내장형 프로세서에서의 미디어 처리를 위한 Shadow Register 기법

[1]안성수, [1]김현규, [1]이성재°, [2]오형철
[1]고려대학교 대학원 전자정보공학과
[2]고려대학교(서창) 공학부
ohyeong@korea.ac.kr

Shadow Register Scheme for Media Processing in Embedded Processors

[1]S.-S. Ahn, [1]H.-G. Kim, [1]S.-J. Lee°, [2]H.-C. Oh
[1]Dept. of Elec. & Info. Engineering, Graduate School, Korea University
[2]School of Engineering, Korea University at Seo-Chang

요 약

비교적 적은 수의 레지스터를 사용하는 내장형 프로세서에서 미디어 데이터를 처리할 때, 레지스터 부족으로 인하여 프로세서의 성능이 현저히 감소하는 경우가 있다. 본 논문에서는 이를 Shadow 레지스터 기법을 사용하여 해결하는 방안을 제안한다. 프로토타입 프로세서를 사용한 비교 실험에서, 제안된 기법은 약 16.7%의 하드웨어 추가로 구현될 수 있으며, 실행시간을 약 16~28% 감소시키고 실행 프로그램의 크기를 약 3.3~5% 감소시킬 수 있음을 보였다. 본 논문의 실험 결과는 이상적인 메모리 모델 하에서 얻어진 것으로서 실제적인 환경에서는 더욱 큰 이득이 예상된다.

1. 서 론

오늘날, 일반 프로세서에서뿐 아니라 내장형 프로세서에서도 미디어 데이터에 대한 처리 수요가 급증하고 있다. 이들 내장형 프로세서들은 흔히 일반 프로세서들에 비하여 보다 적은 수의 범용 레지스터(General-Purpose Register)들을 가지므로[1], 많은 양의 데이터를 동시에 사용하는 미디어 처리에 있어 범용 레지스터의 부족 문제와 이로 인하여 프로세서의 성능이 저하되는 문제를 보다 심각하게 겪는다.

본 논문에서는 내장형 프로세서의 미디어 데이터 처리 능력을 향상시키기 위하여 Shadow 레지스터를 사용하는 기법을 제안한다. Shadow 레지스터는 문맥 교환에 따르는 성능의 저하를 최소화하기 위한 기법으로 제안된 바 있다[2]. 제안된 기법을 평가하기 위하여, EISC (Extendable Instruction Set Computer) 프로세서[3]를 실험의 구체적인 대상으로 정하였다. EISC 프로세서는 저전력 내장형 시스템을 위하여 코드 크기를 압축하여 사용하는[1] 여러 프로세서들 중의 하나로서, 압축된 16 비트 명령어 집합을 정의하여 그대로 사용한다는 점에서, 코드를 압축 코딩하여 저장하였다가 사용 직전에 디코딩하는 다른 프로세서들과 구별된다. 본 논문에서는 기 설계된 프로토타입 프로세서 (이하 **base-p**라고 함)[4]에 제안된 기법을 추가한 프로세서 (이하 **shadow-p**라고 함)를 설계하고 실험으로 성능을 비교하였다. MAD(MPEG Audio Decoder) Lib.[5] 중의 IMDCT (Inverse Modified Discrete Cosine Transform)와 DCT (Discrete Cosine Transform) 및 IDCT(Inverse Discrete Cosine Transform)[6]를 수행하는 실험에서, 제안된 기법은 실행시간을 약 16~28% 감소시키고, 프로그램 크기를 약 3.3~5% 감소시킬 수 있음을 보였다.

본 논문의 구성은 다음과 같다. 제2절에서는 실험에 사용한 알고리즘들에 대해 설명하고, 제3절에서는 제안하는 Shadow Register에 대해 소개 하며, 제4절에서는 실험 방법과 Shadow Register를 사용해 얻어진 성능을 비교 및 평가 한 뒤, 제5절에서는 결론을 맺도록 한다.

2. 미디어 데이터 처리 알고리즘

2.1 IMDCT

MAD Lib.[5]는 ISO/IEC 11172-3 표준문서를 기반으로 프로그래밍된 라이브러리로서, MP3플레이어인 윈앰프(Winamp)등의 다양한 시스템에 널리 사용되고 있다.

표 1은 MPEG2 디코딩 알고리즘에서의 각 블록별 실행에 요구되는 연산 수행 능력을 보여 주고 있다. 표 1에서 볼 수 있듯이 IMDCT의 연산 수행 요구량은 전체 연산량의 약 70% 정도를 차지하고 있다. 즉, IMDCT의 수행 능력의 변화는 전체 MPEG2 디코더의 실행 능력에 상당한 영향을 미칠 수 있다는 것이다[7].

다음은 'layer3.c' 파일[5]에 프로그래밍 되어 있는 'imdct36' 함수의 일부분이다.

```
static inline
void imdct36( mad_fixed_t const X[18],
              mad_fixed_t z[36])
{
    mad_fixed_t t0, t1, t2, t3, t4, t5, t6, t7, t8;
    mad_fixed_t t8, t9, t10, t11, t12, t13, t14, t15;
    register mad_fixed64hi_t hi;
    register mad_fixed64lo_t lo;
```

'imdct36'함수는 18개의 입력 값과 36개의 출력 값 그

리고 16개의 임시 데이터 변수가 사용되고 2개의 누산기용 변수가 있는 것을 볼 수가 있다. 일반적인 내장형 프로세서의 16개의 레지스터로는 사용되는 변수 값을 동시에 처리 할 수가 없기 때문에 추가 비용이 발생한다.

Classification	Function	Required processing Power[MOPS]
Inverse Quantization	Degrouping	0.88
	Requantization	1.44
	Rescalization	0.96
Multichannel Processing	Dematrixing	0.576
	Denormalization	1.44
Synthesis Subband	IMDCT	61.44
	IPQMF	19.22
Total		86.656

표 1. MPEG2 디코딩 알고리즘의 단계별 요구 연산 능력[7]

2.2 DCT/IDCT

DCT와 IDCT 알고리즘은 JPEG 등의 표준 압축 알고리즘들에서 핵심이 되는 알고리즘이다[6,8]. 다음은 [8]에서 소개된 'fdct.c' 파일에 프로그래밍 되어 있는 'dct' 함수의 일부분이다. 'dct'함수 또한 많은 수의 입출력 데이터와 변수가 필요함을 볼 수 있다.

```
void FDCT(short block[8][8])
{
    int i, j, k;
    long dd;
    short temp[8][8]

    for(i=0; i<8; I++)
        for(j=0; j<8; J++)
```

3. Shadow 레지스터 기법

제2절에서 분석한 결과를 보면, IMDCT와 DCT/IDCT 알고리즘에서 동시에 많은 변수 값이 사용되는 것을 볼 수 있다. 일반적인 내장형 프로세서에서 지원되는 레지스터의 수로는 이러한 많은 변수 값을 동시에 처리 할 수가 없기 때문에, PUSH/POP 명령을 사용하여 데이터 값을 외부 메모리로 저장하고 복원하는 과정을 수행한다. 이와 같은 register spilling 과정은 데이터 처리 성능을 저하시킨다.

본 논문에서는 레지스터 부족 문제를 해결하기 위하여, 문맥 교환에 따르는 성능의 저하를 최소화하기 위한 기법으로 제안되어 있는 Shadow 레지스터 [2]를 이용하는 방안을 제안한다. 그림 1은 Shadow 레지스터의 구현 방안을 도시한 것이다. 상태 레지스터(Status Register)에 지정된 Flag(SR Flag) 비트를 두어 Flag의 상태에 따라 사용할 레지스터를 정하게 된다. 본 논문에서는 SR Flag

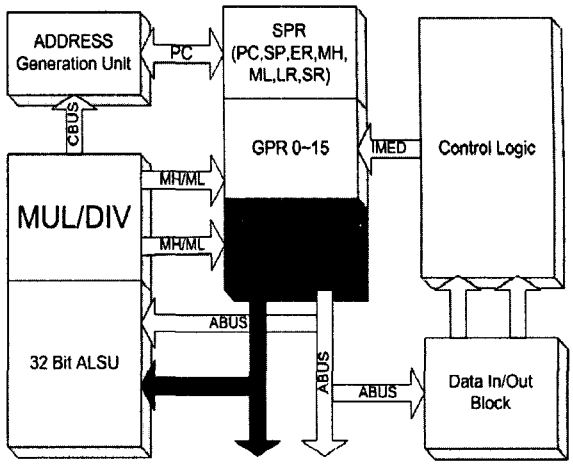

그림 1. Shadow 레지스터 파일의 구현

의 값을 1로 지정하면 Shadow 레지스터 파일을 선택하는 것으로 구현하였다. 본 논문의 실험에서는 Shadow 레지스터 파일을 한 개만 사용하였으나, 필요에 따라 Flag 비트를 추가하면 같은 방법으로 여러 개의 파일을 추가할 수 있다.

표 2는 **base-p**의 범용 레지스터들 중에서 사용이 유보된 레지스터를 보여 주고 있다. 이러한 레지스터들은 빈번히 접근되는 레지스터들로서 Shadow 레지스터를 사용할 때에도 동일하게 사용되므로, 따로 추가하지 않고 공통적으로 사용되도록 하였다.

General Purpose Register	목적
%R5	Frame Pointer
%R7	Index Register
%R8	Argument 0
%R9	Argument 1

표 2. 사용이 유보되어 있는 레지스터들

4. 모의실험 및 비교

본 논문에서 EISC 구조를 기반으로 기 설계된 프로토타입 프로세서인 **base-p** 프로세서를 대상으로, 제안된 기법의 성능을 실험으로 평가하였다. 제안된 기법을 추가한 **shadow-p** 프로세서를 설계하고, Synopsys사의 Design Compiler와 삼성 0.35㎛ 공정 셀 라이브러리를 사용하여 합성을 수행하였다. 임계 경로 지연 시간(critical path delay)의 추정에는 Worst Case Process, 3.3V, Junction Temp=+125℃ 의 동작 환경을 적용하였다.

4.1 IMDCT에 대한 성능 비교

base-p 프로세서와 **shadow-p** 프로세서를 대상으로, IMDCT 알고리즘의 어셈블리 코드를 손으로 각각 작성

하고 수행하였다. 표 3은 정적 실행 명령어 수와 실행 시간을 비교한 것이다. 표 3에서 볼 수 있듯이, 제안된 기법의 사용으로 프로그램의 크기는 약 5% 정도 감소하였고, 실행 시간은 약 16% 정도 감소하였다.

구분	base-p	shadow-p	감소율(%)
프로그램 크기 [Byte]	2864	2723	5%
실행시간 [cycles]	941	792	16%

표 3. IMDCT의 성능 비교

4.2 DCT/IDCT에 대한 성능 비교

DCT 알고리즘과 IDCT 알고리즘에 대해서도, **base-p** 프로세서와 **shadow-p** 프로세서를 대상으로 어셈블리 코드를 손으로 각각 작성하고 수행하였다. 표 4와 표 5는 각 경우에 대하여 정적 실행 명령어 수와 실행 시간을 비교한 것이다. 실행 시간에 있어 제안된 기법이 각각 27%와 28%의 성능 향상을 가져왔으며, 프로그램의 크기를 각각 4.3%와 3.3% 만큼 감소시킴을 알 수 있다.

구분	base-p	shadow-p	감소율(%)
프로그램 크기 [Byte]	1163	1113	4.3%
실행시간 [cycles]	1345	981	27%

표 4. DCT의 성능 비교

구분	base-p	shadow-p	감소율(%)
프로그램 크기 [Byte]	1147	1109	3.3%
실행시간 [cycles]	1212	872	28%

표 5. IDCT의 성능 비교

4.3 합성 결과

표 6은 합성 결과를 비교한 것이다. 확장된 레지스터의 구현에 의한 하드웨어의 증가는 MEM단계에서 반영되었다. Shadow 레지스터 파일의 추가로 전체적으로 약 16.7%의 하드웨어 사용이 증가된 것을 알 수 있다.

블록	base-p	shadow-p
IF_State	11134.66	11158.65
ID_Stage	13886.68	16732.33
EX_Stage	20198.67	20108.33
MEM_Stage	1495.00	6518.66
PControl	55.00	50.00
total	46772.67	54567.97

표 6. 합성 결과 비교 [등가 게이트 수]

5. 결 론

본 논문에서는 내장형 프로세서의 미디어 데이터 처리 능력을 향상시키기 위하여 Shadow 레지스터를 사용하는 기법을 제안하고 실험으로 평가하였다. 프로토타입 프로세서를 사용한 비교 실험에서, 제안된 기법은 약 16.7%의 하드웨어 추가로 구현될 수 있으며, 실행시간을 약 16~28% 감소시키고 실행 프로그램의 크기를 약 3.3~5% 감소시킬 수 있음을 보였다. 실험에 사용된 프로그램들은 음성신호와 영상신호의 처리에 핵심이 되는 프로그램들이다. 본 논문의 실험 결과는 이상적인 메모리 모델 하에서 얻어진 것으로서 실제적인 환경에서는 더욱 큰 이득이 예상된다.

감사의 글

본 연구는 반도체설계교육센터(IDEC)의 설계 도구 지원을 받아 수행되었습니다.

참고문헌

[1] C. Lefurgy, P. Bird, I. Chen, and T. Mudge, "Improving Code Density using Compression Techniques," *MICRO-30*, pp.194~203, Dec. 1997

[2] J. Jayaraj, P.L. Rajendran, and T. Thirumoolam, "Shadow Register File Architecture: A Mechanism to Reduce Context Switch Latency," *HiPC2002*, Poster Presentation, 2002

[3] 조 경연, "확장명령어 32비트 마이크로프로세서에 관한 연구," 전자공학회논문지, 제 36 권, D편, 제 5호, pp.11-20, May. 1999

[4] H.-G. Kim et al.. "AE32000B: A Fully Synthesizable 32-bit Embedded Microprocessor Core," *ETRI Journal*, Vol. 25, No. 5, pp. 337-344, Oct. 2003.

[5] http://www.underbit.com/products/mad

[6] Jae. S. Lim, "Two-Dimensional Signal and Image Processing," Prentice Hall, 1990.

[7] T.H. Tsai et al., "A low-cost architecture design with efficient data arrangement and memory configuration for MPEG-2 audio decoder," *ISCAS'98*, Vol. 4, pp.65-68,1998.

[8] 이문호, 염재훈. "C를 이용한 영상신호처리," 대영사, 1994.