식에 적합하도록 되었다[5]. 다중 채널 대신에 한 개의 채널인 AMBA 버스를 통해서 외부 메모리인 SDRAM과 내부 메모리인 하드웨어 로컬 메모리 사이에 데이터 전송을 한다.

DMA는 16비트용 SDRAM을 사용하므로 DMA는 16비트 고정 단위 DMA 전송이 가능하며, 블록 모드 및 패킷 전송을 지원하고 내부 버퍼 메모리없이 양방향 주소를 이용한 DMA 전송을 한다. 소스와 목적지를 위한 양방향 주소가 필요하고 두 개의 버스 트랜잭션이 필요하다. 먼저, 소스로부터 데이터를 읽어서 목적지에 데이터를 저장하는 트랙잭션이 필요하다. DMA 제어기는 슬레이브 모드에서 ARM 프로세서의 스케쥴러에 의해 ASB 버스를 통해 DMA 레지스터에 데이터 전송할 디바이스 선정, 디바이스 시작 주소, 전송 방향, 전송 길이, 외부 메모리의 시작 주소 등을 초기화한다. DMA의 제어 레지스터인 현재 상태 레지스터인 CCR(Channel Control Register)에 시작 신호를 기동시키면, 시스템 버스 중재기에 버스 요청 신호를 내고(AREQ), 이 요청 신호를 받아서 인가(Grant) 신호를 보낸다. 이 신호를 받으면, DMA는 초기화된 레지스터에 따라 내부 모듈 주소 및 외부 모듈의 주소를 생성한다. DASM(Device Address State Machine)는 내부 모듈 메모리의 주소를 생성하는 블록으로, DMA 레지스터에서 입력 받은 전송 길이, 블록 사이즈, 시작 주소 등을 입력 받아 내부 모듈 메모리의 주소를 형성한다. SASM (SDRAM Address State Machine)는 외부 메모리의 주소를 생성하는 블록으로, DMA 레지스터에서 입력받은 전송 길이, 블록 사이즈, 시작 주소 등의 값을 받아서 외부 메모리의 주소를 생성한다. DASM(Device Address State Machine)에서 만들어진 주소는 AMBA 인터페이스를 통해 각 내부 모듈 메모리의 주소가 된다. 외부 메모리의 주소는 SASM에서 만들어져 EMI를 통해 외부 메모리의 주소가 된다. 주소에 동기되어 데이터 전송이 AMBA 인터페이스를 통해서 각 내부 모듈로 또는 내부 모듈에서 SDRAM으로 전송된다. 마지막 데이터의 전송이 끝났으면 DMADONE 신호를 내고 DMA 마스터 버스의 사용이 끝났음을 시스템 버스 중재기에게 알린다.

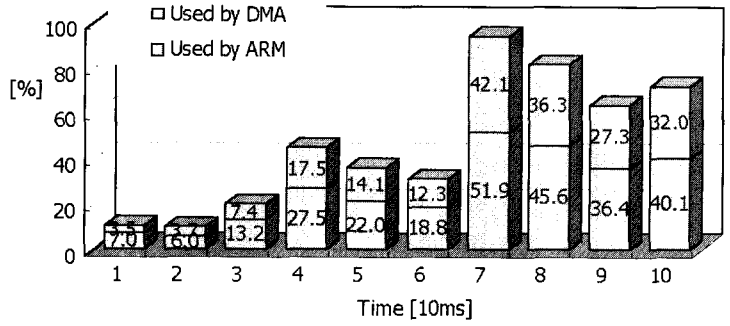
5. ASB 플랫폼 성능 분석과 실험 결과

MPEG-4용 ASB 기반 플랫폼은 기본 ASB 플랫폼을 MPEG-4 영상 코덱을 처리할 수 있는 형태로 ASB 시스템 블록과 주변 장치 블록을 최적화하였으며 MPEG-4 영상 데이터 처리의 성능을 향상시키기 위해서 MPEG-4용 DMA를 설계하여 사용하였다. <표 1>은 ASB 기반 플랫폼의 버스 마스터로 동작하는 두 개의 마스터인 ARM 프로세서와 DMA의 ASB 버스의 점유율을 분석한 데이터이다. 이 데이터는 영상 코덱을 10ms 간격으로 시뮬레이션하여 각 구간당 ARM 프로세서와 DMA가 어느 정도 ASB 버스를 점유하는가 분석한 것이다.

실험 결과, 버스 마스터인 ARM과 DMA가 버스를 최대 94% 이상 점유하고 있는 것을 알 수 있습니다. 따라서 현재 설계된 MPEG-4 영상 코덱은 최대로 버스를 이용하고 있는 것을 알 수 있고 MPEG-4의 성능을 향상시키기 위해서는 ASB보다 성능이 우수한 AHB 버스를 고려할 필요가 있다는 것을 알 수 있다. ASB 기반 플랫폼에서 설계된 영상 코덱 칩은 MPEG-4/H.263 영상 코덱 처리가 가능하며 QCIF(176×144) 영상을 초당 30프레임, CIF(352×288) 영상을 7.5 프레임 처리할 수 있다. 따라서 MPEG-4 영상 코덱이 CIF 25~30 fps를 처리하기 위해서 AMBA 버스에서 고성능 데이터 전송이 가능한 AHB 버스 구조가 필요하다는 것을 알 수 있다.

〈표 1〉 MPEG-4 영상 코덱의 ASB 버스 점유율

Time [10ms]	1	2	3	4	5	6	7	8	9	10
Used by ARM	7.0	6.0	13.2	27.5	22.0	18.8	51.9	45.6	36.4	40.1
Used by DMA	3.5	3.7	7.4	17.5	14.1	12.3	42.1	36.3	27.3	32.0
BUS Usage	10.5	9.7	20.6	44.9	36.1	31.1	94.0	81.9	63.7	72.1



Ⅳ. AHB 기반 MPEG-4 플랫폼

1. AHB 기반 플랫폼을 위한 사양 분석

ASB 기반 플랫폼을 이용한 MPEG-4 설계에서 16비트 버스를 사용하고 27MHz로 QCIF 30fps를 처리하는 경우 ARM 프로세서와 DMA의 ASB 버스의 점유율이 94% 이상이므로 성능이나 사양이 4배 정도 높은 CIF 25~30 fps를 처리하기 위해서는 ASB 기반 플랫폼으로는 처리가 불가능하다는 것을 알 수 있다. 〈표 2〉에서와 같이 CIF 25 fps를 처리하기 위해서는 2배 정도의 동작 주파수가 필요하며, 코덱 영상을 구성하는 CIF 영상의 매크로 블록(MB)의 수와 1 MB를 처리하는 싸이클 수가 약 4배 정도 증가하므로 이에 따른 플랫폼의 구조가 필요한 것을 알 수 있다. 이를 위해서 동작 속도를 54 MHz로 높이고 데이터 전송을 위해서 32비트 버스를 이용하는 구조를 지원하는 AHB 기반 플랫폼 구조가 선택되었다. AHB 기반 플랫폼의 구조에 MPEG-4의 하드웨어와 펌웨어 모듈들을 AHB 인터페이스를 통해서 연결한다.

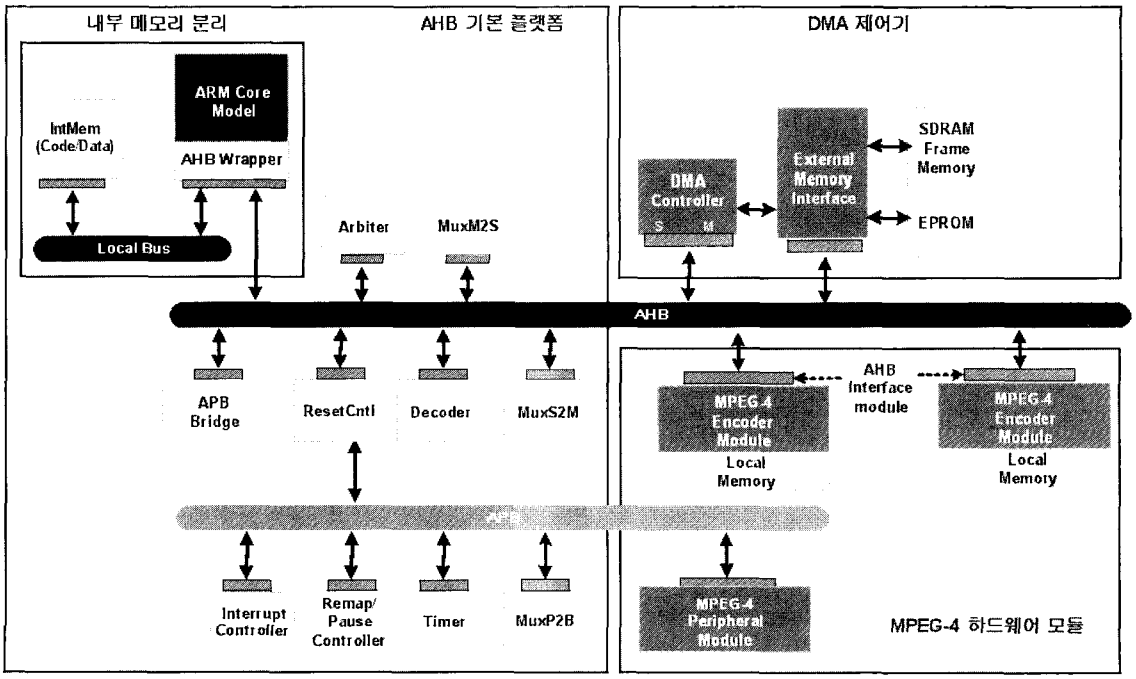
〈표 2〉 MPEG-4 영상 코덱의 사양 분석

	QCIF 30 fps	CIF 25 fps
동작 주파수	27MHz	54 MHz
버스 Width	16 bit	32 bit
파이프라인	Encoder : 3개 Decoder : 4개	Encoder : 7개 Decoder : 4개
MacroBlock의 수	Encoder : 105 MB Decoder : 118 MB	Encoder : 416 MB Decoder : 472 MB
1MB 처리 Cycle 수	Encoder : 4,513 Cycle Decoder : 3,600 Cycle	Encoder : 2,721 Cycle Decoder : 2,177 Cycle
프로세서	ARM7TDMI	ARM7TDMI
버스 구조	ASB/APB 버스	AHB/APB 버스

2. AHB 기반 플랫폼 설계

MPEG-4 CIF 25 fps 영상 코덱 처리를 위한 AHB 기반 플랫폼은 ASB 기반 플랫폼과 같이 ARM 프로세서에 AMBA의 AHB와 APB 버스로 구성되었다. AHB 버스를 근간으로 하는 AHB 기반 플랫폼의 구조는 〈그림 9〉와 같다. AMR 프로세서는 AHB wrapper로 연결되어 있으며, AHB Wrapper는 AHB 버스와 ARM 프로세서 사이의 AHB 데이터 전송 프로토콜을 제어한다. AHB 기반 플랫폼은 AHB 버스에 ARM 프로세서와 AHB 버스의 마스터를 인가하는 Arbiter, 메모리 영역을 디코딩하는 Decoder와 외부 메모리의 내용을 내부 메모리에 다운로드 하는데 사용되는 외부 메모리인 EPROM과 프로그램 코드와 계산 데이터를 저장하기 위한 내부 메모리가 연결되어 있다. 외부 메모리인 SDRAM에 저장된 영상 데이터를 전송하기 위한 SDRAM 제어기와 버스 마스터로 SDRAM과 하드웨어 모듈의 로컬 메모리 사이에 데이터 전송을 전담하기 위한 DMA가 있다. APB Bridge는 AHB 버스와 AHB 버스를 연결하는 인터페이스 모듈이다. APB 모듈에는 타이머와 외부 인터럽트를 제어하기 위한 인터럽트 제어기와 Remap/Pause 제어기가 연결되어 있다.

AHB 기반 플랫폼의 유연성을 향상 시키기 위해서 기존에 MPEG-4에 필요한 하드웨어와 소프트웨어 모듈을 AHB 인터페이스 모듈 형태로 연결되었다. 빈번한 메모리 데이터의 이동을 효과적으로 제어하기 위해 ASB 플랫폼에서 사용한 DMA 제어기를 사용하였으며, AHB 인터페이스 모듈을 통해서 연결되었다. DMA의 버스 점유율을 높이기 위해서 프로그램을 실행하는 ARM 프로세서와 데이터 전송을 위한 DMA의 버스를 분리하여 DMA의 버스의 점유율을 향상시켰다. 그러나 ASB 기반 설계에서는 ASB 버스의 프로토콜을 제어하는 ASB Wrapper 모듈을 수정 및 최적화하여 설계함으로 ASB 플랫폼을 AHB 플랫폼으로 변경하는데 어려움이 있다. AHB 기반 플랫폼에서는 가능한 기본 시스템 블록은 그대로 이용하면서 필요한 블록은 인터페

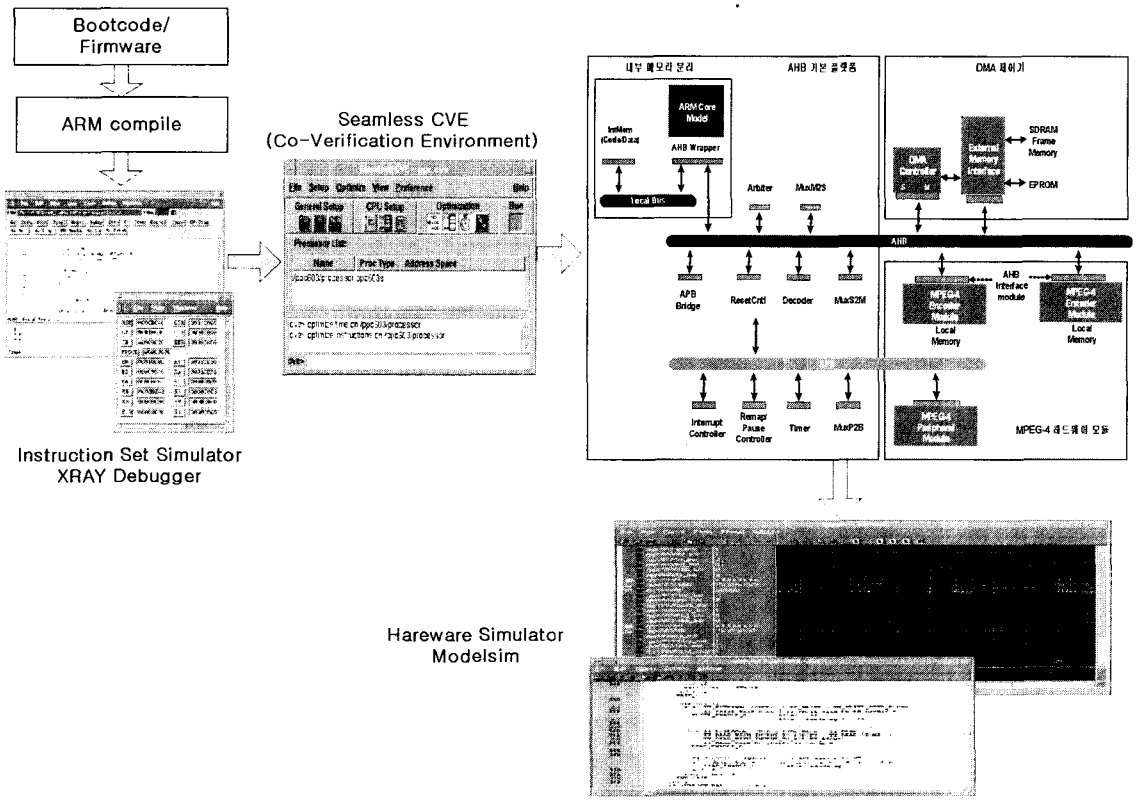


〈그림 9〉 AHB 기반 플랫폼의 구조

이스 회로에서 제어하는 방법을 사용함으로 AHB 플랫폼의 유연성을 확장하였다. AHB 플랫폼에 다양한 종류의 ARM 프로세서를 적용하므로 다양한 형태의 프로세서 기반의 플랫폼의 구성이 가능하다.

3. AHB 기반 플랫폼 검증

AHB 기본 플랫폼에 MPEG-4의 실행 싸이클 분석에 따라서 하드웨어와 펌웨어 모듈로 분할하고 이를 RTL 수준의 AHB 플랫폼에 적용하여 모듈 검증 및 전체 검증을 실행한다. AHB 기본 플랫폼의 하드웨어와 소프트웨어 혼합 검증을 하기 위해서 Mentor Graphic사의 Seamless CVE(Co-Verification Environment)를 사용하였다. ARM7TDMI 프로세서 코어는 Seamless에서 제공하는 ISS(Instruction Set Simulator)를 이용하였고, AHB기반 시스템의 하드웨어 모듈은 AMBA2.0[10]에서 제공하는 RTL 시뮬레이션 모델을 사용하였다. AHB 플랫폼을 Seamless CVE 혼합 설계 환경에 통합하므로 하드웨어와 소프트웨어의 혼합 검증이 가능하게 되었다. AHB 기반 플랫폼의 설계 및 검증 환경은 〈그림 10〉과 같다. 시스템 개발 초기에 하드웨어와 소프트웨어 설계에게 동일한 환경에서 시스템을 개발하고 검증할 수 있는 환경을 제공한다. ASB 기반 플랫폼에서 설계된 MPEG-4 모듈을 AHB 기반 플랫폼에 적용하여 각 모듈을 검증하며, 검증된 모듈을 통합하여 전체 검증을 수행한다. ARM 프로세서에서 실행되는 펌웨어는 AHB 플랫폼을 기동하기 위한 부트 코드와 MPEG-4의 전체 스케줄링을 담당하며 C 코드로 작성되었다. 이는 ARM 컴파일러에 의해 컴파일되어 ARM 프로세서에서 실행되며, XRAY 디버거에 의해서 디버깅할 수 있다. 하드웨어 모



〈그림 10〉 AHB 기반 플랫폼의 검증 환경

듈들은 RTL 수준으로 Modelsim에 의해 검증되며, Seamless CVE에 의해서 하드웨어 및 소프트웨어 혼합 검증을 수행한다.

AHB 플랫폼 검증 환경을 제공하므로 다양한 멀티미디어 응용 분야의 설계를 시스템 개발 초기에 동일한 플랫폼에서 하드웨어와 소프트웨어를 동시에 진행할 수 있고, 시스템 개발 초기에 모듈 검증이 가능하므로 전체 통합시 일어나는 문제를 개발 초기에 점검할 수 있다. AHB 기반 설계 및 검증 환경을 이용하여 하드웨어와 소프트웨어 설계 및 검증을 병행함으로 개발 시간을 단축시킬 수 있으며, SoC 제품의 시장 경쟁력의 요구를 만족시킬 수 있다.

V. 결 론

최근에 새로운 SoC 설계 방법으로 등장한 플랫폼 기반 설계 기법을 적용하여 무선 멀티미디어 응용을 설계하는 방법에 대해서 설명하였다. MPEG-4/H.263 QCIF 30 fps을 처리하기 위한 영상 코덱 모듈을 ASB 플랫폼에 통합, 검증하는 과정을 설명하고, 플랫폼 기반 설계 방법을 적용하여 CIF 25~30 fps를 처리할 수 있는 영상 코덱을 설계하는 방법에 대해서 기술하였다. 성능 분석을 통해서 AHB 플랫폼의 구조를 설계하고, ASB 플랫폼에서 사용된 영상 코덱 모듈을 재사용하였다. 기존의 설계 모듈을 재사용함으로써 설계 시간을 단축할 수 있으며, 영상 코덱 모듈을 AHB 인터페이스를 통해서 AHB 플랫폼에 연결하므로 플랫폼의 유연성을 향상시킬 수 있다. 설계된 AHB 플랫폼은 H.264, 유, 무선 통신 응용 분야인 3G-324M의 기본 플랫폼으로 사용될 수 있다. AHB 플랫폼 확장하여 재구성이 가능한 버스 구조를 기반으로 한 재구성이 가능한 플랫폼(Reconfigurable Platform)으로 연구를 진행해 갈 예정이다[4].

참 고 문 헌

〔1〕 H. Chang, L. Cooke, M. Hunt, G. Martin, A. McNelly and L. Todd, *Surviving the SOC Revolution : A Guide to Platform-Based Design*, Kluwer Academic Publishers : November, 1999.

〔2〕 K. Keutzer, S. Malik, A. R. Newton, J. M. Rabaey, and A. Sangiovanni-Vincentelli, System Level Design : Orthogonalization of Concerns and Platform-Based Design, invited paper, IEEE Transactions on Computer-Aided Design, Vol. 19, No. 12, December 2000.

〔3〕 D. I. August, K. Keutzer, S. Malik and A. R. Newton, A Disciplined Approach to the Development of Platform Architectures. SASIMI, 2001.

〔4〕 M. Baleani, F. Gennari, Y. Jiang, Y. Patel, R. K. Brayton, A. Sangiovanni-Vincentelli, HW/SW Partitioning and Code Generation of Embedded Control Applications on a Reconfigurable Architecture Platform, Proceedings of the 10th International Symposium on Hardware/Software Codesign (CODES), Estes Park, Colorado, USA, May, 2002.

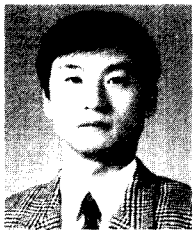
〔5〕 A. Mihal, K. Keutzer, A. Jantsch, H. Tenhunen, *Mapping Concurrent Applications onto Architectural Platforms*, 3, 39-59, Kluwer Academic Publishers, 2003.

〔6〕 *http://www.gigascale.org/pubs/275/design_tech_dsm.pdf* : "The Productivity Gap", The MARCO/DARPA Gigascale Silicon Research Center Collaboration with BWRC.

〔7〕 Jim Tobias (editor), VSI Alliance,

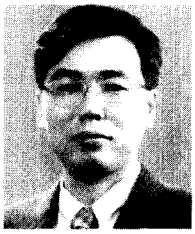
Platform-Based Design DWG, List of Terms and Definitions, Revision 4, 4 March 2002.
〔8〕 Two platform types view for SoC designs : *http://www.eedesign.com/story/OEG20021107S0066*
〔9〕 Juhyun park, Bontae Koo et al, MPEG-4 Video Codec for Mobile Multimedia Applications, IEEE ICCE pp.156-157, Jun. 2001.
〔10〕 ARM Ltd, "AMBA Specification Rev 2.0", Document Number ARM IHI 0011A, 1999.

저 자 소 개



장 준 영

1985년 3월 전남대학교 전산학과 졸업(학사), 1987년 8월 중앙대학교 대학원 전산학과(석사), 1996년 3월 전남대학교 전산학과 졸업(박사), 1996년 3월~1998년 2월 : 대불대학교 전임강사, 1998년 3월~1999년 8월 : 전자통신연구원 회로소자 연구소 POST-DOC, 1999년 9월~현재 : 전자통신연구원 집적회로연구부 선임연구원, 〈주관심 분야 : SoC 설계 및 방법론, HW/SW 통합 설계 및 검증, 내장형 시스템 설계, SoC 플랫폼 설계, 무선 통신 멀티미디어 설계〉

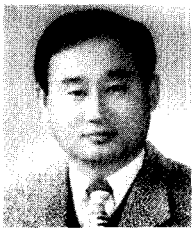


조 한 진

1982년 2월 한양대학교 전자공학과 졸업, 1987년 5월 New Jersey Inst. of Tech 전기공학 석사, 1992년 5월 University of Florida 전기공학 박사, 1992년 11월~1998년 2월 : 한국전자통신연구원 선임연구원, 1998년 3월~현재 : 한국전자통신연구원 책임연구원, 〈주관심 분야 : SOC 설계 방법론, 무선통신, 멀티미디어 설계〉

김 원 종

1989년 2월 전남대학교 공과대학 전자공학과 (공학사), 1992년 2월 한양대학교 대학원 전자공학과 (공학석사), 1999년 2월 한양대학교 대학원 전자공학과 (공학박사), 1999년 2월~2000년 3월 : 한양대학교 공학기술연구소 선임연구원, 2000년 4월~현재 : 한국전자통신연구원 선임연구원, 〈주관심 분야 : SOC 설계 및 설계 방법론, VLSI CAD, 멀티미디어 설계, 하드웨어/소프트웨어 통합 설계, Embedded 소프트웨어 설계, 저전력 설계 방법론 등임〉



김 종 대

1982년 2월 경북대학교 전자공학과 졸업, 1984년 2월 경북대학교 전기공학석사, 1994년 7월 -niversity of New Mexico 전기 및 컴퓨터공학 박사, 1984년 5월~1989년 7월 : 한국전자통신연구원 선임연구원, 1989년 8월~1994년 7월 : University of New Mexico 연구조교, 1994년 11월~현재 : 한국전자통신연구원 연구부장, 책임연구원, 〈주관심 분야 : Power 소자 및 IC 기술, 디스플레이 구동 IC 기술, 나노소자 및 나노 SoC 기술〉