공유자원을 고려한 저전력 스케줄링

피찬호, 신봉식, 인치호*, 정정화
한양대학교 정보통신대학원 미디어통신공학과
*세명대학교 컴퓨터과학과

Low Power Scheduling with Task Synchronization

Chan-Ho Pi , Bong-Sik Sihn, Chi-ho Lin*, Jong-Wha Chong
Department of Information & Communication, Han-Yang University
Department of Computer Science,Semyung University*
E-mail : pks5406@hanmail.net

Abstract

본 논문은 공유자원과 태스크 활용률을 고려한 저전력 스케줄링 알고리즘을 제안한다. 기존에 공유자원을 고려한 논문에서는 선점 영역에서 최악실행시간만을 고려하였기 때문에 높은 전압을 계속 유지하여 전력 소비가 많고, 태스크들이 조기 종료할 경우 남는 시간을 활용하지 못하는 문제가 있다. 본 논문에서는 선점 영역에서는 태스크들이 조기 종료하는 경우 남는 시간을 이용하여 태스크 활용률을 갱신하고 다음 태스크에게 더 낮은 주파수와 전압을 인가하여 전력 소모를 줄이는 방법을 사용하고, 비 선점 영역에서는 최악 실행시간을 기준으로한 최대 태스크 활용률을 이용하여 우선 순위 높은 태스크의 지연 시간을 최소화 시켜주는 방법을 제안한다. 실험 결과는 Arm8 테스트 벤치마크 데이터를 통해 기존의 알고리즘과 제안된 알고리즘의 전력 소모를 비교했을 때 6%의 파워 소모 감소를 보였다.

1. 서론

최근 이동 내장형 시스템들의 이용과 응용이 확대되고 있다. 이들 시스템들은 대부분 제한된 전원 용량을 가지는 베터리를 기반으로 운용된다.

이들 시스템에서는 주어진 전원 용량으로 얼마나 오래 동안 동작할 수 있느냐의 문제가 제품들의 시장 경쟁력을 결정하는 중요한 요소가 된다. 다양한 저전력 설계 기법들 중 동적 전압 조절 (DVS: Dynamic Voltage Scaling)은 CMOS 회로로 구성되는 프로세서의 공급 전압을 온라인 상태에서 조절하여 전력 소모량을 줄이는 기법이다. 일반적으로 CMOS 회로에의해 소모되는 전력(P)는 공급 전압 V_{DD} 의 제곱에 비례하므로 공급 전압을 낮추는 것은 전체 전력 소모를 낮추는 가장 효과적인 방법이다. 하지만, 공급 전압이 낮아질 때에는 회로의 최대 동작 주파수 또한 함께 늦추어져야 하기 때문에 프로세서의 처리량은 줄어들게 된다. 이러한 프로세서 성능과 전력 소모량 간의 이해득실 관계를 이용하여 주어진 작업에 대한 최적의 전압 및 클럭 속도를 설정하는 것을 동적 전압 조절 스케줄링이라 한다.

최근 저전력 이동 내장형 시스템을 위하여 동적 전압 조절 기능을 활용한 가변 전압 프로세서들이 다수 출시되고 있는데 Arm8™, XScale™ 프로세서등을 예로 들 수 있다. 이와 같이 동적 전압 조절이 가능한 내장형 시스템을 위한 프로세서들이 다수 출시됨에 따라, 최근 들어 저전력 시스템에 특화된 다양한 동적 전압 조절 알고리즘들이 제안되고 있다. 하지만 기존의 대부분의 알고리즘들은 공유자원을[2][3] 고려하지 않았고, 공유자원을 고려한 알고리즘들은[4][5]

태스크의 실행시간이 최악실행시간이란 조건하게 구현되었다. 본 논문에서는 태스크들의 실행시간이 조기 종료하는 경우 최악실행 시간이 아닌 실제 실행시간을 기반으로 하여 새로운 태스크 활용률을[1] 계산하고 이를 바탕으로 하여 태스크의 주파수를 조절해주는 알고리즘을 제안한다.

본 논문 구성은 다음과 같다. 2장에서 동적 전압 가변 기법에 대해 기술하고 3장에서 태스크 모델링을 하도록 한다. 4장에서 기존 논문에 있어서 문제점을 언급하고 5장에서 제안한 알고리즘을 기술하고

6장에서 실험결과를 비교하고 7장에서 결론을 맺는다.

2. 동적 전압 조절(DVS)

실시간 시스템에서 전력 소비 문제가 이슈로 대두되면서 기존에 파워다운 방식으로 저전력을 구현하던 방식에서 현재는 동적전압조절 알고리즘에 관한 연구가 활발하게 이루어 지고 있다. 동적전압조절은 식 (1) 에 의해서 전력은 전압의 제곱에 비례한다는 원리를 이용한다. 마감시간을 놓치지 않는 범위내에서 전압과 주파수를 낮추어 주면 제곱 비례에 의해서 전력 소모를 크게 줄일 수 있다. 그림 1 에서 파워다운 방식은 최대 주파수로 태스크의 수행이 10ms 에서 끝나게 되면 데드라인 25ms 까지 파워를 다운해 준다. 동적전압조절은 데드라인까지 태스크의 주파수와 전압을 낮추어서 사용하는 방식으로 파워소모를 비교 했을 때 동적전압조절은 큰 파워 소비 감소 효과를 가져 올 수 있다.

$$P \propto V_{dd}^2 \qquad (1)$$

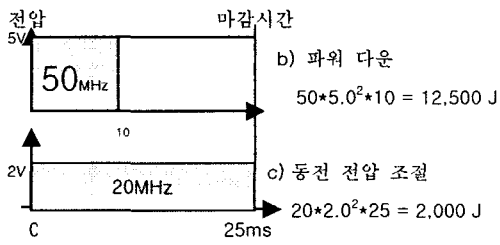


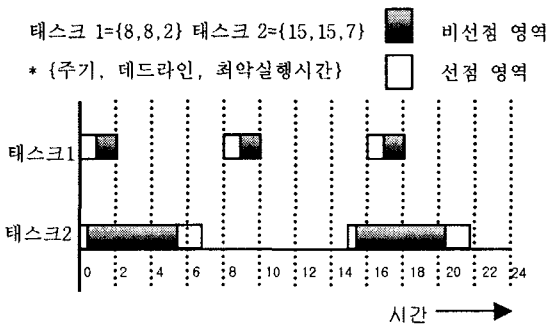
그림 1. 파워다운 방식과 동적 전압 조절 방식간의 파워 소모 비교

3. 태스크 모델링

기존 알고리즘과 제안 알고리즘을 그림 2의 태스크 집합을 모델로 비교하도록 한다. 태스크 1과 태스크 2는 주기, 마감시간, 최악실행시간을 태스크의 파라미터 값으로 가진다. 흰 박스는 공유자원을 사용하지 않는 선점형 영역이고 검정색 박스는 공유자원을 사용하는 비선점형 영역이다. 공유자원에 의해 선점되는 시간(Blocking Time)을 B로 정의하면, 태스크 1은 B=5, 태스크 2 는 B=0이다.

태스크의 우선순위 비교 알고리즘으로는 가장 짧은 마감시간을 가진 태스크가 가장 높은 우선 순위를 가지고 새로운 태스크가 도착할 때마다 마감시간을 비교하여 우선 순위를 조절해 주는 최단 마감시간 우선(Earliest Deadline First) 스케줄링 방식 [1]을 사용한다.

즉 그림 2 에서 태스크 1이 태스크 2에 비해 더 짧은 마감시간을 가지므로 우선 순위가 높다.



. 그림 2. 태스크 집합

공유자원을 사용하지 않는 태스크의 활용율은 다음 식과 같다.

$$\sum_{i=1}^{n} \frac{E_i}{D_i} \le 1 \qquad (2)$$

D_i 는 태스크의 마감시간이고 E_i 는 태스크의 실행 시간이다. 식 (2)의 조건을 만족하는 태스크 집합은 스케줄 가능하다고 말한다.

공유자원을 사용하는 태스크가 있다면 우선 순위가 낮은 태스크에 의해 공유자원이 선점 되었을 경우 우선 순위가 높은 태스크가 공유자원을 사용하기 위해 잠시 대기 하고 있어야 한다. 이 경우 대기 시간이 추가가 되고, 태스크의 활용률은 다음 식과 같다.

$$\sum_{i=1}^{k} \frac{E_i}{D_i} + \frac{B_k}{D_k} \le 1 \qquad (3)$$

식 (3) 과 같이 공유자원에 의해 선점되는 시간(Blocking Time) B_k 가 추가된다.

위의 식 (2)에서 실행시간을 η 로 나누어서 실행시간을 늘려도 $\sum_{i=1}^{k} \frac{E_i/\eta}{D_i} \le 1$ 을 만족한다면 다음 식과 같다.

$$\sum_{i=1}^{n} \frac{E_i}{D_i} \le \eta \qquad (4)$$

식 (4) 와 같이 공유자원을 사용하지 않는 태스크들의 태스크 활용률이 η 보다 작다면 스케줄링 가능하다.

식 (3)에서도 실행시간을 η 로 나누어서 실행시간을 늘려도 $\sum_{i=1}^{k} \frac{E_i/\eta}{D_i} + \frac{B_k/\eta}{D_k} \le 1$ 이 만족을 한다면 다음 식과 같다.

$$\sum_{i=1}^{k} \frac{E_i}{D_i} + \frac{B_k}{D_k} \le \eta \qquad (5)$$

기존 알고리즘과 제안된 알고리즘은 위의 식 (4) 와 식(5) 를 통해 변경된 주파수 값을 이용해 동적 전압 조절을 한다. 위에서 구한 태스크 활용률 η 값은 최대 주파수를 1 로 보았을때의 주파수 비율을 의미한다.

4. 기존 알고리즘

공유자원을 고려한 동적 전압 조절 스케줄링 관련 기존 논문들은 최악 실행 시간을 고려하여 공유자원과 관련없는 선점 영역과 공유자원과 관련된 비 선점 영역에서 주파수값을 다르게 하는 방식을 사용하였다 태스크가 추가될 때 마다 식 (5) 에 의해 가변 주파수 값을 구한다. 공유자원과 관련된 비 선점 영역에서는 태스크가 추가될 때 마다 구해진 가장 큰 주파수 값으로 주파수를 가변하고 공유자원과 관련없는 선점 영역에서는 식 (4) 에 의해 공유자원에 의한 선점이 없는 낮은 주파수 값으로 가변한다.

그림 3 에서 공유자원과 관련없는 영역에서는 식 (4) 에 의해

$$\sum_{i=1}^{n} \frac{E_i}{D_i} \le \eta \rightarrow (\frac{2}{8} + \frac{7}{15}) = 0.716$$

다음과 같이 주파수값을 0.716 을 적용하고 공유자원에 의해 선점되는 영역에서는 식 (5) 에 의해

$$\sum_{i=1}^{k} \frac{E_i}{D_i} + \frac{B_k}{D_k} \le \eta -> (\frac{2}{8} + \frac{5}{8}) = 0.875$$

다음과 같은 주파수값을 적용한다. 그림 3 에서 공유자원에 의해 비선점되는 구간에서는 주파수 값이 0.875 가 되고 나머지 영역에서는 0.716 의 값을 갖는다.

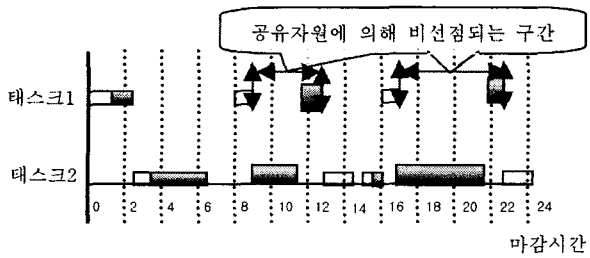


그림 3. 기존 알고리즘에서의 스케줄링

5. 제안된 알고리즘

본 논문에서 제안한 알고리즘은 기존 알고리즘에서의 최악 실행시간만을 고려하것을 개선하여 공유 자원을 고려하지 않는 선점 영역에서는 태스크들이 조기 종료하는 경우 실제 실행시간을 기반으로 하여 새로운 태스크 활용률을 계산하고 이를 바탕으로 하여 태스크의 주파수를 조절한다. 공유자원에 의한 비 선점 영역에서는 최악 실행시간을 기준으로한 최대 태스크 활용률을 이용하여 스케줄링한다. 표 1에서 태스크 1과 태스크 2의 최악 실행 시간과 실제 실행 시간이 주어져 있다.

태스크 1이 수행될때의 태스크 활용률을 식 (5)을 통해 구해보면 η = 0.875, 태스크 2는 η = 0.716 이 된다. 태스크 1은 0.875의 값으로 주파수를 낮추어 주고, 태스크 2는 0.716으로 주파수를 낮추어 준다.

처음 태스크 1이 수행을 하고 나면 최악 실행시간과 실제 실행시간 간의 0.5ms 만큼의 차이가 발생하게 된다. 주파수 값을 식 (5)을 통해 태스크 2는 η =0.654 의 주파수로 가변된다. 실제 실행시간에 의한 태스크 활용률 갱신은 현재 실행된 태스크 보다 우선 순위가 낮은 태스크에 한하여 영향을 미친다. 우선 순위가 높은 태스크에 의해 영향 받지 않는 태스크는 주기 마다 최악 실행시간에 기반한 태스크 활용률에 의해 스케줄링된다. 공유자원에의해 선점 되는 영역에서는 태스크 마다의 태스크 활용률중 가장 큰 값으로 주파수를 높여 준다. 이 영역에서 최대값을 사용하는 이유는 우선 순위가 높은 태스크가 낮은 우선 순위 태스크에 의한 선점으로 지연되는 시간을 최소화 시켜 마감시간을 놓치는것을 방지한다. 위와 같은 방식으로 주파수를 낮추어서 마감시간을 놓치지 않는 범위내에서 낮은 주파수와 낮은 전압으로 태스크를 더 긴 시간동안 수행함으로써 저전력 구현이 가능해 진다.

태스크	최악실행시간	실제실행시간	블록시간	주기
1	2ms	1.5	5	8ms
2	7ms	5	0	15ms

표 1. 태스크 모델링

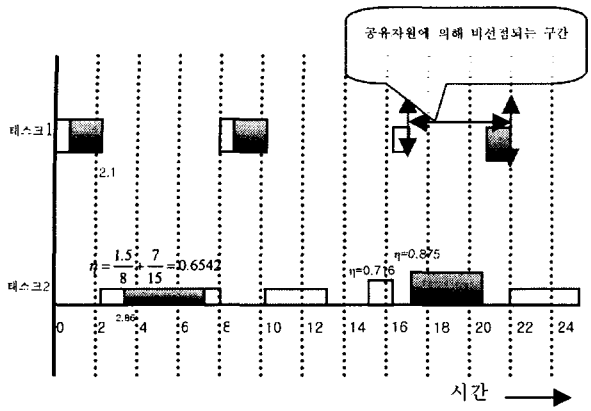


그림 4. 제안한 알고리즘에 의한 스케줄링

6. 시뮬레이션 결과

시뮬레이션 환경은 비쥬얼 C++에서 하였고 파워 측정은 ARM8 태스트 벤치마크 데이터를 사용하여 파워량을 측정하였다. 시뮬레이터는 크게 최단 마감시간 우선(Earliest Deadline First) 스케줄링방식[1] 과 주파수 스케일값을 변경해주는 부분, 공유자원을 고려하는 부분으로 나누어 진다.

표 2 의 태스크들로 시뮬레이션을 하였고, 5000 틱 동안 주기 태스크를 수행하고, 실제 실행시간은 최악실행시간의 90%가 수행된다고 가정하였다. 기존 고리즘과 제안 알고리즘의 총파워 소모량을 구해보았다.

태스크	최악실행시간	블록시간	주기
1	2ms	5	4ms
2	2ms	3	13ms
3	5ms	0	20ms

표 2. 태스크 모델링

알고리즘	총파워소모량
기존 알고리즘	159.638mW
제안 알고리즘	148.182mW

표 3. 기존 알고리즘과 제안 알고리즘의 파워소모량 비교

위의 실험 결과에 의해 기존 알고리즘과 제안 알고리즘의 파워 소모량 비교시 기존의 알고리즘에 비해 6% 파워 소모 감소 효과를 보였다.

결론

동적 전압 조절 기법과 공유자원을 고려한 기존 논문들은 최악 실행 시간만을 고려하였다. 하지만 태스크의 실행 시간은 대부분 최악 실행 시간보다 적은 시간 동안 수행이 끝이 난다. 본 논문에서는 최악 실행 시간이 아닌 실제 실행 시간을 고려하여 최악 실행 시간과 실제 실행 시간의 차이만큼의 더 낮은 주파수 할당을 통해 저전력을 구현하였다.

제안된 알고리즘은 6%의 파워 소모 감소를 보였고, 앞으로의 계획은 공유자원을 고려한 동적 전압 가변 기법에 있어서 정적 그리고 동적인 슬랙값 측정을 통해 더 많은 파워 소모 감소를 시킬 수 있는 알고리즘에 대해 연구하도록 하겠다.

참고문헌

[1] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard real-time environment," *Journal of the ACM*, vol. 20, no. 1, pp. 46-61, 1973

[2] L.Sha,R.Rajkumar,and J.P.Lehoczky. Priority inheritance protocols: An approach to real-time synchronization. In IEEE Transactions on Computers, pages 1175-85,1990.

[3] T.P.Baker. Stack-based scheduling of realtime processes. In RealTime Systems Journal,pages 67-99,1991.

[4] R.Jejurikar and R.Gupta. Energy aware EDF scheduling with task synchronization for embedded real time operating systems. In COLP,2002.

[5] F.Zhang and S.T.Chanson. Processor voltage scheduling for real-time tasks with non-preemptible sections. In Real-Time Systems Symposium,2002.