

저전력 멀티미디어 소프트웨어 기술

숭실대학교 이성수

1. 서론

최근 들어 정보 통신 기술의 급격한 발달에 따라 PCS(personal communication system) 등의 이동 통신 기기, PDA(personal digital assistant) 등의 휴대용 정보 단말기의 중요성은 날로 증가하고 있다. 이러한 이동통신 기기와 휴대용 정보 단말기는 점차 하나로 통합되는 추세를 보이고 있는데, 핸드폰이나 PCS 같은 이동 통신 기기는 점차 대형 LCD 디스플레이를 채택하고 무선 인터넷을 통하여 다양한 멀티미디어 콘텐츠를 제공하기 시작했고, PDA와 포켓PC 같은 휴대용 정보 단말기는 외부 통신 모듈을 장착하여 자동차 운행 정보 시스템(car navigation system)이나 VOD(video-on-demand) 서비스를 지원하기 시작하였다. 이렇게 이동 통신 기기와 휴대용 정보 단말기가 통합된 포스트PC(post-PC)는 2000년부터 시장에 급속하게 보급되기 시작하였으며, IMT-2000이 상용화되는 수년 이내에 폭발적인 시장 수요가 예상된다.

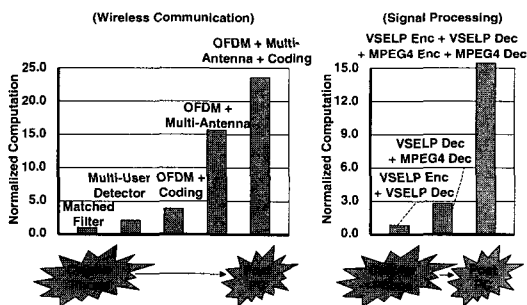


그림 1 휴대폰과 포스트PC의 연산량 비교[2]

이러한 포스트PC는 이동 통신, 무선 인터넷, 멀티미디어 등의 여러 가지 기능을 지원하기 때문에 고성

능, 저전력화되어야 한다[1]. 그림 1은 휴대폰과 포스트PC에서 무선 통신과 멀티미디어 신호 처리의 연산량을 비교한 것인데, 휴대폰이 포스트PC로 대체되면서 연산량이 대략적으로 15배 가량 증가함을 알 수 있다[2].

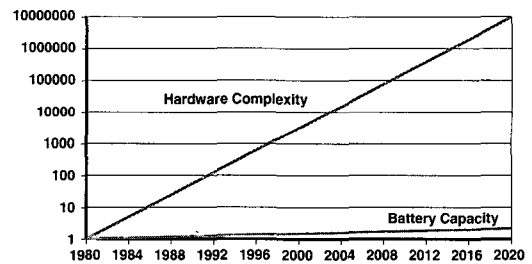


그림 2 마이크로프로세서의 하드웨어 복잡도 증가 및 배터리 전력 용량 개선[2]

문제는 포스트PC가 고성능화 되어감에 따라 전력 소모는 계속 증가하는데 비해서 배터리의 전력 용량은 거의 발전이 없다는 것이다. 그림 2는 1980년부터 2020년까지 마이크로프로세서 하드웨어의 복잡도 증가 및 배터리의 전력 용량 개선을 대략적으로 예측한 것이다[2]. 그림에서 보듯 배터리의 전력 용량은 40년 동안에 3-4배 정도밖에는 증가하지 않을 것으로 예측되는데, 그 이유는 배터리의 전력 용량이 기본적으로 배터리 내부의 화학 물질에 의해서 결정되기 때문이다. 따라서 전력을 획기적으로 줄이는 기술을 개발하지 않는 한 배터리의 연속 동작 가능 시간이 점점 짧아져서 사실상 휴대용으로 사용이 불가능하게 될 수도 있다[3,4,5]. 전력 소모가 증가함에 따라 발생하는 또다른 문제점으로는 마이크로프로세서 내부의 열 발생을 들 수 있다. 그림 3은 세계 반도체 기술 동향을 연구하는 기관인 NTRS(National Tech-

nology Roadmap for Semiconductors)에서 1994년 및 1999년에 발표한 자료인데[6], 1999년 예측 결과는 1994년 예측 결과에 비해서 열 발생이 훨씬 빠르게 증가할 것임을 시사하고 있으며, 획기적인 저전력 기술이 개발되지 않는다면 2009년쯤에는 마이크로프로세서의 열 발생이 200W를 넘어서 기존의 냉각 기술로 감당하기 어려울 정도로 이를 것으로 예측되고 있다. 이러한 열 발생은 칩의 성능을 저하시키고 오 동작의 원인이 되며 심지어는 영구적인 칩 손상을 초래하기도 한다.

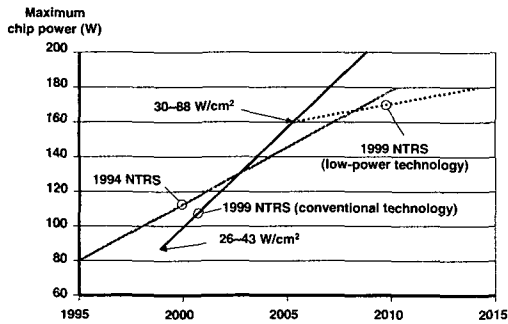


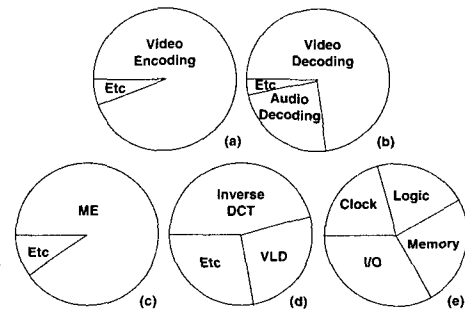
그림 3 마이크로프로세서의 열 발생 예측[6]

이러한 문제점을 해결하기 위하여 많은 저전력 기술이 제안되어왔으나, 이들 기술은 대부분 소프트웨어 쪽보다는 하드웨어 쪽에 집중되어온 것이 사실이다. 최근 들어 포스트PC를 비롯한 각종 휴대 기기의 멀티미디어 기능이 강조되면서 여러 가지 저전력 멀티미디어 소프트웨어 기술이 활발하게 연구되기 시작하였다. 본 논문에서는 현재 널리 사용되고 있는 멀티미디어 신호 처리 기술 및 그 전력 소모 특성을 분석하고, 이를 바탕으로 개발된 각종 저전력 멀티미디어 소프트웨어 기술에 대해 살펴보고자 한다.

2. 멀티미디어 신호 처리 기술 및 전력 소모 분석

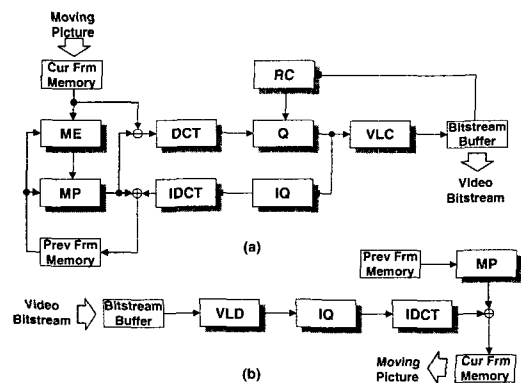
포스트PC를 비롯한 대부분의 휴대 기기에서 사용되는 멀티미디어 관련 기술은 크게 부호화(encoding), 복호화(decoding), 비트열 다중화(multiplexing), 역다중화(demultiplexing), 내용 검색(content retrieval), 사용자 인터페이스(user interface) 등으로 나눌 수 있는데, 이 중에서 대부분의 연산량 및 전력 소모를 차지하는 것은 부호화와 복호화이다. 부호화

는 멀티미디어 데이터를 저장, 전송에 용이하도록 압축하는 기술이며 복호화는 압축된 데이터를 원래의 멀티미디어 데이터로 복원하는 기술이다. 이들 부호화 및 복호화 기술은 MPEG-2[7]나 MPEG-4[8]와 같이 국제 회의에서 표준으로 정의된 기술도 있고 Real Media[9]와 같이 소프트웨어 회사에서 자체 개발된 것이 널리 사용되어 사실상의 표준처럼 쓰이는 기술도 있으나, 이들 기술을 구성하는 기본 요소는 대체적으로 대동소이하다. 부호화 및 복호화는 각각 비디오와 오디오, 시스템의 세 부분으로 나누어지는데, 포스트PC를 비롯한 대부분의 휴대용 멀티미디어 기기에서는 그림 4(a),(b)와 같이 비디오가 전체 연산량 및 전력 소모의 70~90%를 차지하기 때문에, 본 논문에서는 비디오 부호화 및 복호화를 주로 다루기로 한다.



- (a) 부호화 (b) 복호화 (c) 비디오 부호화
- (d) 비디오 복호화
- (e) 비디오 부호화를 수행할 때 각 하드웨어 부분에서 소모되는 전력

그림 4 멀티미디어 프로세서의 전력 소모 분석



- (a) 비디오 부호화 (b) 비디오 복호화

그림 5 MPEG-2 비디오 압축 기술

그림 5는 가장 널리 쓰이는 비디오 부호화 및 복호화 기술인 MPEG-2[7]을 나타낸 것인데, 여기에는 ME[9] (motion estimation), MP[9] (motion prediction), DCT[10] (discrete cosine transform), IDCT [10] (inverse discrete cosine transform), Q (quantization), IQ (inverse quantization), VLC[11] (variable length coding), VLD[11] (variable length decoding), RC (rate control) 등의 기술 요소가 사용된다. MPEG-4[8], Real Media[12], Microsoft Window Media Technology[13] 등의 다른 표준도 기본적으로는 그림 5와 거의 동일한 기술 요소를 사용한다. 그림 4(c),(d)는 이들 기술 요소가 실제 멀티미디어 신호 처리에서 차지하는 전력 소모를 나타낸 것인데, 비디오 부호화에서는 ME가, 비디오 복호화 기술에서는 IDCT 및 VLD가 대부분의 전력을 소모하는 것을 알 수 있다. 이들 ME, IDCT 및 VLD의 전력 소모 특성을 분석해보면 다음과 같다.

2.1 ME(Motion Estimation)

ME는 현재 프레임과 이전 프레임의 시간적 중복성을 제거하여 압축률을 높이는 과정으로서, 이전 프레임에서 현재 프레임의 기준 블록(reference block)과 '가장 닮은' 블록인 정합 블록(best-matched block)을 찾은 후, 기준 블록과 정합 블록의 위치 차이인 움직임 벡터와, 기준 블록과 정합 블록의 화소값 차이인 예측 오차를 생성한다. ME는 그림 6과 같은 수식으로 표현될 수 있다. 이때 $R(x,y)$ 는 현재 프레임의 (x,y) 번째 화소값, $S(x,y)$ 는 이전 프레임의 (x,y) 번째 화소값을 나타내며, 기준 블록과 정합 블록의 차이를 나타내는 척도인 SAD(sum of absolute differences)를 최소화 하는 (u,v) 의 값이 움직임 벡터가 된다.

ME에서는 u 와 v 를 변화시키며 SAD값을 구하는 과정에서 대부분의 전력 소모가 이루어진다. ME는 이론상으로 매 기준 블록마다 동일한 횟수의 화소값 비교가 수행되기 때문에 연산량이 대체적으로 일정해야 하지만, 실제적으로는 화소값을 256번 비교하기 전이라도 지금까지 계산된 sad 값이 $sadmin$ 을 넘을 경우 더 이상의 연산은 이루어지지 않고 (그림 6의 ①) 다음 (u,v) 로 넘어가게 되는데, 이를 조기 중단 (early cut-off)이라고 한다. 따라서 실제 수행되는 연산이 경우에 따라 크게 차이가 나기 때문에 ME

```

sadmin = MAXINT; mvx=0; mvy=0;
for (u=-sx; u<=sx; u++)
  for (v=-sy; v<=sy; v++)
  {
    sad = 0;
    for (i=0; i<16; i++)
    {
      for (j=0; j<16; j++)
        sad += |R(i,j)-S(i+u,j+v)|
      if (sad>=sadmin)
        break;
    }
    if (sad<sadmin)
    {
      sadmin = sad; mvx = u; mvy = v;
    }
  }

```

그림 6 ME 알고리즘

의 연산량은 시간적으로 큰 차이를 보이게 된다. 연산량의 시간적인 변화를 이용하여 공급 전압을 조절함으로써 전력 소모를 줄이는 가변 전압 프로세서 (variable voltage processor)[14]에서는 이러한 조기 중단을 이용하여 전력 소모를 크게 줄일 수 있다.

ME의 다른 특징으로는 메모리 입출력이 매우 자주 일어난다는 것을 들 수 있다. 그림 6에서 보듯이 하나의 기준 블록을 처리하기 위해서는 최대 $4sx \times 4sy \times 256 \times 3$ 번의 덧셈(또는 뺄셈)과 $4sx \times 4sy \times 256 \times 2$ 번의 메모리 입출력이 필요하므로 연산의 횟수와 메모리 입출력 횟수가 비슷하게 된다. 하드웨어적으로는 덧셈기보다 메모리 입출력이 더 많은 전력을 소모하기 때문에, 그림 4(e)와 같이 로직보다 I/O와 메모리에서 소모되는 전력이 더 많게 된다. 따라서 한번 읽어들인 화소값을 나중에 재이용한다는지 하는 방법을 써서 메모리 입출력의 횟수를 줄이는 편이 연산의 횟수를 줄이는 것보다 전력 소모 면에서 더 효과적이다.

2.2 IDCT(inverse discrete cosine transform)

저주파 영역에 에너지를 집중시키는 변환인 DCT와 이의 역변환인 IDCT는 거의 대부분의 동영상 및 정지 영상의 부호화 및 복호화에 폭넓게 사용되고 있다. DCT를 통과한 변환 계수는 저주파 영역에 대부분의 에너지가 집중되므로, 에너지가 적은 고주파 영

역을 양자화시켜서 제거하면 높은 압축률을 얻을 수 있다. 대부분 8×8 화소 블록 단위로 수행되는 DCT 및 IDCT는 행렬 곱셈(matrix multiplication)의 형태를 가지므로 주로 MAC(multiplier-accumulator)로 구현되며, 범용 마이크로프로세서나 DSP(digital signal processor)는 이러한 행렬 곱셈을 손쉽게 수행할 수 있는 전용 하드웨어와 명령어를 대부분 가지고 있다. ME와 달리 DCT 및 IDCT는 메모리 입출력보다는 곱셈 연산의 비중이 훨씬 크기 때문에 메모리 입출력 횟수가 늘어나더라도 연산량을 줄이는 알고리즘을 개발하는 것이 전력 소모를 줄이는 데 더 효과적이다.

DCT 및 IDCT는 ME와 달리 연산량이 시간에 따라 큰 변화를 보이지 않기 때문에 가변 전압 프로세서를 사용하여 전력 소모를 줄이기가 곤란하지만, DCT를 통과한 변환 계수가 양자화를 거치면서 8×8 화소 블록의 변환 계수가 모두 0이 되는 경우가 자주 발생한다. MPEG-2[7]나 MPEG-4[8]를 비롯한 대부분의 비디오 부호화 및 복호화 기술에서는 양자화된 8×8 화소 블록의 변환 계수가 모두 0일 때, 64개의 0을 전송하는 대신에 그 화소 블록 전체를 생략하고 보내는데, 이때 비디오 복호화에서는 생략된 화소 블록에 대해 IDCT를 수행할 필요가 없어진다. 따라서 비디오 복호화에서는 IDCT가 수행되는 경우와 수행되지 않는 경우에 큰 연산량의 차이가 발생하고, 이를 이용하여 가변 전압 프로세서 상에서 전력 소모를 크게 줄일 수 있다.

2.3 VLD(variable length decoding)

비디오 부호화에서는 압축 효율을 높이기 위해서 양자화된 DCT 변환 계수를 가변길이를 갖는 코드워드로 부호화한다. VLD는 비디오 복호화에서 이러한 가변길이 코드워드들을 원래의 DCT 변환 계수로 되돌리는 과정인데, 비트열을 1비트에서 40비트까지 좌우로 쉬프트시켜가면서 코드워드를 찾아내야 하기 때문에 연산량 자체는 별로 많지 않지만 프로그램의 제어흐름이 매우 복잡하다. 비트열을 처리하는 별도 하드웨어를 갖춘 멀티미디어 전용 칩에서는 VLD에서 소모되는 전력이 무시할 만 하지만, 범용 마이크로프로세서나 DSP에서는 복잡한 제어흐름 때문에 많은 사이클 수가 소요되고, 이에 따라서 전력 소모도 상당히 커지게 된다. 범용 마이크로프로세서나

DSP에서 소프트웨어적으로 VLD의 전력 소모를 효과적으로 줄이기 위한 연구가 계속되고 있으나, 아직까지는 큰 성과는 거두지 못하고 있다.

3. 저전력 멀티미디어 소프트웨어 기술

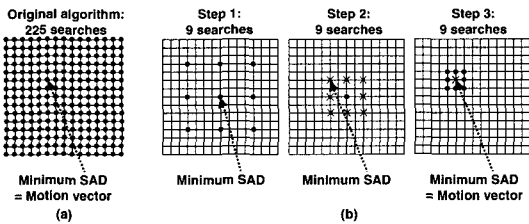
저전력 멀티미디어 소프트웨어 기술은 한마디로 “어떻게 하면 성능을 유지하면서 연산량을 줄이고 전력 소모를 줄일 수 있는가”라고 표현할 수 있다. 이들 기술은 크게 (1) 동일한 성능을 얻는데 필요한 연산량을 줄이는 방법, (2) 메모리에서 화소값을 읽어들이는 입출력의 횟수를 줄이는 방법, (3) 실시간 시스템에서 태스크의 유휴 시간(slack time)을 이용하여 공급 전압을 동적으로 낮추는 방법으로 나눌 수 있다. 이들 기술은 대부분 전력 소모가 많은 ME와 IDCT만을 대상으로 하고 있으나, 최근 들어 비디오 부호화 및 복호화 알고리즘 전체를 전력 소모 측면에서 최적화하기 위해 압축 알고리즘과 비트열 문법(bitstream syntax) 전체를 새로이 설계하려는 시도가 이루어지고 있다.

3.1 연산량 저감

기존에 연구된 저전력 멀티미디어 소프트웨어 기술의 대부분을 차지하는 방법[15-21]으로, 엄밀한 의미의 저전력 기술이라기보다는 저연산량 알고리즘 개발에 가깝다고 할 수 있다. 즉, 동일한 성능을 얻는데 필요한 연산량을 줄임으로서 이에 따른 연산 사이클 수 및 동작 속도 등을 낮추고, 그 결과로 전력 소모가 감소하게 된다. ME에서 연산량을 줄이는 방법은 그림 7과 같이 성능은 조금 떨어지더라도 연산량이 훨씬 작은 고속 탐색 알고리즘[15-17]을 사용하거나 2.1절에서 언급된 조기 중단을 이용하여 불필요한 연산을 수행하지 않도록 하는 방법[18,19]이 사용된다. IDCT에서는 행렬 곱셈에 사용되는 IDCT 계수의 특성을 이용하여 그림 8과 같이 곱셈의 개수를 줄이거나[20], IDCT 변환 계수가 0인 부분에 대해서는 곱셈을 수행하지 않고 건너뛰어서 곱셈의 개수를 줄이는 방법[21]이 사용된다.

이러한 방법들은 범용 마이크로프로세서나 DSP에서 별도의 하드웨어 추가 없이 순수하게 소프트웨어적인 방법으로 구현할 수 있다는 장점이 있기 때문에 널리 사용되고 있지만, 전력 소모라는 측면에서 본다면 반드시 효율이 높다고는 볼 수 없다. 저연산

량이 꼭 저전력으로 연결되는 것이 아니며, 연결된다 해도 연산량과 전력 소모가 대체적으로 선형 비례를 이루기 때문에, 연산량을 1/2로 줄이면 전력 소모는 기껏해야 1/2로 감소할 뿐이며, 대개는 그보다 훨씬 적은 폭으로 감소하게 된다. 이러한 문제점을 극복하기 위해서 최근에는 메모리 대역폭을 줄이는 방법이나 가변 전압 프로세서를 이용하는 방법과 함께 사용하려는 시도가 활발하게 이루어지고 있다.



(a) 기존 ME (b) 저연산량 ME

그림 7 ME의 연산량 저감

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} C_{00} & C_{10} & C_{20} & C_{30} & C_{40} & C_{50} & C_{60} & C_{70} \\ C_{01} & C_{11} & C_{21} & C_{31} & C_{41} & C_{51} & C_{61} & C_{71} \\ C_{02} & C_{12} & C_{22} & C_{32} & C_{42} & C_{52} & C_{62} & C_{72} \\ C_{03} & C_{13} & C_{23} & C_{33} & C_{43} & C_{53} & C_{63} & C_{73} \\ C_{04} & C_{14} & C_{24} & C_{34} & C_{44} & C_{54} & C_{64} & C_{74} \\ C_{05} & C_{15} & C_{25} & C_{35} & C_{45} & C_{55} & C_{65} & C_{75} \\ C_{06} & C_{16} & C_{26} & C_{36} & C_{46} & C_{56} & C_{66} & C_{76} \\ C_{07} & C_{17} & C_{27} & C_{37} & C_{47} & C_{57} & C_{67} & C_{77} \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \end{bmatrix}$$

(a)

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} A & B & A & C \\ A & C & -A & -B \\ A & -C & -A & B \\ A & -B & A & -C \end{bmatrix} \begin{bmatrix} X_0 \\ X_2 \\ X_4 \\ X_6 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} D & E & F & G \\ E & -G & -D & -F \\ F & -D & G & E \\ G & -F & E & -D \end{bmatrix} \begin{bmatrix} X_1 \\ X_3 \\ X_5 \\ X_7 \end{bmatrix}$$

$$\begin{bmatrix} x_7 \\ x_6 \\ x_5 \\ x_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} A & B & A & C \\ A & C & -A & -B \\ A & -C & -A & B \\ A & -B & A & -C \end{bmatrix} \begin{bmatrix} X_0 \\ X_2 \\ X_4 \\ X_6 \end{bmatrix} - \frac{1}{2} \begin{bmatrix} D & E & F & G \\ E & -G & -D & -F \\ F & -D & G & E \\ G & -F & E & -D \end{bmatrix} \begin{bmatrix} X_1 \\ X_3 \\ X_5 \\ X_7 \end{bmatrix}$$

(b)

(a) 기존 IDCT (b) 저연산량 IDCT

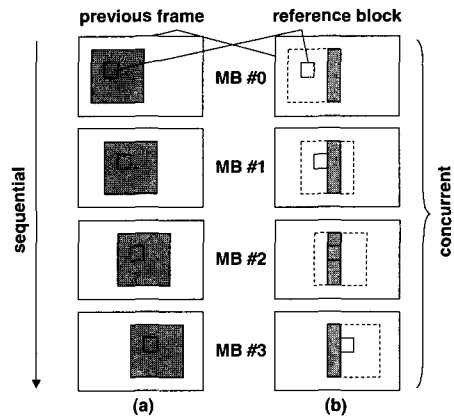
그림 8 IDCT의 연산량 저감

3.2 메모리 입출력 저감

2.1절에서 언급되었듯이 ME의 경우에는 화소값 비교 연산에 못지 않게 메모리 입출력의 전력 소모가 큰 비중을 차지한다. 그림 4(e)에서 볼 수 있듯이 비디오 부호화에서 메모리와 I/O의 소모 전력은 전체의 절반이 넘고, 이중에서 많은 부분이 ME가 화소값을 읽어오는 과정에서 소모된다. 이러한 문제점을 해

결하기 위해서는 다음과 같은 방법들이 사용된다.

가장 많이 쓰이는 것은 기존의 소프트웨어 기법을 사용하여 루프를 최적화하는 방법이다. 그림 6에서 볼 수 있듯이 ME 연산은 여러 개의 루프로 구성되어 있는데, 이때 작업 변수를 레지스터에 재할당하거나 (register reallocation), 여러 개의 루프를 하나로 합치거나 (loop fusion) 반대로 하나의 루프를 여러 개로 나누거나 (loop distribution) 루프를 반복된 코드로 풀거나 (loop unrolling) 동일한 수행 결과를 갖는 다른 루프로 변환하거나 (loop transformation) 캐쉬 특성을 고려하여 타일 형태의 루프를 만들면 (loop tiling) 수행 속도를 높임은 물론, 루프 내부에서 읽어들이는 데이터를 효과적으로 재사용할 수 있다[22]. 이 방법의 가장 큰 장점은 상용화된 루프 분석 도구 및 최적화 도구를 이용할 수 있다는 점이다.



(a) 기존 ME (b) 매크로블록 폴딩 ME

그림 9 매크로블록 폴딩 [23]

그림 9와 같이 여러 개의 기준 블록을 한꺼번에 비교하는 방법(macroblock folding)[23]도 메모리 입출력 횟수를 줄이는데 효과적이다. 진한 색으로 표시된 부분이 이전 프레임에서 읽어와야 하는 화소값인데, 기존의 방법에서는 MB#0, MB#1, MB#2, MB#3...과 같은 식으로 순차적으로 움직임 벡터를 계산하지만, 매크로블록 폴딩에서는 MB#0, MB#1, MB#2, MB#3에 공통으로 해당되는 이전 프레임 영역을 결정 한 후, 이들 매크로블록에 해당하는 움직임 벡터를 한꺼번에 계산한다. 이 경우, 읽어들이는 이전 프레임의 화소값 개수가 크게 감소하여 메모리 입출력 횟수를 크게 줄일 수 있다. 이외에도 프로그램의 구조와 수행

순서를 분석하여 한 번 읽어들이는 데이터가 캐쉬 상에 가능한 한 오래 머물러 있으면서 자주 재사용되도록 하는 방법[24,25]도 있으나, 데이터 재사용이 효율적인 반면에 프로그래밍이 어려워지는 단점이 있다.

이러한 방법을 잘 사용하면 메인 메모리에서 읽어오는 데이터의 양을 크게 줄일 수 있으며 특히 비교적 큰 용량의 캐쉬를 가지고 있는 마이크로프로세서나 DSP에 적합하나, 읽어들이야 할 데이터 숫자가 비교적 적은 IDCT에는 사용할 수 없다는 단점이 있다.

3.3 동적 전압 제어

실시간 시스템에서 처리해야 할 연산량이 최악(worst-case execution) 연산량보다 작은 경우에는 주어진 제한 시간(deadline) 내에 간신히 끝나도록 프로세서의 동작 속도를 늦추고, 동작 속도가 늦어진 만큼 공급 전압(supply voltage)을 낮출 수 있다. 디지털 회로의 동작 속도는 대체적으로 공급 전압에 비례하고 소비전력은 대체적으로 공급 전압의 제곱에 비례하기 때문에, 동작 속도를 1/N으로 늦추면 소비 전력은 대략적으로 1/N²로 감소하게 된다. 그림 10에서 최악 연산량이 1.25×10⁹ 사이클, 클록 주파수가 50MHz, 공급 전압이 5.0V라고 가정하고 동작 속도가 공급 전압에 비례한다고 가정한다면, 실제 처리해야 할 연산량이 5×10⁸ 사이클일 때에는 공급 전압을 2.0V까지 줄일 수 있으며, 이때의 전력 소모는 공급 전압이 5.0V일 때의 1/6.25로 감소하게 된다. 이러한 방법을 동적 전압 제어(dynamic voltage scaling)[3]라고 한다. 동적 전압 제어를 위해서는 공급 전압을 소프트웨어적으로 조절할 수 있는 가변 전압 프로세서가 필요한데, 이미 인텔과 트랜스메타에서 각각 Xscale[26]과 Crusoe[27]라는 가변전압 프로세서를 출시하여 현재 핸드폰, PDA, 노트북 컴퓨터 등에 널리 사용되고 있으며, 가변 전압 프로세서 없어도 기존의 마이크로프로세서 보드에 간단한 전원 제어 하드웨어를 덧붙여서 사용하는 방법[4]도 제안되었다.

동적 전압 제어에서는 최악 연산량과 평균 연산량의 차이인 연산량 변화(workload fluctuation)가 클수록 전력 소모를 크게 줄일 수 있으며, 특히 멀티미디어 소프트웨어, 특히 비디오 부호화 및 복호화에 적합하다고 알려져 있다. 이론상으로는 ME의 경우 평균 연산량이 최악 연산량의 1/10 정도에 불과하기 때

문에 전력 소모를 1/10~1/100 정도로 줄일 수 있고, 비디오 부호화 및 복호화의 다른 과정보다 1/3~1/10 정도로 줄일 수 있다. 3.1절과 3.2절에서 설명한 방법과는 달리, 동적 전압 제어는 비디오를 제외한 멀티미디어 소프트웨어의 다른 연산에도 쉽게 적용할 수 있고, 전력 소모 개선도 훨씬 크다는 장점을 가지고 있다.

동적 전압 제어의 핵심은 멀티미디어 소프트웨어의 실행 상황을 주시하면서 어떻게 하면 주어진 제한 시간을 어기지 않으면서 전력 소모가 최소가 되도록 공급 전압을 결정하느냐이다. 현재까지 프로그램의 구조를 분석하는 방법[28], 프로그램 수행 시간에 대한 프로파일을 분석하는 방법[29,30], 여러장의 비디오 프레임 버퍼를 사용하여 제한 시간 조건을 완화시키는 방법[31], 성능과 전력 소모의 절충(tradeoff between performance and power)을 고려하여 공급 전압을 결정하는 방법[32] 등이 있다. 현재 동적 전압 제어는 저전력 멀티미디어 소프트웨어 기술 분야에서 가장 효과적인 방법 중 하나로 평가받고 있으며, 최근 들어 매우 활발하게 연구되고 있다.

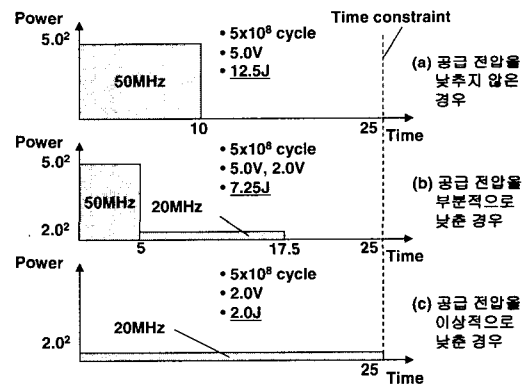


그림 10 동적 전압 제어

4. 결론

본 논문에서는 저전력 멀티미디어 소프트웨어 기술의 필요성을 살펴보고, 현재 널리 사용되고 있는 멀티미디어 신호 처리 기술 및 그 전력 소모 특성을 분석하였다. 또한 이러한 분석을 바탕으로 개발된 각종 저전력 멀티미디어 소프트웨어 기술을 비디오 부호화 및 복호화를 중심으로 살펴보았다. 이러한 저전력 기술은 포스트PC를 비롯한 각종 휴대용 멀티미디어

어 기기의 전력 소비를 획기적으로 줄임으로서 기기의 소형화, 경량화, 연속 동작 시간 연장에 크게 이바지할 수 있을 것으로 생각된다.

참고문헌

- [1] S. Lee, "Low-power Technologies in Wireless Communication," Proceedings of EIST BK21 Annual Symposium, pp. 1-40, 2001.
- [2] J. Rabaey, "Wireless beyond the Third Generation," Proceedings of International Symposium on Low-Power Electronics and Design, pp. 1-3, 2001.
- [3] A. Chandrakasan and R. Brodersen, *Low Power Digital CMOS Design*, Kluwer Academic Publishers, 1995.
- [4] S. Lee and T. Sakurai, "Run-time Voltage Hopping for Low-Power Real-Time Systems," Proceedings of Design Automation Conference, pp. 806-809, 2000.
- [5] D. Shin, J. Kim and S. Lee, "Intra-Task Voltage Scheduling for Low-Energy Hard Real-Time Applications", IEEE Design and Test of Computers, vol. 18, no 2. pp. 20-30, Mar. 2001.
- [6] C. Belady, "Cooling and Power Considerations for Semiconductors into the Next Century," Proceedings of International Symposium on Low-Power Electronics and Design, pp. 100-105, 2001.
- [7] ISO/IEC JTC1/SC29/WG11 13818, "Coding of Moving Pictures and Associated Audio," Nov. 1994.
- [8] ISO/IEC JTC1/SC29/WG11 14496, "Coding of Audio-Visual Objects," Oct. 1998.
- [9] J. Jain and A. Jain, "Displacement Measurement and Its Application in Interframe Image Coding," IEEE Transactions on Communications, vol. 29, no. 12, pp. 1799-1808, Dec. 1981.
- [10] N. Ahmed, T. Natarajan, and K. Rao, "Discrete Cosine Transform," IEEE Transactions on Computer, vol. 23, no. 1, pp. 90-93, Jan. 1974.
- [11] D. Huffman, "A Method for the Construction of Minimum Redundancy Codes," Proceedings of IRE, vol. 40, pp. 1098-1101, 1952.
- [12] <http://www.real.com>
- [13] <http://www.microsoft.com/windows/windows-media>
- [14] T. Burd et al., "A Dynamic Voltage Scaled Microprocessor System," Proceedings of International Solid-State Circuits Conferences, pp. 294-295, 2000.
- [15] T. Koga, K. Iinuma, K. Hirano, Y. Iijima, and T. Ishiguro, "Motion Compensated Interframe Coding for Video Conferencing," Proceedings on National Telecommunication Conference, G.5.3.1-5.3.5, 1981.
- [16] H. Jong, L. Chen, and T. Chiueh, "Accuracy Improvement and Cost Reduction of 3-step Search Block Matching Algorithm for Video Coding," IEEE Transactions on Circuits and Systems for Video Technology, vol. 4, no. 1, pp. 88-90, Jan. 1994.
- [17] J. Kim and T. Choi, "A Fast Motion Estimation for Software Based Real-Time Video Coding," IEEE Transactions on Consumer Electronics, vol. 45, no. 2, pp. 417-426, May 1999.
- [18] H. Huang, Y. Hung, and W. Hwang, "Adaptive Early Jump-out Technique for Fast Motion Estimation in Video Coding," Proceedings of International Conference on Pattern Recognition, pp. 864-868, 1996.
- [19] V. Moshnyaga, "A New Computationally Adaptive formulation of Block-Matching Motion Estimation," IEEE Transactions on Circuits and Systems for Video Technology, vol. 11, no. 1, pp. 118-124, Jan. 2001.
- [20] W. Chen, C. Smith, and S. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform," IEEE Transactions on Communication, vol. 25, no. 9, pp. 1004-1009, Sep. 1977.
- [21] T. Xanthopoulos and A. Chandrakasan, "A

- Low-Power DCT Core Using Adaptive Bitwidth and Arithmetic Activity Exploiting Signal Correlations and Quantization," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 5, pp. 740-750, May 2000.
- [22] W. Wolf, *Computers as Components*, Morgan Kaufmann, 2000.
- [23] E. Brockmeyer, L. Nachtergaele, F. Catthoor, J. Bormans, and H. De Man, "Low Power Memory Storage and Transfer Organization for the MPEG-4 Full Pel Motion Estimation on a Multimedia Processor," *IEEE Transactions on Multimedia*, vol. 1, no. 2, pp. 202-216, Jun. 1999.
- [24] J. Diguët, S. Wuytack, F. Catthoor, and H. De Man, "Formalized Methodology for Data Reuse Exploration for Low-Power Hierarchical Memory Mapping," *IEEE Transactions on VLSI Systems*, vol. 6, no. 4, pp. 529-537, Dec. 1998.
- [25] E. Greef, F. Catthoor, and H. De Man, "Program Transformation Strategies for Memory Size and Power Reduction of Pseudoregular Multimedia Subsystems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 6, pp. 719-733, Oct. 1998.
- [26] <http://www.intel.com/design/intelxscale>
- [27] <http://www.transmeta.com/technology>
- [28] D. Shin, J. Kim, and S. Lee, "Low-Energy Intra-Task Voltage Scheduling Using Static Timing Analysis," *Proceedings of Design Automation Conference*, 2001.
- [29] D. Shin and J. Kim, "A Profile-Based Energy-Efficient Intra-Task Voltage Scheduling Algorithm for Hard Real-Time Applications," *Proceedings of International Symposium on Low-Power Electronics and Design*, 2001.
- [30] J. Pouwelse, K. Langendoen, and H. Sips, "Energy Priority Scheduling for Variable Voltage Processors," *Proceedings of International Symposium on Low-Power Electronics and Design*, pp. 28-33, 2001.
- [31] C. Im, H. Kim and S. Ha, "Dynamic Voltage Scheduling Technique for Low-Power Multimedia Applications using Buffers," *Proceedings of International Symposium on Low-Power Electronics and Design*, 2001.
- [32] S. Lee and T. Sakurai, "Energy-Constrained VDD/VTH Hopping Scheme with Run-Time Power Estimation for Low-Power Real-Time VLSI Systems," To appear in *Journal of Circuits, Systems, and Computers*, Oct. 2002.

이 성 수



1993 서울대학교 전기공학부(석사)
1998 서울대학교 전기공학부(박사)
1998~2000 동경대학교 국제산학공동연구센터 Research Associate
2000~02 이화여자대학교 정보통신학과 전임강사
2002~현재 숭실대학교 정보통신전자공학부 전임강사
관심분야: low-power system design, multimedia signal processing, high-speed signalling

Email: sslee@e.ssu.ac.kr
