

# 멀티 프로세서 임베디드 시스템에서 여유시간 예측에 의한 저전력 태스크 스케줄링

준회원 서 범 식\*, 정회원 황 선 영\*

## An Energy-Efficient Task Scheduling Algorithm for Multi Processor Embedded System by Laxity Estimation

Beom-Sik Suh\* Associate Member, Sun-Young Hwang\* Regular Member

### 요 약

본 논문은 멀티프로세서 임베디드 시스템에서 사용하는 어플리케이션을 구성하는 태스크를 수행하는데 필요한 전력과 태스크간의 의존성에 의해서 발생하는 통신 비용을 감소시킬 수 있는 스케줄링 알고리즘을 제안한다. 제안된 알고리즘은 기존 알고리즘에서는 통신비용은 고려하였으나 스케줄링 단계에서는 동적 전력 관리 결과를 고려하지 않는 점을 보완하여 스케줄링 단계에서 여유시간(Laxity) 사용을 예측하여 동적 전압 관리의 효율을 높이는 방법을 사용한다. 이를 통해 복잡한 멀티미디어 어플리케이션에 동적 전력 관리를 적용하여 수행 시 에너지 효율을 높였다. 실험 결과 멀티미디어 어플리케이션인 HD MPEG4, MotionJPEG codec, MP3, Wavelet 프로그램을 멀티프로세서 환경에서 제안된 알고리즘을 이용하여 저전력 스케줄링 결과 기존 방식에 비해 평균 11.2%의 에너지 감소를 얻었다.

**Key Words** : Laxity, 태스크 스케줄링, 동적 전력 관리, 동적 전압 스케일링, 멀티 프로세서

### ABSTRACT

This paper proposes a scheduling algorithm that can reduce the power consumed for execution of application programs and the communication cost incurred due to dependencies among tasks. The proposed scheduling algorithm can increase energy efficiency of the DVS(Dynamic Voltage Scaling) by estimating laxity usage during scheduling, making up for conventional algorithms that apply the DVS after scheduling. Energy efficiency can be increased by applying the proposed algorithm to complex multimedia applications. Experimental results show that energy consumptions for executing HD MPEG4, MotionJPEG codec, MP3, and Wavelet have been reduced by 11.2% on the average, when compared to conventional algorithms.

### I. 서 론

최근 모바일 기기들에 집적되는 임베디드 시스템은 멀티미디어 어플리케이션에서 대용량 데이터를 처리해야 하는 프로그램의 실시간 수행이 요구되며, 사용자의 더 높은 질에 대한 요구와 다양한 기능의 결합으

로 인해 점점 복잡해지고 있다. 모바일 기기에서 이용하는 멀티미디어 어플리케이션 중 하나인 영상신호 처리는 알고리즘이 복잡하고 대용량의 데이터 처리를 수행 하며, 이미지 처리에 대한 사용자의 요구가 HD 급 화질이 되면서 영상신호 처리는 멀티프로세서를 가진 MPSoC (Multi-Processor System on Chip) 환

\* 본 연구는 교육과학기술부 및 한국연구재단의 지원으로 수행되었습니다(#2010-0008043).

\* 서강대학교 전자공학과 CAD & ES. 연구실 (sbs1986@sogang.ac.kr)

논문번호 : KICS2010-07-712, 접수일자 : 2010년 7월 12일, 최종논문접수일자 : 2010년 11월 4일

경에서 수행될 수밖에 없다. 이러한 어플리케이션이 배터리 동작을 하는 모바일 기기에서 수행되기 위해서는 많은 에너지를 소모하는 PE (Processing Element) 간의 데이터 통신과 PE에서의 연산을 에너지 효율적으로 수행하는 것이 중요하다. MPSoC를 구성하는 PE에서 소모되는 에너지와 PE간 데이터 통신에서 소모되는 에너지를 감소시키기 위하여 회로 수준, 운영 체제, 미들웨어, 응용 수준에 이르기까지 다양한 추상화 단계에서 저전력 설계 기법들이 제안되었다<sup>14)</sup>.

지난 몇 년간 두드러지게 발전한 고집적 반도체 기술은 시스템 내부에서 저전력 수행을 가능하게 함으로써 위와 같은 문제점을 해결한다. 그럼에도 불구하고 현재 시스템에 대한 다양한 요구를 충족하기 위해 시스템 내부 전력 소모를 최소화할 수 있는 저전력 관리가 중요하며, 저전력 관리 기법은 크게 정적인 기법과 동적인 기법으로 분류된다<sup>15)</sup>. 정적인 기법은 설계 혹은 컴파일 시에 적용하는 것으로 전력 소모가 적도록 하드웨어를 합성하는 방법, 소프트웨어를 컴파일할 때 코드의 재구성을 통해 전력 소모가 적도록 하는 방법 등이 있다. 동적인 기법은 실행 시간에 적용하는 것으로 작업 부하량의 변동을 이용하여 시스템의 전력 소모를 줄여준다. 정적인 기법을 통해서도 시스템의 에너지 소모를 상당히 줄일 수 있으나 시스템이 실행되는 가운데 계속해서 변하는 수행 환경에 대처할 수 없는 단점이 있다. 그렇기 때문에 최근 동적인 기법에 더욱 초점이 맞추어지고 있으며, 하드웨어 자원 관리 기능과 수행할 어플리케이션의 총체적인 정보를 갖고 운영 체제 관점에서 전력 관리를 수행 할 수 있도록 하는 연구가 주목받고 있다<sup>16,7)</sup>.

PE간 데이터 통신시 발생하는 통신 비용을 줄이기 위해서 주파수 조절이 가능한 병렬 링크와 직렬 링크도 제안되었다<sup>18,9)</sup>. 링크의 데이터 전송 에너지를 줄이기 위하여 데이터 전송시간 제약을 넘지 않는 범위 내에서 주파수와 전압을 낮춤으로써 대역폭을 조절한다. 또한 PE 간 데이터 전송량을 줄이기 위하여 어플리케이션의 총체적인 정보를 갖고 데이터 의존성이 큰 태스크끼리 같은 PE에 맵핑하는 방법이 제안되었다<sup>10)</sup>.

동적 전력 관리(DPM : Dynamic Power Management)는 시스템의 실행 시간 중 작업 부하량의 변동을 이용하여 하드웨어의 동작 전압을 동적으로 조절하는 방법이다. 동적 전압 조절(Dynamic Voltage Scaling: DVS) 기법이 프로세서 클럭 주파수와 전압을 동적으로 변경하여 프로세서에서 소모되는 전력을 줄이는 기법으로 제안되었으며, 마이크로프로세서 기반의 시스템에서 가장 주목 받는 전력 관리 기술 중

하나이다<sup>11)</sup>. 동적 전압 스케일링은 여유시간을 활용하여 전압 및 동작 주파수를 낮춤으로써 시스템의 에너지 효율성을 높이는 것이며 이미 많은 모바일 프로세서는 전압 조절기능을 갖는다<sup>12)</sup>. 동적 전력 관리 기법에 있어서 중요한 문제의 하나는 마감 시간(deadline)을 지키면서 전력 소모를 최소화 하는 것이다. 임베디드 시스템의 경우 대부분 마감 시간 제약 조건이 존재하는 실시간 시스템이므로, 동적 전력 관리를 적용할 경우 전력 상태 전환에 따른 시간적 오버헤드가 존재하기 때문에 마감 시간을 어기지 않도록 주의하여야 한다.

지금까지 연구된 동적 전압 조절 기법<sup>13)</sup>은 단일 프로세서 시스템을 모델로 하여 제안되었으며 멀티프로세서 시스템을 대상으로 한 연구는 진행이 미비하였다. 최근에 들어서야 시작되고 있으나 기존의 단일 프로세서 칩에서 사용되던 기법들을 단순하게 병렬처리가 가능함을 이용하여 확대 적용하는 차원이어서, 태스크간의 데이터 의존성이 높은 멀티미디어 어플리케이션을 멀티프로세서 시스템에 에너지 효율적으로 적용할 수 있는 동적 전압 조절 기법이 필요하게 되었다.

본 논문에서는 기존에 제안된 동적 전압 조절과 태스크간의 의존성을 고려하여 계산한 여유시간 활용률을 통하여 멀티프로세서 시스템에 적용할 수 있는 스케줄링 알고리즘을 제안한다. 먼저 알고리즘을 구현하는 과정에서 태스크의 우선순위를 정하기 위해서 여유시간 활용률을 계산하고, 이를 활용하여 여유시간 활용률을 최대로 하는 동적 전압 조절 기법을 이용하여 스케줄링하였으며, PE간 데이터 통신 비용을 고려하여 맵핑하였다. 시스템 환경은 동일한 프로세서들과 프로세서를 연결하고 있는 버스로 구성이 되어 있으며, 시스템의 구조와 위상은 결정되어져 있다고 가정하였다. 또한 실제 시스템과 유사한 환경을 구축하기 위하여 어플리케이션의 모형은 data flow와 control flow를 모두 고려하였다.

본 논문의 구성은 다음과 같다. 제 2절에서는 에너지 모델과 기존의 연구를 소개하고 제 3절에서는 여유시간 활용률을 이용한 새로운 동적 전압 조절 알고리즘을 기술하며 제안한 알고리즘과 기존 기법과의 차이를 보인다. 제 4절에서는 제안한 알고리즘의 효율성에 대한 실험 결과를 제시하고, 마지막으로 제 5절에서는 결론을 맺는다.

## II. 관련 연구

최근 복잡한 멀티미디어 어플리케이션을 임베디드

환경에서 실시간으로 처리하기 위하여 멀티프로세서 환경의 임베디드 시스템을 구성한다. 임베디드 시스템에서는 해당 어플리케이션을 수행하는데 소모하는 에너지를 최소화할 필요에 따라 동적 전력 관리 기법을 적용하여 시스템에서 소모되는 에너지를 최소화하는 연구가 진행되고 있다<sup>13)</sup>. 동적 전력 관리 방식의 목적은 프로세서의 수행 주파수를 조절하여 어플리케이션을 에너지 효율적으로 수행시키는데 있다. 동적 에너지 소비는 시스템의 총 에너지 소비의 큰 부분을 차지하고 있으며, 시스템의 동적 전력 소비는 크게 PE에서 사용하는 에너지와, PE간 혹은 PE와 메모리 간의 통신에 사용되는 에너지로 구성된다.

식 (1)은 PE에서의 동적 전력 소비를 나타낸다. 여기서  $p$ 는 전력 소모,  $f$ 는 클럭 주파수,  $\alpha$ 는 switching activity,  $V_{dd}$ 는 공급 전압을 의미한다. 식 (1)에서 보듯이 동적 전력 소비는 프로세서가 소모하는 동적 전압 값의 지수 함수에 비례하여 증가한다. 식 (2)는 공급 전압과 문턱 전압( $V_t$ )의 변화에 따른 클럭 주파수를 구하는 식으로,  $k$ 는 공정에서 결정된 상수이며 지수  $\beta$ 는 1과 2사이의 값을 가진다. 식 (2)를 통해 프로세서의 클럭 주파수는 공급되는 전압 값에 비례함을 알 수 있다. 따라서 동적 전압 관리를 통하여 프로세서에 공급되는 전압을 조정함으로써 시스템에서 소모되는 에너지를 감소시킬 수 있다.

$$p = \frac{1}{2} \alpha f V_{dd}^2 \quad (1)$$

$$f = \frac{k(V_{dd} - V_t)^\beta}{V_{dd}} \quad (2)$$

각 태스크가 PE에 맵핑되어 수행될 때 태스크들의 의존성에 따라서 PE간 통신이 발생한다. 프로세서간 통신의 경우 많은 전력 소모를 유발하기 때문에 데이터양이 많은 멀티미디어를 시스템에서 수행중인 경우 많은 전력 소모가 데이터 통신에서 발생하게 된다<sup>14)</sup>. 통신비용을 감소시키기 위해서는 태스크를 PE에 맵핑하는 단계에서 태스크간의 의존성을 고려해야 한다. 각 메모리를 갖고 있기 때문에 통신비용이 큰 태스크 끼리는 같은 PE에 맵핑을 함으로써 통신비용을 감소시켜 전체 시스템의 에너지를 감소시키는 연구가 수행되었다<sup>10)</sup>. 태스크 간의 의존성을 고려하여 동적 전압 관리 기법을 어플리케이션에 적용하기 위하여 어플리케이션을 Synchronous Data Flow Graph(SDF)

를 이용하여 태스크-통신 그래프  $G(V,E)$ 로 모델링한다. SDF는 각 태스크의 control flow와 data path를 효과적으로 나타낼 수 있는 방법이다<sup>15)</sup>. 여기서 각 vertex  $v_i, v_j \in V$ 는 태스크  $T_i, T_j$ 를 나타내며, 각각의 edge  $e(i,j) \in E$ 는 source vertex  $v_i$ 와 destination vertex  $v_j$ 간의 제어 및 데이터 의존성을 의미하고, edge의 weight는 태스크간 통신량을 의미한다. 멀티프로세서 환경에서 SDF를 이용할 때, 어플리케이션을 구성하는 태스크들 중 병렬화 수행이 가능한 태스크들을 병렬수행이 가능하도록 SDF를 재구성하여 이용한다. 그림 1은 입력으로 받은 SDF와 재구성된 SDF의 예를 보인다.

식(3)은 전체 통신에 소모되는 에너지를 나타낸다.  $PE_i$ 와  $PE_j$ 간의 통신 비용은 각 PE에 맵핑된 태스크가 의존성이 있을 때  $PE_i$ 에서  $PE_j$ 로 한 비트를 전송하는데 필요한 에너지이며  $p_{bit}(PE_i, PE_j)$ 로 나타낸다.  $w(e_{i,j})$ 는 그래프상 edge weight로 통신의 양을 나타내는 정도이다. 의존성이 있는 모든 태스크들에 대해서  $p_{bit}(PE_i, PE_j) \times w(e_{i,j})$ 의 값을 더하면 전체 통신 비용이 된다.

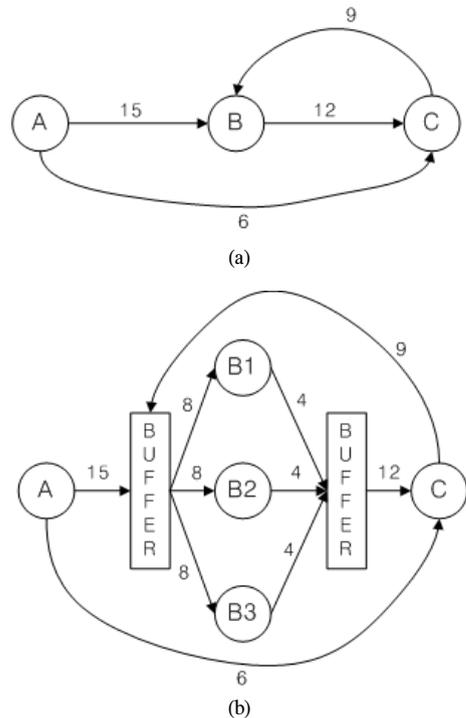


그림 1. 태스크 수준의 SDF 모델의 예. (a) 태스크 수준의 SDF 모델 예, (b) 재구성된 태스크 수준의 SDF 모델의 예

$$p_{comm} = \sum_{\forall e_{i,j} = (src, dest_j)} w(e_{i,j}) \cdot p_{bit}(PE_i, PE_j) \quad (3)$$

기존 연구로 어플리케이션을 태스크-통신 그래프로 모델링하고 하드웨어 정보를 이용하여 태스크의 의존성을 고려하여 통신 비용을 최소화하도록 스케줄링을 한 후, 최적화 이론을 적용하여 에너지 효율적인 태스크 수행 주파수를 결정하는 알고리즘이 제안되었으며 태스크의 수행이 병렬적으로 이루어지기 때문에 이를 고려하여 최대 병렬화 수준을 이용하여 스케줄링시 동적 전압 관리 기술을 적용할 때 에너지 소모를 최소화하는 방법이 제안되었다<sup>[16,17]</sup>. 기존의 방법에서는 멀티프로세서 환경에서 사용하는 프로세서의 클럭 주파수가 연속적이지 않고 이산적이라는 것을 고려하지 않고 스케줄링과 맵핑을 수행한 후에 동적 전압 관리를 통하여 태스크의 수행 주파수를 정하였기 때문에 비효율적인 부분이 발생한다. 이를 해결하기 위하여 스케줄링 단계에서 미리 변화하는 태스크의 수행 시간을 고려하여 맵핑함으로써 에너지 효율을 최대화할

수 있다.

### III. 제안된 동적 전력 관리 기법

제안된 전력 관리 기법은 시스템을 구성하는 하드웨어의 정보와 어플리케이션 프로그램에 관한 정보를 이용하여 여유시간(Laxity)을 예측하며, 여유시간 사용 예측을 통하여 에너지 효율적인 스케줄링 결과를 찾는다. 하드웨어 정보는 프로세서에서 지원하는 클럭 주파수 정보와 프로세서간 통신에 사용되는 통신 구조에 관한 정보이며, 어플리케이션 정보는 어플리케이션을 구성하는 각 태스크들의 수행시간과 제어 및 데이터 의존성 정보이다. 하드웨어 정보와 어플리케이션 정보를 이용하여 스케줄링 단계에서 미리 동적 전압 관리를 적용한 경우의 소요시간을 예측하여 고려한다. 그림 2는 제안된 동적 전력 관리 기법의 의사코드를 보인다. 스케줄링 시간 t에서 수행 가능한 태스크 집합(Released\_TS)에서 각 태스크별 여유시간 활용도를 계산하여 가장 여유시간 활용도가 큰 태스크와 태스

```

Procedure LP_Scheduler(SDF){
  AvailPE = Set of available PEs;           // 선택된 태스크를 수행 할 수 있는 PE 집합
  t = 0;                                     // scheduling time
  Divide given SDF into sub_graphs for using parallelism;
  Laxity_usage = 0;                          // 여유시간 사용율
  Released_TS = Set of executable tasks; // 수행가능한 태스크 집합
  AvailFreq= {f1, f2, ..., fn};           // 프로세서에서 지원하는 클럭 주파수 집합   clk_FS = {f1, f2, ..., fk};
  // 마감 시간을 만족하는 클럭 주파수 집합

  for (t = 0 ; t ≤ Deadline ; t++){
    Released_TS = Released task at t ;
    for each task T ∈ Released_TS {
      clk_FS = Applicable clock frequencies among the elements in AvailFreq ;
      for each clock frequency f ∈ clk_FS {
        Laxity_Estimation( sub_graph, T, f );
        // Proposed laxity estimation

        Calculate Laxity_usage;
      }
      fs = Select clock frequency with the highest Laxity_usage;
    }
    Ts = Select a task with the highest Laxity_usage;
    for each PE ∈ AvailPE {
      cost = α • communication_cost + β • estimation_power ;
    }
    PEs = Select the PE which has the least cost for executing the selected task Ts;
    Assign task Ts to PEs which runs at fs ;
  }
}
    
```

그림 2. 제안된 동적 전력 관리 기법의 의사코드

크의 수행 클록 주파수를 선택하고 통신 비용과 태스크 수행 시 소모되는 에너지를 고려하여 PE를 선택한다. 선택된 태스크가 선택된 PE에서 수행될 때 마감 시간을 고려하여 마감 시간을 어기게 되는 경우는 태스크의 수행 클록 주파수를 다시 결정하게 되며, 새로운 여유시간 활용도에 따라서 스케줄링하게 된다. 어플리케이션이 종료되는 시점까지 반복하면 최종적으로 최소의 에너지 소모를 갖는 태스크 스케줄링 결과를 얻을 수 있다.

### 3.1 여유시간 사용 예측을 이용한 태스크 선택

우선 순위에 기반을 둔 태스크 스케줄링은 불필요한 의존적인 패스의 길이를 작게 만드는 장점이 있다<sup>[18-20]</sup>. 이를 통하여 태스크가 이용할 수 있는 여유시간을 최대화함으로써 동적 전압 스케일링을 통하여 태스크 집합 전체의 에너지 소모를 감소시킬 확률을 높인다. 어플리케이션을 태스크 수준의 SDF로 모델링한 태스크-통신 그래프를 이용하여 스케줄링 시 발생하는 태스크간의 의존성 문제를 고려한다. 스케줄링 시간  $t$ 에서 수행 가능한 태스크 집합에서 전체 여유시간 활용률을 최대로 되게 하는 태스크를 선택하여 스케줄링한다. 여유시간 활용률을 구하기 위하여 어플리케이션을 profiling하여 만든 태스크-통신 그래프를 이용하여 태스크  $T_i$ 의 가장 늦은 완료시간(Latest Finish Time :  $lft_i$ )과 가장 이른 시작시간(Earliest Start Time :  $est_i$ )을 계산한다. 태스크  $T_i$ 의 가장 늦은 완료시간  $lft_i$ 는 식 (4)와 같이 정의된다.

$$lft_i = \min( dl_i , ( lft_j - TC_j ) ) \quad (4)$$

태스크-통신 그래프에서 자식 태스크는 제어 및 데이터 의존성으로 인하여 해당 태스크가 수행이 완료된 후에 수행할 수 있는 태스크이다. 태스크  $T_i$ 의 마감 시간을  $dl_i$ ,  $T_i$ 의 자식 태스크를  $T_j$ , 통신 비용과 수행시간의 합은  $TC_j$ 로 나타낸다. 태스크  $T_i$ 의  $lft_i$ 는 태스크-통신 그래프에서 가장 늦게 끝나는 태스크의 마감 시간을 그 태스크의  $lft$ 로 하여 통신에 걸리는 시간과 수행시간을 빼면 이전 태스크의  $lft$ 가 나옴을 이용하여 태스크-통신 그래프 상의 모든 태스크들의  $lft_i$ 를 계산할 수 있다. 태스크  $T_i$ 의 가장 이른 시작시간  $est_i$ 는 식 (5)와 같이 정의된다.

$$est_i = \max( r_i , \min( t_{PE_j} ) ) \quad (5)$$

태스크-통신 그래프에서 부모 태스크는 해당 태스크가 수행되기 위해서 제어 및 데이터 의존성으로 인하여 반드시 수행이 완료되어 있어야 하는 태스크를 뜻한다. 태스크  $T_i$ 의 준비 완료 시간  $r_i$ (Task Ready Time)는 태스크-통신 그래프에서  $T_i$ 의 모든 부모 태스크들의 수행이 완료된 시점을 뜻한다. 또한  $PE_j$ 의 준비 완료 시간  $t_{PE_j}$ (PE Ready Time)는  $T_i$ 가 PE에 맵핑되어 수행을 시작할 수 있는 시간을 뜻한다. 각 태스크의  $est$ 는 처음 태스크의 수행 시작 시간을 0으로 놓고 각 태스크가 최대 클록 주파수로 수행될 때의 수행시간을 의존성 정보에 맞추어 더해 나감으로써 계산할 수 있다.

각 태스크의  $lft$ ,  $est$ 와 어플리케이션을 profiling하여 얻은 태스크의 수행시간과 의존성 정보를 이용하여 태스크-통신 그래프상의 모든 태스크에 대하여 의존성을 고려하여 활용 가능한 여유시간을 계산할 수 있다. 병렬 처리가 가능한 태스크들과의 의존성이 있는 태스크의 경우, 이 태스크를 높은 클록 주파수로 수행하면 병렬처리 가능한 태스크들은 높은 주파수로 수행하는 태스크에 비하여 병렬화 정도만큼의 전체 여유시간 활용도가 올라간다. 따라서 병렬화 가능한 태스크들을 낮은 클록 주파수로 수행하여 에너지 소모를 감소시킬 수 있다. 최대 여유시간은 모든 태스크가 PE의 최대 클록 주파수로 수행이 될 경우의 어플리케이션의 마감 시간까지 존재하는 PE들의 여유시간의 합이다. 스케줄링 시간에 수행 가능한 태스크집합(released\_TS)에 존재하는 태스크들에 대하여 각 태스크 별로 여유시간 활용도를 최대화할 수 있는 주파수를 계산한다. 한 태스크를 스케줄링 하면 다른 태스크들의  $est$ 와 사용가능한 여유시간이 바뀌게 되므로 이를 다시 계산해 주어야한다.

### 3.2 태스크 맵핑

태스크의  $lft$ 와  $est$ , 그리고 어플리케이션 profiling 작업을 통하여 얻은 태스크의 수행시간과 의존성 정보를 이용하여 선택한 태스크를 부모 태스크와의 통신비용과 수행할 때 소모되는 에너지를 계산하여 가장 적은 에너지를 소모하게 하는 PE에 맵핑하게 된다. PE  $P_j$ 에 태스크  $T_i$ 를 맵핑하여 수행할 때 소모되는 에너지는  $P_j$ 가  $T_i$ 를 수행하는데 소모되는 에너지와  $T_i$ 의 의존성으로 인해서 발생하는 데이터 통신에 소모되는 에너지의 합이다.

PE 선택의 과정은 다음과 같다. 하나의 태스크  $T_i$ 를 스케줄링 시간  $t$ 에서 사용 가능한 PE에 맵핑하는

비용이 현재 다른 태스크를 수행중인  $PE_j$ 에 해당 태스크가 종료된 후에  $T_i$ 를  $PE_j$ 에 맵핑하는 비용보다 큰 경우 현재 시간에 스케줄링하지 않고  $PE_j$ 의 수행이 끝나면  $PE_j$ 에 맵핑한다. 이러한 경우에는 수행중인 태스크를 기다리는 시간을 고려하여 다시 태스크를 수행할 클럭 주파수를 계산한다. 이 때 새로운 주파수와 기존의 주파수를 비교하여 더 적은 에너지를 소비하는 주파수로 태스크가 수행되며, PE를 다시 선택한다. PE 선택 결과 수행중인 태스크의 완료를 기다렸다가 해당 PE에서 수행을 하는 경우에 더 높은 주파수로 태스크를 수행해야하고, 다른 PE에서 수행하는 경우에는 기존의 주파수로 수행해도 마감 시간을 만족하는 경우 두 경우의 에너지 소모를 비교하여 더 적은 에너지가 드는 PE에 맵핑하고, 태스크를 수행할 클럭 주파수를 선택하게 된다. 이 결과 마감 시간 제약조건 만족 여부를 조사하여 마감 시간 제약 조건에 걸리는 경우는 다시 스케줄링 과정을 수행하며, 스케줄링과 맵핑 결과 break-even time 보다 큰 시간 동안 태스크의 수행이 없는 PE는 shutdown 시킨다.

그림 3은 어플리케이션의 SDF에 대하여 기존의 알고리즘을 적용하여 스케줄링한 결과와 제안한 알고리즘을 적용한 결과와의 차이를 보인다. SDF 그래프에서 vertex 내부의 숫자는 동적 전압 스케일링이 적용되지 않은 태스크의 단위 수행 시간을 의미한다.

시스템을 구성하는 프로세서에서 지원 가능한 클럭 주파수가  $f$ 와  $\frac{1}{2}f$  만 있는 경우에 참고문헌[16]에서 제안한 방법의 결과는 그림 3(b)와 같다. 태스크의 의존성을 고려하여 스케줄링과 맵핑을 한 후, ILP 솔루션을 이용하여 최적의 주파수를 선택한 결과  $T_4$ ,  $T_5$  태스크가  $\frac{1}{2}f$ 의 클럭 주파수로 수행되며, 나머지 태스크들은  $f$ 의 클럭 주파수로 수행되어 6단위 시간만큼의 여유시간을 사용한다. 참고문헌[16]에서 제안한 방법의 경우 태스크의 수행 클럭 주파수를 구하기 위해 ILP 솔루션을 이용할 때 각 PE에 수행될 태스크가 정해져 있기 때문에 태스크간의 의존성과 마감 시간 제약 조건으로 인하여 동적 전압 스케일링을 적용할 수 없는 태스크가 발생한다. 본 논문에서 제안한 동적 전압 스케일링 기법을 적용한 경우의 스케줄링 결과는 그림 3(c)와 같다.  $T_2$ ,  $T_4$ ,  $T_5$  태스크가  $\frac{1}{2}f$ 의 클럭 주파수로 수행되며, 나머지 태스크들은  $f$ 의 클럭 주파수로 수행되어 9단위 시간만큼의 여유시간을 사

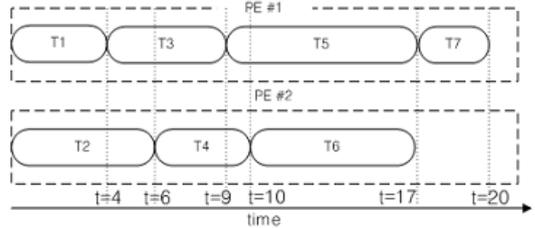
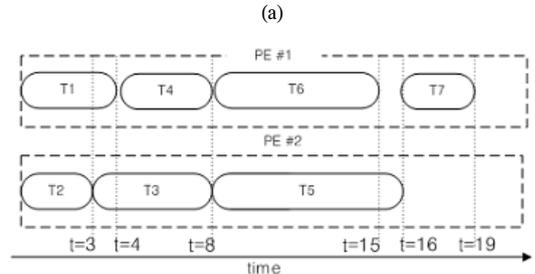
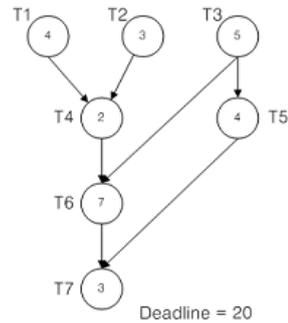


그림 3. 태스크 맵핑 및 스케줄링의 예. (a) 어플리케이션의 SDF의 예, (b) 참고문헌[16]에서 제안한 스케줄링을 적용한 결과, (c) 제안한 스케줄링을 적용한 결과.

용한다. 스케줄링 단계에서 전체 여유시간을 계산하고 각 태스크의 수행시간이 늘어나는 것을 예상하여 동적 전압 관리 기법을 사용할 수 있는 다른 PE에 맵핑을 하는 동작이 있기 때문에 더 효율적인 동적 전압 스케일링할 수 있다.

#### IV. 실험 결과

제안된 알고리즘은 C언어를 이용하여 구현되었으며, 어플리케이션은 MPEG4, MP3, Wavelet, Motion-JPEG reference code<sup>[21-24]</sup>와 GNU profiler를 이용하여 태스크 그래프를 구성하고, 기존논문<sup>[25,26]</sup>에 근거하여 태스크 그래프의 노드를 분할하여 병렬화 하였으며, 각 어플리케이션의 마감 시간은 멀티미디어 어플리케이션의 특성을 반영하여 설정했다. MPEG4,

MotionJPEG은 HD 30 frame/s를 만족할 수 있도록 마감 시간을 설정하였으며, MP3는 1.7Mbyte의 음악 파일을 수행할 수 있도록, Wavelet 은 1628 × 1024 이미지를 실시간으로 변환할 수 있도록 마감 시간을 설정하였다. 연산 장치로 ARM11 프로세서 4개를 사용하였고 각각의 프로세서는 개별 메모리 소자를 가지며 버스 구조를 이용하여 프로세서간 통신이 이루어지게 구성하였다. 전체 어플리케이션의 소스 코드가 저장될 글로벌 메모리를 갖는다. 멀티프로세서에 채택된 PE의 에너지 소모에 관련된 데이터는 참고문헌 [19]에 언급된 ARM11 프로세서의 데이터에 기반한다. 이 경우 전압에 따른 클럭 주파수가 달라지면서 수행시간과 소모 전력이 각각 달라진다. high frequency 모드의 경우와 low frequency 모드의 경우 각각 40pJ/cycle과 13.3pJ/cycle의 에너지가 소모된다. 또한 정적 에너지의 경우 low frequency 모드의 20%만큼을 소모한다. 태스크가 새로 맵핑되는 경우의 오버헤드는 10단위 시간으로 가정한다.

그림 4는 참고문헌[16]에서 제안한 알고리즘과 본 논문에서 제안한 알고리즘을 이용하여 각각의 어플리케이션을 수행할 때의 에너지 소모를 보인다. 표 1은 기존의 알고리즘과 제안된 알고리즘의 에너지 소모 차이를 보인다. 멀티미디어 어플리케이션에 대하여 제안한 알고리즘을 이용하여 수행한 결과, 기존의 알고리즘을 이용한 결과에 비하여 소모되는 에너지가 평균 11.2% 감소함을 보인다. 실험 결과 가장 큰 에너지 효율 감소를 보인 어플리케이션은 MotionJPEG으로 12.8%의 에너지 소모 감소를 보이며, 가장 적은 에너지 효율 감소를 보인 어플리케이션은 MP3로 9.8%의 에너지 소모 감소를 보인다. 복잡한 어플리케이션일수록 의존성이 크고 병렬화 수준이 높기 때문에 스케줄링 단계에서 동적 전력 관리 기법을 이용하지 않고, 스케줄링 이후에 동적 전력 관리 기법을 적용하게 되

표 1. 어플리케이션 별 기존 알고리즘과 제안된 알고리즘의 에너지 소모 비교

어플리케이션	기존 알고리즘 (mW)	제안한 알고리즘 (mW)	비교
MPEG4	135.3	120.1	-11.2 %
MotionJPEG	242.5	219.9	-12.8 %
MP3	23.2	20.7	- 9.8 %
Wavelet	18.8	16.4	-10.8 %
Average	-	-	-11.2 %

면 낮은 클럭 주파수를 선택하지 못하는 경우가 많이 발생하기 때문에 더 많은 에너지 감소율을 얻었다.

제안된 알고리즘은 스케줄링 단계에서 동적 전력 관리 기법을 적용하며, 여유시간 예측과 태스크의 의존성과 병렬화 정보를 이용하여 태스크 수행시간 변화를 종합적으로 분석하여 스케줄링한다. 그 결과 불필요한 에너지 소모를 감소시킬 수 있는 기회가 증가하여 기존의 알고리즘을 적용한 결과에 비하여 어플리케이션을 수행하는데 필요한 에너지가 감소하였다.

### V. 결론 및 추후과제

본 논문에서는 멀티프로세서 시스템에서 복잡한 멀티미디어 어플리케이션을 수행할 때 에너지 효율을 높이기 위하여 기존에 제안된 동적 전압 조절 기법과 태스크간의 의존성을 이용하여 멀티프로세서 시스템에 적용할 수 있는 스케줄링 알고리즘을 제안하였다. 어플리케이션을 profiling하여 얻은 태스크들의 수행시간과 의존성 정보를 이용하여 여유시간 사용을 예측하여 여유시간 활용률을 최대로 하는 태스크를 선택하고, 이를 수행할 클럭 주파수는 여유시간 예측을 바탕으로 하여 결정하였다. 이후 선택된 태스크를 통신 비용을 고려하여 여유시간 활용도를 최대로 할 수 있는 PE에 맵핑하는 방식을 사용하였다. 제안된 스케줄링 알고리즘을 이용하여 MPEG4, MotionJPEG, MP3, Wavelet과 같은 멀티미디어 어플리케이션에 기반한 태스크 그래프를 대상으로한 실험 결과에서 에너지 소모를 효율적으로 줄일 수 있음을 확인하였다.

추후 연구로 동종 프로세서들로 구성된 멀티프로세서 환경을 타겟으로 하는 제안된 알고리즘을 이종 프로세서들로 구성된 멀티프로세서 환경도 지원할 수 있도록 확장하는 연구가 필요하다.

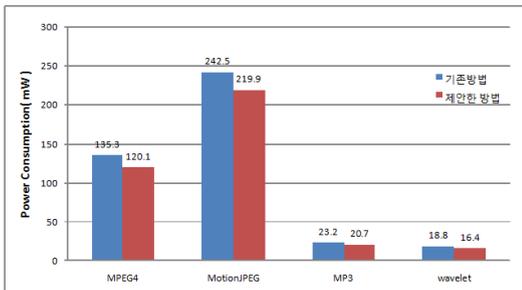


그림 4. 어플리케이션에 기존 및 제안된 알고리즘을 적용한 결과 에너지 소모

참 고 문 헌

- [1] M. Pedram and J. Rabaey, Power Aware Design Methodologies, Kluwer Academic Pub., 2002.
- [2] International Technology Roadmap for Semiconductors 2002, <http://public.itrs.net>.
- [3] Berkeley Predictive TechnologyModels and BSIM4, <http://www-device.eecs.berkeley.edu/research.html>.
- [4] R. Jejurikar, C. Pereira, and R. Gupta, "Leakage Aware Dynamic Voltage Scaling for Real-time Embedded Systems," in Proc. Design Automation Conference, San Diego, CA, pp.275-280, June, 2004.
- [5] J. Rabaey and M. Pedram, Low Power Design Methodologies, Kluwer Academic Pub., 1996.
- [6] Y. Lu, L. Benini, and G. DeMicheli, "Power-Aware Operating Systems for Interactive Systems", IEEE Trans. on VLSI Systems, Vol.10, No.2, pp.119-134, Apr. 2002,
- [7] Y. Lu, L. Benini, and G. DeMicheli, "Operating-system Directed Power Reduction," in Proc. Int. Symp. Low Power Electronics and Design, Rapallo, Italy, pp.37-42, July 2000.
- [8] G. Wei, J. Kim, D. Liu, S. Sidiropoulos, and M. Horowitz, "A Variable-frequency Parallel I/O Interface with Adaptive Power-supply Regulation," IEEE Journal of Solid-State Circuits. Vol.35, No.11, pp.1600-1610, Nov. 2000.
- [9] L. Shang, L. Peh, and N. Jha, "Dynamic Voltage Scaling of Processors and Communication Links in Real-Time Distributed Embedded Systems," IEEE Trans. on VLSI Systems, Vol.15, No.4, pp.427-437, Nov. 2000.
- [10] S. H. Lee, Y. C. Yoon, and S. Y. Hwang, "Communication-Aware Task Assignment Algorithm for MPSoC Using Shared Memory," Journal of Systems Architecture, Vol.56, No.7, pp.233-241, July 2010.
- [11] A. Chandrakasan, A. Sheng, and R. Brodersen, "Low-Power CMOS Digital Design," IEEE Journal of Solid-State Circuits, Vol.27, No.4, pp.473-484, Apr. 1992.
- [12] Y. Shin, K. Choi, and T. Sakurai, "Power Optimization of Real-Time Embedded Systems on Variable Speed Processors," in Proc. Int. Conf. Computer-Aided Design, San Jose, CA, pp.365-368, Nov. 2000.
- [13] M. Pedram and J. Rabaey, Power Aware Design Methodologies, Kluwer Academic Pub., 2002.
- [14] A. Jerraya and W. Wolf, Multiprocessor Systems-on-Chips, Morgan Kaufmann Pub., 2005.
- [15] W. Wolf, High-Performance Embedded Computing, Morgan Kaufmann Pub., 2007.
- [16] 김현진, 홍혜정, 김홍식, 강성호, "멀티프로세서 상의 에너지 소모를 고려한 동적 전압 스케일링 및 전력 셧다운을 이용한 태스크 스케줄링", 한국통신학회논문지, 46권 7호, pp.22-28, 2009년 7월.
- [17] 노 경우, 박창우, 김석윤, "멀티프로세서 시스템을 위한 동적 전압 조절 기반의 효율적인 스케줄링 기법," 대한전기학회논문지, 제 57권 제3호, pp.421-428, 2008년 3월.
- [18] Y. Zhang, X. Hu, and D. Chen, "Task Scheduling and Voltage Selection for Energy Minimization," in Proc. Design Automation Conference, New Orleans, LA, pp.183-188, June. 2008.
- [19] G. Varatkar and R. Marculescu, "Communication-aware Task Scheduling and Voltage Selection for Total Systems Energy Minimization," in Proc. Int. Conf. Computer-Aided Design, San Jose, CA, pp.510-517, Nov. 2003.
- [20] H. Kim, H. Hong, H. S. Kim, J. H. Ahn, and S. Kang, "Total Energy Minimization of Real-time Tasks in an On-chip Multiprocessor Using Dynamic Voltage Scaling Efficiency Metric," IEEE Trans. on CAD, Vol.27, No.11, pp.2088-2092, Nov. 2008.
- [21] 2D Fast Wavelet Transform Library for Image Processing, The Code Project, <http://www.codeproject.com>.
- [22] Mpeg3Play (Release 0.9.6). MP3' Tech., <http://www.mp3-tech.org/programmer/decoding.html>.
- [23] H.264/AVC Software Coordination, Fraunhofer, Institut Nachrichtentechnik Heinrich-Hertz-Institut, <http://iphome.hhi.de/suehring/tml>.
- [24] Motion JPEG decoder Application. JPEG Committee, Standardized in ISO/IEC IS 10918-1/2,

<http://www.jpeg.org>.

- [25] E. Tol, E. Jaspers, and R. Gelderblom, "Mapping of H.264 Decoding on Multi-processor Architecture," in Proc. SPIE Conf. Image and Video Communications and Processing, Santa Clara, CA, Vol.5022, pp.707 - 718, Jan. 2003.
- [26] O. Lehtoranta, E. Salminen, A. Kulmala, M. Hannikainen, and T. Hamalainen, "A Parallel MPEG-4 Encoder for FPGA based Multi-processor SoC," in Proc. Int. conf. Field Programmable Logic and Application, Tampere, Finland, pp.380 - 385, Aug. 2005.

**황 선 영 (Sun-Young Hwang)**

정회원



1976년 2월 서울대학교 전자공학과  
 1978년 2월 한국 과학원 전기 및 전자공학과 공학석사  
 1986년 10월 미국 Stanford대학 전자공학 박사  
 1976~1981 삼성반도체 주식회사 연구원, 팀장  
 1986~1989 Stanford 대학 Center for Integrated System 연구소 책임 연구원 Fairchild Semiconductor Palo Alto Research Center 기술 자문  
 1989~1992 삼성전자(주) 반도체 기술 자문  
 1989년 3월~현재 서강대학교 전자공학과 교수  
 <관심분야> SoC 설계 및 framework 구성, CAD시스템, Com. Architecture 및 DSP System Design 등

**서 범 식 (Beom-Sik Suh)**

준회원



2009년 2월 서강대학교 전자공학과  
 2009년 3월~현재 서강대학교 전자공학과 대학원 CAD&Embedded Systems 연구실 석사과정  
 <관심분야> 저전력 임베디드 시스템 설계, 저전력 SoC 설계.