韓國컴퓨터情報學會 論文誌
第16卷 第11號, 2011. 11.

2011-16-11-1-1

가상 플랫폼을 이용한 JPEG 디코더 IP의 구현 및 검증

정 용 범*, 김 용 민**, 황 철 희***, 김 종 면***

Implementation and Verification of JPEG Decoder IP using a Virtual Platform

Yong-Bum Jung*, Yong-Min Kim**, Chul-Hee-Hwang***, Jong-Myon Kim****

요 약

하나의 제품에 다양한 기능들이 복합적으로 통합하는 단일칩시스템 (System-on-a-Chip, SoC)의 설계 요구가 증가하는 반면, 시장이 요구하는 적기 출하 시점은 점점 짧아지고 있다. 따라서 이러한 요구를 만족시키기 위해서 소프트웨어와 하드웨어를 통합하여 검증하는 것이 무엇보다 중요하다. 이러한 하드웨어-소프트웨어 통합 검증을 조기에 수행하는 방법으로 IP(intellectual property) 재사용을 통한 가상 플랫폼 기반 설계 방법이 널리 연구되고 있다. 본 논문에서는 기존 ARM프로세서 기반 S3C2440A 시스템을 가상 플랫폼을 이용하여 재설계하고, JPEG 디코더를 S3C2440A 가상 플랫폼에 구현하여 성능을 평가하였다. 또한, ARM 프로세서 기반 인라인 어셈블리어를 이용하여 JPEG 디코더를 최적화하는 기법을 소개하였고, 이를 가상 플랫폼에 구현하여 성능 향상을 검증하였다. 이러한 가상 플랫폼 기반 설계를 통해 하드웨어 및 소프트웨어의 통합 검증이 가능하고, 시장 적기 출하(Time-to-Market) 요구에 신속히 대처할 수 있다.

▸ Keyword : 가상 플랫폼, 단일칩시스템, JPEG 디코더, 인라인 어셈블리

Abstract

The requirement of a system-on-a-chip (SoC) design is increasing, which combines various and complex functional units on a single device. However, short time to market prohibits to release the device. To satisfy this shorter time-to-market, verification of both hardware and software at the same time is important. A virtual platform-based design method supports faster verification of

• 제1저자 : 정용범 • 교신저자 : 김종면
• 투고일 : 2011. 02. 05, 심사일 : 2011. 03. 03, 게재확정일 : 2011. 04. 19.
* 울산대학교 전기공학부(School of Electrical Engineering, University of Ulsan) 석사과정
** 울산대학교 전기공학부(School of Electrical Engineering, University of Ulsan) 석사과정
*** 울산대학교 전기공학부(School of Electrical Engineering, University of Ulsan) 석사과정
**** 울산대학교 전기공학부(School of Electrical Engineering, University of Ulsan) 교수
※ 이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2011-0017941).
툴은 반도체설계교육센터(IDEC)의 지원을 받아 수행되었음

these combined software and hardware by reusing pre-defined intellectual properties (IP). In this paper, we introduce the virtual platform-based design and redesign the existing ARM processor based S3C2440A system using the virtual platform-based method. In addtion, we implement and evaluate the performance of a JPEG decoder on the S3C2440A virtual platform. Furthermore, we introduce an optimized technique of the JPEG decoder using the ARM based inline assembly language, and then verify the performance improvement on the virtual platform. Such virtual platform-based design allows to verify both software and hardware at the same time and can meet the requirement of the shorter time-to-market.

▸ Keyword : Virtual platform, system on a chip, JPEG decoder, inline assembly

I. 서 론

최근 기술이 발전함에 따라 소형화 및 다양한 기능이 복합적으로 통합되는 전자 제품의 수요가 급증하고 있다. 소형화를 통한 휴대용 모바일 제품의 출시가 가능하고, 복합화를 통하여 하나의 제품이 다양한 기능을 제공하게 되면서 다수 제품을 휴대해야 하는 번거로움이 줄어들기 때문이다. 이러한 전자 제품의 출현은 하나의 시스템을 하나의 칩 상에 구현되는 SoC (System-on-a-Chip, SoC) 설계의 보편화 덕분에 가능하다[1]. 하지만 SoC 설계에서 칩의 복잡도와 SoC 제품의 생산성 차이가 계속해서 증가함에 따라 현재의 SoC 설계 방법으로는 SoC 제품의 성능과 요구의 변화를 만족시킬 수 없다. 또한, 프로토타입 시스템을 하드웨어로 구현하고 이를 이용하여 소프트웨어를 개발 및 검증하고 다시 하드웨어를 수정하는 기존의 방법은 제품의 개발 주기가 길어지기 때문에 날이 갈수록 짧아지는 시장 적기 출하 (Time-to-Market)의 요구를 해결할 수 없다. 이러한 문제를 해결하기 위해서 하드웨어-소프트웨어를 통합하여 검증할 수 있는 방법으로 IP (intellectual property)재사용을 기반으로 한 플랫폼 기반 설계가 대안으로 부각하고 있다[2][3].

플랫폼 기반 설계는 재사용이 가능한 IP 또는 VC (virtual component)를 이용한 플랫폼 기반 설계(platform-based design) 방법으로 SoC 제품을 빠르게 개발하기 위한 응용 기반 통합 플랫폼이다. 이 새로운 설계 방법의 가장 큰 장점이 되는 세 가지 요소는 재사용 (reuse), 유연성 (flexibility), 및 효율성 (efficiency)이다. 재사용이란 주어진 응용 분야에 공통으로 적용될 수 있는 아키텍처를 정의하여 같은 아키텍처를 다시 사용함을 의미한다. 이러한 아키텍처의 재사용을 광범위한 영역에서 적용하기 위해서는 범용성 혹은 유연성 (flexibility)이 필수이며, 이를 위해서는 프로그래머빌러티 (programmability)가 높은 아키텍처 구조가 적합하다. 결국 프로그래머빌러티가 높은 아키텍처의 재사용을 통하여 효율적인 생산이 가능하다. 따라서 플랫폼 기반 설계 방법은 IP 재사용을 통해서 설계 시간을 단축하며, 시스템 수준에서의 최적화를 통해서 제품의 시장 경쟁력을 높이는 수단이 된다[4].

본 논문에서는 SoC 디자인 설계 및 검증을 위한 하드웨어-소프트웨어 통합 검증 방법 중 Carbon사의 SoC Designer 툴[5]을 이용하여 가상 플랫폼 (virtual platform) 환경에서 시스템을 설계 및 검증하는 방법을 소개한다. SoC Designer 툴은 기존의 하드웨어 IP를 재사용함으로써 설계 비용 및 디자인 수정시간을 단축할 수 있다. 또한, SoC Designer 툴을 이용하여 기존 ARM프로세서 기반 S3C2440A 시스템[6]을 가상 플랫폼으로 재설계하고, JPEG 디코더를 이 가상 플랫폼에 구현하고 성능을 평가하였다. 또한, 본 논문에서는 ARM프로세서 기반 인라인 어셈블리어를 이용하여 JPEG 디코더의 성능을 향상시키는 방법을 소개하였고, 이를 가상 플랫폼에 구현하여 성능 향상을 검증하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구들을 소개하고, 3장에서는 제안하는 가상플랫폼 설계 기법 및 인라인 어셈블리어를 이용한 JPEG 디코더 구현 방법을 소개한다. 4장에서는 SoC Designer 툴을 이용하여 ARM기반 S3C2440A 시스템을 재설계하고, 인라인 어셈블리 기반 JPEG 디코더를 S3C2440A 가상 플랫폼에 구현하고 성능을 평가한다. 5장에서는 본 논문의 결론을 맺는다.

II. 관련 연구

1. 일반적인 하드웨어 설계 방법

그림 1은 일반적인 하드웨어 설계 방법을 보여준다. 알고리즘 수준의 하드웨어 사양은 부동소수점 모델과 고정소수점 모델을 이용하여 검증이 이루어지며, 이로부터 구조 설계와 RTL (register transfer level) 설계가 이루어진다. RTL

설계는 다양한 설계 검증 툴을 이용하여 충분한 검증을 수행하고 합성과정을 거쳐 게이트 수준의 설계로 변환된다. 또한, 게이트 수준에서도 다양한 검증과 분석이 수행되고, 추후 배치와 배선 (place and routing)을 통하여 FPGA (field programmable gate array)나 ASIC (application-specific integrated circuit)으로 구현된다[7].

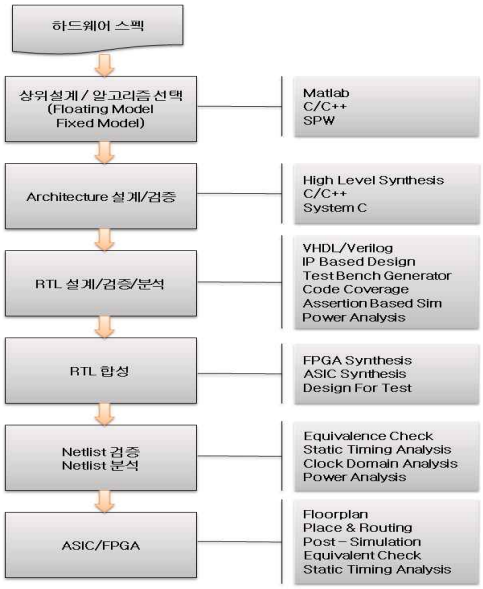

그림 1. 일반적인 하드웨어 설계 방법
Fig. 1. General hardware design flow

일반적인 하드웨어 설계 방법은 FPGA 또는 ASIC을 소프트웨어와 독립적으로 검증함으로써 통합 구현 시 문제가 발생할 확률이 높고, 시스템의 동작에 대한 분석을 정확하게 할 수 없는 단점이 있다. 따라서 일반적인 하드웨어 설계 방법을 이용하여 다양하고 복잡한 기능을 수행하는 SoC를 설계할 때 시스템 검증에서 요구되는 시간적 비용과 재수정에 필요한 비용은 상당하다. 이러한 문제를 해결하기 위해 IP 재사용을 기반으로 한 플랫폼 기반 설계 방법론이 대두되고 있다[8].

2. 플랫폼 기반 설계

플랫폼이란 SoC 설계에서 어떤 작업 또는 기술 구현이 이루어질 수 있는 공통·공용의 표준화된 하드웨어 및 소프트웨어 환경 및 설계 도구를 말한다. 플랫폼의 라이브러리와 설계 방법에 따라서 구조적 탐색을 통해 플랫폼의 구조를 결정하고 기존의 정의된 IP 블록이나 모듈을 적용 또는 수정 보완하여 시스템을 검증한다. 플랫폼 기반 설계는 복잡한 SoC 제품을 개발하기 위해 IP 재사용을 강조한 플랫폼 기반 통합 설계 접근 방법이다.

그림 2는 플랫폼 기반 설계 방식의 흐름도를 보여준다. 크게는 응용 분야의 요구 사항에 따라 플랫폼을 구성하여 설계하는 과정과 검증 및 구현하는 과정으로 나누어진다. 먼저 플랫폼을 구성하는 측면에서는 응용 분야와 제품에 대한 방향을 정의하여 설계한다. 초기 플랫폼의 설계 시 응용 분야를 가속시키는 기능적 IP들을 명시하고 구조적인 측면에서는 시스템에서 요구되는 코어와 데이터 전송을 위한 온칩 (on-chip) 버스 형태 그리고 메모리 등을 고려하여 설계한다. 응용 알고리즘을 플랫폼 구조에 대응되도록 하기 위해서 초기 설계는 결정된 IP와 외부 인터페이스를 고려하여 필요한 래퍼 (wrapper)를 구성하여 플랫폼 통합과정을 수행한다. 플랫폼을 구성하고 설계하는 최종 단계로써 메모리 전송 폭과 플랫폼에서 소비하는 전력 및 성능을 고려하여 요구되는 통합 플랫폼 시스템을 구성한다.

플랫폼 구현 및 검증 단계에서는 구성된 플랫폼 통합시스템을 응용 부분의 개발을 고려하여 재조정 할 수 있다. 또한, 시스템의 파라미터인 클록 속도, 버스 프로토콜 및 크기를 고려하여 각 모듈을 최적화한 후, 소프트웨어 수행부분과 하드웨어 수행부분으로 나누어 구현한다. 최종 단계에서는 플랫폼을 기반으로 하여 하드웨어-소프트웨어 통합 검증을 수행한다[9].

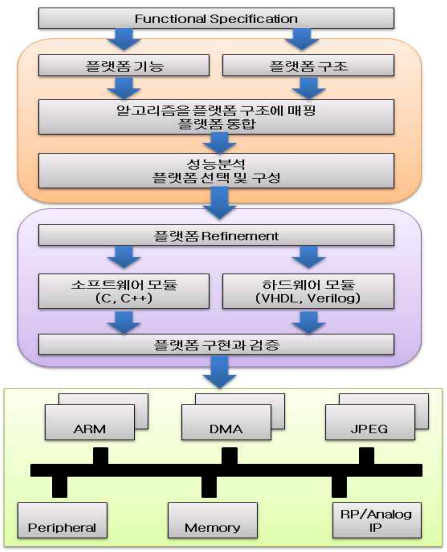

그림 2. 플랫폼 기반 설계 방법
Fig. 2. Platform-based design methodology

플랫폼 기반 설계는 미리 검증된 모듈을 조립하여 설계하는 것이기 때문에 설계, 제조, 테스트에 필요한 시간과 비용

을 대폭 절감하고 시장적시성(Time-To-Market) 단축에 유리하며 하드웨어 및 소프트웨어 재활용도 가능하여 확장성에 따른 시장이나 응용분야에 신속히 대처할 수 있다.

3. 가상 플랫폼 기반 설계

기존 하드웨어나 SoC 설계에서 일반적으로 많이 사용되는 RTL 추상화 수준은 설계적인 측면에서 자동화된 합성 툴을 직접 이용할 수 있다는 장점이 있다. 하지만, 규모가 큰 설계에서는 성능 검증 및 분석에 소요되는 시간이 상당히 커서 Time-to-Market의 요구를 만족시키지 못하는 단점이 있다.

RTL은 주로 Verilog와 같은 HDL (hardware description language)로 하드웨어를 표현하고 FPGA 에뮬레이션 (emulation) 혹은 타겟 IP의 실리콘 시제품을 장착한 평가 보드 (evaluation board)를 통해 소프트웨어를 포함한 전체 SoC의 시뮬레이션과 에뮬레이션을 수행한다. 이와 달리 가상 플랫폼 은 기존의 HDL로 표현된 SoC의 추상화 레벨 (abstraction level)을 높여서 주로 C, C++, 혹은 SystemC로 표현된다. 가상 플랫폼 (virtual platform)에서는 시뮬레이션 속도 향상이 주된 목적이므로 기존 RTL에서 사용되는 HDL과 같은 비트 단위의 시그널을 사용하는 이벤트 기반 표현 대신, 바이트 혹은 워드 단위의 데이터 타입을 사용하고 사이클 기반의 표현을 사용한다. 이러한 표현을 기존 RTL과 대비되는 상위 개념의 추상화 레벨로써 트랜잭션 레벨 (transaction level)이라고 명명한다. 이러한 트랜잭션 레벨 모델 (TLM)의 사용을 통해 RTL 대비 정확도는 희생하지만 시뮬레이션 시간은 현격히 향상시킬 수 있다[10].

III. 가상 플랫폼을 이용한 JPEG 디코더 구현 및 검증

1. SoC Designer 툴을 이용한 시스템 설계

가상 플랫폼 기반 설계를 위한 툴로서 Carbon사의 SoC Designer는 아키텍처를 명세한 단계와 명세된 아키텍처의 프로세서 위에 컴파일된 이미지를 올리고 시뮬레이션 하는 단계로 나뉘고, TLM 레벨에서 하드웨어와 소프트웨어의 통합 시뮬레이션을 수행한다. 이러한 시스템 시뮬레이터는 충분한 프로세서 컴포넌트 라이브러리가 제공되면 쉽게 활용될 수 있지만, 라이브러리로 제공되지 않는 프로세서 시뮬레이터를 붙여서 동작하는 것은 불가능하거나 큰 노력이 드는 것이 보통이다.

SoC Designer 툴은 다양한 형태의 ARM프로세서, 메모리, AMBA (Advanced Microcontroller Bus Architecture) 버스 및 I/O 페리퍼럴 (peripheral) IP들을 제공하고 있어 기존의 시스템을 재설계하는데 용이할 뿐만 아니라, 새로이 요구되는 컴포넌트를 쉽게 제작하고 추가하여 검증할 수 있는 환경을 제공한다. 그림 3은 SoC Designer 툴을 이용하여 가상 플랫폼 환경에서 하드웨어와 소프트웨어의 동시 검증을 위한 설계 방법론을 보여준다. 가상 플랫폼 기반의 C를 모델링을 하여 프로파일링을 진행함으로써 소프트웨어와 하드웨어의 설계가 이루어진 상태에서 통합하여 동작이 정확히 이루어지는지를 판단할 수 있다.

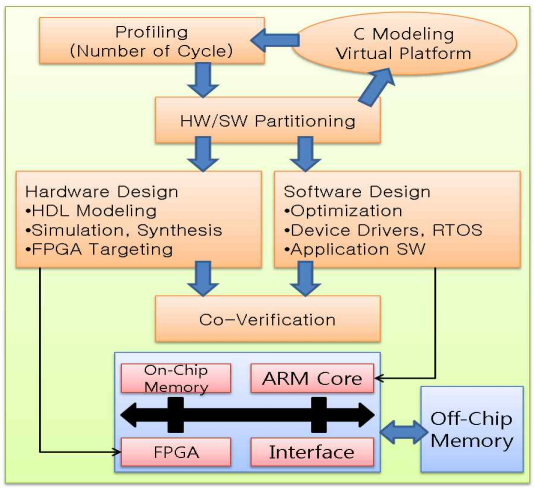

그림 3. SoC Designer 툴을 이용한 설계 방법
Fig. 3. Design methodology using a SoC designer tool

2. S3C2440A 플랫폼 설계

그림 4는 SoC Designer 툴을 이용하여 재설계한 ARM 프로세서 기반 S3C2440A 가상 플랫폼을 보여준다. S3C2440A 가상 플랫폼은 ARM920T 프로세서, AMBA 버스, 메모리 및 각종 I/O 등으로 구성되며 기존에 정의되어 있는 IP를 재사용하여 설계하였다.

각 컴포넌트를 적절하게 배치하여 마스터/슬레이브 (Master/Slave) 포트를 연결하고 시스템 파라미터를 설정하였다. 시스템 파라미터는 스택 (stack)과 힙 (heap)을 할당하는 시작 주소와 크기에 따라 결정되며 이를 바탕으로 메모리 맵 (memory map)을 설정할 수 있다. 표 1은 S3C2440A 가상 플랫폼에서 설정된 메모리 맵을 보여준다.

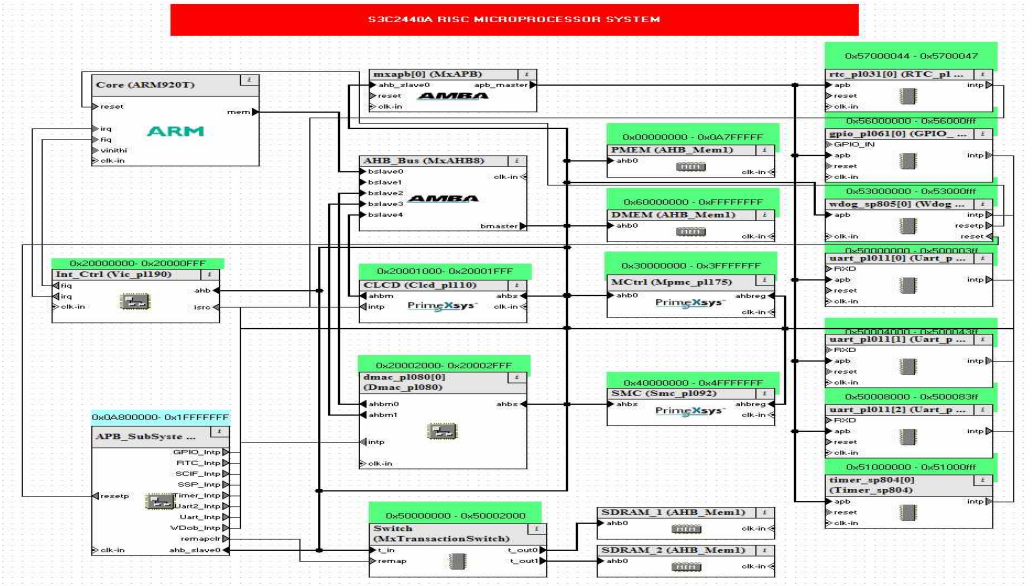

그림 4. S3C2440A 가상 플랫폼
Fig. 4. S3C2440A virtual platform

표 1. S3C2440A 가상 플랫폼의 메모리 맵
Table 1. Memory map of S3C2440A virtual platform

Component	Port	Start Address	End Address	Size
PMEM	ahb	0x0	0xa7fffff	0xa800000
APB_SubSystem	ahb_slave	0xa800000	0xa808fff	0x9000
mxapb	ahb_slave	0xa900000	0xa908fff	0x9000
Int_Ctrl	Ahb	0x20000000	0x20000fff	0x1000
CLCD	ahbs	0x20001000	0x20001fff	0x1000
dmac_pl080	ahbs	0x20002000	0x20002fff	0x1000
MCtrl	ahb	0x30000000	0x30007fff	0x8000
MCtrl	Ahbreg	0x3ffff000	0x3fffffff	0x1000
SMC	ahbs	0x40000000	0x4fffefff	0xffff000
SMC	ahbreg	0x4ffff000	0x4fffffff	0x1000
Switch	t_in	0x50009000	0x5000afff	0x2000
wdog_sp805	apb	0x53000000	0x53000fff	0x1000
DMEM	ahb	0x60000000	0xffffffff	0xa0000000

3. 인라인 어셈블리를 이용한 JPEG 디코더설계

JPEG 디코더는 그림 5와 같이 크게 3개의 단계를 거쳐 압축된 JPEG 데이터를 복원한다.

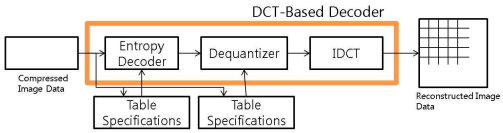

그림 5. JPEG 디코더 블록 다이어그램
Fig. 5. A block diagram of the JPEG decoder

본 논문에서는 JPEG 디코더의 IDCT (inverse DCT), dequantization (역 양자화), huffman decoder를 ARM 기반 어셈블리어 (assembly language)를 사용하여 성능 향상을 꾀하였다.

먼저 어셈블리어로 변환하기 전에 몇 가지 규칙을 준수하였다. C언어로 기술된 JPEG 디코더 알고리즘 자체를 최적화하지 않았으며, 또한 이미지는 단색보다는 여러 가지 색이 들어있는 사진으로 선정하여 사용하였다.

본 논문에서는 어셈블리어 변환을 다음 두 단계로 나누었다. 첫 번째는 C언어에 기반을 둔 인라인 어셈블리어 변환이고, 두 번째는 레지스터를 이용한 인라인 어셈블리어 변환이다.

3.1 C언어 기반 인라인 어셈블리

그림 6은 C언어에 기반을 둔 인라인 어셈블리어 사용의 예이다. C언어에서 사용하는 변수명을 그대로 사용하는 방법이 특징이다. 단순하게 C언어 코드를 가지고 어셈블리로 번역하는 과정이며 성능보다는 어셈블리어로 제어를 하기 위해서 사용하는 방법 중 하나이다. C언어 코드에서 필요한 변수명을 가져와서 ARM 어셈블리어 규칙에 맞게 삽입하면 된다.

```
void my_strcpy(const char *src, char *dst)
{
    int ch;
    __asm
    {
        loop:
        // ARM version
        LDRB ch, [src], #1
        STRB ch, [dst], #1
    }
```

그림 6. C프로그램 기반 인라인 어셈블리 예
Fig. 6. An example of inline assembly based on a C program

3.2 어셈블리어 기반 인라인 어셈블리

3.1절의 C언어 코드에 기반을 둔 인라인 어셈블리 방법과는 다르게 어셈블리어에 기반을 둔 인라인 어셈블리는 C언어의 변수명을 사용하지 않고, ARM 내부의 레지스터를 주로 사용한다. 그림 7은 어셈블리어 기반 인라인 어셈블리 사용의 예를 보여준다.

```
        AREA globals,CODE,READONLY
        EXPORT asmsubroutine
        IMPORT globvar
asmsubroutine
        LDR r1, =globvar ; read address of globvar into
        ; r1 from literal pool
        LDR r0, [r1]
        ADD r0, r0, #2
        STR r0, [r1]
        MOV pc, lr
        END
```

그림 7. 어셈블리 프로그램 기반 인라인 어셈블리어 예
Fig. 7. An example of inline assembly based on an assembly program

C언어 코드를 분석해서 최대한 많은 레지스터를 사용하였다. C언어의 변수를 사용해야 될 경우에는 초기에 mov 명령어를 사용해서 그 변수명에 맞는 레지스터를 설정하였다. 이를 이용해서 모든 C코드의 변수명을 레지스터로 지정해주면 그 만큼 성능이 향상된다. 본 논문에서는 위의 두 가지 방법 중에서 두 번째 방법을 사용하여 성능 향상을 꾀하였다.

또한, S3C2440A 가상 플랫폼에서 인라인 어셈블리어 기반 JPEG 디코더를 구현하고 성능을 평가하기 위해 ADS (ARM Developer Suite)툴[11]을 이용하여 디코더 소스 코드 내부에 그림 8과 같이 메모리 맵을 선언하였다. 소프트웨어 구현에서 정의된 메모리 맵은 함수 호출에 따라 하드웨어 컨트롤 시그널을 보내며 인터럽트 (interrupt)를 수행한다.

```
#define CTRL                ((volatile unsigned*)0x70000000)
#define RESULT              ((volatile unsigned*)0x30001000)
#define OP                  ((volatile unsigned*)0x30000000)
…
#define block_CTRL          ((volatile unsigned*)0x50000000)
#define block_addr          ((volatile unsigned*)0x50000010)

#define Quanti_CTRL         ((volatile unsigned*)0x60000000)
#define Quanti_addr         ((volatile unsigned*)0x60000010)
```

그림 8. 소프트웨어 명세 - 메모리 맵
Fig. 8. Software specification - Memory map

IV. 실험 및 결과

본 장에서는 SoC Designer 툴을 이용하여 ARM 프로세서 기반 S3C2440A 가상 플랫폼을 설계하고, 인라인 어셈블리어 기반 JPEG 디코더를 구현하여 성능을 평가하였다. 또한, ARM 프로세서를 이용한 JPEG 디코더의 성능과 S3C2440A 가상 플랫폼을 이용한 JPEG 디코더의 성능을 비교 분석하였다.

1. 성능 평가

1.1 ARM 프로세서의 인라인 어셈블리어 기반 JPEG 디코더 성능

그림 9는 ADS 툴을 이용하여 JPEG 디코더에서 복잡도가 가장 높은 IDCT (inverse DCT), dequantization (역양자화), huffman decoder를 인라인 어셈블리어를 사용하여 구현한 경우와 그렇지 않은 경우의 프로파일링 분석 결과를 보여 준다. 인라인 어셈블리어를 이용함으로써 IDCT에서 35.8%, 역 양자화에서 71.1%, huffman decoder에서 8.6%의 수행시간이 감소되었다. 각 커널 부분에서 사용되는 메모리 접근 패턴을 단일 접근 패턴에서 다중 접근 패턴으로 수행한 결과, 연산 요구량이 줄어들어 소프트웨어적인 최적화가 이루어졌음을 알 수 있다.

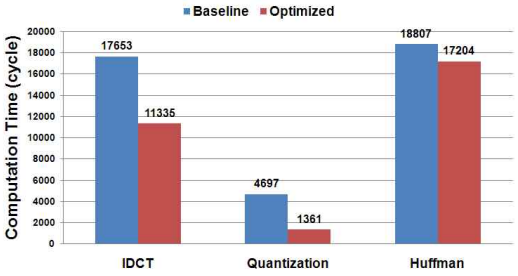

그림 9. 인라인 어셈블리어를 이용한 JPEG 커널별 성능
Fig. 9. Performance of JPEG kernels using inline assembly

1.2 ARM 프로세서와 S3C2440A 플랫폼을 사용한 JPEG 디코더의 성능 비교

표 2는 ARM 프로세서와 S3C2440A 가상 플랫폼을 이용하여 baseline JPEG 디코더 및 인라인 어셈블리어를 이용한 optimized JPEG 디코더를 수행한 결과에 대한 성능(cycle count) 비교를 보여준다. ARM 프로세서를 이용한 결과는 단순히 ARM 프로세서 코어에서 JPEG 디코더를 수행한 성능 결과이며, S3C2440A 가상 플랫폼을 이용한 결과는 ARM 프로세서, AMBA 버스, 메모리 및 입출력장치 등 통합된 시스템에서 JPEG 디코더를 수행한 결과이다. 따라서 S3C2440A 가상 플랫폼을 이용하여 JPEG 디코더를 구현한 경우는 ARM 프로세서, AMBA버스, 메모리 및 입출력 장치 사이에서 발생하는 이벤트 처리 때문에 더 많은 클럭 사이클이 요구됨을 알 수 있다.

표 2. ARM 프로세서와 S3C2440A 가상 플랫폼을 이용한 JPEG 디코더 성능 비교
Table 2. Performance comparison of JPEG decoder using ARM processor only and S3C2440A virtual platform

	Baseline JPEG	Optimized JPEG
ARM 프로세서	62,876,892	51,226,699
S3C2440A 가상 플랫폼	164,777,112	134,886,972

그림 10은 S3C2440A 가상 플랫폼에서 IDCT 커널 수행 시 발생하는 버스 내부 신호의 사용 빈도를 보여주며, 원으로 표시된 부분이 버스를 통해 메모리를 접근할 때 발생하는 버스 내부의 충돌을 나타낸다. 초기 데이터 입력 부분에서 발생하는 버스 충돌보다 IDCT 커널에서 발생하는 충돌이 더 많음을 알 수 있다. 이는 Chen방식의 IDCT 커널이 과거 데이터의 연산 결과 값을 사용하는 동안, 코어는 연산 결과들이 완료되기 전에 읽기 요청을 수행함으로써 많은 데이터 충돌이 발생하기 때문이다.

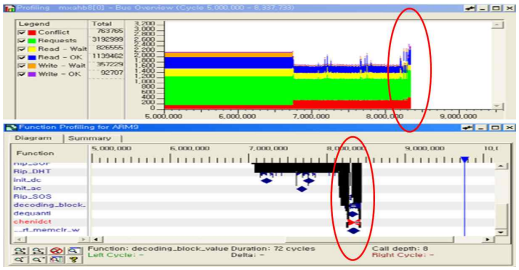

그림 10. IDCT 수행시 발생하는 버스 내부 신호
Fig. 10. Internal bus signals during IDCT

그림 11은 SoC Designer 툴을 이용하여 S3C2440A 가상 플랫폼 구현하고 이를 이용하여 JPEG 디코더를 수행한 결과 이미지를 보여준다.

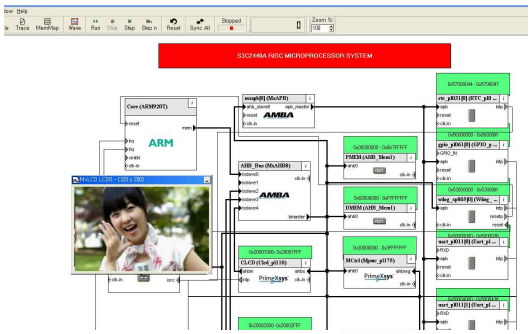

그림 11. 가상 플랫폼을 이용한 JPEG 디코더 결과 화면
Fig. 11. A screen shot of JPEG decoder using the virtual platform

V. 결 론

시스템이 지능화되고 복잡해짐에 따라 SoC 설계 과정 중 검증에 필요한 시간과 비용이 증가하는 반면, 시장적시성(Time-To-Market)에 대한 요구는 오히려 단축되고 있다. 따라서 기존의 하드웨어와 소프트웨어를 분리해서 설계하고 검증하는 방식으로는 이러한 시장 적기 출하 요구를 만족시키지 못한다. 이러한 문제를 해결하기 위한 방법으로 본 논문에서는 미리 검증된 모듈 (혹은 IP)을 이용하여 설계 및 검증 시간을 단축시킬 수 있는 플랫폼 기반 설계 방법론을 소개하였다. 또한, 기존의 ARM 프로세서 기반 S3C2440A 시스템을 가상 플랫폼 환경에서 재설계하였고, JPEG 디코더를 S3C2440A 가상 플랫폼에 구현하고 성능을 검증하였다. 더불어, 인라인 어셈블리어를 이용하여 JPEG 디코더의 성능을

약 19% 향상시켰으며, 이를 S3C2440A 가상 플랫폼에 구현하여 검증한 결과, 인라인 어셈블리어용 JPEG 디코더가 baseline JPEG 디코더보다 약 18%의 성능이 향상되었다. 이러한 플랫폼 기반 설계 방식을 이용함으로써 하드웨어와 소프트웨어의 통합 검증이 가능하며, 이를 통해 시장이 요구하는 적기 출하 시점을 만족시킬 수 있을 것으로 기대한다.

참고문헌

[1] J. Um, S. Hong, Y. Kim, E. Chung, K. Choi, J. Kong, and S. Eo, "ViP: a practical approach for HW/SW co-design", Journal of Semiconductor Technology and Science, vol. 5, no. 2, pp. 89-101, June 2005.

[2] A. Sangiovanni-Vincentelli and G. Martin, "Platform-based design and software design methodology for embedded systems", IEEE Design & Test of computers, vol. 18, no. 6, pp. 23-33, Nov. 2001.

[3] P. Magarshack, "Improving SoC design quality through a reproducible design flow", IEEE Design & Test of computers, vol. 19, no. 1, pp. 76-83, Jan. 2001.

[4] H. Kim and B. Moon, "A research improving IP reusability and minimizing latency in NoC architecture," The Proceedings of Institute of Electronics Engineers of Korea Fall Conference, vol. 30, no. 1, pp. 699-700, 2007.

[5] Carbon's SoC Designer Tool: http://carbondesignsystems.com/SocDesignerPlus.aspx

[6] S3C2440A User's Manual: www.rockbox.org/twiki/pub/Main/DataSheets/um_s3c2440a_rev10.pdf

[7] O. Adeluyi and J. Lee, "CHARMS: A mapping heuristic to explore an optimal partitioning in HW/SW co-design," The Journal of Korea Society of computer and information, vol. 15, no. 9, pp. 1-8, 2010.

[8] A. Sangiovanni-Vincentelli, "Quo vadis, SLD? Reasoning about the trends and challenges of system level design," Proceedings of the IEEE, vol. 95, no. 3, pp. 467-506, March 2007.

[9] K. Keutzer, S. Malik, A. R. Newton, J. M. Rabaey, and A. Sangiovanni Vincentelli, "System level design : orthogonalization of concerns and platform-based design," IEEE Transactions on Computer Aided Design, vol. 19, no. 12, pp. 1523-1543, Dec. 2000.

[10] I. Lee, H. Kim, P. Yang, S. Yoo, E. Chung, K. Choi, J. Kong, and S. Eo, "PowerViP: SoC power estimation framework at transaction level", Proc. Asia South Pacific Design Automation Conference, 8 pages, Jan. 2006.

[11] ARM Development Tools: http://www.arm.com/products/tools/index.php

저 자 소 개

정 용 범

2009 : 울산대학교 정보통신공학사.

2009 : 울산대학교 컴퓨터정보통신공학부 석사과정 입학.

관심분야 : 임베디드 SoC, 컴퓨터 구조, 병렬처리

Email: smartnow@nate.com

김 용 민

2009 : 울산대학교 컴퓨터공학사.

2009 : 울산대학교 컴퓨터정보통신공학부 석사과정 입학.

관심분야 : 임베디드시스템, 컴퓨터구조, 병렬처리

Email: jafstar@nate.com

황 철 희

2008: 울산대학교 컴퓨터공학사.

2008: 울산대학교 컴퓨터정보통신공학부 석사과정 입학.

관심분야: 임베디드 SoC, 컴퓨터 구조, 고장예측, 병렬처리

Email: irnus@hanmail.net

김 종 면

1995 : 명지대학교 전기공학사

2000 : University of Florida ECE 석사

2005 : Georgia Institute of Technology ECE 박사

2005~2007 : 삼성종합기술원 전문연구원

2007~현재 : 울산대학교 컴퓨터정보통신공학부 교수

관심분야 : 프로세서 설계, 임베디드 SoC, 컴퓨터구조, 병렬처리

Email: jongmyon.kim@gmail.com