

QplusME/Esto를 이용한 H.264 디코더의 함수별 전력 측정

채송아^o, 김두현, 임채덕*, 우덕균*, 정창희*
건국대학교 인터넷미디어공학부, *한국전자통신연구원 임베디드S/W연구단
{ sachae^o, doohyun }@konkuk.ac.kr, {cdlim, dkwu, chung}@etri.re.kr

Experimental Power Consumption Measurement of H.264 Decoder Functions using Qplus/Esto

Song-Ah Chae^o, Doo-Hyun Kim, Chae-Duk lim*, Duk-Kyun Woo*, Chang-hee Jung*
School of Internet and Multimedia Eng., Konkuk Univ.,

*ETRI(Electronics and Telecommunications Research Institute) Embedded S/W Research Division

요약

본 논문에서는 임베디드 리눅스와 그에 따른 통합 개발 환경인 QplusME/Esto에서 제공하는 전력소모량 분석도구를 이용하여 H.264 디코더의 함수별 전력 소모량을 측정하여 공개함으로써 향후 연구에 기반 정보로 사용하도록 한다. 또한 이를 이용하여 MacroBlock 디코딩시, 전력을 줄일 수 있는 다양한 방법 중 하나의 예를 제시하고 그 실험 결과를 제시한다.

1. 서론

최근 휴대폰, PDA, PMP 등의 다양한 휴대용 단말의 사용이 보편화되고 발전함에 따라 이에 따른 임베디드 시스템은 기존의 시스템보다 고성능화되고 응용이 복잡화되며 사용자와의 상호작용이 가능하도록 요구되고 있다. 또한 휴대기기의 특성상 배터리 기반의 임베디드 시스템에 대한 저전력 요구가 증대되고 있다. 기존의 전력 관리 기법은 주로 하드웨어적으로 구현되었으나 최근 휴대기기를 통하여 동영상, 게임등의 다양한 멀티미디어 서비스를 제공함에 따라 소프트웨어 차원의 저전력 관리 요구가 커지고 있다.

이에 본 논문에서는 한국전자통신연구원(ETRI)에서 개발한 임베디드 리눅스와 그에 따른 통합 개발 환경인 QplusME/Esto에서 제공하는 전력소모량 분석도구를 이용하여 H.264 디코더의 함수별 전력 소모량을 측정하여 공개함으로써 향후 연구에 기반 정보로 사용하도록 한다. 또한 이를 이용하여 소프트웨어의 전력을 줄일 수 있는 다양한 방법 중 하나의 예를 제시한다.

2. QplusME/Esto[1]

Qplus는 한국전자통신연구원(ETRI)에서 개발한 임베디드 리눅스기반의 운영체제로 기존의 리눅스 버전에서 임베디드 시스템에 사용하기에 부족한 실시간 스케줄링, 저전력 관리, 멀티미디어 처리 기능, 관련 디바이스 드라이버 등을 보완하여 확장 개발한 임베디드 시스템용 리눅스 운영체제이다. QplusME는 스마트폰, 텔레텍스 단말 등을 지원하고자 하는 Qplus의 Mobile Edition으로 다른 임베디드 리눅스에 비해 사용자 및 애플리케이션 환경에 최적화된 향상된 전력관리를 제공하여 모바일 단말의 에너지 소모를 최소화하고 데이터 입출력에서의 실시간 성능을 증대시킨 멀티미디어 파일 시스템을 제공한다.

Esto(Embedded Systems TOol)는 Qplus 플랫폼에서

수행되는 펌웨어, 디바이스 드라이버, 응용 등의 임베디드 소프트웨어 개발을 보다 쉽고 빠르게 개발할 수 있는 통합 개발 환경(IDE, Integrated Development Environment)이다. 본 논문에서 사용하는 Esto는 Eclipse기반의 Esto로 사용자에게 하나의 단일화된 통합 환경을 제공한다.

2.1 전력 측정 장치

Esto에서 제공하는 전력 분석 도구는 타겟 시스템과 멀티미터, 멀티미터와 호스트 시스템, 타겟 시스템 및 호스트 시스템 간의 인터페이스로 구성된다. 전력 측정을 위한 환경은 아래의 그림 1과 같다.

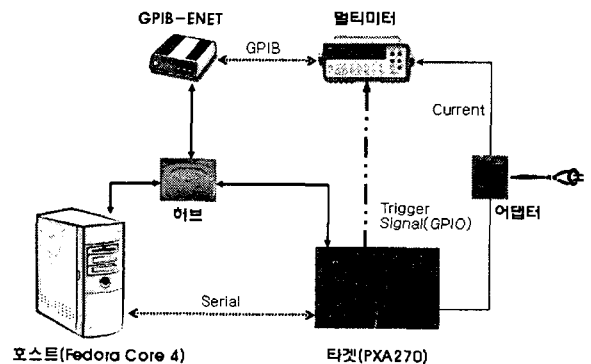


그림 1 전력 측정 구조도

타겟 시스템은 Hybus사의 X-HYPER270 보드를 사용하고 호스트 시스템에는 Fedora Core 4를 설치하였다. 전력 측정 장치는 기존의 호스트와 타겟 보드와 함께 전력 측정을 할 수 있는 멀티미터(Agilent사의 Digital Multimeter 34401A)와 멀티미터가 인터넷과 연결될 수 있도록 하는 GPIB-ENET(National Instrument사의

NI-488.2), 호스트-타겟-멀티미터가 네트워크로 연결되도록 하는 허브로 구성되어 있다. 타겟 시스템과 멀티미터, 멀티미터와 호스트 시스템, 타겟 시스템 및 호스트 시스템 간의 관계는 그림 2와 같고 이들 간의 세부 흐름은 아래와 같이 요약된다.

Esto에서 분석하여 출력하는 측정치는 전력 소모량뿐만 아니라 Execution Time, I-cache 미스율(Miss Rate), D-cache 미스율 등이 있다. 이 중 I-cache 미스율과 D-cache 미스율은 전력 소모량과 아울러 소프트웨어 최적화를 위해 중요한 단서를 제공한다.

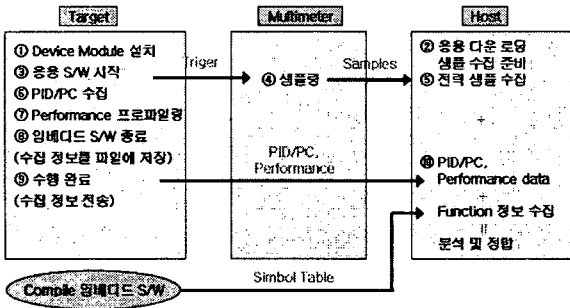


그림 2 전력 분석 도구의 기능 흐름도

- ① 전력 분석을 위한 디바이스 모듈을 설치한다.
- ② 응용 프로그램을 타겟 시스템에 다운로드하고 호스트 시스템은 멀티미터로부터 샘플링 준비를 한다.
- ③ 타겟 시스템은 타겟 임베디드 S/W를 수행하면서 일정한 주기의 타이머 인터럽트를 발생시키고 인터럽트 처리 루틴에서 GPIO를 통해 멀티미터에 트리거 입력을 준다.(현재 10ms로 설정)
- ④ 트리거 입력을 받은 멀티미터는 현재 전류를 샘플링하고 호스트 시스템에 전류 샘플을 송신한다.
- ⑤ 호스트는 멀티미터로부터 측정된 전류 샘플을 수신하여 이를 파일에 기록한다.
- ⑥ 타겟 시스템은 또한 인터럽트 처리 루틴에서 현재 PID(Process Identifier)와 PC(Program Counter) 및 성능 지수를 수집한다.
- ⑦ 성능 모니터는 임베디드 S/W가 수행되면서 수행 코드 내에 포함된 시스템 콜 호출을 통해 성능 모니터링 data를 프로파일링 한다.
- ⑧ 임베디드 S/W가 수행 완료되면 메모리 버퍼에 수집된 S/W 수행 정보들을 파일 시스템에 저장한다.
- ⑨ 타겟 시스템은 이 파일을 호스트 시스템으로 송신한다.
- ⑩ 호스트 시스템은 타겟 S/W의 컴파일 시에 생성된 심볼 테이블과 S/W 수행 정보 및 전류 샘플에서 계산된 전력 값들을 비교 분석하여 연계하고 함수 별 코드 수행에 따른 전력 소모 정보를 출력한다.

3. H.264 디코더

본 논문에서 사용한 H.264 디코더는 H.264/AVC의 참

조 소프트웨어인 Joint Model(JM) 7.6버전이다. H.264 디코더의 기본 구조는 아래의 그림 3과 같다. 디코더는 한 장의 픽처를 읽어들이고, 6단계 계층 구조 frame, slice, macroblock(16X16)단위의 순서로 영상을 나누어 디코딩 한다. 먼저, 인코딩된 하나의 프레임을 메모리로 읽어 들이고 디코더 설정파일을 확인하여 어떤 형태로 디코딩 할 것인지 결정한다. 그런 다음 데이터 시퀀스의 헤더를 읽어 시퀀스의 슬라이스 타입(I,P,B), 프로파일, 레벨, 픽처 전체의 부호화 모드(엔트로피 부호화 모드, 양자화 파라미터 초기값) 등의 정보를 얻는다. 디코딩을 하기 위한 준비 과정이 모두 끝나면 매크로블럭 단위로 압축된 영상을 복원하며 복원 과정이 끝나면 메모리에 저장되고 복원된 영상은 화면으로 출력된다.

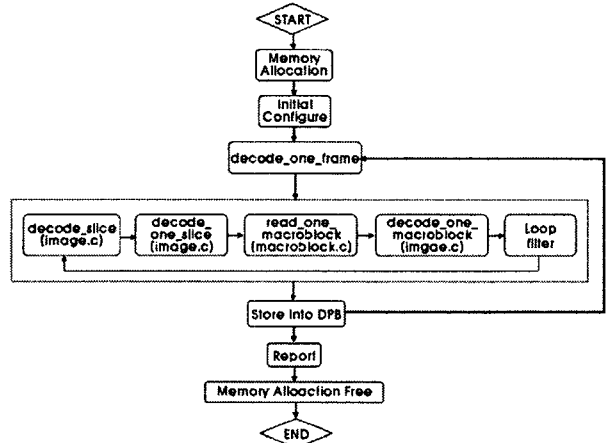


그림 3 H.264 디코더 주요 함수 흐름도

3.1 H.264 디코더의 전력 측정

H.264 디코더의 전력 소모량 측정은 Qplus/Esto에서 제공되는 전력 분석 도구를 이용하여 압축된 영상을 읽어 들여 복원하고 출력하는 과정 상의 주요함수별 전력소모량을 측정하는 방법으로 이루어졌다.

전력 측정에 사용된 파일은 "carphone.yuv", "foreman.yuv", "salesman.yuv", "suzie.yuv", "trevor.yuv"의 총 5개 yuv파일이 사용되었으며 각각 인코딩된 H.264파일은 총 프레임 수가 60 frame이고 시퀀스 타입은 I-P-B-P-B 형태로 모두 동일하다. 측정된 결과는 아래의 그림 4와 같은 형태로 나타나며 그림 4와 표 1에 나타난 측정 결과에 사용된 파일은 "foreman.yuv" 이다.

Function	Source	Code Size	Energy Consumption(mJ)	Execution Time(ms)
decode_nrio_size	image.c:1230	N/A	N/A	N/A
decode_frame_slice	image.c:1204	N/A	N/A	1.4645
decode_ipcm_rmb	macroblock.c:3005	N/A	N/A	N/A
decode_one_frame	image.c:1319	N/A	N/A	19.4458
decode_one_macroblock	macroblock.c:3066	N/A	14.5967	5.515 8594
decode_one_slice	image.c:1367	N/A	N/A	1.2405

그림 4 Esto 전력 분석 결과

H.264 디코더의 전력 소모량 측정 결과, 표 1과 같이

decode_one_macroblock(), dpb_split_field(), get_block(), itrans(), init_macroblock() 등의 특정 함수가 전력 소모에 큰 영향을 미치는 것으로 나타났다.

표 1 전력 소모에 영향을 주는 함수

주요 함수	전력 소모량(mJ)
decode_one_macroblock()	14.5967
dpb_split_field()	9.8862
get_block()	22.1242
itrans()	8.6893
init_macroblock()	3.7947

3.2 decode_one_macroblock() 함수 저전력화 실험

본 논문에서는 H.264 디코더의 전력을 줄일 수 있는 기법 중 한가지 예로 다음과 같은 방법을 제시한다. 위의 측정 결과에 따라 전력 소모에 영향을 주는 함수 중에 하나인 decode_one_macroblock() 함수의 형태를 변형하여 함수의 전력 소모량을 측정하고, 약간의 전력 감소가 발생하였다. 함수의 변형 과정은 아래의 그림 5와 같다.

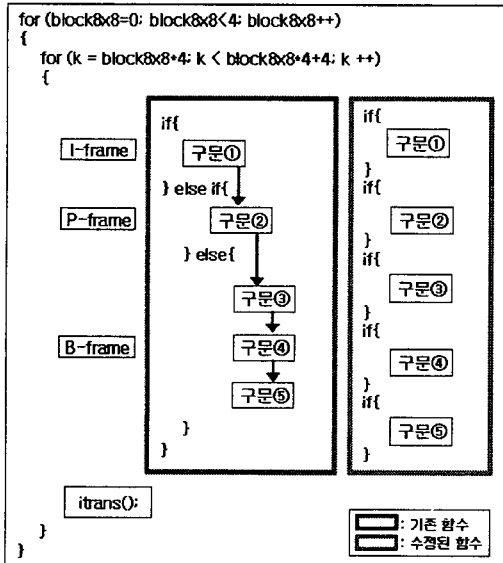


그림 5 수정된 decode_one_macroblock()

decode_one_macroblock() 함수는 압축된 정보를 복호화하는 주요 기능인 움직임 예측하고 보상하는 과정이 모두 포함된 중요한 함수로서 MB단위(16X16)로 영상을 복호화한다. 복호화 과정 시, Block 타입인 I, P, B 타입에 따라 각각 다른 과정으로 복호화가 이루어지는데 기존의 디코더 소스는 이 과정이 하나에 for() 문안에 if-else()의 제어문으로 구성되어있다. 이 제어문을 그림 5와 같이 여러 개로 나누어 decode_one_macroblock() 함수를 수정한 뒤 디코딩하여 전력 소모량을 측정하였다. 표 2는 yuv 파일별 디코딩 저전력소모량 측정값을 평

균을 내어 나타낸 표이다. Qplus/Esto로 디코더를 실행시킬 때마다 측정된 전력소모량의 편차가 있어 총 10번 수행 후, 평균을 내었다. 그 결과, 평균 15% 정도의 전력 소모량이 감소되었다.

표 2 decode_one_macroblock() 함수의 평균 전력 소모량(단위: mJ)

	수정 전	수정 후	감소 비율(%)
carphone.yuv	11.8040	10.6410	9.9
foreman.yuv	12.8114	10.8815	15.0
salesman.yuv	10.9679	8.7184	20.5
suzie.yuv	11.4031	10.6386	6.7
trevor.yuv	10.2740	8.0080	22.0
평균			14.8

이는 만약 디코더가 P-프레임을 처리하는 중이라고 가정할 때, 코드 수정 이전의 경우 함수 내부의 첫 번째 조건문은 'false'가 되어, P-프레임 처리를 위한 구문②는 물론, B-프레임 처리를 위한 구문③,④,⑤ 코드 부분의 사용 가능성에 대한 예측에 영향을 주게 된다. 따라서 현재 처리해야 할 프레임이 P-프레임이라서 구문②만 수행될지라도 페이지 선인출 매커니즘에 의하여 구문③,④,⑤에 해당하는 페이지를 메인메모리에 적재하는 경우가 발생할 수 있다. 그러나 수정 이후의 경우, 구문①~⑤이 코드 상으로 볼 때 병행관계이므로 이러한 상황이 발생할 가능성이 적어지게 되기 때문에 전력 소모가 감소되는 것으로 풀이된다.

4. 결 론

본 논문에서는 한국전자통신연구원에서 개발한 Qplus/Esto에서 제공하는 전력소모량 분석도구를 이용하여 H.264 디코더의 함수별 전력 소모량을 측정하고 이를 바탕으로 소프트웨어의 전력을 줄일 수 있는 방법 중 한가지 방법을 간단하게 실험하여 어플리케이션의 전력 소모가 줄어들 수 있다는 것을 확인해보았다. 본 논문을 통하여 얻어진 H.264 디코더의 함수별 전력 소모량 측정 결과를 이용하여 앞으로 소프트웨어차원의 저전력화에 대한 다양한 연구 방향을 제시하는데 도움이 될 것이라는 점에 의의를 둔다.

5. 참고문헌

- [1] <http://www.qplus.or.kr>
- [2] 임채덕, 우덕균, 김영국, "임베디드 S/W 전력 분석 및 최적화 도구," 전자공학회 논문지, 2006.2.
- [3] Iain E.G. Richardson "H.264 and MPEG-4", WILEY, 2004.
- [4] 角野 眞也, 菊池 義浩, 鈴木 輝彦, "H.264/AVC 비디오 압축 표준," 홍릉과학출판사, 2005.10.
- [5] 김지홍, "휴대용 임베디드 시스템에서의 전력 관리," ITFIND 주간기술동향 제 1195호, 2005.5.